

# **TIBCO Substation ES™**

## **Configuration and Resources**

*Software Release 2.10*  
*May 2016*

## Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO BusinessEvents, TIBCO Business Process Management, TIBCO BusinessWorks, TIBCO Enterprise Message Service, TIBCO Rendezvous, TIBCO Mainframe WebUI, TIBCO Mainframe RED, and TIBCO Substation ES are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2001-2016 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

# Contents

---

<b>Figures .....</b>	<b>8</b>
<b>TIBCO Documentation and Support Services .....</b>	<b>9</b>
<b>Configuration of Services .....</b>	<b>11</b>
Overview of Data Transformation .....	11
Transformation of Inbound Messages .....	11
Transformation of Outbound Messages .....	11
Definitions of Configuration Metadata .....	11
Metadata Relationship .....	12
Allocation of Configuration File .....	13
Configuring Substation ES Transformer .....	13
Defining the Back-End System (BES) .....	14
Defining Network Parameters (ESB Endpoint) .....	15
Using the COBOL Copybook Converter .....	16
Defining Messages .....	17
Defining Buffers .....	17
Defining Conversion Rules .....	18
Defining Recipes .....	19
Defining Triggers .....	21
Defining Groups .....	22
Transformer DCUI Panels and Usage .....	23
Main Entry Panel .....	23
Transformer Configuration Panel .....	24
Message Definitions Panel .....	25
Define Fields in a Message Panel .....	26
Conversion Rules using Message Definition Panel .....	28
Buffer Definitions Panel .....	28
Define Fields in a Buffer Panel .....	29
Buffer Field Details Panel .....	31
Conversion Rules using Buffer Definition Panel .....	32
Conversion Rules Panel .....	33
Select a Message Definition Panel .....	34
Select a Buffer Definition Panel .....	35
Define Msg-Field or Buffer-Field Conversion Panel .....	35
Define Message Fields to Conversion Rule Panel .....	36
Recipes & Triggers using a Conversion Rule Panel .....	37
Define Processing System Resources Panel .....	38

Define ESB Endpoints Panel .....	39
ESB Endpoint Extensions Panel .....	40
ESB Endpoint – Connection Factory Extensions Panel .....	41
ESB Endpoint SSL Extensions Panel .....	43
ESB Administration Extensions Panel .....	44
Recipe List Panel .....	45
Select ESB Endpoint Panel (Recipe) .....	46
Select Processing System Resources Panel (Recipe) .....	47
Select Conversion Rules Panel (Recipe) .....	48
Recipe Details Panel .....	48
MOI Usage .....	54
CICS Service Details Panel .....	55
IMS Service Details Panel .....	57
EMS Recipe Details Extension Panel .....	58
EMS Properties (Recipe) Panel .....	62
EMS Selector Specification Panel .....	63
Define Triggers Panel .....	64
Select ESB Endpoint Panel (Triggers) .....	65
Select Output Conversion Rule Panel (Triggers) .....	65
Trigger Details Panel .....	66
EMS Trigger Details Extension Panel .....	69
EMS Properties (Trigger) Panel .....	71
Group Definitions Panel .....	72
Select Triggers and Recipes for Group Panel .....	73
Display Configuration File Detail Panel .....	73
Select Desired Entry Panel .....	74
Summary of Definition Panel (Recipe) .....	74
Summary of Definition Panel (Trigger) .....	83
Summary of Definition Panel (Conversion) .....	88
Summary of Definition Panel (Group) .....	91
Copybook Converter Panel .....	92
Log Viewer Panel .....	93
Log Entries Panel .....	94
Single Message Values Panel .....	95
<b>Additional Configuration .....</b>	<b>97</b>
Adding an Additional Substation ES CICS Interface .....	97
Adding an Additional ESB Interface .....	97
Adding a Rendezvous ESB Interface with SSL .....	98
Adding Substation ES Instances to a Single CICS Region .....	98

Using Connection Factories .....	98
Using RVDQ with Multiple Substation ES Instances .....	99
Using RVDQ with One Substation ES Instance .....	100
Using Opaque Fields and Message Sizes .....	101
<b>Utilities for Configuration File .....</b>	<b>102</b>
Overview .....	102
Conversion Utility .....	102
Function .....	102
JCL Sample (SXSCFC26) .....	102
Parameter Values .....	103
Configuration File Utility .....	104
Functions .....	104
JCL Sample (SXSCFIMP and SXSCFEXP) .....	104
Parameter Values .....	105
Selective Export Utility .....	106
Functions .....	106
JCL Sample (SXSCFEXS) .....	106
Parameter Values .....	107
Control Parameters .....	107
<b>Examples .....</b>	<b>111</b>
Overview .....	111
Substation ES Installation IVPs .....	112
Request or Reply .....	112
Triggers .....	112
Substation ES Communication Programs .....	112
Rendezvous IVP Sender Listener .....	112
Enterprise Message Service IVP Sender Listener .....	115
IVP Request or Reply EMS Consumer Producer .....	118
IVP Generic Listener .....	119
IVP IMS Conversational Sender or Listener .....	120
IVP Generic Error Listener .....	120
Example of ESB Request – Reply DPL .....	121
Component Summary .....	121
Execution Parameters of IVP Sender Listener .....	121
Example of ESB Request – Reply DPL Container .....	122
Component Summary .....	122
Execution Parameters of IVP Sender Listener .....	123
Example of ESB Request – Reply DPL Storage .....	123
Component Summary .....	123

Execution Parameters of IVP Sender Listener .....	123
Example of ESB Request – Reply DPL TSQ .....	124
Component Summary .....	124
Execution Parameters of IVP Sender Listener .....	124
Example of CICS Trigger .....	125
Component Summary .....	125
CICS Execution Command .....	125
Example of CICS High Volume Trigger .....	125
Component Summary .....	126
CICS Execution Command .....	126
Example of RED CICS Trigger .....	126
Component Summary .....	127
CICS Execution Command .....	127
Example of ESB Request DPL XCTL .....	128
Component Summary .....	128
Execution Parameters of IVP Sender Listener .....	128
Example of ESB Invoking a CICS Transaction .....	128
Component Summary .....	128
Execution Parameters of IVP Sender Listener .....	129
Example of ESB to TDQ .....	129
Component Summary .....	129
Execution Parameters of IVP Sender Listener .....	129
Example of CICS TSQ Entry .....	130
Component Summary .....	130
Execution Parameters of IVP Sender Listener .....	130
Example of ESB Request – Reply Complex DPL .....	131
Component Summary .....	131
Execution Parameters of IVP Sender Listener .....	131
Example of CICS-Initiated Request or Reply .....	131
Component Summary .....	132
CICS Execution Command .....	132
Example of RED CICS-Initiated Request or Reply .....	133
Component Summary .....	133
CICS Execution Command .....	133
Example of ESB Request – Reply IMS .....	134
Component Summary .....	134
Execution Parameters of IVP Sender Listener .....	134
Example of ESB Request – Reply and Trigger .....	134
Component Summary .....	135

Execution Parameters of IVP Sender Listener .....	135
Example of IMS BMP Trigger .....	135
Component Summary .....	135
Execution Parameters of JCL .....	136
Example of IMS Synchronous Callout — Request or Reply .....	136
Component Summary .....	136
Example of IMS Conversational Functions .....	136
Component Summary .....	137
Execution Parameters of IVP IMS Conversational Sender or Listener .....	137
<b>Communication of Application Messages .....</b>	<b>139</b>
Communication Fields .....	139
System Fields .....	139
Definitions .....	140
Usage .....	140
Defining System Fields on an Inbound Conversion Rule .....	142
System Fields Used as Property Fields .....	143
Trigger Service Example .....	145
Recipe Service Example .....	145
Dynamic Transaction IDs .....	145
User Fields .....	146
Definitions .....	146
Attributes .....	147
Application Notifications .....	147
Information Types .....	148
Structure .....	148
<b>IVP Resources and Configuration File Utilities .....</b>	<b>149</b>
Sample IVP Resources .....	149
Summary of Substation ES ESB Configuration File Utilities .....	152
<b>Conversion Formats .....</b>	<b>154</b>
Conversion Formats from Rendezvous .....	154
Conversion Formats from TIBCO Enterprise Message Service .....	156
<b>BusinessWorks Sample Projects for Substation ES .....</b>	<b>159</b>
IMS Project .....	159
CICS Project .....	160

# Figures

---

Structure of Application Notification Message ..... 148

Nested Structure of Substation ES Error Message ..... 148

Structure of Substation ES Error Messages as Properties Fields ..... 148



# TIBCO Documentation and Support Services

---

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

<https://docs.tibco.com>

## Product-Specific Documentation

Documentation for TIBCO products is bundled with the software. It is also available on the TIBCO Documentation site at <https://docs.tibco.com/products/tibco-substation-es>. To directly access documentation for this product, double-click the following file:

`TIB_substation_version_docinfo.html`

where this file is shipped with the software package .zip file.

The following documents for this product can be found on the TIBCO Documentation site or after extracting the documentation .zip file:

- *TIBCO Substation ES Concepts*
- *TIBCO Substation ES Installation*
- *TIBCO Substation ES Operations and Administration*
- *TIBCO Substation ES Configuration and Resources*
- *TIBCO Substation ES Messages and Codes*
- *TIBCO Substation ES Release Notes*

The following documents provide additional information and can be found on the TIBCO Documentation site:

- *TIBCO Rendezvous for z/OS Installation and Configuration*
- *TIBCO Rendezvous for z/OS COBOL Reference and TIBCO Rendezvous C Reference*
- *TIBCO Enterprise Message Service User's Guide*
- *TIBCO Enterprise Message Service C & COBOL API Reference*
- *TIBCO Mainframe RED User's Guide*
- *TIBCO Mainframe RED Installation*

## How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

**How to Join TIBCO Community**

TIBCO Community is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCO Community offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://community.tibco.com>

# Configuration of Services

---

You have to configure some services before Substation ES™ can exchange data between systems that run in the z/OS environment and external TIBCO messaging applications.

Before you can pass data between an external TIBCO messaging application and a transaction processing application, you must establish communications and determine how data is transformed between these environments. Data is transformed by Substation ES Transformer (referred to as Transformer) that you configure through the transformer data configuration user interface (DCUI) panels.

## Overview of Data Transformation

Substation ES subscribes to and publishes TIBCO messages on behalf of CICS and IMS transaction processing applications that run on the z/OS system.

### Transformation of Inbound Messages

You can configure Substation ES to subscribe to and process requests generated by a TIBCO messaging application, for example, inbound messages.

To configure Substation ES to subscribe to and process requests generated by a TIBCO messaging application, you have to specify the following items:

- The message subject to listen for
- How data is transformed for delivery to a transaction processing application
- The method of invocation (MOI)
- Whether to return data (or some type of reply) to the calling application

If return data or an acknowledgement is expected from the transaction processing application, you must specify how the data or acknowledgement is transformed for the calling application.

For information about messaging flow and different types of MOIs, see *TIBCO Substation ES Concepts*.

### Transformation of Outbound Messages

You can configure Substation ES to publish data generated by transaction processing applications, for example, outbound messages, by using the trigger MOI.

In this case, you assign a subject to the data. With this subject, the data can be delivered to TIBCO messaging applications subscribing to this subject. You can also specify how data in outbound messages is transformed to be suitable for delivery to TIBCO messaging applications. Since transaction processing applications cannot use a message subject to indicate the configuration to use when transforming the data in the buffer, you must instead specify a text string starting at a given offset.

## Definitions of Configuration Metadata

Definitions of configuration metadata are stored in a file that is read by Substation ES. These definitions determine how Substation ES Transformer processes and transforms message data.

Definitions that are configured for the Transformer can be organized into the following categories:

- System
- Inbound or outbound message data
- Metadata

For information about relationships between definitions of configuration metadata, see [Metadata Relationship](#).

## System

The following are system definitions:

- Back-end system (BES): defines the interfaces to mainframe systems where transaction processing applications are located.
- Network: defines ESB endpoints, ports, daemons, or TIBCO Enterprise Message Service servers and IP networks.

## Inbound or Outbound Message Data

The following are inbound or outbound message data definitions:

- Message: defines inbound Rendezvous or TIBCO Enterprise Message Service messages and their data fields, name and value pairs.
- Buffer: defines outbound transaction processing application buffers and their data fields, name and value pairs.

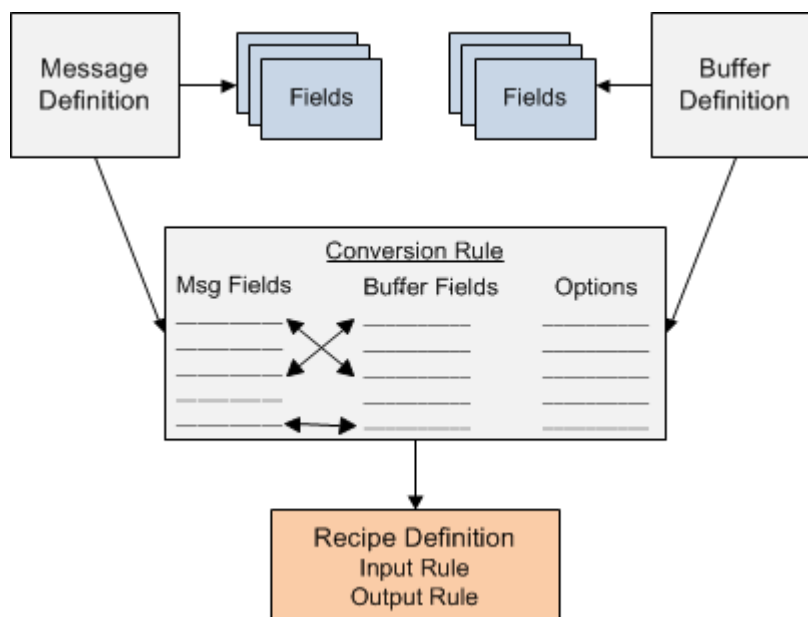
## Metadata

The following are metadata definitions:

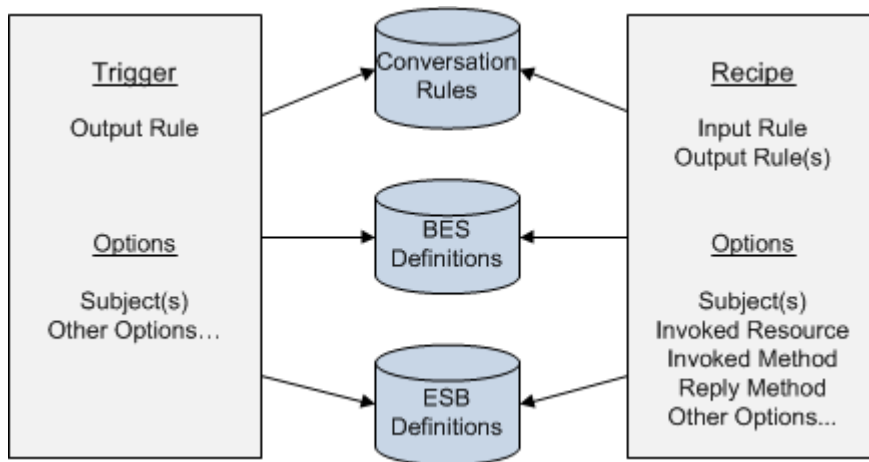
- Conversion rule: defines the conversion between inbound and outbound messages, for example, the mapping between message fields and buffer fields.
- Recipe or trigger: defines the decisions for routing inbound and outbound messages to or from specific mainframe processes. A recipe defines both inbound and outbound message routing; a trigger defines outbound message routing only.
- Group: defines groupings of recipes and triggers that achieve customized routing goals.

## Metadata Relationship

The following figure shows a conversion rule definition and its relationship with a message definition and a buffer definition:



The following figure shows a recipe definition, a trigger definition, and their components:



## Allocation of Configuration File

You can use File Creation JCL to create a new version of the Substation ES Transformer Configuration file for use with Substation ES.

You can define and allocate a VSAM linear file, and then use this file as the repository for all the transformation configuration settings that are read by Substation ES.

The Transformer Configuration file can be shared among multiple users under the following conditions:

- The Transformer Configuration file is a catalogued data set shared between LPARs.
- VSAM **SHAREOPTIONS (2 3)** must be used to limit the update authority to one user.
- The first user who opens the configuration file for editing has update authority.
- The second and subsequent users have read-only authority.

The JCL and control statements that are required for defining and allocating a Transformer configuration file are as follows:

```
//JS010 EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//DD1 DD UNIT=SYSDA,VOL=SER=xxxxxx TARGET VOLUME
//SYSIN DD *
        DEFINE CLUSTER { -
                NAME(USERHLQ.CONFIG) -
                VOLUMES(xxxxxx) -
                SHAREOPTIONS(2 3) -
                TRACKS(10 10) -
                CISZ(4096) -
                LINEAR -
        }
//
```

You must change the data set name, DASD volume name, and DASD volume device according to your environment. Also, replace USERHLQ with High Level Qualifier for your Substation ES libraries.

## Configuring Substation ES Transformer

You can configure Substation ES Transformer from the Transform Configuration panel.

To start the Transformer DCUI panel, see [Main Entry Panel](#) for details. The following examples show how to configure Substation ES Transformer:

- [Defining the Back-End System \(BES\)](#)

- [Defining Network Parameters \(ESB Endpoint\)](#)
- [Using the COBOL Copybook Converter](#)
- [Defining Messages](#)
- [Defining Buffers](#)
- [Defining Conversion Rules](#)
- [Defining Recipes](#) and [Defining Triggers](#)
- [Defining Groups](#)

After doing each task, you exit back to the Transformer Configuration panel. You can review the messages displayed in the upper-right corner of the panel to verify your operations:

- If your changes have been saved successfully, the message `Changes saved` is displayed.
- If you cancel your operation with the **CAN** command, the message `Changes Cancelled` is displayed.
- If you make a wrong selection while editing a panel, a related error message is displayed.
- If a file opened in read-only mode is edited, the message `Change` and `Readonly` is displayed.

For more information about the Transformer Configuration panel, see [Transformer Configuration Panel](#).

## Defining the Back-End System (BES)

You can define or modify the processing system resources or BES to be used with the recipe or trigger definition.



Do not consider any dependencies before performing the following steps.

### Procedure

1. Select option 4 - `Define Processing System Resources` to go to the Define Processing System Resources panel.  
For more information about this panel, see [Define Processing System Resources Panel](#).
2. On the command line, enter `ADD`, and then press Enter to create a new Substation ES interface.
3. Select a Substation ES interface:

- `C(ics)`: CICS Interface.
- `I(ms)`: IMS Interface.
- `A(dmin)`: Administration Interface.

4. In the **Identifier** field, enter up to 16 characters to name the interface.
5. In the **Interface Id** field, enter up to 8 characters to identify the assigned interface. Press Enter.



The interface ID must match the ID that is defined in the Interface SIP member.

6. Exit the Define Processing System Resources panel in either of the following ways:
  - Press PFK3.
  - On the command line, enter `END`, and then press Enter.
7. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.

## Defining Network Parameters (ESB Endpoint)

You can define or modify an ESB definition for a recipe or trigger definition.

See *TIBCO Rendezvous Administration* or the TIBCO Enterprise Message Service documentation for details and the default settings of the parameters in this procedure.



Do not consider any dependencies before performing the following steps.

### Procedure

1. Select option 5 - Define ESB Endpoints and Parameters to go to the Define ESB Endpoints panel.

For more information about this panel, see [Define ESB Endpoints Panel](#).

2. On the command line, enter ADD, and then press Enter to create a new ESB definition.
3. In the ESB Endpoint Extensions panel, set field values to define the ESB. The configurations vary depending on the messaging protocol that is used by Substation ES to communicate with TIBCO applications.

- For TIBCO Enterprise Message Service:

- a) In the **Type** field, enter E for TIBCO Enterprise Message Service.
- b) In the **Startup** field, specify (Y or N) whether this ESB starts during the initialization of Substation ES.
- c) In the **URL** field, specify the host name or host IP address that Substation ES uses to find the TIBCO Enterprise Message Service server and establish communication.



For details on setting up EMS ECF connection, see [Using Connection Factories](#).

- d) In the **Port** field, specify (if applicable) the value of the TCP port used by the Substation ES ESB.
- e) In the **User** field, specify (if applicable) a user ID if TIBCO Enterprise Message Service requires a specific ID.
- f) In the **Password** field, specify (if applicable) a password if TIBCO Enterprise Message Service requires a specific password.
- g) In the **TCPIP Stack Name** field, specify the IBM stack name to be used with this connection.

- For TIBCO Rendezvous:

- a) In the **Type** field, enter R for TIBCO Rendezvous.
- b) In the **Startup** field, specify (Y or N) whether or not this ESB starts during the initialization of Substation ES.
- c) In the **URL** field, specify (if applicable) the host name or host IP address that Substation ES uses to find the Rendezvous daemon and establish communication.
- d) In the **Port** field, specify the daemon port if necessary. The default value is 7500.
- e) In the **Service** field, specify the UDP service port.
- f) In the **Network** field, specify (if applicable) the network IP address.
- g) In the **Multicast** field, specify (Y or N) whether this ESB is to use multicast addressing.
- h) In the **Multicast Address** field, specify (if applicable) the multicast addresses.



You must specify this field if you set the Multicast field to Y.

- i) In the **TCPIP Stack Name** field, specify the IBM stack name to be used with this connection.
- For more information about the ESB Endpoint Extensions panel, see [ESB Endpoint Extensions Panel](#).

4. Press Enter before exiting the panel.

5. Exit the ESB Endpoint Extensions panel in either of the following ways:

- Press PFK3.
- On the command line, enter END, and then press Enter.

When you return to the Define ESB Endpoint panel, you can change the identifier for this new ESB endpoint by overtyping the value in the **Identifier** field.

6. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.

## Using the COBOL Copybook Converter

You can use Copybook Converter to read a COBOL copybook and create configuration definitions for Substation ES Transformer.

In the Copybook Converter panel, you can specify whether a buffer definition and a Rendezvous message definition are created. If you do not want the converter to take an action, leave the panel selection blank. If you specify both a buffer identifier and a message identifier, you can also define a conversion rule that creates a one-to-one mapping for data conversion.

To create buffer, message, and conversion rule names that are identical to the copybook name, insert an asterisk (\*) in the panel field that corresponds to the selection.

### Prerequisites

Ensure that you have checked the dependencies on the COBOL copybook members.

### Procedure

1. Select option 10 - COBOL Copybook Converter to go to the Copybook Converter panel.  
For more information about this panel, see [Copybook Converter Panel](#).
2. In the **Copy Member** field, enter the COBOL copybook member name to use for conversion.
3. In the **Buffer Identifier** field, enter up to 16 characters as a name that identifies the buffer definition.
4. In the **Message Identifier** field, enter up to 16 characters as a name that identifies the message definition.
5. In the **Convert Rule Id** field, enter up to 16 characters as a name that identifies the conversion rule definition.
6. In **Library containing Copy Member** field, enter the data source name (DSN) of the partitioned data set (PDS) that contains the copybook member.
7. In the **Additional libraries for COPY** field, enter the DSNs of the PDS that can contain other copybooks that are included within the copybook member. Press Enter.
8. Exit the Copybook Converter panel in either of the following ways:
  - Press PFK3.
  - On the command line, enter END, and then press Enter.
9. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.





In cases where features not supported by the converter are required for the definitions, you must edit the definitions to change the default attributes, clear the mapping selections, and apply the changes. Note the following behavior of the converter:

- Levels **66**, **77**, and **88** items are ignored.
- Pictures with **BLANK WHEN ZERO**, **DATE FORMAT**, **EXTERNAL**, **GLOBAL** are ignored.
- **OCCURS** clauses only generate the first occurring field items.
- **REDEFINE** clauses select the 01 level group of the redefined item.
- **FILLER** fields are not converted; they are processed only to compute the proper buffer position.
- All **Group level** structures are not converted; only elementary items are converted.

## Defining Messages

You can define a new message definition or select an existing message definition for a conversion rule for modification.



Do not consider any dependencies before performing the following steps.

### Procedure

1. Select option 1 - Define Message Contents to go to the Message Definitions panel.  
For more information about this panel, see [Message Definitions Panel](#).
2. Type either of the following commands:
  - To create a new entry, enter ADD on the command line, and then press Enter.
  - To select an existing definition to modify, enter S in the **Sel** field, and then press Enter.
3. In the **Message Id** field, enter up to 16 characters to name the message definition.
4. In the **Type** field, enter a valid data type.
5. In the **Dec** field, enter (if applicable) the implied number of digits after a decimal point.
6. In the **Name** field, enter up to 54 characters to name the message field. Press Enter.
7. Press Enter before exiting the panel.
8. Exit the Define Message Contents panels in either of the following ways:
  - Press PFK3.
  - On the command line, enter END, and then press Enter.
9. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.

## Defining Buffers

You can define a new buffer definition or select an existing buffer definition for modification.



Do not consider any dependencies before starting these steps.

### Procedure

1. Select option 2 - Define Buffer Areas to go to the Buffer Definitions panel.  
For more information about this panel, see [Buffer Definitions Panel](#).

2. If you want to select an existing buffer definition to modify, enter S in the **Sel** field, and then press Enter.
3. If you want to create a new buffer definition, enter ADD on the command line, and then press Enter. Define the following fields in the Define Fields in a Buffer panel:
  - a) In the **Buffer Id** field, enter up to 16 characters to name the message definition.
  - b) Optional: In the **Maximum Size** field, enter the maximum size of the buffer.
  - c) In the **Type** field, enter the mainframe data type that the invoked application expects.
  - d) In the **Start** field, enter the mainframe data type that the invoked application expects.
  - e) In the **Len** field, enter the value to use for the length of the field.  
The length is in bytes, not decimal characters retained. Thus, a COBOL PIC S9(9) COMP retains nine characters but is four bytes in length.
  - f) In the **Dec** field, enter (if applicable) the implied number of digits after a decimal point.
  - g) In the **Name** field, enter up to 45 characters to name the buffer field.  
For more information about the Define Fields in a Buffer panel, see [Define Fields in a Buffer Panel](#).
4. Enter S under **Sel**, and then press Enter to go to the Buffer Field Details panel, where you can configure the details of a buffer definition field:
  - a) In the **Pad Character** field, enter (if applicable) the character to use for padding purposes.
  - b) In the **Separate Sign** field, enter (if applicable) Y for the sign selection.
  - c) In the **Justify Right** field, enter (if applicable) Y.
  - d) In the **Leading Sign** field, enter (if applicable) Y for the sign selection.
  - e) Enter the value to be used to fill in the field if the field is missing in the inbound message and the field required option is No. Then the initial value in the buffer field is defined in the **Initial Value** field.
  - f) In the **Initial Value Length** field, enter (if applicable) the value for the length of the Initial Value field. Press Enter.  
For more information about the Buffer Field Details panel, see [Buffer Field Details Panel](#).
5. Exit the Define Buffer Areas panels in either of the following ways:
  - Press PFK3.
  - On the command line, enter END, and then press Enter.
6. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.

## Defining Conversion Rules

You can define a new conversion rule definition or select an existing conversion rule definition for modification.

### Prerequisites

Ensure that you have done the following tasks:

- [Defining Messages](#)
- [Defining Buffers](#)

### Procedure

1. Select option 3 - Define Conversion Relationships to go to the Conversion Rules panel.  
For more information about this panel, see [Conversion Rules Panel](#).
2. Type either of the following commands:

- To create a new entry, enter ADD on the command line, and then press Enter.
  - To select an existing definition to modify, enter S in the **Sel** field, and then press Enter.
3. In the **Sel** field, enter S next to the message definition you want to use, and then press Enter.  
For more information about this panel, see [Select a Message Definition Panel](#).
  4. In the **Sel** field, enter S next to the buffer definition you want to use, and then press Enter.  
For more information about this panel, see [Select a Buffer Definition Panel](#).
  5. In the **Conversion ID** field, enter up to 16 characters to name the conversion rule definition.
  6. In the **Max Buffer Size** field, enter (if applicable) a value to use for the maximum buffer size.
  7. In the **Trunc** field, enter Y or N to truncate data that extends beyond the specified buffer length.
  8. In the **Reqd** field, enter Y or N to indicate if the field is required.
  9. In the **BfFld** field, enter the sequence number of the buffer field that maps to the message field for the transformation.
  10. Press Enter before exiting the panel.
  11. Exit the Define Conversion Rules panels in either of the following ways:
    - Press PFK3.
    - On the command line, enter END, and then press Enter.
  12. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved and that no error messages are displayed.

## Defining Recipes

You can define a new recipe definition or modify an existing recipe definition.

### Prerequisites

Ensure that you have done the following tasks:

- [Defining the Back-End System \(BES\)](#)
- [Defining Network Parameters \(ESB Endpoint\)](#)
- [Defining Conversion Rules](#)

### Procedure

1. Select option 6 - Define Recipes -Inbound / Outbound / Destinations/BES to go to the Recipe List panel.  
For more information about this panel, see [Recipe List Panel](#).
2. Type any of the following commands:
  - To create a new recipe definition, enter ADD on the command line, and then press Enter.
  - To select an existing definition to modify, enter S in the **Sel** field, and then press Enter.
  - To set TIBCO Enterprise Message Service specific settings, enter E in the **Sel** field, and then press Enter.
3. In the **Sel** field, enter S to select ESB you want to use and press Enter to continue the process.  
For more information about this panel, see [Select ESB Endpoint Panel \(Recipe\)](#).
4. In the **Sel** field, enter S to select a processing system definition, and then press Enter to continue the process.  
For more information about this panel, see [Select Processing System Resources Panel \(Recipe\)](#).

5. In the **Sel** field, select a conversion rule for a recipe, and then press Enter to continue the process.
  - Enter **B** to use the same rule for both input and output.
  - Enter **I** to specify a rule for input, and then repeat the process selecting **O** to specify a different rule for output.

For more information about the Conversion Rules panel, see [Select Conversion Rules Panel \(Recipe\)](#).

6. Configure the recipe definition in the Recipe Details panel:
  - a) In the **Recipe Identifier** field, enter up to 31 characters to name the recipe.
  - b) In the **Listen to** field, enter the invoke subject name that Substation ES listens for to invoke a given transformation.
  - c) In the **Reply to** field, enter (if applicable) the outbound subject name that Substation ES uses to publish the reply message for this process.
  - d) In the **Reply To Len (Dyn)** field, enter (if applicable) the number of bytes in the buffer to be used for a dynamic reply subject name.
  - e) In the **Start** field, enter (if applicable) the starting offset in the buffer to be used for the dynamic reply subject name.
  - f) In the **Usage Limit** field, enter (if applicable) the maximum number of concurrent messages to be processed by an ESB for this recipe. To specify no limit, enter 0.  
This parameter is only valid with EMS transports.
  - g) In the **Error Subject** field, enter a subject name for the error message subject.
  - h) In the **Error Threshold** field, enter (if applicable) the number of errors allowed before the recipe is disabled.
  - i) In the **Error Percentage** field, enter (if applicable) the percentage of errors allowed before the recipe is disabled.
  - j) In the **Resource Name** field, enter the name of the BES resource to be invoked upon receipt of a request.
  - k) In the **Method of Invocation** field, select the process method that Substation ES invokes for this transformation.
  - l) In the **Reply Method** field, select the option of outbound data that is expected for this process. Press Enter.

7. Set the CICS Service details in the CICS Service Details panel:
  - a) On the command line, enter **CICS**, and then Press Enter.
  - b) In the CICS Service Details panel, configure CICS Service Details, and then press Enter.
  - c) Press PFK3 to exit this panel.

For more information about this panel, see [CICS Service Details Panel](#)

8. Set the IMS Service details in the IMS Service Details panel:
  - a) On the command line, enter (if applicable) **IMS**, and then Press Enter.
  - b) In the IMS Service Details panel, configure the IMS Service details, and then press Enter.
  - c) Press PFK3 to exit this panel.

For more information about this panel, see [IMS Service Details Panel](#).

9. Set the EMS details for a recipe in the EMS Recipe Details Extension panel:
  - a) On the command line, enter (if applicable) **EMS**, and then press Enter.
  - b) In the EMS Recipe Details Extension panel, configure EMS Details, and then press Enter.
  - c) Press PFK3 to exit this panel.

For more information about this panel, see [EMS Recipe Details Extension Panel](#)

10. Set the EMS properties in the EMS Properties panel:
  - a) On the command line, enter (if applicable) **PROP**, and then press Enter.

- b) In the EMS Properties panel, configure the EMS properties, and then press Enter.
- c) Press PFK3 to exit this panel.

For more information about this panel, see [EMS Properties \(Recipe\) Panel](#).

11. Set the EMS Selector specification in the EMS Selector Specification panel:
  - a) On the command line, enter (if applicable) `SEL`, and then press Enter.
  - b) In the EMS Selector Specification panel, set the EMS Selector specification, and then press Enter.
  - c) Press PFK3 to exit this panel.

For more information about this panel, see [EMS Selector Specification Panel](#).

12. Exit the Define Recipes panel in either of the following ways:

- Press PFK3.
- On the command line, enter `END`, and then press Enter.

13. Review the message displayed in the upper-right corner to verify that your changes have been saved.

## Defining Triggers

You can define a new trigger or modify an existing trigger.

### Prerequisites

Ensure that you have done the following tasks:

- [Defining Network Parameters \(ESB Endpoint\)](#)
- [Defining Conversion Rules](#)

### Procedure

1. Select option 7- Define Triggers - Outbound Convert / Destinations to go to the Define Triggers panel.

For more information about this panel, see [Define Triggers Panel](#).

2. Type any of the following commands:

- To create a new entry, enter `ADD` on the command line, and then press Enter.
- To select an existing definition to modify, enter `S` in the **Sel** field, and then press Enter.
- To set TIBCO Enterprise Message Service specific settings, enter `E` in the **Sel** field.

3. In the **Sel** field, enter `S` next to the ESB definition that you want to use, and then press Enter to continue the process.

For more information about this panel, see [Select ESB Endpoint Panel \(Triggers\)](#).

4. In the **Sel** field, enter `S` or `O` next to the conversion rule definition, and then press Enter to continue the process.

For more information about this panel, see [Select Output Conversion Rule Panel \(Triggers\)](#).

5. Configure the recipe definition in the Trigger Details panel:

- a) In the **Trigger Identifier** field, enter up to 31 characters to name the trigger.
- b) In the **Publish to** field, enter the name of the outbound subject on which a message is published for a given trigger transaction.
- c) In the **Publish To, Sec** field, enter (if applicable) the optional second outbound subject that a message is published on for a given trigger transaction.
- d) In the **Reply to** field, enter (if applicable) the outbound subject name that Substation ES uses to publish the reply message for this trigger process.

- e) In the **Publish To Len (Dyn)** field, enter (if applicable) the number of bytes in the buffer to be used for a dynamic reply subject name.
- f) In the **Start** field, enter (if applicable) the starting offset in the buffer for the dynamic reply subject name.
- g) In the **Error Subject** field, enter a subject name for error messages.
- h) In the **Error Threshold** field, enter (if applicable) the number of errors allowed before the trigger is disabled.
- i) In the **Error Percentage** field, enter (if applicable) the percentage of errors allowed before the trigger is disabled.
- j) In the **Method of Invocation** field, select the option of outbound data process method that is expected for this process.
- k) In the **Reply to** field, select the option of outbound data that is expected for this process.
- l) In the **Start** field, enter the starting offset in the buffer that contains the text to be used for the search criteria in a trigger process.
- m) In the **Length** field, enter the number of bytes of the text to be used for the search criteria in a trigger process.
- n) In the **Value** field, enter the text to be used for the search criteria in a trigger process.
- o) In the **Undelivered Requeue** field, enter (if applicable) Y so that the reliable trigger processes will requeue message if delivery fails. Press Enter.

For more information about the Recipe Triggers panel, see [Trigger Details Panel](#).

- 6. Press Enter before exiting the panel.
- 7. Set the EMS details for a trigger in the EMS Trigger Details Extension panel:
  - a) On the command line, enter (if applicable) EMS, and then Press Enter.
  - b) In the EMS Trigger Details Extension panel, configure the EMS details, and then press Enter.
  - c) Press PFK3 to exit this panel.

For more information about this panel, see [EMS Trigger Details Extension Panel](#).

- 8. Set the EMS properties in the EMS Properties panel:
  - a) On the command line, enter (if applicable) PROP, and then press Enter.
  - b) In the EMS Properties panel, configure the EMS properties, and then press Enter.
  - c) Press PFK3 to exit this panel.

For more information about this panel, see [EMS Properties \(Trigger\) Panel](#).

- 9. Exit the Define Triggers panels in either of the following ways:
  - Press PFK3.
  - On the command line, enter END, and then press Enter.
- 10. Review the message displayed in the upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.

## Defining Groups

You can define a new group definition or modify an existing group definition consisting of previously defined recipe and trigger definitions.

The Group definition combines recipe and trigger definitions to be used by Substation ES. You can specify the group names in the transformer SIP member to tell Substation ES which groups of recipe and trigger definitions you want to load.

With grouping, one configuration file can be used by multiple instances of Substation ES at the same time, thus reducing the requirement for defining duplicate recipe and trigger definitions.

### Prerequisites

Ensure that you have done the following tasks:

- [Defining Recipes](#)
- [Defining Triggers](#)

### Procedure

1. Select option 8 - Define Groups.  
For more information about this panel, see [Group Definitions Panel](#).
2. Type either of the following:
  - To create a new entry, enter ADD on the command line, and then press Enter .
  - To choose an existing definition to modify, enter S in the **Sel** field, and then press Enter.
3. Type a group name.  
The group name can be up to 15 characters long.
4. In the **Sel** field, enter S next to each trigger and recipe that you want to select into the group. Press Enter to continue the process.  
For more details for this panel, see [Select Triggers and Recipes for Group Panel](#).
5. Press Enter before exiting the panel.
6. Exit the Define Groups panel by doing either of the following:
  - Pressing PFK3.
  - Entering END on the command line and pressing Enter.
7. Review the message displayed in upper-right corner of the Transformer Configuration panel to verify that your changes have been saved.

## Transformer DCUI Panels and Usage

You can configure Substation ES Transformer to transform data through the Transformer data configuration user interface (DCUI) panels.

### Main Entry Panel

Substation ES Main Entry Panel provides the main entry to access the Transformer Configuration panels.

To access Substation ES the Transformer Configuration panels, perform the following steps:

1. Go to the ISPF Command Shell (Option 6).
2. Enter the `ex 'USERHLQ.CLIST(SXGSSPNL)'` command.  
where *USERHLQ* is the High Level Qualifier for your Substation ES installed libraries. Then Main Entry Panel is displayed.
3. Type 1 in the **Select option** field, and then press Enter.  
The Transformer Configuration panel is displayed.



To see a panel ID, enter PANELID on the command line. You can use the **PANELID** command to toggle the display of panel IDs on and off.

### Panel Layout

```

SXGMP001 ----- TIBCO Software Inc. -----
                                Main Entry Panel                                2004/10/12
                                                                                   09:49

      SSSS  u  u  b      ssss  tttt  aaaa  tttt  iiii  oooo  nn  n
      SS    u  u  b      ss    tt   a  a   tt   ii   o  o  nn  n
  
```



```

      S   u   u   bbbb   s       tt   aaaa   tt   ii   o   o   n   n   n
      SS  u   u   b   b   ss      tt   a   a   tt   ii   o   o   n   nn
      SSSS uuuu   bbbb   ssss     tt   a   a   tt   iiii  oooo  n   nn

              ((   EEEE   SSSS   ))
              ((   E     SS     ))
              ((   EE     S      ))
              ((   E      SS     ))
              ((   EEEE   SSSS   ))

Select option      ==>
      1      Configuration
      2      LTA Search and Display Facility
ENTER = PROCEED   END PFKey = RETURN

```

### Field Description

The following table lists the fields in the Substation ES Main Entry Panel:

Field	Description
<b>Select option</b>	<p>You can select a panel. Two options are available:</p> <ul style="list-style-type: none"> <li>Configuration: enter 1 to select this option, press Enter, and then the Transformer Configuration panel is displayed, where you can start configuring Substation ES transformer.</li> </ul> <p>For more information about this panel, see <a href="#">Transformer Configuration Panel</a>.</p> <ul style="list-style-type: none"> <li>LTA Search and Display Facility: enter 2 to select this option, press Enter, and then the <a href="#">Log Viewer Panel</a> is displayed.</li> </ul>

## Transformer Configuration Panel

The Transformer Configuration panel provides options for defining a new set of conversion records or for working with an existing set of conversion records.

### Panel Layout

```

SXTPMAIN ----- TIBCO Substation (ES) with RED V1 -----
                  Transformer Configuration 2.9

SELECT OPTION  ==>

File Name ==> USERHLQ.DATASET
Read Only ==> N (Yes/No)
Debug      ==> 0 (0 - 5)

Select One of the Following:
  1 - Define Message Contents
  2 - Define Buffer Areas
  3 - Define Conversion Relationships
  4 - Define Processing System Resources
  5 - Define ESB Endpoints and Parameters
  6 - Define Recipes - Inbound/Outbound/Destinations/BES
  7 - Define Triggers - Outbound Convert/Destinations
  8 - Define Groups
  9 - Show Related Structures
 10 - COBOL Copybook Converter

Enter END command to terminate.

```

### Field Description

The following table lists the fields in the Transformer Configuration panel:



Field	Description
<b>SELECT OPTION</b>	<p>You can select the panels and functions to perform through the Transformer DCUI panels. The following options are available:</p> <ul style="list-style-type: none"> <li>• 1 - Define Message Contents</li> <li>• 2 - Define Buffer Areas</li> <li>• 3 - Define Conversion Relationships</li> <li>• 4 - Define Processing System Resources</li> <li>• 5 - Define ESB Endpoints and Parameters</li> <li>• 6 - Define Recipes - Inbound/Outbound/Destinations/BES</li> <li>• 7 - Define Triggers - Outbound Convert/Destinations</li> <li>• 8 - Define Groups</li> <li>• 9 - Show Related Structures</li> <li>• 10 - COBOL Copybook Converter</li> </ul>
<b>File Name</b>	The name of the Substation ES Transformer configuration VSAM file. You must specify the file name before selecting an option.
<b>Read Only</b>	<p>If you select N, updates made to the Transformer configuration VSAM file are saved. If you select Y, the updates are not saved.</p> <p>Valid selections: Y(es) and N(o). Default selection: N.</p>
<b>Debug</b>	<p>Required. The level of informational messages recorded to the screen. This value must be 0 unless TIBCO Support require the change.</p> <p>Valid values: 0 - 5. Default value: 0.</p>

## Message Definitions Panel

You can define or modify a message definition in the Message Definitions panel.

### Panel Layout

```

SXTPO1 ----- Message Definitions ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> CSR

(ADD=Create new Message, CAN=Cancel updates)
Sel Flds  Message Id
-----
      0  New-Msg1
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Message Definitions panel:

Field	Description
<b>Command</b>	<p>You can enter ADD to create a new message definition, and enter CAN to cancel all modifications just done to the existing message definition.</p> <p>Valid commands: <b>ADD</b> and <b>CAN</b>.</p>

Field	Description
<b>Sel</b>	You can select an existing message definition to modify.  Valid selections: S(elect), D(elete), R(epro), and U(sedon).  If you select U, you can access the panel described in <a href="#">Conversion Rules using Message Definition Panel</a> .
<b>Flds</b>	The total number of fields in the message definition.
<b>Message Id</b>	Required. The user defined name of the message definition.  Default value: New-Msg1.

## Define Fields in a Message Panel

You can modify the fields of a message definition in the Define Fields in a Message panel.

### Panel Layout

```

SXTPO1F ----- Define Fields in a Message ----- Row 1 to 20 of 20
Command ==>                                     Scroll ==> CSR
(Types: DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64,
      F32, F64, POR, IPA)
Message Id      ==> New-Msg1

Sel  Type Dec Id   Seq Name
-----
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
      0   0  0  0
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Define Fields in a Message panel:

Field	Description
<b>Message Id</b>	The name of the message definition.
<b>Sel</b>	You can modify a previously defined field in a message definition.  Valid selections: S(elect), D(elete), and R(epro).

Field	Description
<b>Type</b>	<p>The data type for the field.</p> <p>The following are valid data types for TIBCO Enterprise Message Service (EMS) and Rendezvous (RV):</p> <ul style="list-style-type: none"> <li>OPA: an opaque byte sequence for RV, and bytes for EMS.</li> <li>STR: a character string for RV, and an EMS or JMS UTF8 data type for EMS.</li> <li>BOL: Boolean for both RV and EMS.</li> <li>I8: an 8-bit integer for RV, and a byte for EMS.</li> <li>I16: a 16-bit integer for RV, and an EMS or JMS short data type for EMS.</li> <li>U16: an unsigned 16-bit integer for RV, and the wchar data type for EMS.</li> <li>I32: a 32-bit integer for RV, and bytes for EMS.</li> <li>U64: an opaque byte sequence for RV, and an EMS or JMS integer data type for EMS.</li> <li>F32: a 32-bit floating point for both RV and EMS.</li> <li>F64: a 64-bit floating point for RV, and an EMS or JMS double data type for EMS.</li> <li>MSG: a message for RV, and an EMS-mapped message for EMS.</li> </ul> <p>The following are valid data types for only Rendezvous:</p> <ul style="list-style-type: none"> <li>DAT: the Rendezvous date-time format.</li> <li>U8: an unsigned 8-bit integer.</li> <li>U32: an unsigned 32-bit integer.</li> <li>I64: a 64-bit integer.</li> <li>POR: a 2-byte IP port.</li> <li>IPA: a 4-byte IP address.</li> </ul>
<b>Dec</b>	<p>The implied number of digits after the decimal point. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.</p> <p>This field is also used to indicate a bytes type message. For more information, see <a href="#">Using Opaque Fields and Message Sizes</a>.</p>
<b>Id</b>	<p>A message field identifier. Zero (0) is a special value that signifies no field identifier. You must not add a field that contains both a null field name and a nonzero field identifier. This field is not implemented in the current release.</p>
<b>Seq</b>	<p>The sequence in which each field is displayed in the Conversion Rules panel. Numeric only.</p> <p>Valid values: 0 - 9999. Default value: 0.</p>
<b>Name</b>	<p>Required. The user defined name of the field.</p>

## Conversion Rules using Message Definition Panel

The Conversion Rules using Message Definition panel displays the conversion rules where a message definition is used.

You access this panel by selecting U in the **Sel** field of the Message Definitions panel. For more information, see [Message Definitions Panel](#).

### Panel Layout

```
SXTP01U ----- Conversion Rules using Message Definition -----
Command ==>                                     Scroll ==> CSR

      Message Id ==> New-Msg1

      Used in these Conversion Rules:
      ***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Conversion Rules using Message Definition panel:

Field	Description
<b>Message Id</b>	Required. The name of the message definition used in the conversion rule. Default value: New-Msg1.
<b>Used in these Conversion Rules</b>	The names of the conversion rule definitions that use the message definition.

## Buffer Definitions Panel

You can define or modify a buffer definition in the Buffer Definitions panel.

### Panel Layout

```
SXTP02 ----- Buffer Definitions ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

(ADD=Create new Buffer, CAN=Cancel updates)
Sel Flds Buffer Id
-----
      0 New-Buffer1
      ***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Buffer Definitions panel:

Field	Description
<b>Command</b>	You can enter ADD to create a new buffer definition, and enter CAN to cancel all modifications just done to the existing buffer definition.  Valid commands: <b>ADD</b> and <b>CAN</b> .

Field	Description
<b>Sel</b>	You can select an existing buffer definition to modify. Valid selections: S(elect), D(elete), R(epro), and U(sedon). If you select U, you can access the panel described in <a href="#">Conversion Rules using Buffer Definition Panel</a> .
<b>Flds</b>	The total number of fields in the buffer definition.
<b>Message Id</b>	Required. The user defined name of the message definition. Default value: New-Buffer1.

## Define Fields in a Buffer Panel

You can modify the fields of a buffer definition in the Define Fields in a Buffer panel.

### Panel Layout

```

SOTP02F ----- Define Fields in a Buffer ----- Row 1 to 20 of 20
Command ==>                                     Scroll ==> CSR
  (Types are Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, Opa)
  Buffer Id    ==> New-Buffer1
  Maximum Size ==> 0
Sel  Type  Start  Len  Dec  Name
-----
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
      0      0      0
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Define Fields in a Buffer panel:

Field	Description
<b>Message Id</b>	The name of the buffer definition.

Field	Description
<b>Maximum Size</b>	<p>The maximum size of the buffer (data area) that is used for communication.</p> <p>Valid values: 0 - 4194304. Default value: 0.</p> <p>If the value is 0, the size of the buffer is determined by the field definitions. Otherwise, this value overrides the determination from the fields. If specified, the value must be greater than or equal to the field determination.</p>
<b>Sel</b>	<p>You can modify a previously defined field in a buffer definition.</p> <p>Valid selections: S(elect), D(elete), and R(epro).</p>
<b>Type</b>	<p>Required. The type of mainframe data expected by the invoked application for the field.</p> <p>The following are valid data types:</p> <ul style="list-style-type: none"> <li>• Str: the variable-length CHARACTER data string. Always contains a null terminator at the end such as ALPHANUMERIC, NUMERIC and ALPHABETIC.</li> <li>• Text: the fixed-length CHARACTER data string such as ALPHANUMERIC, NUMERIC, and ALPHABETIC.</li> <li>• Pdec: the PACKED-DECIMAL or COMP-3 data format.</li> <li>• Updec: the UNSIGNED PACKED-DECIMAL or COMP-3 data format.</li> <li>• Zdec: the EXTERNAL DECIMAL data format such as USAGE or DISPLAY.</li> <li>• Uzdec: the UNSIGNED ZONED DECIMAL data format.</li> <li>• Ubin: the UNSIGNED BINARY, COMP or COMP-5 data format.</li> <li>• Bin: the SIGNED BINARY or COMP data format.</li> <li>• Float: the FLOATING-POINT data format such as COMP-1 and COMP-2.</li> <li>• Opa: a block of unchanged characters (no ASCII/EBCDIC conversion).</li> </ul>
<b>Start</b>	<p>The number of starting bytes that the field is offset in the buffer. The buffer offset positions the data at a given byte position.</p>
<b>Len</b>	<p>Required. The maximum size of the field represented in storage or memory. This size must be greater than 0.</p> <p>Valid values: 0 - 4194304.</p>
<b>Dec</b>	<p>The implied number of digits after the decimal point. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.</p> <p>This field is also used to indicate a bytes type message. For more information, see <a href="#">Using Opaque Fields and Message Sizes</a>.</p>
<b>Name</b>	<p>Required. The user defined name of the field.</p>

## Buffer Field Details Panel

You can modify the details of a field in a buffer definition in the Buffer Field Details panel.

### Panel Layout

```
SXTP02D ----- Buffer Field Details -----
Command ==>

      Field Name ==> string field


      Field Type ==> STR      (STR, ...
Buffer Start ==> 0
Buffer Length ==> 8
Decimal Places ==> 0  (0-8)
Pad Character ==>      or hex( )
Separate Sign ==> N  (N/Y)
Leading Sign ==> N  (N/Y)
Justify Right ==> N  (Y/N)
Initial Value ==>
Initial Value Length ==> 0

Press  END to save
```

### Field Description

The following table lists the fields in the Buffer Field Details panel:

Field	Description
<b>Field Name</b>	The name of the field.
<b>Field Type</b>	Required. The type of data expected by the invoked application for the field.  Valid data types: Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, and Opa. For more information about these types, see <a href="#">Type</a> in the Define Fields in a Buffer panel.
<b>Buffer Start</b>	The number of starting bytes that the field is offset in the buffer. The buffer offset positions the data at a given byte position.
<b>Buffer Length</b>	The expected length of the data for the field. If the data exceeds this length and is not truncated as specified in the <b>Trunc</b> field of the Define Msg-Field / Buffer-Field Conversion panel, the transformation fails.
<b>Decimal Places</b>	The implied number of digits after the decimal point. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Pad Character</b>	The character to use for padding. You can use keyboard accessible characters, which are at the beginning or the end according to the justify configuration.  Default value: space.

Field	Description
<b>or hex( )</b>	The fill character defined as a hex value instead of an alpha value.
<b>Separate Sign</b>	This field indicates whether a byte is reserved for the sign or the position of the sign.  Valid selections: N(one) and Y(es). Default selection: N.
<b>Leading Sign</b>	The field contains a leading sign as opposed to a trailing sign.  Valid selections: N(one) and Y(es). Default selection: N.
<b>Justify Right</b>	The positioning of the string. You can fill in the data on either side with padding characters, as appropriate.  Valid selections: Y(es) and N(es). Default selection: N.
<b>Initial Value</b>	<p>The value or characters that fill in the field if the data is not available in the buffer.</p> <div>  <p>Setting initial values is only for mapped buffer fields, for example, fields that have corresponding message field names selected on a conversion rule.</p> <p>If a message field is not present in the input message, apply the following rules:</p> <ul style="list-style-type: none"> <li>• If the buffer field type is Text and the BES application expects the entire field to be initialized with a defined character, such as LOW-VALUES (00), specify the desired value on the pad character in the <b>or HEX</b> input field. Do not specify a value in the <b>Initial Value Length</b> field.</li> <li>• If the BES application expects the buffer field to be filled with an array of characters, specify values for both the <b>Initial Value</b> and the <b>Initial Value Length</b> fields. If you set the initial value length to 99, Substation ES uses the first character of the initial value to initialize the entire buffer field. The default value for initial value is a low value.</li> <li>• If the buffer field type is Text and the field is missing from the request message, and the value in <b>Initial Value Length</b> field is 0, Substation ES initializes the entire field with spaces by default.</li> </ul> </div>
<b>Initial Value Length</b>	The length of the <b>Initial Value</b> field when used.

## Conversion Rules using Buffer Definition Panel

The Conversion Rules using Buffer Definition panel displays the conversion rules in which a buffer definition is used.

You access this panel by selecting U in the **Sel** field of the Buffer Definitions panel. For more information, see [Buffer Definitions Panel](#).

### Panel Layout

```
SXTP02U ----- Conversion Rules using Buffer Definition-----
Command ==> Scroll ==> CSR
```



Buffer Id ==> New-Buffer1

Used in these Conversion Rules:

\*\*\*\*\* Bottom of data \*\*\*\*\*

### Field Description

The following table lists the fields in the Conversion Rules using Buffer Definition panel:

Field	Description
<b>Buffer Id</b>	Required. The name of the buffer definition used in the conversion rule. Default value: New-Msg1.
<b>Used in these Conversion Rules</b>	The name of the conversion rule definitions that uses the buffer definition.

## Conversion Rules Panel

You can create or modify a conversion rule in the Conversion Rules panel.

### Panel Layout

SXTPO3 ----- Conversion Rules ----- Row 1 to 6 of 6						
Command ==>						
(ADD=Create new Conversion Rule, CAN=Cancel updates)						
(Use line cmd 'M' to relate Message fields to conversion rules)						
	Max	EMS	S			
Sel	Flds	Bufsz	Type	Z	Conversion Id	Message Id
						Buffer Id
	1	0	M	B	CVR-I-ADMIN	MSG-IN-ADMIN
	9	0	M	B	CVR-I-IMSCVS-01	MSG-IN-IMSCVS
	13	0	M	B	CVR-I-TSRR-01	MSG-IO-RR-01
	12	0	M	B	CVR-I-01	MSG-IN-01
	20	0	M	B	CVR-I-02	MSG-IN-02
	1	0	B	M	CVR-IO-BYTES	MSG-JMS-BYTES

### Field Description

The following table list the fields in the Conversion Rules panel:

Field	Description
<b>Command</b>	You can enter ADD to create a new conversion rule, and enter CAN to cancel all modifications just done to the existing conversion rule. Valid commands: <b>ADD</b> and <b>CAN</b> .
<b>Sel</b>	You can select an existing conversion rule to modify. Valid selections: S(elect), M(essages), D(elete), R(epro), and U(sedon). If you select M, you can access the panel described in <a href="#">Define Message Fields to Conversion Rule Panel</a> . If you select U, you can access the panel described in <a href="#">Recipes &amp; Triggers using a Conversion Rule Panel</a>
<b>Flds</b>	The number of fields that have been mapped between messages and buffers.

Field	Description
<b>Max Bufsz</b>	Optional. The maximum size of the buffer (data area) that is used for communication. Numeric only.  Default value: 0.
<b>EMS Type</b>	The message type of a nested message.  Valid selections: M(ap), S(tream), T(ext), and B(ytes). Default selection: M.
<b>SZ</b>	This field is used only with TIBCO Enterprise Message Service messages and if the message type is either BYTES or TEXT.  Valid selections: M(ap) and B(ytes). Default selection: B.  You can determine how Substation ES calculates the size for data communication with the BES application (CICS or IMS): <ul style="list-style-type: none"> <li>• If you select M, the payload (TIBCO Enterprise Message Service message) size determines the data communication area size. Both the input and output data communication sizes are set by the size of the input message. The output message data size can only be equal to or less than the input message size.</li> <li>• If you select B, the data communication size is determined by the area, fields, or both in the conversion rules buffer definition. This is the existing method for Substation ES versions earlier than 2.5.</li> </ul>
<b>Conversion Id</b>	Required. The user defined name of the conversion.  Default value: New_Rule1.
<b>Message Id</b>	The name of the message definition used in the conversion rule.
<b>Buffer Id</b>	The name of the buffer definition used in the conversion rule.

## Select a Message Definition Panel

You can select a message definition for a new conversion rule in the Select a Message Definition panel.

### Panel Layout

```

SXTPO3M ----- Select a Message Definition ----- Row 1 to 1 of 1
Command ==>                                         Scroll ==> CSR

Sel Flds   Message Id
-  ----   -
      0   New-Msg1
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Select a Message Definition panel

Field	Description
<b>Sel</b>	You can select a message definition for a Msg-Field or Buffer-Field conversion.  Valid selection: S(elect).

Field	Description
<b>Flds</b>	The number of fields in the message definition.
<b>Message Id</b>	The name of the message definition.

## Select a Buffer Definition Panel

You can select a buffer definition for a new conversion rule in the Select a Buffer Definition panel.

### Panel Layout

```
SXTP03M ----- Select a Message Definition ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Sel Flds   Buffer Id
-   -   -   -
      0   New-Buffer1
***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Select a Buffer Definition panel

Field	Description
<b>Sel</b>	You can select a buffer definition for a Msg-Field or Buffer-Field conversion. Valid selection: S(elect).
<b>Flds</b>	The number of fields in the buffer definition.
<b>Buffer Id</b>	The name of the buffer definition.

## Define Msg-Field or Buffer-Field Conversion Panel

You can map the fields of a message definition with the fields of a buffer definition for a conversion rule in the Define Msg-Field / Buffer-Field Conversion panel.

### Panel Layout

```
SXTP03F ----- Define Msg-Field / Buffer-Field Conversion - Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Conversion Id   ==> New_Rule1
Using Message   ...: New-Msg1
Using Buffer     ...: New-Buffer1
Max Buffer Size ==> 0

Trunc Reqd BfFld Msg Field | Num Buffer Field
-   -   -   -   -   -   -   |   -   -   -   -   -   -   -
  Y   Y   1   String-field | 1 STRING-FIELD
***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Define Msg-Field / Buffer-Field Conversion panel:

Field	Description
<b>Conversion Id</b>	The name of the conversion rule.
<b>Using Message</b>	The name of the message definition.
<b>Using Buffer</b>	The name of the buffer definition.
<b>Max Buffer Size</b>	<p>Optional. The maximum size of the buffer (data area) for communication. Numeric value only.</p> <p>Default value: 0. When this value is 0, the size of the buffer is determined by the field definitions. When this value is not 0, this value overrides the determination from the fields. The size you specify must be greater than or equal to the field determination.</p>
<b>Trunc</b>	<p>You can specify whether to truncate data that extends beyond the specified buffer length allocated for a message field. If the inbound data is longer than the available space in the destination buffer and the truncate option is set to Y, the string is limited to the length specified in the Buffer Length field on Buffer Field Details panel. If the data exceeds the specified length and the truncate option is set to N, the conversion fails.</p> <p>Valid selections: Y(es), N(o) and S(yetem). Default selection: Y.</p> <p>S specifies that the field is a system field, in which case no mapping to the buffer field is required.</p>
<b>Reqd</b>	<p>Required. You can specify whether the field is required for a conversion.</p> <p>Valid selections: Y(es) and N(o). Default selection: Y.</p> <p>For an inbound process, if you set this field to Y, the transaction is not completed successfully unless a field with this name is encountered in the inbound Rendezvous message. If you set this field to N, no specific check is made as to the availability of that message field in the Rendezvous message.</p> <p>For an outbound process, if you set this field to N, and the data type is numeric and the contents of the field is not initialized by the host-side application, the field is initialized by the Transformer and no editing errors occur.</p>
<b>BfFld</b>	The sequence number of the buffer field that is to be mapped to the message field.
<b>Msg Field</b>	The name of the field in the message definition.
<b>Num</b>	The sequence number of the field in the buffer definition.
<b>Buffer Field</b>	The name of the field in the buffer definition.

## Define Message Fields to Conversion Rule Panel

You can define a nested message field (RVMSG) to a conversion rule in the Define Message Fields to Conversion Rule panel.

You can access this panel by selecting **M** in the **Sel** field of the Conversion Rules panel, selecting a conversion rule that contains nested message fields. For more information about the Conversion Rules panel, see [Conversion Rules Panel](#).

## Panel Layout

```

SXTPO3G ----- Define Message Fields to Conversion Rule -- Row 1 to 1 of 1
Command ==>                                         Scroll ==> CSR

    Conversion Id ...: New_Rule1
    Using Message ...: New-Msg1
Sel Trunc Reqd CvRul Msg Field (type MSG) | Num Conversion Rule
-      -      -      -      -      -      -      -      -      -
    Y      Y      0      -      -      -      -      -      -
***** Bottom of data *****

```

## Field Description

The following table lists the fields in the Define Message Fields to Conversion Rule panel:

Field	Description
<b>Conversion Id</b>	The name of the conversion rule that the nested message field is defined to.
<b>Using Message</b>	The name of the message definition containing the nested message field.
<b>Sel</b>	You can select a conversion rule that the nested message field can be defined to.  Valid selection: S(select).
<b>Trunc</b>	Not applicable for nested messages. You can ignore it.
<b>Reqd</b>	Required. You can specify whether the field is required for a conversion.  Valid selections: Y(es) and N(o). Default selection: Y.  For an inbound process, if you set this field to Y, the transaction is not completed successfully unless a field with this name is encountered in the inbound Rendezvous message. If you set this field to N, no specific check is made as to the availability of that message field in the Rendezvous message.  For an outbound process, if you set this field to N, and the data type is numeric and the contents of the field is not initialized by the host-side application, the field is initialized by the Transformer and no editing errors occur.
<b>CvRul</b>	The sequence number of the conversion rule that the nested message field is defined to.
<b>Msg Field</b>	The name of the nested message field.
<b>Num</b>	The sequence number of the conversion rule.
<b>Conversion Rule</b>	The name of the conversion rule.

## Recipes & Triggers using a Conversion Rule Panel

You can view the recipe and trigger definitions for a conversion rule in the Recipes & Triggers using a Conversion Rule panel.

You can access this panel by selecting U in the **Sel** field of the Conversion Rules panel, selecting a conversion rule that contains nested message fields. For more information about the Conversion Rules panel, see [Conversion Rules Panel](#).

## Panel Layout

```
SXTP03U ----- Recipes & Triggers using a Conversion Rule-----
Command ==>                                     Scroll ==> CSR
      Conversion Rule ==> New_Rule1
      Used in these Recipes and Triggers:
***** Bottom of data *****
```

## Field Description

The following table lists the fields in the Recipes & Triggers using a Conversion Rule panel:

Field	Description
<b>Conversion Rule</b>	The name of the conversion rule used in the recipe and trigger definition.
<b>Used in these Recipes and Triggers</b>	The name of the recipe and trigger definitions that use the conversion rule.

## Define Processing System Resources Panel

You can create or modify a Substation ES interface in the Define Processing System Resources panel.

## Panel Layout

```
SXTP04 ----- Define Processing System Resources ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR
      Type: C(ics) I(ms) A(dmin)
      (ADD=Create new Processing System, CAN=Cancel updates)
Sel Type  Identifier  Interface Id.
-----
      I      New-BES1  IIIMS
***** Bottom of data *****
```

## Field Description

The following table lists the fields in the Define Processing System Resources panel:

Field	Description
<b>Command</b>	You can enter ADD to create a new Substation ES interface, and enter CAN to cancel all modifications just done to the existing Substation ES interface.  Valid commands: <b>ADD</b> and <b>CAN</b> .
<b>Sel</b>	You can select an existing Substation ES interface to modify.  Valid selections: S(elect), D(elete), and R(epro).
<b>Type</b>	The type of interface.  Valid types: A(dmin), C(ics), and I(ms)
<b>Identifier</b>	Required. The user defined description of the Substation ES interface. The value must be unique.  Default value: New-BES1.

Field	Description
<b>Interface Id</b>	<p>Required. The name, up to eight characters, assigned to the Substation ES interface for communication with the transaction processing systems, such as CICS and IMS, which run in the z/OS environment.</p> <p>This identification must match the interface ID that is configured through the <b>INTF-ID</b> parameter in the interface system initialization parameter (SIP) member. The SIP member is located in the <i>USERHLQ</i>.INTF library.</p>

## Define ESB Endpoints Panel

You can create or modify an ESB endpoint in the Define ESB Endpoints panel.

### Panel Layout

```

SXTP07 ----- Define ESB Endpoints ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> CSR
(ADD=Create new ESB Endpoint, CAN=Cancel updates)
Sel Identifier      Service Port Daemon
-----
      New-ESB1           0      0
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Define ESB Endpoints panel:

Field	Description
<b>Command</b>	<p>You can enter <b>ADD</b> to create a new ESB endpoint, and enter <b>CAN</b> to cancel all modifications last made to an existing ESB endpoint.</p> <p>Valid commands: <b>ADD</b> and <b>CAN</b>.</p>
<b>Sel</b>	<p>You can select an existing ESB endpoint to modify.</p> <p>Valid selections: S(elect), D(elete), and R(epro).</p> <p>If you select S, you can access the panel described in <a href="#">ESB Endpoint Extensions Panel</a>.</p>
<b>Identifier</b>	<p>Required. The user defined name of the ESB endpoint.</p> <p>Default value: New-ESB1.</p>
<b>Service</b>	<p>The value for the UDP service port number that defines the service group.</p> <p>Default value: 0.</p>
<b>Port</b>	<p>The TCP port for the ESB endpoint that distinguishes both Substation ES and the role of the ESB endpoint within it.</p> <p>Default value: 0.</p>
<b>Daemon</b>	<p>The host name or IP address that Substation ES uses to find the Rendezvous daemon or TIBCO Enterprise Message Service server and to establish communication.</p> <p>Default value: a blank space.</p>

## ESB Endpoint Extensions Panel

You can create or modify ESB endpoint extensions in the ESB Endpoint Extensions panel.

### Panel Layout

```

SXTPO7D ----- ESB Endpoint Extensions -----
Command ==>                                     (ECF,SSL ADMIN)

ESB Id      ==> New-ESB1
Type        ==> E (E/R EMS, or RV)
Startup     ==> Y (Y/N Start during Substation Initialization)
Port        ==> 0
URL         ==>

Rendezvous Values:
Service     ==> 0
Network     ==> 0 0 0 0
Multicast   ==> N (N/Y)
Multicast Address ==>

EMS Server Connection Credentials:
User        ==>
Password    ==>

TCPIP Stack Name ==> (Name of TCP/IP Stack)

```

### Field Description

The following table lists the fields in the ESB Endpoint Extensions panel:

Field	Description
<b>Command</b>	<p>You can enter ECF to access the panel described in <a href="#">ESB Endpoint – Connection Factory Extensions Panel</a>, and enter SSL to access the panel described in <a href="#">ESB Endpoint SSL Extensions Panel</a>.</p> <p>Valid commands: <b>ECF</b> or <b>SSL</b>.</p> <p>These two commands are not supported when the ESB endpoint type is RV.</p>
<b>ESB Id</b>	The name of an ESB endpoint.
<b>Type</b>	<p>One or more types of ESB to be defined.</p> <p>Valid selections: E(EMS) or R(endezvous). Default selection: E.</p>
<b>Startup</b>	<p>You can specify whether to start this ESB during Substation ES startup.</p> <p>Valid selections: Y(es) and N(o). Default selection: Y.</p>
<b>URL</b>	For Rendezvous, the daemon URL; for TIBCO Enterprise Message Service, the server URL.
<b>Port</b>	<p>For Rendezvous, the daemon port; for TIBCO Enterprise Message Service, the port portion of the TIBCO Enterprise Message Service server URL.</p> <p>Default value: 0.</p>



Field	Description
<b>Service</b>	The value for the UDP service port number that defines the service group. Default value: 0.
<b>Network</b>	The service group. You can select a local network by network IP address. Default value: 0.0.0.0.
<b>Multicast</b>	You can specify whether the address is multicast enabled. Valid selections: Y(es) and N(o). Default selection: N.
<b>Multicast Address</b>	The multicast addresses if the host computer has multiple network interfaces.
<b>User</b>	The user identifier if a special identification is to be used in the session. The maximum character length is 59.
<b>Password</b>	The password associated with Conn User if a special identification is used. The maximum character length is 59.
<b>TCPIP Stack Name</b>	The IBM stack name to be used with this connection. If this TCP/IP stack name is different from the default name defined with the <b>STACK-NAME</b> parameter in the Substation SIP member, then the stack name specified here is used for this connection.

When defining the fields related to Rendezvous, note the following conditions:

- The service value is used if the value specified for **Service** is nonzero.
- The network value is used if the value specified for **Network** is nonzero. If the value specified for **Multicast** is Y, a colon symbol (:) precedes the network value.
- The URL value is used as specified. If the value specified for **Port** is nonzero, a colon symbol (:) and the port number are appended to the URL value.

When defining fields related to TIBCO Enterprise Message Service, note that the URL value is used as specified. If the value specified for **Port** is nonzero, a colon symbol (:) and the port number are appended to the URL value.

## ESB Endpoint – Connection Factory Extensions Panel

You can create or modify ESB endpoint connection factory extensions in the ESB Endpoint – Connection Factory Extensions panel.

### Panel Layout

```

SXTPO7F ----- ESB Endpoint - Connection Factory Extensions -----
Command ===>

ESB Id      ===> New-ESB1

ECF User    ===>
ECF Psd     ===>
ECF Name    ===>
Use SSL     ===> N (N/Y=Using url below and details from SSL panel)
ECF URL     ===>

ECF Metric  ===> 0 (0=None, 1=Connections, 2=Byte Rate)

Connect      Reconnect

```

Attempts	====>	9999999	====>	9999999	(0=Use Default)
Delay (ms)	====>	5000	====>	5000	
Timeout (ms)	====>	15000	====>	15000	

### Field Description

The following table lists the fields in the ESB Endpoint – Connection Factory Extensions panel:

Field	Description
<b>ESB Id</b>	The name of an ESB endpoint.
<b>ECF User</b>	The connection object presents this user identity to the server. The maximum character length is 60.
<b>ECF Pswd</b>	The password used by the connection object to authenticate the user identity. The maximum character length is 40.
<b>ECF Name</b>	The name of the connection factory. The maximum character length is 60.
<b>Use SSL</b>	You can specify whether this ESB uses an SSL connection to the server. The SSL details are setup in <a href="#">ESB Endpoint SSL Extensions Panel</a> . If you set the field to Y, Substation ES uses the SSL definitions to connect to the ECF Server using the ECF URL. The same SSL definition is used for both ECF Lookup Connection and Data Connection.  Default value: N.
<b>ECF URL</b>	The URL string that connections created by this factory is connected to.
<b>ECF Metric</b>	The load balancing metric of this connection factory. Valid selections: <ul style="list-style-type: none"> <li>• 0: absence of any load balancing metric.</li> <li>• 1: the connection factory balances the connection load among several servers by creating a connection to the server with the fewest number of connections.</li> <li>• 2: the connection factory balances the connection load among several servers by creating a connection to the server with the lowest total byte rate (input and output).</li> </ul>
<b>Attempts</b>	Connect attempts: limits the number of times that Substation ES attempts to establish a connection to the TIBCO Enterprise Message Service (EMS) server when the server is not present at Substation ES startup time.  Reconnect attempts: limits the number of times that Substation ES attempts to reestablish the connection to the EMS server.  Default value: 9999999. If 0 is specified, the value is determined by the connection factory definition, or the default setting in EMS.

Field	Description
<b>Delay (ms)</b>	<p>Connect delay: determines the time (in milliseconds) between connection attempts.</p> <p>Valid values: 250 - 20000. Default value: 500.</p> <p>Reconnect delay: determines the time (in milliseconds) between reconnection attempts.</p> <p>Valid values: 0, and 250 - 20000. Default value: 500.</p> <p>If 0 is specified, the value is determined by the connection factory definition, or the default setting in EMS.</p>
<b>Timeout (ms)</b>	<p>Connect timeout: determines the maximum time (in milliseconds) that Substation ES will wait for a connection to the server to be established.</p> <p>Valid values: 15000 - 30000. Default value: 15000.</p> <p>Reconnect timeout: determines the maximum time (in milliseconds) the Substation ES will wait for reconnection to the server to be established.</p> <p>Valid values: 0, and 15000 - 30000. Default value: 15000.</p> <p>If 0 is specified, the value is determined by the connection factory definition, or the default setting in EMS.</p>

## ESB Endpoint SSL Extensions Panel

You can create or modify ESB endpoint SSL extensions in the ESB Endpoint SSL Extensions panel.

### Panel Layout

```

SXTPO7S ----- ESB Endpoint EMS SSL Extensions -----
Command ==>

ESB Id          ==> New-ESB1
Use SSL         ==> N   (N/Y - Use SSL)
Authorization Only ==> N   (N/Y - SSL Authorization only)

Key Ring File   ==>
Key Ring Label  ==>
Cipher          ==>
LDAP URL        ==>
  LDAP USERID   ==>
  LDAP PASSWORD ==>
Enable FIPS 140-2 ==> N   (N/Y)
Enable Trace    ==> N   (N/Y)      Enable Debug Trace ==> N (N/Y)
Enable SSLV3    ==> N   (N/Y)      Enable TLS1 ==> N (N/Y)
Exp. EMS Host Name ==>
Verify Expect   ==> N   (Y/N)

```

### Field Description

The following table lists the fields in the ESB Endpoint SSL Extensions panel:

Field	Description
<b>ESB Id</b>	The name of an ESB endpoint.
<b>Use SSL</b>	<p>You can specify whether this ESB uses an SSL connection to Data Server.</p> <p>Default value: N.</p>

Field	Description
<b>Authorization Only</b>	You can specify whether this ESB connection uses SSL only for authentication. If the field is set to Y, after establishing the EMS Server's identity through SSL, the data transfer is not encrypted.  Default value: N.
<b>Key Ring File</b>	The Resource Access Control Facility (RACF) ring name.
<b>Key Ring Label</b>	The RACF identity name.
<b>Cipher</b>	The IBM cipher codes for encryption. For example: 35363738392F303132
<b>LDAP URL</b>	If LDAP is used, the URL to used to access the LDAP server.
<b>LDAP USERID</b>	If LDAP is used, the user ID used to access the LDAP server.
<b>LDAP PASSWORD</b>	If LDAP is used and requires a password, enter the password.
<b>Enable FIPS 140-2</b>	You can specify Y(es) to use FIPS 140-2, or N(o) not to use FIPS 140-2.  Default value: N.
<b>Enable Trace</b>	You can specify Y(es) to turn on SSL Trace, or N(o) to disable SSL Trace.  Default value: N.
<b>Enable Debug Trace</b>	You can specify Y(es) to turn on SSL Debug Trace, or N(o) to disable SSL Debug Trace.  Default value: N.
<b>Enable SSLV3</b>	You can specify Y(es) to turn on SSL V3, or N(o) to disable SSL V3.  Default value: N.
<b>Enable TLS1</b>	You can specify Y(es) to turn on TLS 1, or N(o) to disable TLS 1.  Default value: N.
<b>Exp. EMS Host Name</b>	The name of the TIBCO Enterprise Message Service server name that is being interfaced with.
<b>Verify Expect</b>	If you specify Y(es), the name of the TIBCO Enterprise Message Service server is matched against the value specified in the <b>Exp. EMS Host Name</b> field.

## ESB Administration Extensions Panel

You can create or modify ESB administration connection extensions in the ESB Administration Extensions panel.

In the ESB Administration Extensions panel, you can set up an administrative connection for this TIBCO Enterprise Message Service (EMS) ESB. You can also use this panel to get EMS server statistics and information.

## Panel Layout

```
SXTP07E ----- ESB Administration Extensions -----
Command ==>

  ESB Id      ==> New-ESB1

  Setup Admin ==> N (N/Y Create Admin connection)
  User        ==>
  Password    ==>
```

## Field Description

The following table lists the fields in the ESB Administration Extensions panel:

Field	Description
<b>ESB Id</b>	The name of an ESB endpoint.
<b>Startup Admin</b>	You can specify whether to create this ESB Admin connection during Substation ES startup.  Valid selections: Y(es) and N(o). Default selection: N.
<b>User</b>	The user identifier that has EMS server administrator privileges. The maximum character length is 16.
<b>Password</b>	The password associated with the user. The maximum character length is 16.



If the ESB Endpoint EMS SSL Extensions panel is used to define an SSL connection for data, the same SSL parameters are used for the administrative connection. For more information about this panel, see [ESB Endpoint SSL Extensions Panel](#).

## Recipe List Panel

You can create or modify a recipe in the Recipe List panel.

## Panel Layout

```
SXTP05 ----- Recipe List ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> CSR

  (ADD=Create new Recipe, CAN=Cancel updates)

Sel Resourc  MOI  Reply  Processor      Identification
-----
           2    2      New-BES1      New-Recipe1
***** Bottom of data *****
```

## Field Description

The following table lists the fields in the Recipe List panel:

Field	Description
<b>Command</b>	You can enter ADD to create a new recipe, and enter CAN to cancel all modifications just done to the existing recipe.  Valid commands: <b>ADD</b> and <b>CAN</b> .

Field	Description
<b>Sel</b>	<p>You can select an existing recipe to modify.</p> <p>Valid selections: S(elect), D(elete), R(epro), and E(MS).</p> <p>If you select E, you can access the panel described in <a href="#">EMS Recipe Details Extension Panel</a>.</p>
<b>Resource</b>	<p>The name of the resource. For details, see the <b>Resource Name</b> field on the <a href="#">Recipe Details Panel</a>.</p>
<b>MOI</b>	<p>The method of invocation that initiates the configured transformation.</p> <p>The following are valid selections:</p> <ul style="list-style-type: none"> <li>• 2 DPL</li> <li>• 3 Program</li> <li>• 4 Task/IMS Tran</li> <li>• 5 TDQ Write</li> <li>• 6 TSQ Write</li> <li>• 7 IMS Conversational Tran</li> <li>• 8 Command</li> <li>• 9 DPL MRO</li> <li>• 10 Container</li> <li>• 11 Storage</li> <li>• 12 DPL TSQ</li> </ul>
<b>Reply</b>	<p>The type of data expected as output from the transaction processing application. For details, see the <b>Reply Method</b> field in the <a href="#">Recipe Details Panel</a>.</p>
<b>Processor</b>	<p>The description of the Substation ES interface specified in the <b>Identifier</b> field of the <a href="#">Define Processing System Resources Panel</a>.</p>
<b>Identification</b>	<p>Required. The user defined name of the recipe.</p> <p>Default value: New-Recipe1.</p>

## Select ESB Endpoint Panel (Recipe)

You can select an ESB endpoint for a recipe in the Select ESB Endpoint panel.

### Panel Layout

```

SXTPO5T ----- Select ESB Endpoint ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

(Use S to select an ESB Endpoint)
Sel Service Port  Identifier      Daemon      Network
-----
      0      0      New-ESB1
***** Bottom of data *****

```

## Field Description

The following table lists the fields in the Select ESB Endpoint panel:

Field	Description
<b>Sel</b>	You can select the ESB endpoint for a recipe, one ESB per recipe. Valid selection: s(elect).
<b>Service</b>	The value for the UDP service port number that defines the service group.
<b>Port</b>	For Rendezvous, the daemon port; for TIBCO Enterprise Message Service, the port portion of the EMS server URL.
<b>Identifier</b>	The name of an ESB endpoint to be used in the recipe.
<b>Daemon</b>	The host name or IP address used by Substation ES to find the Rendezvous daemon or TIBCO Enterprise Message Service server and to establish communication.
<b>Network</b>	The service group selected by network IP address or multicast addresses.

## Select Processing System Resources Panel (Recipe)

You can select a processing system resource for a recipe in the Select Processing System Resources panel.

### Panel Layout

```
SXTP05B ----- Select Processing System Resources ----- Row 1 to 1 of 1
Command ==>                                         Scroll ==> CSR

      Use S to select a Processing System
Sel Type  Identifier      Interface Id.
-----
  I      New-BES1        IIIMS
***** Bottom of data *****
```

## Field Description

The following table lists the fields in the Select Processing System Resources panel:

Field	Description
<b>Sel</b>	You can select the processing system resource for a recipe. Valid selection: s(elect).
<b>Type</b>	The type of interface. Valid types: A(dmin), C(ics), and I(ms).
<b>Identifier</b>	The description of the Substation ES interface.
<b>Interface Id.</b>	The name assigned to Substation ES interface to communicate with the transaction processing systems.

## Select Conversion Rules Panel (Recipe)

You can select a conversion rule for a recipe in the Select Conversion Rules panel.

### Panel Layout

```
SXTP05C ----- Select Conversion Rules ----- Row 1 to 6 of 6
Command ==>                                     Scroll ==> CSR
(I for input converter, O for output, B for both)
Max
Sel Flds Bufsz Conversion Id      Message Id      Buffer Id
-----
   1 0    CVR-I-ADMIN      MSG-IN-ADMIN    BUF-IO-ADMIN
  13 0    CVR-I-TSRR-01    MSG-IO-RR-01    BUF-IO-BF01
  12 0    CVR-I-01         MSG-IN-01       BUF-IO-BF01
  20 0    CVR-I-02         MSG-IN-02       BUF-IO-BF02
   4 0    CVR-IO-BYTE      MSG-JMS-BYTE    BUF-BYTES
   1 0    CVR-IO-STRING    MSG-IO-TEXT     BUF-STRING
***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Select Conversion Rules panel:

Field	Description
<b>Sel</b>	You can select the conversion rule for a recipe. Valid selections: I(nput), O(output), and B(oth).
<b>Flds</b>	The number of fields in the conversion rule.
<b>Max Bufsz</b>	The maximum buffer size to be used for the conversion rule.
<b>Conversion Id</b>	The name of the conversion rule to be used in the recipe.
<b>Message Id</b>	The name of the message definition used in the conversion rule.
<b>Buffer Id</b>	The name of the buffer definition used in the conversion rule.

## Recipe Details Panel

You can create or modify the details of a recipe in the Recipe Details panel.

### Panel Layout

```
SXTP05N ----- Recipe Details -----
Command ==>                                     (CICS,IMS,EMS,PROP,SEL)

Recipe Identifier    ==> SXC-DPL-C
Listen To           ==> tibss.CICS.C.Request
Reply To            ==>
Reply To Always      ==> Y
Error Subject        ==> tibss.error.out
Error Threshold      ==> 0      (0 - 99999)      Error Percentage ==> 0
Reply To Len (Dyn)   ==> 0      Start ==> 0      Usage Limit ==> 0
Input Conversion (?) ==> CVR-I-01      (SEE CICS OPT FOR NO CONVERSION)
Output Conversion(?) ==> CVR-O-01
Back End System (?)  ==> Interface-CICS1  Alt BES(?): Interface-CICS2
ESB Endpoint (?)     ==> EMS-SERVER      SS-Id Filter: N (N or Y)
Resource Name        ==> SXC3I001
Method of Invocation ==> 2      2 DPL  3 PGM   9 DPL MRO  12 DPL TSQ
DPL > 32K Invocations 10 Container  11 Storage
```



```


                    5 TDQ Write      6 TSQ Write 8 Command
                    4 Task/IMS Tran 7 IMS Conversational Tran
Trace Level ==> 0 0-5 (Debugging)
Reply Method ==> 3 0 None          2 Message
                  1 Status        3 Both Status and Message

```

## Field Description

The following table lists the fields in the Recipe Details panel:

Field	Description
<b>Command</b>	<p>The commands used to access panels:</p> <p><b>CICS</b>: enter CICS to access the panel described in <a href="#">CICS Service Details Panel</a>.</p> <p><b>IMS</b>: enter IMS to access the panel described in <a href="#">IMS Service Details Panel</a>.</p> <p><b>PROP</b>: enter PROP to access the panel described in <a href="#">EMS Properties (Recipe) Panel</a>.</p> <p><b>SEL</b>: enter SEL to access the panel described in <a href="#">EMS Selector Specification Panel</a>.</p>
<b>Recipe Identifier</b>	The name of the recipe.
<b>Listen To</b>	Required. When configuring Substation ES to subscribe to data from a TIBCO messaging application, this field specifies the subject that Substation ES listens to invoke a given transformation for.
<b>Reply To</b>	The outbound subject that a message is published on by a transaction process. See the <b>Reply To Always</b> field description for information on the rules regarding published messages.

Field	Description
<b>Reply To Always</b>	<p>You can determine whether a response from a transaction process is sent by Substation ES only to the reply ID specified in a message, to the value specified in the <b>Reply To</b> field (the outbound destination name in TIBCO Enterprise Message Service), or to both.</p> <p>The reply ID in a message is set by the application that issues the request. The reply ID is extracted from the message by Substation ES and the response relating to that request could be delivered to the reply ID, depending on the setting of the <b>Reply To Always</b> field.</p> <p>Valid selections: Y(es) and N(o). Default selection: Y.</p> <p>If you select Y, the response is sent as follows:</p> <ul style="list-style-type: none"> <li>• To the reply ID in the message: if the reply ID is specified and the Reply To subject or Destination is not specified.</li> <li>• To the Reply To subject or Destination: if the Reply To subject or Destination is specified and the reply ID is not specified.</li> <li>• To both the reply ID in the message and Reply To subject or Destination: if both are specified.</li> </ul> <p>If you select N, the response is sent as follows:</p> <ul style="list-style-type: none"> <li>• To only the reply ID in the message: if the reply ID is specified and the Reply To subject or Destination is or is not specified.</li> <li>• To the Reply To subject or Destination: if the Reply To subject or Destination is specified and the reply ID is not specified.</li> <li>• Not sent: if both have not been specified.</li> </ul>
<b>Error Subject</b>	<p>The subject that an error message is published on if Transformer conversion or BES encounters errors.</p> <div>  <p>If you use TIBCO Enterprise Message Service as your ESB, you can now select the TIBCO Enterprise Message Service message type of the error message. See the <b>ERROR-MSG-TYPE</b> keyword in <i>TIBCO Substation ES Installation</i>.</p> </div>
<b>Error Threshold</b>	The number of errors allowed before the recipe is disabled.
<b>Error Percentage</b>	The percentage of errors allowed before the recipe is disabled.
<b>Reply To Len (Dyn)</b>	<p>The value in this field allocates the outbound reply subject dynamically within the buffer. You can specify the number of bytes to be used for the dynamic Reply To subject length.</p> <p>The value of this field must be greater than zero for the dynamic Reply To to work. When 0 (zero) is specified, the feature is disabled.</p> <p>The area specified by <b>Reply To Len (Dyn)</b> and <b>Start</b> must be in the buffer, or an error is issued.</p> <p>Default value: 0.</p>

Field	Description
<b>Start</b>	<p>The starting offset in the buffer to be used for the dynamic subject name.</p> <p>The area defined by <b>Reply To Len (Dyn)</b> and <b>Start</b> is concatenated to form the Reply To subject. This is the name that the message is published to. If this dynamic Reply To name matches the name of either the Reply To subject in the message or the error subject, then only one message is written to the subject. The dynamic Reply To message must adhere to TIBCO Enterprise Message Service or Rendezvous subject structure, or an error is issued by either messaging service when the message is published.</p> <p>By default, the first character in the buffer is location 0.</p>
<b>Usage Limit</b>	<p>The maximum number of concurrent messages to be processed by an ESB endpoint for this recipe. This parameter is only valid with TIBCO Enterprise Message Service transports.</p> <p>Default value: 0.</p>
<b>Input Conversion (?)</b>	<p>The name of the conversion rule used for input conversion.</p> <p>If you specify ?, a list of all conversion rules is displayed where you can select a conversion rule.</p>
<b>Back End System (?)</b>	<p>The name of the conversion rule used for output conversion.</p> <p>If you specify ?, a list of all interfaces is displayed where you can select an interface.</p>
<b>Alt BES(?)</b>	<p>Required. The description of the Substation ES interface.</p> <p>If you specify ?, a list of all interfaces is displayed where you can select an interface.</p>
<b>ESB Endpoint (?)</b>	<p>Optional. The description of an alternate Substation ES interface.</p> <p>If you specify ?, a list of all interfaces is displayed where you can select an interface. If the specified BES is not available during the process, then the messages attempt to process on this BES.</p>
<b>SS-Id Filter</b>	<p>The SS-Id Filter applies to TIBCO Enterprise Message Service (EMS) and Rendezvous (RV) ESB. With this feature, EMS TOPIC type destinations and RV subjects can be filtered by the <code>tibss-ss-id</code> value so that, while multiple Substation ES might receive the message, only one processes the request.</p> <p>Default value: N (no).</p> <p>If you specify Y, then each message on this recipe is checked for the EMS property or RV field <code>tibss-ss-id</code>.</p> <ul style="list-style-type: none"> <li>• If the field is not present, then Substation ES processes the incoming message.</li> <li>• If the field is present, the <code>tibss-ss-id</code> value is compared to the Substation ES ID. Only those messages that match are processed by Substation ES.</li> </ul> <p>For Admin recipes (MOI 8) with topic type destinations, each message is forced to be checked.</p>

Field	Description
<b>Resource Name</b>	<p>The name of the resource. If you are configuring Substation ES to subscribe to data on behalf of a transaction processing application, this is the name of the transaction or program that is invoked upon receipt of a request.</p> <p>If your MOI is TDQ Write or TSQ Write, this field specifies the name of the Transient Data Queue (TDQ) or Temporary Storage Queue (TSQ). The length of the TDQ name can be four characters or less. The maximum length of the TSQ name is eight characters.</p>

Field	Description
<b>Method of Invocation</b>	<p>The method that initiates the transformation that is being configured.</p> <p>If you are configuring Substation ES to publish data on behalf of a CICS or IMS transaction processing application, the sole choice is the trigger method of invocation (MOI).</p> <p>If you are configuring Substation ES to subscribe to data generated by a TIBCO messaging application, you can choose one or more of the other MOIs.</p> <p>The following are valid selections:</p> <ul style="list-style-type: none"> <li>• 2 DPL: the transaction invoked with the Dynamic Program Link (DPL) MOI. By using this MOI, a response or return data can be sent back to the calling TIBCO messaging application.</li> <li>• 3 PGM: the transaction invoked with the Transfer Control (XCTL) mechanism of CICS. Only a status return is available with this option.</li> <li>• 4 Task/IMS Tran: invokes a CICS task or transaction (four-character resource name) as opposed to a CICS program. Can be a status return when this method is in use.</li> </ul> <p>Alternatively, invokes an IMS transaction to be processed by the BES so that a response or return data can be sent back to the calling TIBCO messaging application.</p> <ul style="list-style-type: none"> <li>• 5 TDQ Write: the transaction invoked by a write to the TDQ from which the transaction is already polling. The length of the TDQ name can be four characters or less.</li> <li>• 6 TSQ Write: the transaction invoked by a write to TSQ. The maximum length of the TSQ name is eight characters.</li> <li>• 7 IMS Conversational Tran: an IMS conversational transaction is invoked for processing.</li> <li>• 8 Command: invokes a Substation ES administration task for processing operation commands, Substation ES Console commands, or IMS commands that form a TIBCO message.</li> <li>• 9 DPL MRO: the transaction invoked through CICS LINK with a COMMAREA. Useful for MRO routing.</li> <li>• 10 Container: the transaction invoked through CICS LINK with a Container for input and output. The Container name is the same as the program name. Useful for MRO routing and greater than 32 KB input and output.</li> <li>• 11 Storage: the transaction invoked through CICS LINK with a communication area (COMMAREA). The COMMAREA contains a pointer to a storage area and its length. Useful for greater than 32 K input and output.</li> <li>• 12 DPL TSQ: the transaction invoked through CICS LINK with a COMMAREA, which contains an eight-character TSQ ID, which in turn contains the input data and which can also be used for output.</li> </ul> <p>For more information about the use of MOIs, see <a href="#">MOI Usage</a>.</p>

Field	Description
<b>Trace Level</b>	<p>The user trace debug level for an individual recipe. The value must always be 0 or 1, unless TIBCO Support requires the output. A certain amount of overhead and many lines of output are produced when this value is greater than 1.</p> <p>Valid values: 0 - 5. Default value: 0.</p> <p>The following are valid values and their descriptions:</p> <ul style="list-style-type: none"> <li>1: shows message entry into Substation ES.  <pre>SXT5468I U-Trace(R) UoW:1584586364 Recipe:DPL-BYTES to BES:CICSA, Subject:tibss.CICS.Bytes.Request</pre> </li> <li>2: shows message entry and interface flow.  <pre>SXT5468I U-Trace(R) UoW:1584586364 Recipe:DPL-BYTES to BES:CICSA, Subject:tibss.CICS.Bytes.Request SXT5465I U-Trace(I) UoW:1584586364 Recipe:DPL-BYTES BES:CICSA, Subject:tibss.CICS.Bytes.Request SXG1885I IIId:CICSA U-Trc(I) UoW:1584586364 SXG1887I IIId:CICSA U-Trc(O) UoW:1584586364 SXT5303I U-Trace(O) UoW:1584586364 Rcp/Trig:DPL-BYTES, Respond:NULL, Reply:tibss.CICS.Bytes.Reply</pre> </li> <li>3 and above: shows all of level 2 plus data buffers.</li> </ul>
<b>Reply Method</b>	<p>You can specify whether data is expected to be output from the transaction processing application. If configuring Substation ES to subscribe to data generated by a TIBCO messaging application using the DPL MOI, you can specify whether return data or a status is expected back. If configuring Substation ES to publish data for a transaction processing application, you can use this option to indicate whether data and a status is being generated.</p> <p>The following are valid selections:</p> <ul style="list-style-type: none"> <li>0 None: no reply method applies for this recipe.</li> <li>1 Status: a status message is sent to the requesting application through the reply subject name, and the subject is specified in the <b>Reply To</b> field.</li> <li>2 Message: the return data is to be transformed and published as a message to the TIBCO messaging application.</li> <li>3 Both Status and Message: for this process, status and return data set are to be transformed and published as a message to the TIBCO messaging application.</li> </ul>

## MOI Usage

The table lists the method of invocation (MOI) used to initiate the transformation.

MOI	Requirements	Interface	Inbound Destination
CICS			

MOI	Requirements	Interface	Inbound Destination
2	Resource Name; 1 - 32000 bytes in size	EXCI	User program
3, 4, 5, 6, 9	Resource Name; 1 - 32000 bytes in size	EXCI	DPL server (SXCOSRVD)
10, 11, 12	Resource Name; 1 - 4 MB in size	EXCI	DPL server for > 32000 (SXCOSRVG)
2	User Tran = Y; Mirror Tran present; 1 - 32000 bytes in size	CSMI	User transaction
<b>IMS</b>			
4, 7, 8	Resource Name; 1 - 4 MB in size	OMTA XCF	User transaction

## CICS Service Details Panel

You can create or modify CICS service values for a recipe in the CICS Service Details panel.

You can access this panel by entering CICS in the Recipe Details panel. For details, see [Recipe Details Panel](#).

### Panel Layout

```

SXTPO5I ----- CICS Service Details -----
Recipe Identifier      ===> SXC-DPL-C-TSQ
Resource Name         ===> SXC3I012
Method of Invocation   ===> 12 Method of Reply ===> 3

Mirror Transaction     ===>      Default SXEX
CSMI User Transaction  ===> N      N/Y

Container (CTN) Extended Options
Channel Name          ===>
Single CTN Name       ===>
Container Invocation   ===> P P/T - Program / Transaction
Container Process      ===> S S/M Single CTN, Message
FWD Destination Names ===> N N/Y as Containers

Remote Region Execution
Transaction Id        ===>
System Name           ===>

Multi-Execution
  Resource 2:          3:          4:          5:

```

### Field Description

The following table lists the fields in the CICS Service Details panel:

Field	Description
<b>Recipe Identifier</b>	The name of the recipe.
<b>Resource Name</b>	The name of the resource.

Field	Description
<b>Method of Invocation</b>	The method used to initiate the transformation that is being configured.
<b>Method of Reply</b>	This field indicates whether data is expected to be output from the transaction processing the application.
<b>Mirror Transaction</b>	<p>A CICS-specific field used to invoke different transactions for Substation ES to CICS communications during run time. The following are typical usages of the mirror transaction:</p> <ul style="list-style-type: none"> <li>• Substation ES uses transaction SXEX by default that passes data by way of the COMMAREA to programs compiled with AMODE(ANY). In this way, data that is passed to CICS programs can use the desired method CICS extended storage.</li> <li>• Substation ES uses transaction SXBX by default that passes data by way of the COMMAREA to programs compiled with AMODE(24). Transaction SXBX has been supplied during Substation ES installation so that older programs compiled with AMODE(24) can access data below the 16 M storage line.</li> <li>• If different security options for transactions are required, you can specify a mirror transaction per invocation of the CICS resource without complicating RACF/ACF2 definitions.</li> </ul>
<b>CSMI User Transaction</b>	<p>You can specify whether this is a CSMI user transaction.</p> <p>Default selection: N (no).</p>
<b>Channel Name</b>	<p>The name of the channel to be set when container data input and output is used.</p> <p>Default value: SXCOSRVGCHANNEL.</p>
<b>Single CTN Name</b>	<p>The name of the container to be set when single container input and output is used.</p> <p>The default value is the resource name.</p>
<b>Container Invocation</b>	<p>The method of container invocation:</p> <ul style="list-style-type: none"> <li>• P: for a program link.</li> <li>• T: to start a transaction in CICS.</li> </ul> <p>Default value: P.</p>
<b>Container Process</b>	<p>The type of container:</p> <ul style="list-style-type: none"> <li>• S: for single container</li> <li>• M: for multiple containers, driven by message fields. The number of containers is determined by message fields.</li> </ul> <p>Default value: S</p>



Field	Description
<b>FWD Destination Names</b>	If this parameter is enabled and the process has multiple containers without conversion rules, then Substation ES creates two additional containers named tibss-ListenTo and tibss-ReplyTo, and sends them to CICS. These containers contain the corresponding destination or subject names of this service.
<b>Transaction Id</b>	<p>A four-character transaction identifier responsible for running the program on the remote CICS system.</p> <p>Only the following MOIs are supported:</p> <ul style="list-style-type: none"> <li>• 9 DPL MRO</li> <li>• 10 Container</li> <li>• 12 DPL TSQ</li> </ul>
<b>System Name</b>	<p>A four-character system identifier for the CICS region where the program will be run. This is referred to as the SYSID in CICS terms.</p> <p>Only the following MOIs are supported:</p> <ul style="list-style-type: none"> <li>• 9 DPL MRO</li> <li>• 10 Container</li> <li>• 12 DPL TSQ</li> </ul>
<b>Resource N</b>	You can specify two to five additional resources. Each of these fields can indicate an additional resource name for special CICS execution.

## IMS Service Details Panel

You can create or modify IMS service values for a recipe in the IMS Service Details panel.

You can access this panel by entering IMS in the Recipe Details panel. For details, see [Recipe Details Panel](#).

### Panel Layout

```

SXTPO5J ----- IMS Service Details -----
Command ==>

Recipe Identifier      ==> SXI-TRAN
Resource Name         ==> SXICIT01
Method of Invocation   ==> 4

Variable Length Tran   ==> N   (N/Y)      Transaction Length ==> 0 (0 - 8)

Segment Break (8 hex) ==> 00000000      (No 0x00 or 0x40)
MFS Map Name          ==>
Natural Stack Cmd      ==>
SAF Checking           ==> N             (N)None (U)User (F)Full
Segment Length         ==> 0             0 - 32524

```

### Field Description

The following table lists the fields in the IMS Service Details panel:

Field	Description
<b>Recipe Identifier</b>	The name of the recipe.
<b>Resource Name</b>	The name of the resource.
<b>Method of Invocation</b>	The method used to initiate the transformation that is being configured.
<b>Variable Length Tran</b>	This field indicates whether the IMS transaction ID provided in <b>Resource Name</b> is less than eight characters.  Valid selections: Y(es) or N(o). Default selection: N.
<b>Transaction Length</b>	This field value indicates the length of the Transaction Field that IMS is using.
<b>Segment Break</b>	The eight hex characters that represent the segment separator in the message.
<b>MFS Map Name</b>	The MFS map name (1 - 8 characters) to use.
<b>Natural Stack Cmd</b>	You can use this field to input a natural command (1 - 29 characters) that must be passed from a recipe to the OTMS interface.
<b>SAF Checking</b>	The IMS recipe service SAF check indicator. This field indicates the type of SAF check performed by the OTMA interface. Valid values are: <ul style="list-style-type: none"> <li>• N: none, or no SAF check.</li> <li>• U: user ID is checked.</li> <li>• F: full user ID and password check.</li> </ul>
<b>Segment Length</b>	The segment length used to segment request messages. If the value is 0, the length used is the <b>BUFFLEN-SRB</b> value, which is defined on the IMS SIP member.  Valid values: 0 - 32524.

## EMS Recipe Details Extension Panel

You can create or modify TIBCO Enterprise Message Service values for a recipe in the EMS Recipe Details Extension panel.

You can access this panel in two ways:

- By selecting E in the **Sel** field in of the Recipe List panel. For details, see [Recipe List Panel](#).
- By entering EMS in the Recipe Details panel, for details, see [Recipe Details Panel](#).

### Panel Layout

```
SXTP05E----- EMS Recipe Details Extension -----
Command ==>
```

```
Recipe Identifier    ==> New-Recipe1
Input Dest Type     ==> T Name: EMS
Output Dest Type    ==> T Name:
Error Dest Type     ==> Q Name: tibss.error.ivp.out
```

```

Durable Subscriber    ==> N  (N/Y)
Durable Client Id    ==>


                                (Service Levels G=Guaranteed, R=Reliable)
SMSL Mode             ==> RA  (RA Auto-Ack, RN No-Ack, GA All, GS SS Recv)
Delivery Mode         ==> P  (P Persistent, N Non-Persistent, R Reliable)


Output Message properties:
Priority               ==> N  (N/Y) Value ==> 4      (0 - 9)
Expiration             ==> N  (N/Y) Value ==> 0      Unit ==> S (S,M,H,D)
Compression            ==> N  (N/Y)
Preserve               ==> N  (N/Y, undelivered messages)
Body Trace             ==> N  (N/Y)
Message Type          ==> M  M(ap)/S(tream)/T(ext)/B(ytes)
Copy JMSCorrelation   ==> N  N(o)/Y(es)/E(nvelope)/M(sgId)/W(CorId or MsgId)

```

## Field Description

The following table lists the fields in the EMS Recipe Details Extension panel:

Field	Description
<b>Recipe Identifier</b>	The name of the recipe.
<b>Input Dest Type</b>	The input destination type. Valid selections: T(opic) and Q(ueue). Default selection: T.
<b>Input Dest Name</b>	The destination that Substation ES listens to invoke a given transformation for.
<b>Output Dest Type</b>	The output destination type. Valid selections: T(opic) and Q(ueue). Default selection: T.  When the output destination type is set to T, the SMSL mode must be either RA (Auto-Ack) or RN (No-Ack).
<b>Output Dest Name</b>	The outbound destination that a message is published on for a given transaction process. See the <b>Publish Always</b> field description in the Recipe Details panel for information on the rules regarding published messages.
<b>Error Dest Type</b>	The error destination type. Valid selections: T(opic) and Q(ueue). Default selection: Q.
<b>Error Dest Name</b>	The destination where an error message is published when Transformer conversion or BES encounter errors.
<b>Durable Subscriber</b>	You can specify whether to connect the session as a durable subscriber. Once this has been turned on, Substation ES does not destroy the subscriber. To destroy a durable subscriber, you must use the TIBCO Enterprise Message Service administration tool to destroy the subscriber manually from the TIBCO Enterprise Message Service server. Valid selections: N(o) and Y(es). Default selection: N.
<b>Durable Client Id</b>	The client identifier in a durable connection. If you do not specify a value, the recipe name is used.

Field	Description
<b>SMSL Mode</b>	<p>The SMSL mode for TIBCO Enterprise Message Service (EMS).</p> <p>The following are valid selections. Default selection: RA.</p> <ul style="list-style-type: none"> <li>• RA (Auto Ack) : the EMS client library automatically confirms the receipt of a message. The Substation ES service level for Auto Ack is Reliable. Note that Substation ES cannot determine if the receipt succeeded in the event of a network outage or the loss of connectivity.</li> <li>• RN (No-Ack) : the EMS client library does not send a receipt of the message that was received. The Substation ES service level for No-Ack is Reliable, no acknowledgement. The server destroys the message after it has been sent.</li> <li>• GA (GA All): Substation ES calls the EMS client library and explicitly confirms receipt of a message only once after all the Substation ES processes and executions controlled by Substation ES for a given message are completed successfully. The Substation ES service level for GA All is Guaranteed.</li> <li>• GS (SS Recv): Substation ES calls the EMS client library and explicitly confirms receipt of a message after the Substation ES processes are executed successfully. Processing is considered complete when a response message is successfully delivered to its destination. This is regardless of types of message that are delivered back by Substation ES, error, or application type messages. The Substation ES service level for SS-Recv is Guaranteed Substation Processing.</li> </ul> <div>  <p>When the SMSL mode is set to GA or GS, the output destination type must be set to Q(ueue) and the delivery mode must be set to P(ersistent).</p> </div>
<b>Delivery Mode</b>	<p>The delivery mode for TIBCO Enterprise Message Service.</p> <p>Valid modes: P(ersistent), N(on-Persistent) and R(eliable). Default mode: P.</p> <p>When the delivery mode is set to N or R , the SMSL mode must be set to either RA (Auto-Ack) or RN (No-Ack).</p> <p>When the delivery mode is set to P:</p> <ul style="list-style-type: none"> <li>• For topics, the SMSL mode must be either RA (Auto-Ack) or RN (No-Ack).</li> <li>• For queues, no restrictions are set on the SMSL mode setting.</li> </ul>
<b>Output Message properties</b>	<p>The output message properties.</p> <p>For additional information, see <i>TIBCO Enterprise Message Service User's Guide</i>.</p>
<b>Priority</b>	<p>You can set the TIBCO Enterprise Message Service message priority to the value specified in the <b>Priority Value</b> field.</p> <p>Valid selections: N(o) and Y(es). Default selection: N.</p>
<b>Priority Value</b>	<p>The TIBCO Enterprise Message Service message priority value to be used.</p> <p>Valid values: 0 – 9. Default value: 4.</p>

Field	Description
<b>Expiration</b>	<p>You can set the TIBCO Enterprise Message Service message expiration to the value specified in the <b>Expiration Value</b> field.</p> <p>Valid selections: N(o) and Y(es). Default selection: N.</p>
<b>Expiration Value</b>	<p>The TIBCO Enterprise Message Service message expiration value to be used.</p> <p>Valid values: 0 - 99999. Default value: 0.</p>
<b>Expiration Unit</b>	<p>The TIBCO Enterprise Message Service message expiration unit to be used.</p> <p>Valid selections: S(econd), M(inute), H(our), and D(ay). Default selection: S.</p>
<b>Compression</b>	<p>You can specify whether to compress the TIBCO Enterprise Message Service message for storage on the server.</p> <p>Valid selections: N(o) and Y(es). Default selection: N.</p>
<b>Preserve</b>	<p>If a message is removed from its queue by the TIBCO Enterprise Message Service server for reasons other than being consumed, the server checks this message property. If this value is set to Y, the message is placed on the server undelivered message queue.</p> <p>Valid selections: N(o) and Y(es). Default selection: N.</p>
<b>Body Trace</b>	<p>Tracing for this destination generates trace messages that include the message body.</p> <p>Valid selections: N(o) and Y(es). Default selection: N.</p>
<b>Message Type</b>	<p>The type of TIBCO Enterprise Message Service output message.</p> <p>Valid selections: M(ap), S(tream), T(ext), and B(ytes). Default: M.</p>
<b>Copy JMSCorrelation</b>	<p>Copies the JMSCorrelationID value from an inbound message and put it in the output message.</p> <p>Valid selections: N(o), Y(es), E(nvelope), M(sgId), and w(CorId or MsgId). Default selection: N.</p> <p>If Y is selected, Substation ES uses the original JMSCorrelationID value from the inbound message.</p> <p>If N is selected, Substation ES performs no action.</p> <p>If E is selected, Substation ES sets the JMSCorrelationID to the Hex String Envelope on the output message.</p> <p>If M is selected, Substation ES sets the JMSCorrelationID to the Recipe Incoming Message ID.</p> <p>If w is selected, Substation ES sets the JMSCorrelationID depending on whether a JMSCorrelationID is present in the inbound message:</p> <ul style="list-style-type: none"> <li>• If a JMSCorrelationID is present in the incoming message, Substation ES uses the original value of that field for the outbound message.</li> <li>• If no JMSCorrelationID is present, Substation ES sets JMSCorrelationID to the Recipe Incoming Message ID.</li> </ul>

## EMS Properties (Recipe) Panel

You can specify EMS properties to be inserted in output messages or map the EMS properties fields from the input message to the buffer fields in the EMS Properties (Recipe) panel.

You can specify up to 15 properties, including the JMSCorrelationID header field. For more information on how to set up Substation ES system fields as property fields, see [System Fields Used as Property Fields](#).

You can access this panel by entering PROP in the Recipe Details panel. For details, see [Recipe Details Panel](#).

### Panel Layout

```

SXTP05PR ----- EMS Properties (Recipe) ----- Row 1 to 11 of 15
Command ==>                                     Scroll ==> CSR

      Output Buffer Id: Out-Buffer
      Input Buffer Id: In-Buffer
Property Name      B/E/V Req Value (100 char)
-----
Property-Out-1      B   _   PGM-ID
Property-Out-2      V   _   This is the property field
Property-In          E   Y   HOST-NAME

***** Bottom of data *****

```

### Field Description

The following table lists the fields in the EMS Properties (Recipe) panel:

Field	Description
<b>Output Buffer ID</b>	If you select property values from a buffer, this is the buffer name associated with an output conversion rule from which a selection can be made.
<b>Input Buffer ID</b>	If you map a property value to a buffer, this is the buffer name associated with an input conversion rule from which a selection can be made.
<b>Property Name</b>	The name of the property.
<b>B/E/V</b>	<p>You can specify where the property value is from:</p> <ul style="list-style-type: none"> <li>B: the property value is selected from a buffer field.</li> <li>E: the property value is selected from an ESB input message field.</li> <li>v: the property value is defined by the user in the <b>Value</b> field.</li> </ul> <p>Valid selections: B(uffer), E(SB) or V(alue).</p>

Field	Description
<b>Req</b>	<p>This field indicates whether this property field is mandatory in the request message:</p> <ul style="list-style-type: none"> <li>Y: required.</li> <li>N: optional.</li> </ul> <p>The default value depends on the setting of the <b>B/E/V</b> field:</p> <ul style="list-style-type: none"> <li>If the property field value is from the B(uffer) field, the default is Y(es).</li> <li>If the property field value is from the E(SB) incoming request message field, the default is N(o).</li> <li>If the property field value is from the user input v(alue), the default is Y(es).</li> </ul>
<b>Value</b>	<p>If the property value is selected from a buffer field by specifying B in the <b>B/E/V</b> field, enter the question mark (?) and then press Enter. A list of buffer field names is displayed in the buffer where you can select one.</p> <p>If the property value is user defined by specifying v in the <b>B/E/V</b> field, enter the desired value.</p>

## EMS Selector Specification Panel

You can specify a TIBCO Enterprise Message Service message selector in the EMS Selector Specification panel.

You can access this panel by entering SEL in the Recipe Details panel. For details, see [Recipe Details Panel](#)

### Panel Layout

```
SXTP05S----- EMS Selector Specification -----
Command ==>


Selector ==>

(End of Selector)
```

### Field Description

The following table lists the fields in the EMS Selector Specification panel:

Field	Description
<b>Command</b>	Currently not used.

Field	Description
<b>Selector</b>	<p>A free-form field for specifying a selector statement. A message selector is a string that contains an expression. The syntax of the expression is based on a subset of the SQL92 conditional expression syntax.</p> <p>A message selector specifies a set of messages based on the values of message headers and properties. A selector matches a message if, after substituting header and property values from the message into the selector string, the string evaluates to true.</p> <div>  <p>For information on the message selector syntax, see the section on the message class in the relevant TIBCO Enterprise Message Service API reference documentation.</p> </div>

## Define Triggers Panel

You can create or modify a trigger in the Define Triggers panel.

### Panel Layout

```

SXTP06 ----- Define Triggers ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> CSR

(ADD=Create new Trigger, CAN=Cancel updates)

Sel Output Identification                     At      Value
-----
-      -      -----
2      2      New-Trigger1                     0      MATCH
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Define Triggers panel:

Field	Description
<b>Command</b>	You can enter ADD to creates a new trigger, and enter CAN to cancel all modifications just done to the existing trigger.
<b>Sel</b>	<p>You can select an existing trigger to modify.</p> <p>Valid selections: S(elect), D(elete), R(epro), and E(MS).</p> <p>If you select E, you enter the panel described in <a href="#">EMS Trigger Details Extension Panel</a>.</p>
<b>Output</b>	<p>The type of data expected as output from the transaction processing application.</p> <p>Valid selections: 1 (Status), 2 (Message), and 3 (Both Status and Message).</p> <p>For details, see the <b>Output Method</b> field in the <a href="#">Trigger Details Panel</a>.</p>
<b>Identification</b>	Required. The name you specify for the trigger.
<b>At</b>	The starting offset in the buffer that contains the text to be used for the search criteria in a trigger process.



Field	Description
<b>Value</b>	The text value associated with the transformation that tells the Transformer the transformation to perform on the data deposited in the TDQ by a transaction processing application.

## Select ESB Endpoint Panel (Triggers)

You select an ESB endpoint for a trigger in the Select ESB Endpoint panel.

### Panel Layout

```

SXTP06T----- Select ESB Endpoint ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

(Use S to select an ESB endpoint)
Sel Service Port Identifier Daemon Network
-----
0 0 New-ESB1 0 0 0 0
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Select ESB Endpoint panel:

Field	Description
<b>Sel</b>	You can select an ESB endpoint for a trigger. Select one ESB endpoint for each trigger. Valid selection: s(select).
<b>Service</b>	The value for the UDP service port number that defines the service group.
<b>Port</b>	The TCP port given to an ESB endpoint that distinguishes both Substation ES and the role of the ESB endpoint within it.
<b>Identifier</b>	The name of an ESB endpoint in the trigger.
<b>Daemon</b>	The host name or host IP address used by Substation ES to find the Rendezvous daemon or TIBCO Enterprise Message Service server and establish communication.
<b>Network</b>	The service group selected by IP network number or multicast addresses.

## Select Output Conversion Rule Panel (Triggers)

You can select a conversion rule for a trigger in the Select Output Conversion Rule panel.

### Panel Layout

```

SXTP06C----- Select Output Conversion Rule ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

          Max
Sel Flds Bufsz Conversion Id Message Id Buffer Id
-----
0 0 New_Rule1 New-Msg1 New-Buffer1
***** Bottom of data *****

```

## Field Description

The following table lists the fields in the Select Output Conversion Rule panel:

Field	Description
<b>Sel</b>	You can select a conversion rule for a trigger. Valid selections: S(elect) and O(utput).
<b>Flds</b>	The number of fields in the conversion rule.
<b>Max Bufsz</b>	The maximum buffer size for the conversion rule.
<b>Conversion Id</b>	The name of the conversion rule in the trigger.
<b>Message Id</b>	The name of the message definition in the conversion rule.
<b>Buffer Id</b>	The name of the buffer definition in the conversion rule.

## Trigger Details Panel

You can create or modify the details of a trigger in the Trigger Details panel.

### Panel Layout

```

SXTPO6N ----- Trigger Details -----
Command ==>                                     (EMS,PROP)

Trigger Identifier    => New-Trigger1
Publish To           => New-Trigger1-subject
Publish To, Sec      =>
Reply To             =>
Error Subject        =>
Error Threshold       => 0      (0 - 99999) Error Percent => 0 (0 - 99)
Publish To Len (Dyn) => 0      Start => 0
Output Conversion(?) => New_Rule1
ESB Endpoint (?)     => New-ESB1
Trace Level          => 0      0-5
Output Method        => 2      1 Status    2 Message
                        3 Both Status and Message

Compare Text Contain in Buffer
Start => 0
Length => 5
Value => MATCH
Undelivered Requeue  => Y      (Y Requeue, N Discard - Reliable Only)

```


## Field Description

The following table lists the fields in the Trigger Details panel:

Field	Description
<b>Command</b>	You can enter EMS to access the panel described in <a href="#">EMS Trigger Details Extension Panel</a> , and enter PROP to access the panel described in <a href="#">EMS Properties (Trigger) Panel</a> .  Valid commands: <b>EMS</b> and <b>PROP</b> .

Field	Description
<b>Trigger Identifier</b>	The name of the trigger.
<b>Publish To</b>	Required. The outbound subject that a message is published on for a given trigger transaction. The field is required when using the trigger MOI.
<b>Publish To, Sec</b>	Optional. The optional second outbound subject that a message is published on for a given trigger transaction.
<b>Reply To</b>	Optional. The reply subject that an external messaging application returns a response to Substation ES (for a CICS-initiated request reply process).
<b>Error Subject</b>	The subject that an error message is published on if Transformer conversion or BES encounters errors.
<b>Error Threshold</b>	The number of errors allowed before the trigger is disabled.
<b>Error Percent</b>	The percentage of errors allowed before the trigger is disabled.
<b>Publish To Len (Dyn)</b>	Optional. The outbound subject name can be specified dynamically within the buffer. You can the number of bytes to be used for the dynamic Publish To subject name.  Default value: 0.
<b>Start</b>	The starting offset in the buffer to be used for the dynamic Publish To subject name.  The area defined by dynamic <b>Publish To Len (Dyn)</b> and <b>Start</b> is concatenated to form the Publish To subject. This is the name that the message is published to. If this dynamic Publish To subject name matches the name of the message or the error subject, then only one message is written to the subject. The dynamic message must adhere to EMS or RV subject structure, whichever messaging service is used, or an error is issued by EMS or RV when the message is published.
<b>Output Conversion (?)</b>	Required. The name of the conversion rule used for output conversion. If you specify ? in this field, a list of all conversion rules id displayed where you can select a conversion rule.
<b>ESB Endpoint (?)</b>	Required. The name of an ESB endpoint. If you specify ? in this field, a list of all ESB endpoints is displayed where you can select an ESB endpoint.

Field	Description
<b>Trace Level</b>	<p>The user trace debug level for an individual trigger process. Must always be 0 or 1, unless TIBCO Support require the output. A certain amount of overhead and many lines of output are produced when this value is greater than 2.</p> <p>Valid values: 0 - 5. Default value: 0.</p> <ul style="list-style-type: none"> <li>0 - 1: nothing.</li> <li>2 - 5: shows message exit from Substation ES.</li> </ul> <pre>SXT5472I U-Trace(T) UoW:1178777278 Trigger:TRIGGER-SAAS-BYTES         matched SXT5303I U-Trace(T) UoW:1178777278 Rcp/Trig:TRIGGER-SAAS-         BYTES,         Respond:tibss.trigger, Reply:this.is.a.test.reply.to</pre>
<b>Output Method</b>	<p>You can specify whether data is expected to be output from the transaction processing application:</p> <ul style="list-style-type: none"> <li>If you are configuring Substation ES to subscribe to data generated by a TIBCO messaging application with the DPL MOI, you can specify whether return data and a status is expected back.</li> <li>If you are configuring Substation ES to publish data for a transaction processing application, you can use this option to indicate whether some type of data or a status is being generated.</li> </ul> <p>The following are valid selections:</p> <ul style="list-style-type: none"> <li>1 Status: a status message is sent to the awaiting TIBCO messaging application by way of the value specified in the <b>Reply To</b> field.</li> <li>2 Message: outbound data is to be transformed and published as a message to the TIBCO messaging application.</li> <li>3 Both Status and Message: for this process return code, reason code, and outbound data set are to be transformed and published as a message to the TIBCO messaging application.</li> </ul>
<b>Compare Text Contain in Buffer</b>	You can compare texts contained in a buffer.
<b>Start</b>	The starting offset in the buffer that contains the text to be used for the search criteria in a trigger process (valid numbers 0-size of buffer). Remember that the first character in the buffer is offset 0 (not 1).
<b>Length</b>	The size of the search text in the buffer of a TDQ record. If you specify 0 , there is no matching on buffer content and the only way to select this trigger is through the Service Name parameter on a programs call to SXCQWRIT. For details on implementing HVT and SXCQWRIT, see <i>TIBCO Substation ES Operations and Administration</i> .
<b>Value</b>	The text value that is associated with the transformation. This text tells the Transformer what transformation to perform on the data that is deposited in the TDQ by a transaction processing application.

Field	Description
<b>Undelivered Requeue</b>	This option indicates whether the reliable trigger processes will requeue a message if delivery fails.
	RED Interface is not support this function.

## EMS Trigger Details Extension Panel

You create or modify TIBCO Enterprise Message Service values for a trigger in the EMS Trigger Details Extension panel.

You can access this panel in the following two ways:

- Select E in the **Sel** field of the Define Triggers panel.  
For details, see [Define Triggers Panel](#).
- Enter EMS in the Trigger Details panel.  
For details, see [Trigger Details Panel](#).

### Panel Layout

```

SXTPO6E----- EMS Trigger Details Extension -----
Command ==>

Trigger Identifier      => New-Trigger1
Output Dest Type       => T  Name: New-Trigger1-subject
Output Dest Type, Sec  => T  Name:
Reply Dest Type        => T  Name:
Error Dest Type        => Q  Name:

                               (Service Levels G=Guaranteed, R=Reliable)
SMSL Mode              ==> RN (GA All, RN No-Ack)
Delivery Mode          ==> P (P Persistent, N Non-Persistent, R Reliable)


Output Message properties:
Priority               ==> N (N/Y) Value ==> 4      (0 - 9)
Expiration            ==> N (N/Y) Value ==> 0      Unit ==> S (S,M,H,D)
Compression           ==> N (N/Y)
Preserve              ==> N (N/Y, undelivered messages)
Body Trace            ==> N (N/Y)
Message Type          ==> M  M(ap)/S(tream)/T(ext)/B(ytes)

```

### Field Description

The following table lists the fields in the EMS Trigger Details Extension panel:

Field	Description
<b>Trigger Identifier</b>	The name of the trigger.
<b>Output Dest Type</b>	The output destination type. Valid selections: T(opic) and Q(ueue). Default selection: T.
<b>Output Dest Name Type, Sec</b>	The outbound destination that a message is published on for a given trigger transaction. Required when using the trigger method of invocation.
<b>Output Dest</b>	The optional second output destination type.

Field	Description
<b>Output Dest Name, Sec</b>	The optional second outbound destination that a message is published on for a given trigger transaction.
<b>Reply Dest Type</b>	The reply destination type. Valid selections: T(opic) and Q(ueue). Default: T.
<b>Reply Dest Name</b>	The reply destination where an external messaging application returns a response to Substation ES (for a CICS-initiated request reply process).
<b>Error Dest Type</b>	The error destination type. Valid selections: T(opic) and Q(ueue). Default: Q.
<b>Error Dest Name</b>	The destination where an error message is published if Transformer conversion or BES encounter errors.
<b>SMSL Mode</b>	<p>The SMSL mode for TIBCO Enterprise Message Service.</p> <p>Valid selections:</p> <p>GA (GA All): Substation ES specific BES interface explicitly issues a commit type call of the transaction to the BES when all the Substation ES processes and executions for a given message are completed successfully. In case of an error, the message is moved to the Dead Message Queue. The Substation ES service level for GA All is Guaranteed.</p> <p>RN (No-Ack): the TIBCO Enterprise Message Service client library does not send a receipt of the message that was received. The server destroys the message after it has been sent. The Substation ES service level for No-Ack is Reliable, no acknowledgement.</p> <p>Default selection: RN.</p> <div>  <p>When the SMSL Mode is set to GA, not the following requirements:</p> <ul style="list-style-type: none"> <li>• The <b>Output Dest Type</b> must be set to Q(ueue).</li> <li>• The <b>Delivery Mode</b> must be set to P(ersistent).</li> <li>• The <b>Undelivered Requeue</b> option must be set to Y.</li> </ul> </div>
<b>Delivery Mode</b>	<p>The delivery mode for TIBCO Enterprise Message Service.</p> <p>Valid modes: P(ersistent), N(onpersistent), and R(eliable). Default: P.</p> <p>For additional information on Output Message properties, see <i>TIBCO Enterprise Message Service User's Guide</i>.</p>
<b>Priority</b>	<p>You can set the EMS message priority to the value specified in the <b>Priority Value</b> field.</p> <p>Valid selections: N(o) and Y(es). Default selection: N.</p>
<b>Priority Value</b>	<p>The EMS message priority value.</p> <p>Valid values: 0 - 9. Default value: 4.</p>

Field	Description
<b>Expiration</b>	You can set the EMS message expiration to the value specified in the Expiration Value field. Valid selections: N(o) and Y(es). Default selection: N.
<b>Expiration Value</b>	The EMS message expiration value. Valid values: 0 - 99999. Default value: 0.
<b>Expiration Unit</b>	The message expiration unit. Valid selections: S(econd), M(inute), H(our), and D(ay). Default selection: S.
<b>Compression</b>	You can specify whether to compress the EMS message for storage on the server. Valid selections: N(o) and Y(es). Default selection: N.
<b>Preserve</b>	If a message is removed from its queue by the EMS server for reasons other than being consumed, the server checks this value. If this value is set to Y, the message is placed on the server's the undelivered message queue. Valid selections: N(o) and Y(es). Default selection: N.
<b>Body Trace</b>	Tracing for this destination generates trace messages that include the message body. Valid selections: N(o) and Y(es). Default selection: N.
<b>Message Type</b>	The type of EMS output message. Valid selections: M(ap), S(tream), T(ext), and B(ytes). Default: M.

## EMS Properties (Trigger) Panel

You can specify up to fifteen EMS properties to be inserted in output messages in the EMS Properties (Trigger) panel.



You can also specify the JMSCorrelationID header field, tibss-BES, tibss-cfg-id, and tibss-ss-id in this panel. See [System Fields Used as Property Fields](#) for more information on how to set up Substation ES system fields as property fields.

You can access this panel by entering PROP in the Trigger Details panel. For details, see [Trigger Details Panel](#).

### Panel Layout

```

SXTPO5PT ----- EMS Properties (Trigger) ----- Row 1 to 11 of 15
Command ==>                                         Scroll ==> CSR

      Output Buffer Id: New-Buffer1

Property Name      B/V Value
-----
Property-1         B   IVP-BES-PGM-NAME
Property-2         V   This is the property field

```

```

tibss-BES          V  $$SYSTEM

tibss-ss-id        V  $$SYSTEM

***** Bottom of data *****

```

### Field Description

The following table lists the fields in the EMS Properties (Trigger) panel:

Field	Description
<b>Output Buffer ID</b>	If you select property values from a buffer, this is the buffer name associated with a output conversion rule that a selection can be made from.
<b>Property Name</b>	The name of the property.
<b>B/V</b>	You can specify where the property value is from: <ul style="list-style-type: none"> <li>B: the property value is selected from a buffer field.</li> <li>V: the property value is defined by the user in the <b>Value</b> field.</li> </ul> Valid selections: B(uffer) or v(alue).
<b>Value</b>	If you select this field from the buffer, enter the question mark (?) and then press Enter. A list of field names in the buffer is displayed where you can select a field name.  If this field is user defined, enter the desired value.

## Group Definitions Panel

You can create or modify a group definition in the Group Definitions panel.

### Panel Layout

```

SOTP08 ----- Group Definitions ----- Row 1 to 1 of 1
Command ==>> Scroll ==> CSR

(ADD=Create new Group, CAN=Cancel updates)
Sel Group Id
-----
New-Group1
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Group Definitions panel:

Field	Description
<b>Command</b>	You can enter ADD to create a group definition, and enter CAN to cancel modifications just done to the existing group definition.  Valid commands: <b>ADD</b> and <b>CAN</b> .



Field	Description
<b>Sel</b>	You can select an existing group definition to modify. Valid selections: S(elect), D(elete), and R(epro).
<b>Group Id</b>	Required. The name of the group specified by you. The maximum character length is 15. Default value: New-Group1.

## Select Triggers and Recipes for Group Panel

You can select the recipes and triggers to be included in a group in the Select Triggers and Recipes for Group panel.

### Panel Layout

```
SXTP08G ----- Select Triggers and Recipes for Group      Row 1 to 2 of 2
Command ==>                                         Scroll ==> CSR

      Group Name ==> New-Group1

      Use S to select each Trigger and/or Recipe into the Group.
Sel Type      Identifier
-----
      Recipe    New-Recipe1
      Trigger    New-Trigger1
***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Select Triggers and Recipes for Group panel:

Field	Description
<b>Group Name</b>	The name of the group specified by you.
<b>Sel</b>	You can select triggers or recipes for the group. Valid selection: S(elect).
<b>Type</b>	Trigger or recipe.
<b>Identifier</b>	The name of the trigger or recipe.

## Display Configuration File Detail Panel

You can create or modify a group definition in the Display Configuration File Detail panel.

### Panel Layout

```
SXTP09 ----- Display Configuration File Detail -----
Command ==>

      Select one of the following:
      1 Expand a Recipe
      2 Expand a Trigger
      3 Expand a Conversion
      4 Expand a Group Definition
```

```

Identification(?) ==> ?
                        (Recipe/Trigger/Convert Id/Group Id)

Display in Hex      ==> N (Yes/No)

```

### Field Description

The following table lists the fields in the Display Configuration File Detail panel:

Field	Description
<b>Command</b>	You can enter ADD to create a group definition, and enter CAN to cancel all modifications just done to the existing group definition.  Valid selections: S(elect), D(elete), and R(epro).
<b>Sel</b>	You can select an existing group definition to modify.
<b>Group Id</b>	Required. The name of the group specified by you. The maximum character length is 15.  Default value: New-Group1.

### Select Desired Entry Panel

You can select an existing recipe, trigger, conversion rule, or group definition for detailed display in the Select Desired Entry panel.

#### Panel Layout

```

SOTP09S----- Select Desired Entry ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

    Use S to select the proper Entry
Sel   Identifier
-----
      New-Recipe1
***** Bottom of data *****

```

### Field Description

The following table lists the fields in the Select Desired Entry panel:

Field	Description
<b>Sel</b>	You can select a specific entry from a list of recipes, triggers, conversion rules, or group definitions.  Valid selection: S.
<b>Identifier</b>	The name of the recipe, trigger, conversion rule or group definition entry.

### Summary of Definition Panel (Recipe)

The Summary of Definition panel displays a recipe in detail.

#### Panel Layout

```

SOTP09D ----- Summary of Definition ----- Row 1 to 77 of 77
Command ==>                                     Scroll ==> CSR

|Recipe name   : Substation-Admin-IMS

```

```

Subject : tibss.admin.sxi.request
Response : , Always: Y
Err Subj : tibss.error.out
Error Thresholds: Count: 0, Percent: 0
Transcode: , MOI:8, Trace: 0, Reply: 3
Dynamic Subject Leng:0, Start: 0, Limit: 0, SSID Filter:No
CICS Fields:
Mirror tran: User-tran: N, Tranid: System-name:
Channel Name :
Single CTN Name :
Container Invocation:P
Container Process :S
Forward Destinations:No
Resource 2 : Resource 3 :
Resource 4 : Resource 5 :
IMS Fields:
Variable Length Tran : N Transaction Len:0
Segment Esc Seq : 00000000
MFS Map Name :
EMS Extensions:
Destination Types: Input: Q, Output: Q, Error: Q
Output: Prty: N Expir: N, Compr: N, Presv: N, Trace: N
Expire: 0, Units: S, Priority: 4
Durable Subscriber: N, Clientid:
SMSL Mode: RA, Delivery Mode: R
Message type: M, Copy Correlation id: N

Back End System: Interface-IMS
Type: I, Target: IMS
ESB RV Endpoint id: RENDEZVOUS, Network: 127.0.0.1
Daemon:
Service: 7555, Port: 0, Multicast: N, Startup: Y
Multicast-addr:
SSL Required: N, SSL Authonly: N
KeyRingFile :
KeyRingLabel :
Cipher :
LDAP URL :
LDAP Userid :
LDAP Password:
Enable FIPS : N
Enable Trace : N Enable Debug Trace: N
Enable SSLV3 : N Enable TLS1 : N
Exp Host Member:
Ver Ex Host : N
ConnFact Name: , Password: Null, Userid:
USE SSL Parms for ECF:N
ECF URL :
ConnFactory Metric: 0
Attempts Delay Timeout
Connect: 0 0 0
Reconnect: 0 0 0
Admin Con:No Admin Userid:

Input Conversion: CVR-I-ADMIN
Max buffer size: 0
Receiving Buffer max: 0, id: BUF-IO-ADMIN
Input Message id: MSG-IN-ADMIN

From Message id: 0, name: SXS-COMMAND
To offset: 00000000, length: 00008192, name: COMMAND-STR
In decimal: 0, Out decimal: 0, Buf-type: STR
Truncate: Y, Required: Y, Lead-sign: N
Sep-sign: N, Right-just: N, Pad: 40
Init len: 0, Init value:

Output Conversion: CVR-O-ADMIN
Max buffer size: 0
Sending Buffer max: 0, id: BUF-IO-ADMIN
Output Message id: MSG-OUT-ADMIN

From offset: 00000000, length: 00008192, name: COMMAND-STR

```

```

|      Buffer decimal: 0, Message decimal: 0, Buf-type: STR
|      To Message id: 0, Type: STR, Req'd: Y, name: SXS-RESULT

```

```

***** Bottom of data *****

```

## Field Description

The following table lists the fields in the Summary of Definition panel:

Field	Description
<b>Recipe name</b>	The name of the recipe.
<b>Subject</b>	The subject Substation ES listens for to invoke a given transformation.
<b>Response</b>	The subject that a message is published on for a given transaction.
<b>Always</b>	This field determines where the response from a transaction process is sent. For details, see the <b>Publish Always</b> field of the <a href="#">Recipe Details Panel</a> .
<b>Err Subj</b>	The subject that an error message is published on if Transformer conversion or BES encounters errors.
<b>Count</b>	The number of errors allowed before the recipe is disabled.
<b>Percent</b>	The percentage of errors allowed before the recipe is disabled.
<b>Transcode</b>	The name of the resource.
<b>MOI</b>	The method of invocation that initiates the configured transformation. For details, see the <b>Method of Invocation</b> field of the <a href="#">Recipe Details Panel</a> .
<b>Trace</b>	The user trace debug level.
<b>Reply</b>	The type of data expected as output from the transaction processing application. For details, see the <b>Reply Method</b> field of the <a href="#">Recipe Details Panel</a> .
<b>Dynamic Subject Leng</b>	The number of bytes to be used for the dynamic subject name.
<b>Start</b>	The starting offset in the buffer to be used for the dynamic subject name.
<b>Limit</b>	The maximum number of concurrent messages to be processed by an ESB endpoint for this recipe. This parameter is only valid with EMS transports.
<b>SSID Filter</b>	The SSID filter used to filter messages that are to be processed by the Substation ES instance with a matching SSID.
<b>Mirror tran</b>	The CICS-specific field used to invoke different transactions for Substation ES to
<b>User-tran</b>	This field indicates whether it is a CSMI user transaction.

Field	Description
<b>Tranid</b>	A four-character identifier that specifies the transaction responsible for running the program on the remote CICS system.
<b>System-name</b>	A four-character system identifier that specifies the CICS region where the program will ultimately be run. This is referred to as the SYSID in CICS terms.
<b>Channel Name</b>	The name of the channel to be set when container data input and output is used.
<b>Single CTN Name</b>	The name of the container to be set when single container input and output is used.
<b>Container Invocation</b>	The method of container invocation. The valid selections are P(for a program link) and T(to start a transaction in CICS).
<b>Container Process</b>	The type of container channel. The valid selections are S(single container) and M(multiple containers).
<b>Forward Destinations</b>	If this parameter is enabled and the process has multiple containers without conversion rules, then Substation ES creates two additional containers named <b>tibss-ListenTo</b> and <b>tibss-ReplyTo</b> and sends them to CICS. These containers contain the corresponding destination or subject names of this service.
<b>Resource N</b>	Two to five additional resources can be specified. Each of these fields can specify an additional resource name for special CICS execution.
<b>Variable Length Tran</b>	This field indicates whether the IMS transaction ID provided in the resource name is less than eight characters.
<b>Segment Esc Seq</b>	The eight hex characters that represent the segment separator in the message.
<b>MFS Map Name</b>	The MFS map name to use.
<b>Input</b>	The destination of the subject that Substation ES listens to invoke a given transformation for: T(topic) or Q(queue).
<b>Output</b>	The destination of an outbound subject that a message is published on for a given transaction process: T(topic) or Q(queue).
<b>Error</b>	The subject destination where an error message is published when Transformer conversion or BES encounters errors: T(topic) or Q(queue).
<b>Prtty</b>	This field indicates whether the priority is set for this message.
<b>Expir</b>	This field indicates whether the expiration is set for this message.
<b>Compr</b>	This field indicates whether compression is set for this message.
<b>Presv</b>	This field indicates whether the message is to be preserved on the server.

Field	Description
<b>Trace</b>	This field indicates whether messages that include the message body are generated.
<b>Expire</b>	The EMS expiration value for the message.
<b>Units</b>	The EMS legal unit value for the message.
<b>Priority</b>	The EMS priority value for the message.
<b>Durable Subscriber</b>	This field indicates whether to connect the session as a durable subscriber.
<b>ClientId</b>	The client identifier in a durable connection.
<b>SMSL Mode</b>	The SMSL mode used for this message by Substation ES.
<b>Delivery Mode</b>	The delivery mode used for this message by Substation ES.
<b>Message type</b>	<p>The types of the TIBCO Enterprise Message Service output message: M(ap), S(ream), T(ext), or B(ytes).</p> <p>This selection only applies to the output message (not for the input message). The input message can be any types (Map, Stream, Text, or Bytes). Substation ES will attempt to convert it properly.</p>
<b>Copy Correlation id</b>	<p>Copies the JMSCorrelationID value from an inbound message and puts it in the output message. The following are valid settings:</p> <ul style="list-style-type: none"> <li>• N(o)</li> <li>• Y(es)</li> <li>• E(nvelope)</li> <li>• M(sgId)</li> <li>• w(CorId or MsgId)</li> </ul>
<b>EMS Selector</b>	A free-form field for specifying a selector statement. For details, see the <a href="#">EMS Selector Specification Panel</a> .
<b>Back End System</b>	The description of the Substation ES interface, both Primary and Alternate.
<b>Type</b>	The type of interface: A(dmin), C(ICS), or I(MS).
<b>Target</b>	The identification of the interface that Substation ES communicates with the transaction processing application through. Must match the interface ID that is configured through the <b>INTF-ID</b> parameter in the system initialization member.
<b>ESB EMS/RV Endpoint id</b>	The name of the ESB endpoint.
<b>Daemon / URL</b>	The host name or host IP address used by Substation ES to find the Rendezvous daemon or EMS Server and establish communication.

Field	Description
<b>Network</b>	The service group selected by network IP network number or multicast addresses.
<b>Service</b>	The value provided for the UDP service port number that defines the service group.
<b>Port</b>	The TCP port given to an ESB endpoint that distinguishes both Substation ES and the role of the ESB endpoint within it.
<b>Multicast</b>	This field indicates whether the address is multicast enabled.
<b>Startup</b>	This field indicates whether to start this ESB during Substation ES startup.
<b>Multicast-addr</b>	The multicast addresses, if the host computer has multiple network interfaces.
<b>SSL Required</b>	This field indicates whether this ESB uses an SSL connection to server.
<b>SSL Authonly</b>	This field indicates whether this ESB connection use SSL only for authentication.
<b>KeyRingFile</b>	The Resource Access Control Facility (RACF) ring name.
<b>KeyRingLabel</b>	The RACF identity name.
<b>Cipher</b>	The IBM cipher codes for encryption.
<b>LDAP URL</b>	The URL to address LDAP server.
<b>LDAP Userid</b>	The user ID used to access the LDAP server.
<b>LDAP Password</b>	The password used to access the LDAP server.
<b>Enable FIPS</b>	This field indicates whether to use FIPS 140-2.
<b>Enable Trace</b>	This field indicates whether to turn on SSL Trace.
<b>Enable Debug Trace</b>	This field indicates whether to turn on SSL Debug Trace.
<b>Enable SSLV3</b>	This field indicates whether to turn on SSL V3.
<b>Enable TLS1</b>	This field indicates whether to turn on TLS1.
<b>Exp Host Member</b>	The name of the EMS server name that is being interfaced with.
<b>Ver Ex Host</b>	This field indicates whether to verify the name of the EMS server is matched against the value specified in the <b>Exp. EMS Host Name</b> field.
<b>ConnFact Name</b>	The name of the connection factory.
<b>Password</b>	The connection object authenticates the user identity using this password.

Field	Description
<b>Userid</b>	The connection object presents this user identity to the server.
<b>USE SSL Parms for ECF</b>	This field indicates whether to use SSL parameters from SSL panel.
<b>ECF URL</b>	The URL string to which connections created by this factory will be connected.
<b>ConnFactory Metric</b>	The load balancing metric of this connection factory.
<b>Connect Attempts</b>	This field limits the number of times that Substation ES attempts to establish a connection to the EMS server when the EMS server is not present at Substation ES startup.
<b>Reconnect Attempts</b>	The time (in milliseconds) between connection attempts.
<b>Connect Delay</b>	The time (in milliseconds) between reconnection attempts.
<b>Reconnect Delay</b>	The maximum time (in milliseconds) the Substation ES will wait for a connection to the server to be established.
<b>Connect Timeout</b>	The maximum time (in milliseconds) the Substation ES will wait for a reconnection to the server to be established.
<b>Reconnect Timeout</b>	This field indicates whether to create this ESB Admin connection during Substation ES startup.
<b>Admin Con</b>	This field indicates whether to create this ESB Admin connection during Substation ES startup.
<b>Admin Userid</b>	The user identifier that has EMS server administrator privileges.
<b>Input Conversion</b>	The name of the conversion rule for the input conversion.
<b>Max buffer size</b>	The maximum input buffer size defined for the conversion rule.
<b>Receiving Buffer max</b>	The maximum size of the inbound buffer.
<b>id</b>	The name of the buffer associated with the transformation for the receiving buffer.
<b>Input Message id</b>	The name of the message.
<b>From Message id</b>	A message field identifier. Zero (0) is a special value that signifies no message field identifier.
<b>name</b>	The field name you specify in the message definition associated with the transformation from which data is to be extracted.
<b>To offset</b>	The number of bytes for the field that is the starting offset in the inbound buffer. The buffer offset positions the data at a given byte position.



Field	Description
<b>length</b>	The length of the data type field in the buffer definition associated with the transformation.
<b>name</b>	The name you specify for the field in the buffer definition associated with the transformation.
<b>In decimal</b>	The implied number of digits after the decimal point in the input. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Out decimal</b>	The implied number of digits after the decimal point in the output. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Buf-type</b>	The mainframe type of data that the invoked application expects: Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, and Opa. For type definitions, see <a href="#">Type</a> .
<b>Truncate</b>	<p>This field indicates whether data that extends beyond the specified buffer length allocated for a message field was truncated. By default, the truncate option is Yes.</p> <p>The <b>S</b> field serves as a system field, which means no mapping to the buffer field is required.</p>
<b>Required</b>	<p>This field indicates whether the field is required for a conversion. The default value is Y.</p> <p>If you specify Y, the transaction is not considered complete unless a field with this name is encountered in the inbound Rendezvous message. If you specify N, no specific check is made as to the availability of that message field in the Rendezvous message.</p>
<b>Lead-sign</b>	The field contains a leading sign, as opposed to a trailing sign.
<b>Sep-sign</b>	<p>This field indicates whether a byte is reserved for the sign.</p> <p>Valid selections: N(one), L(eading), and T(railing). Default selection: N.</p>
<b>Right-just</b>	The positioning of the string. You can use padding characters to fill in the data on either side, as applicable. The default value is NO and strings are left justified.
<b>Pad</b>	The character for padding. You can use keyboard accessible characters as a pad, which are placed at the beginning, at the end, or in both positions, depending on the configuration. The default pad character is a space.
<b>Init len</b>	The length of the <b>Init Value</b> field when used.
<b>Init value</b>	The value or characters for filling in the field if the data is not available in the buffer.
<b>Output Conversion</b>	The name you specify for the output conversion associated with the transformation.

Field	Description
<b>Max buffer size</b>	The maximum outbound buffer size of all fields defined in the Msg-Field or Buffer-Field conversion rule.
<b>Sending Buffer max</b>	The maximum size of the outbound buffer for this recipe process to be sent.
<b>id</b>	The name you specify for the buffer associated with the transformation for the sending buffer.
<b>Output Message id</b>	The name of the message definition associated with the transformation that was entered on the Message Definitions panel.
<b>From offset</b>	The number of bytes for the field that is the starting offset in the outbound buffer. The buffer offset positions the data at a given byte position.
<b>length</b>	The length of the data type field in the buffer definition associated with the transformation.
<b>name</b>	The name you specify for the field in the buffer definition associated with the transformation.
<b>Buffer decimal</b>	The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Message decimal</b>	The implied number of digits after the decimal point in the message field. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Buf-type</b>	The mainframe type of data that the invoked application expects. The valid buffer types are Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, and Opa. For the definitions of the buffer types, see <a href="#">Type</a> .
<b>To Message id</b>	A message field identifier. Zero (0) is a special value that signifies no field identifier.
<b>Type</b>	The TIBCO message data type for the current field. The current valid message types are DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64, F32, F64, POR, and IPA. For the definitions of the message types, see <a href="#">Type</a> .
<b>Reqd</b>	<p>This field indicates whether the field is required for a conversion. The default value is Y.</p> <p>If you set this field to N, if the data type is numeric, and if the content of the field has not been initialized by the host-side application, the Transformer initializes the field and no editing errors occur.</p>
<b>name</b>	The name you specify for the field in the message definition associated with the transformation that was entered in the <b>Define Fields</b> in a message panel.

## Summary of Definition Panel (Trigger)

The Summary of Definition panel displays a trigger definition in detail.

### Panel Layout

SXTP09D ----- Summary of Definition ----- Row 1 to 61 of 61  
Command ==> Scroll ==> CSR

```

Trigger name : CICS-RR-REQUEST
  Output Method : 2, Trace: 2
  Publish To : tibss.CICS.RR.trigger.Request
  Publish To, Sec:
  Reply To :
  Error Subject : tibss.error.out
  Error Thresholds: Count: 0, Percent: 0
  Dynamic Subject Leng:0, Start: 0
  Match at: 0, Length: 6, Value: 'SXCIRR'
  Undelivered Requeue: No
EMS Extensions:
  Destination Types: Output: T, Reply: T, Error: Q, 2nd Resp:T
  Output: Prty: N Expir: N, Compr: N, Presv: N, Trace: N
  Expire: 0, Units: S, Priority: 4
  SMSL Mode: RN, Delivery Mode: N
  Message type: M

ESB RV Endpoint id: RENDEZVOUS, Network: 127.0.0.1
Daemon:
  Service: 7555, Port: 0, Multicast: N, Startup: Y
  Multicast-addr:
SSL Required: N, SSL Authonly: N
  KeyRingFile :
  KeyRingLabel :
  Cipher :
  LDAP URL :
  LDAP Userid :
  LDAP Password:
  Enable FIPS : N
  Enable Trace : N Enable Debug Trace: N
  Enable SSLV3 : N Enable TLS1 : N
  Exp Host Member:
  Ver Ex Host : N
ConnFact Name: , Password: Null, Userid:
USE SSL Parms for ECF:N
ECF URL :
ConnFactory Metric: 0
      Attempts Delay Timeout
Connect: 0 0 0
Reconnect: 0 0 0
Admin Con:No Admin Userid:

Output Conversion: CVR-O-TSRR-01
  Max buffer size: 0
  Sending Buffer max: 0, id: BUF-IO-BF01
  Output Message id: MSG-IO-RR-01
  From offset: 00000000, length: 00000008, name: IVP-ID
    Buffer decimal: 0, Message decimal: 0, Buf-type: TEXT
    To Message id: 0, Type: STR, Req: Y, name: IVP-ID

  From offset: 00000008, length: 00000056, name: IVP-DESCR
    Buffer decimal: 0, Message decimal: 0, Buf-type: TEXT
    To Message id: 0, Type: STR, Req: Y, name: IVP-DESCR

  From offset: 00000176, length: 00000008, name: IVP-START-LILSECS
    Buffer decimal: 0, Message decimal: 0, Buf-type: FLOAT
    To Message id: 0, Type: F64, Req: Y, name: IVP-START-LILSECS

  From offset: 00000236, length: 00000004, name: IVP-BTCH-SEQ-NO
    Buffer decimal: 0, Message decimal: 0, Buf-type: BIN

```

```
|      To Message id: 0, Type: I32, Req'd: N, name: IVP-BTCH-SEQ-NO
***** Bottom of data *****
```

## Field Description


The following table lists the fields in the Summary of Definition panel:

Field	Description
<b>Trigger name</b>	The user defined trigger name associated with a particular transformer configuration.
<b>Output Method</b>	The data expected to be output from the transaction processing application. For further clarification, see the <b>Output Method</b> field in the <a href="#">Trigger Details Panel</a> .
<b>Trace</b>	The user trace debug level assigned on the Trigger Details panel. The user trace debug level assigned on the Trigger Details Panel.  Valid values: 0 - 5.
<b>Publish To</b>	A subject that a message is published on for a given trigger transaction.
<b>Publish To, Sec</b>	A subject that a message is published on for a given trigger transaction.
<b>Reply To</b>	The reply subject or destination message property for the published message.
<b>Error Subject</b>	A subject that an error message is published on if Transformer conversion or BES encounters errors.
<b>Count</b>	The number of errors allowed before the trigger is disabled.
<b>Percent</b>	The percentage of errors allowed before the trigger is disabled.
<b>Dynamic Subject Leng</b>	Optionally, you can specify the outbound subject name dynamically within the buffer. This field specifies the number of bytes for the dynamic subject name. The default value is 0.
<b>Start</b>	The starting offset in the buffer for the dynamic subject name.
<b>Match at</b>	The number of character positions that a string is offset in the buffer to indicate that it is a TDQ search string. The Transformer searches for a string with this offset and evaluates the string to determine the kind of transformation to perform.
<b>Length</b>	The size of the search string in the buffer of a TDQ.
<b>Value</b>	The string value that is associated with the transformation. This string tells the Transformer what transformation to perform on the data that is deposited in the TDQ by a transaction processing application.
<b>Undelivered Requeue</b>	This field indicates whether the reliable trigger processes will re-queue the message if delivery fails.
<b>Output</b>	The output destination type: T(opic) or Q(ueue).

Field	Description
<b>Reply</b>	The reply destination type: T(opic) or Q(ueue).
<b>Error</b>	The subject destination in which an error message is published if Transformer conversion or BES encounters errors. The destination type is T(opic) or Q(ueue).
<b>2nd Resp</b>	The optional second output destination type: T(opic) or Q(ueue).
<b>Prtty</b>	This field indicates whether the priority has been set for this message.
<b>Expir</b>	This field indicates whether the expiration has been set for this message.
<b>Compr</b>	This field indicates whether compression has been set for this message.
<b>Presv</b>	This field indicates whether the message is to be preserved on the server.
<b>Trace</b>	This field indicates whether trace generates messages that include the message body.
<b>Expire</b>	The TIBCO Enterprise Message Service expiration value for the message.
<b>Units</b>	The TIBCO Enterprise Message Service legal unit value for the message.
<b>Priority</b>	The TIBCO Enterprise Message Service priority value for the message.
<b>SMSL Mode</b>	The Substation Messaging Services Levels mode used for this message by Substation ES.
<b>Delivery Mode</b>	The delivery mode used for this message by Substation ES.
<b>Message type</b>	The type of TIBCO Enterprise Message Service output message. Valid selections: M(ap), S(tream), T(ext), and B(ytes).
<b>ESB EMS/RV Endpoint id</b>	The name you specify for a ESB definition for a particular transformer configuration entered in the Define ESB Endpoint panel.
<b>Daemon</b>	The host name or host IP address used by Substation ES to find the Rendezvous daemon or TIBCO Enterprise Message Service server and establish communication.
<b>Network</b>	The service group selected by IP network number or multicast addresses.
<b>Service</b>	The value for the UDP service port number that defines the service group.
<b>Port</b>	The TCP port given to an ESB that distinguishes both Substation ES and the role of the ESB within it.
<b>Multicast</b>	This field indicates whether the address is multicast enabled.
<b>Startup</b>	This field indicates whether to start this ESB during Substation ES startup. Valid selections: Y(es) or N(o).

Field	Description
<b>Multicast-addr</b>	The multicast addresses to use, if the host computer has multiple network interfaces.
<b>SSL Required</b>	This field indicates whether this ESB uses an SSL connection to server. Valid selections: Y(es) or N(o).
<b>SSL Authonly</b>	This field indicates whether this ESB connection uses SSL only for authentication. Valid selections: Y(es) or N(o).
<b>KeyRingFile</b>	The Resource Access Control Facility (RACF) ring name.
<b>KeyRingLabel</b>	The RACF identity name.
<b>Cipher</b>	The IBM cipher codes for encryption.
<b>LDAP URL</b>	The URL of the LDAP server.
<b>LDAP Userid</b>	The user ID used to access the LDAP server.
<b>LDAP Password</b>	The password used to access the LDAP server.
<b>Enable FIPS</b>	This field indicates whether to use FIPS 140-2. Valid selections: Y(es) or N(o).
<b>Enable Trace</b>	This field indicates whether to turn on SSL Trace. Valid selections: Y(es) or N(o).
<b>Enable Debug Trace</b>	This field indicates whether to turn on SSL Debug Trace. Valid selections: Y(es) or N(o).
<b>Enable SSLV3</b>	This field indicates whether to turn on SSL V3. Valid selections: Y(es) or N(o).
<b>Enable TLS1</b>	This field indicates whether to turn on TLS1. Valid selections: Y(es) or N(o).
<b>Exp Host Member</b>	The name of the TIBCO Enterprise Message Service server name that is being interfaced with.
<b>ConnFact Name</b>	This field indicates whether to verify the name of the TIBCO Enterprise Message Service server matches the value specified in the <b>Exp. EMS Host Name</b> field. Valid selections: Y(es) or N(o).
<b>Password</b>	The name of the connection factory.

Field	Description
<b>Userid</b>	The password to be used by the connection object when authenticating the user to the server.
<b>USE SSL Parms for ECF</b>	The user ID to be used by the connection object when connecting to the server.
<b>ECF URL</b>	This field indicates whether to use SSL parameters from the SSL panel. Valid selections: Y(es) or N(o).
<b>ConnFactory Metric</b>	The URL string that connections created by this factory will connect to.
<b>Connect Attempts</b>	The load balancing metric of this connection factory.
<b>Reconnect Attempts</b>	This field limits the number of times that Substation ES attempts to establish a connection to the TIBCO Enterprise Message Service server when the server is not present at Substation ES startup time.
<b>Connect Delay</b>	This field limits the number of times that Substation ES attempts to reestablish the connection to the TIBCO Enterprise Message Service server.
<b>Reconnect Delay</b>	The time (in milliseconds) between connection attempts.
<b>Connect Timeout</b>	The time (in milliseconds) between reconnection attempts.
<b>Reconnect Timeout</b>	The maximum time (in milliseconds) the Substation ES waits for a connection to the server to be established.
<b>Admin Con</b>	This field indicates whether to create this ESB Admin connection during Substation ES startup.
<b>Output Conversion</b>	The name you specify for the output conversion associated with the transformation.
<b>Max buffer size</b>	The maximum outbound buffer size defined for the Msg-Field or Buffer-Field conversion rule.
<b>Sending Buffer max</b>	The maximum size of the outbound buffer for this trigger process to be sent.
<b>id</b>	The user defined name of the buffer definition associated with the transformation used for the sending buffer.
<b>Output Message id</b>	The name of the message definition associated with the transformation that was entered on the Message Definitions panel.
<b>From offset</b>	The number of bytes that the response message is offset by in the buffer.
<b>length</b>	The length of the data type field in the buffer definition associated with the transformation.

Field	Description
<b>name</b>	The user defined name for the field in the buffer definition associated with the transformation.
<b>Buffer decimal</b>	The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Message decimal</b>	The implied number of digits after the decimal point in the message field. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Buf-type</b>	The mainframe type of data that the invoked application expects. The valid buffer types are Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, and Opa. For the definitions of the buffer types, see <a href="#">Type</a> .
<b>To Message id</b>	A message field identifier. Zero (0) is a special value that signifies no field identifier.
<b>Type</b>	The TIBCO message data type for the current field. The current valid message types are DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64, F32, F64, POR, and IPA. For the definitions of the message types, see <a href="#">Type</a> .
<b>Reqd</b>	<p>This field indicates whether the field is required for a conversion. The default value is Y.</p> <p>If you set this field to N, if the data type is numeric, and if the content of the field has not been initialized by the host-side application, the Transformer initializes the field and no editing errors occur.</p>
<b>name</b>	<p>The name you specify for the field in the message definition associated with the transformation that data is to be extracted from.</p> <div>  <p>The display <b>No buffer field</b> stipulates that this field is a system field.</p> </div>

## Summary of Definition Panel (Conversion)

The Summary of Definition panel displays a conversion rule definition in detail.

### Panel Layout

SXTP09D----- Summary of Definition ----- Row 1 to 85 of 85  
 Command ==> Scroll ==> CSR

```

| Conversion Identification: SXC-RR-I-CR01
|   Max buffer size: 0
|   Related Buffer max: 0, id: SXG-IO-BF01
|   Related Message id: SXC-IO-RR-MS01
|
| Message Field id: 0, type: OPA, name: tibss-envelope
|   Truncate: S, Required: Y
|
| Message Field id: 0, type: STR, name: IVP-ID
|   Buffer offset: 0000, length: 008, name: IVP-ID
|   Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
|   Truncate: Y, Required: Y

```



```

      Sep-sign: N, Right-just: N, Pad: 40
      Init len: 0, Init value:

Message Field id: 0, type: STR, name: IVP-DESCR
  Buffer offset: 0008, length: 056, name: IVP-DESCR
  Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
  Truncate: Y, Required: Y
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: STR, name: IVP-RESULT
  Buffer offset: 0064, length: 080, name: IVP-RESULT
  Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: STR, name: IVP-BES-NAME
  Buffer offset: 0144, length: 008, name: IVP-BES-NAME
  Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: STR, name: IVP-BES-TRAN-ID
  Buffer offset: 0152, length: 008, name: IVP-BES-TRAN-ID
  Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: STR, name: IVP-BES-PGM-NAME
  Buffer offset: 0160, length: 008, name: IVP-BES-PGM-NAME
  Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: STR, name: IVP-USERID
  Buffer offset: 0168, length: 008, name: IVP-USERID
  Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: F64, name: IVP-REPLY-LILSECS
  Buffer offset: 0184, length: 008, name: IVP-REPLY-LILSECS
  Message decimal: 0, Buffer decimal: 0, Buf-type: FLOAT
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: F64, name: IVP-START-LILSECS
  Buffer offset: 0176, length: 008, name: IVP-START-LILSECS
  Message decimal: 0, Buffer decimal: 0, Buf-type: FLOAT
  Truncate: Y, Required: Y
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: I32, name: IVP-BTCH-SEQ-NO
  Buffer offset: 0236, length: 004, name: IVP-BTCH-SEQ-NO
  Message decimal: 0, Buffer decimal: 0, Buf-type: BIN
  Truncate: Y, Required: N
  Sep-sign: N, Right-just: N, Pad: 40
  Init len: 0, Init value:

Message Field id: 0, type: STR, name: tibss-userid
  Truncate: S, Required: N

Message Field id: 0, type: STR, name: tibss-password
  Truncate: S, Required: N

```

\*\*\*\*\* Bottom of data \*\*\*\*\*

## Field Description

The following table lists the fields in the Summary of Definition panel:

Field	Description
<b>Conversion Identification</b>	The name you specify for the conversion definition associated with the transformation
<b>Max buffer size</b>	The maximum buffer size defined for the Msg-Field or Buffer-Field conversion rule.
<b>Related buffer max</b>	The maximum size of the buffer definition associated with the transformation that was entered on the Define Fields in a Buffer panel.
<b>id</b>	The name of the buffer definition associated with the transformation that was entered on the Buffer Definitions panel.
<b>Related Message id</b>	The name of the message definition associated with the transformation that was entered on the Message Definitions panel.
<b>Message Field id</b>	A message field identifier. Zero (0) is a special value that indicates no field identifier.
<b>type</b>	The TIBCO message data type for the current field. The current valid message types are DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64, F32, F64, POR, and IPA.  For the definitions of the message types, see <a href="#">Type</a> .
<b>name</b>	The field name you specify in the message definition associated with the transformation that data is to be extracted from.
<b>Buffer offset</b>	The number of bytes for the field that is the starting offset in the buffer. The buffer offset positions the data at a given byte position.
<b>length</b>	The length of the data type field in the buffer definition associated with the transformation.
<b>name</b>	The field name you specify in the message definition associated with the transformation that data is to be extracted from.
<b>Message decimal</b>	The implied number of digits after the decimal point in the message field. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.
<b>Buffer decimal</b>	The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.

Field	Description
<b>Buffer-type</b>	<p>The mainframe type of data that the invoked application expects. The valid buffer types are Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, and Opa.</p> <p>For the definitions of the buffer types, see <a href="#">Type</a>.</p>
<b>Truncate</b>	<p>You can specify whether data that extends beyond the specified buffer length allocated for a message field was truncated. By default, the truncate option is Yes.</p> <p>The S field serves as a system field, which means no mapping to the buffer field is required.</p>
<b>Required</b>	<p>You can specify whether the field is required for a conversion.</p> <p>Valid selections: Y(es) and N(o). Default selection: Y.</p> <p>For an inbound process, if you set this field to Y, the transaction is not considered complete unless a field with this name is encountered in the inbound Rendezvous message. If you specify N, no specific check is made as to the availability of that message field in the Rendezvous message.</p> <p>For an outbound process, if you set this field to N, if the data type is numeric, and if the content of the field has not been initialized by the host-side application, the field is initialized by the Transformer and no editing errors occur.</p>
<b>Lead-sign</b>	The field contains a leading sign, as opposed to a trailing sign.
<b>Sep-sign</b>	<p>You can specify whether a byte is reserved for the sign.</p> <p>Valid selections: N(one), L(eading), and T(railing). Default selection: N.</p>
<b>Right-just</b>	<p>The positioning of the string. You can use padding characters to fill in the data on either side, as applicable.</p> <p>The default value is NO and strings are left justified.</p>
<b>Pad</b>	<p>The character for padding. You can use keyboard accessible characters as a pad, which are placed at the beginning or at the end, depending on the configuration.</p> <p>The default pad character is a space.</p>
<b>Init len</b>	The length of the <b>Init Value</b> field when used.
<b>Init value</b>	The value or characters for filling in the field if the data is not available in the buffer.

## Summary of Definition Panel (Group)

The Summary of Definition panel displays the recipes and triggers in a group definition.

### Panel Layout

```

SXTP09D----- Summary of Definition ----- Row 1 to 3 of 3
Command ==>                                     Scroll ==> CSR

|Group name : IVP Group

```

```
|      Recipe id: Substation-Admin-IMS
|      Trigger id: CICS-RR-REQUEST
***** Bottom of data *****
```

### Field Description

The following table lists the fields in the Summary of Definition panel:

Field	Description
<b>Group name</b>	The name of the group definition.
<b>Recipe id</b>	The name of the recipe identifier associated with the group.
<b>Trigger id</b>	The name of the trigger identifier associated with the group.

## Copybook Converter Panel

In the Copybook Converter panel, you can select and convert a COBOL copybook member from a PDS into a buffer definition within the configuration file.

Optionally, you can create a message and a conversion rule during the copybook parsing and conversion process.

### Panel Layout

```
SXTP10----- Copybook Converter -----
Command

Copy Member          ===>

Buffer Identifier     ===>          (* to use Member name)
Message Identifier    ===>          (* to use Member name)
Convert Rule Id       ===>          (* to use Member name)

Library containing Copy Member:
    ===>
Additional libraries for COPY:
    ===>
    ===>
    ===>
    ===>

(Parser Debug level) ===> 0
```

### Field Description

The following table lists the fields in the Copybook Converter panel:

Field	Description
<b>Copy Member</b>	Required. The name of the copybook member to be selected from a PDS.
<b>Buffer Identifier</b>	Optional. The 16-character name for the buffer identifier. If you enter an asterisk (*), the copy member name is used.
<b>Message Identifier</b>	Optional. The 16-character name for the message identifier. If you enter an asterisk (*), the copy member name is used.

Field	Description
<b>Convert Rule Id</b>	Optional. The 16-character name for the convert rule identifier. If you enter an asterisk (*), the copy member name is used.  You can specify a convert rule identifier only if both a buffer identifier and a Rendezvous message identifier exist.
<b>Library containing Copy Member</b>	Required. The primary PDS that contains the selected copybook member.
<b>Additional libraries for Copy</b>	The DSNs of the PDS that can contain other copybooks that are included within the copybook member.
<b>Parser Debug level</b>	The parser debug level. Retain the default value zero (0) unless TIBCO Support requests otherwise.

## Log Viewer Panel

In the Log Viewer panel, you can select the recorded log and trace information written to a preallocated disk file.

### Panel Layout

```

SXLPO01----- Log Viewer -----
Command ==>

ISPF Log:
  Project . . . . . _____
  Group . . . . . _____
  Type . . . . . _____

Other VSAM Cluster Name:
  Cluster Name . . . _____

Display Format ==>          (1, 2)
Direction      ==>          (F-Forward B-Backward)
Active Log?    ==>          (Y/N)
Number to Read ==>          (Number to read initially)

VSAM Debug ==> (0 - 4)
Pgm Debug ==> (0 - 4)

Enter END command to terminate.
```

### Field Description

The following table lists the fields in the Log Viewer panel:

Field	Description
<b>ISPF Log</b>	You can specify a log to view by specifying the values for <b>Project</b> , <b>Group</b> , and <b>Type</b> ; or a fully qualified cluster name in the <b>Other VSAM Cluster Name</b> field. If this field is blank, the value of the <b>Other VSAM Cluster Name</b> field is used.

Field	Description
<b>Display Format</b>	<p>The format to display entries.</p> <p>The valid values are 1 and 2.</p> <ul style="list-style-type: none"> <li>1: one recorded entry per line (truncated when necessary).</li> <li>2: a two-line display where the description is on the second line.</li> </ul>
<b>Direction</b>	<p>The direction of reading entries. The Backward direction starts with the most recently recorded entries and reads backward for the specified number of records, thus showing the most current activity.</p> <p>Valid values: Forward and Backward.</p>
<b>Active Log?</b>	<p>If you specify Y, the log is treated as active (currently being updated by Substation ES). When the log is active, the index record is reread for each screen refresh to obtain the most current records.</p> <p>Valid selections: Y or N.</p>
<b>Number to Read</b>	<p>The number of recorded entries to display. If you do not specify a value, a default value of 200 is used.</p>
<b>VSAM Debug</b>	<p>The debugging level for the VSAM access routines. Do not change this value unless requested by TIBCO Support.</p> <p>The default value is zero (0).</p>
<b>Pgm Debug</b>	<p>The debugging level for the ISPF interface program. Do not change this value unless requested by TIBCO Support.</p> <p>The default value is zero (0).</p>

## Log Entries Panel

In the Log Entries panel, you can select a log or trace record written to the log or trace disk file.

### Panel Layout

SXLP002----- Log Entries -----					Row 1 of 29	
Command ==>					Scroll ==> PAGE	
Sel	Time	Type	Msg Id	RC	Rsn	Message
	11:09:54.0919	154	SXG1600I	0	0	Log Agent Starting - Logging to TIBLOG
	11:09:54.1126	154	SXG1800I	0	0	Parameter Log Agent Starting - Logging
	11:09:54.4368	154	SXS1000I	0	0	Starting ~ TIBCO Substation (ES) for 0
	11:09:54.4596	154	SXS1009I	0	0	Substation (ES) - Version 2.0.0
	11:09:54.7894	154	SXG2801I	0	0	Initialized ~ Substation ES - Admin &
	11:09:55.0454	154	SXG2800I	0	0	Starting ~ Substation ES - Transformer
	11:09:55.0479	0	SXT5717I	0	0	Trace level 1, 10 work threads; Cfg:TI
	11:09:55.2871	154	SXG2801I	0	0	Initialized ~ Substation ES - Transfor
	11:09:55.3874	154	SXG2800I	0	0	Starting ~ Substation ES - CICS Interf
	11:09:55.4673	154	SXC3000I	0	0	CICSTS4G - Connect IICICS to CICS Regi
	11:09:55.5539	154	SXC3001I	0	0	CICSTS4G - Connect IICICS to CICS Regi
	11:09:55.5590	154	SXG2801I	0	0	Initialized ~ Substation ES - CICS Int
	11:09:55.5950	156	SXC3400I	0	0	CICSTS4G - Started CICS EXCI Session (
	11:09:55.6130	156	SXC3400I	0	0	CICSTS4G - Started CICS EXCI Session (
	11:09:55.6352	154	SXG2800I	0	0	Starting ~ Substation ES - IMS Interfa
	11:09:55.6612	154	SXI4000I	0	0	A01IMSG - Connect IIIMS to IMS Region
	11:09:55.7283	154	SXI4001I	0	0	A01IMSG - Connect IIIMS to IMS Region
	11:09:55.7326	154	SXG2801I	0	0	Initialized ~ Substation ES - IMS Inte

```

11:09:55.7336 154 SXS1001I 0 0 Initialized ~ Substation ES for IVP's
11:09:55.8350 154 SXG1801I 0 0 Parameter Log closed
11:09:55.8362 154 SXS2120I 0 0 (B) HiMark:0 Occ:1024 Incr:0 Curr:0
11:09:55.8363 154 SXS2121I 0 0 (F) Stress Levels Strt:768 Slow:819 St
11:09:55.8408 154 SXS2120I 0 0 (A) HiMark:410 Occ:1024 Incr:102 Curr:
11:09:55.8504 154 SXG2806I 0 0 Activated ~ Substation ES - CICS Inter
11:09:55.8521 154 SXG2806I 0 0 Activated ~ Substation ES - Transforme
11:09:55.8525 0 SXT5036I 0 0 All transformer tasks are Good To Go
11:09:55.8525 154 SXG2806I 0 0 Activated ~ Substation ES - Admin & Op
11:09:55.8531 154 SXG2806I 0 0 Activated ~ Substation ES - IMS Interf
11:09:55.8610 154 SXG2806I 0 0 Activated ~ Substation ES for IVP's
***** Bottom of data *****

```

## Field Description

The following table lists the fields in the Log Entries panel:

Field	Description
<b>Sel</b>	If you specify s or x in the input column, a panel with all the values that relate to the selected message is displayed.  Valid selections: s and x.
<b>Time</b>	The time that this message was recorded at by Substation ES LTA Agent.
<b>Type</b>	The internal message type code.
<b>Msg Id</b>	The message identifier.
<b>RC</b>	The return code associated with the message.
<b>Rsn</b>	The reason code associated with the message.
<b>Message</b>	The initial part of the message description. The entire message is available if you select the panel with all the values that relate to the selected message.

## Single Message Values Panel

The Single Message Values panel displays the details of a log or trace record written to the log or trace disk file.

### Panel Layout

```

SXLP004----- Single Message Values ----- Row 1 of 4
Command ==>

Msg Id: SXG1600I          Union id: 0          Stck: B9EC624552B80481
Type: 154                GRIN: 0              RC: 0
Date: 2003/08/25         Proc Id: 67111444      Reason: 0
Time: 11:09:54.0919      Thread: 0

Message-Dependent Values:
  Buffer length - 40
-----
| No additional information
|
| Message:
| Log Agent Starting - Logging to TIBLOGF1
***** Bottom of data *****

```

## Field Description

The following table lists the fields in the Single Message Values panel:

Field	Description
<b>Msg Id</b>	An eight-character message identifier.
<b>Type</b>	The internal message type.
<b>Date</b>	The date that this message was recorded on by Substation ES LTA Agent.
<b>Time</b>	The time that this message was recorded at by Substation ES LTA Agent.
<b>Union id</b>	A representation of what message fields were recorded for this entry. See <i>TIBCO Substation ES Messages and Codes</i> for description.
<b>GRIN</b>	The internal Global Resource Identification Number.
<b>Pro Id</b>	The internal process identifier for identifying a Substation ES task or subtask.
<b>Thread</b>	The Substation ES thread that issues the message.
<b>Stck</b>	The internal 64-bit store clock value that uniquely identifies this message.
<b>RC</b>	The return code associated with the message.
<b>Reason</b>	The reason code associated with the message.
<b>Buffer Length</b>	A complete message description. If necessary, the message is displayed in multiple lines.



## Additional Configuration

You can use Substation ES to specify different interface configurations to meet various business requirements.

For example, you can configure a single Substation ES instance to simultaneously communicate with multiple CICS regions. Alternatively, you can configure multiple Substation ES instances to communicate with the same CICS region. The configuration choice depends on your system requirements.

### Adding an Additional Substation ES CICS Interface

You can add an additional Substation ES CICS Interface connection to a CICS region.

The current version of Substation ES supports about thirty interfaces per single started task.

#### Procedure

1. Add the Substation ES CICS resource definitions to the CICS region if this has not yet been done.
  - Repeat the step of the CICS Installation that updates the CSD.
  - Assign the correct CSD file to your JCL definition.
2. Create a new Substation ES CICS interface member by copying the default CICS SIP member located in `USERHLQ.INTF(SXCINTF)` to a new name.  
This new member must have a different `INTF-ID` keyword value. If connecting to a different CICS region, the `APPLID` keyword value must be changed.
3. Edit the Substation ES SIP member.  
The default member is `USERHLQ.INTF(SXSSIP$1)`.
4. Add a new `INTF-MEMBER` keyword, specifying the newly created CICS member as the keyword value.  
For example: `INTF-MEMBER=NEWINTF`
5. Go to the Substation ES configuration panels, `Option 4`, and add the interface to the existing list.



- Ensure that the interface Id is the same as specified in the interface member.
- Never use identical CICS `INTF-IDs` within a single region of Substation ES.

### Adding an Additional ESB Interface

You can add an additional TIBCO ESB Interface (formerly referred to as the Transformer) connection to a CICS region.

#### Procedure

1. Add a Substation ES CICS Interface by performing the steps in [Adding an Additional Substation ES CICS Interface](#).
2. Create a new ESB Interface member by copying the default ESB SIP member located in `USERHLQ.INTF(SXSIEMS1` or `SXSIRV1)` to a new name.  
This new member must have a different `INTF-ID` keyword value.
3. Change the new Substation ES CICS interface member value of `INTF-PARTNER` to match the new value of `INTF-ID` in [Step 2](#).
4. Edit the Substation ES SIP member.  
The default value is `USERHLQ.INTF(SXSSIP$1)`.

5. Add a new `INTF-MEMBER` keyword, specifying the newly created ESB SIP member as the keyword value.

For example: `INTF-MEMBER=NEWINTF`



Never specify identical `INTF-ID` values in the ESB SIP members.

## Adding a Rendezvous ESB Interface with SSL

Adding SSL for a Rendezvous interface is done in the Rendezvous daemon definition, not in Substation ES.

You can ask the Rendezvous administrator to set up a Rendezvous SSL interface.

## Adding Substation ES Instances to a Single CICS Region

You can configure multiple Substation ES instances that communicate to a single CICS region.

To share one ESB configuration file with multiple instances of Substation ES, distinguish between recipes or triggers definitions for the Substation ES instances with groups. For details, see [Defining Groups](#).

The current version of Substation ES supports up to five Substation ES instances connected to a single CICS region.

For each Substation ES instance, perform the following steps:

### Procedure

1. Create a copy of the Substation ES SIP for CICS interface member `USERHLQ.INTF(SXCINTF)`. In the new member specify a new `INTF-ID` keyword value.
2. Create a copy of Substation ES SIP ESB interface member `USERHLQ.INTF(SXSIEMS1 or SXSIIRV1)`. Make parameter changes that are necessary.
3. Optional: Create a copy of the Substation ES SIP for the Admin interface member `USERHLQ.INTF(SXSIADM)`. In the new member, specify the corresponding ESB `INTF-ID` from [Step 2](#) to the new `ESB-INTF-ID` keyword value.  
This step is required only if Substation ES Heartbeat communications is used.
4. Create a new copy of the Substation ES system initialization parameters SIP member `USERHLQ.INTF(SXSSIP$1)`.
5. Change the `INTF-MEMBER` entries in the Substation ES SIP member for each of the newly created CICS and ESB and Admin Interface member names.
6. Change the Substation ID in the Substation ES SIP.



Each active Substation ES must have a unique Substation ID.

The number of HVT workers for Reliable and Guaranteed must be the same in all Substation ES definitions going to the same CICS.

7. Create a new copy of the Substation ES system startup parameters (SSP) member `USERHLQ.INTF(SXSSSP$1)`.
8. Change the `SIPMEM` keyword value and point it to the newly created `SXSSIP$1` member.
9. In the Substation ES startup JCL or Procedure, change the `PARM` card value to point to the newly created Substation ES SSP member, for example: `'-SSPMEM NEWSSP$1'`

## Using Connection Factories

Substation ES uses the TIBCO Enterprise Message Service connection factory for all EMS connections. Substation ES creates a default Connection Factory Object when the ESB interface starts, and then

populates the Connection and Re-Connection Parameters using the definition from the Substation ES Configuration File.

### Implementing Automatic Reconnect

To implement Auto Reconnect to the EMS ESB endpoints, you must update the EMS Server URL definition in the Configuration File and specify this in the same way you would for any EMS URL definition. For example, with a non FT server, specify the URL twice separated by a comma:

```
ems_server_url,ems_server_url
```



There are no spaces between the URL names. For example:  
192.168.1.100:7888,192.168.1.100:7888

The Reconnect parameters can be customized on the ECF panel. These parameters are listed as follows:

- **Attempts**
- **Delay**
- **Timeout**


Use caution when setting these values or changing them from the product default settings because Mainframe values must be a little higher than a local connection.

## Using RVDQ with Multiple Substation ES Instances

This section describes the process taken to configure multiple Substation ES instances to use RV distributed queue processing to communicate with a single CICS region.

The following steps configure three Substation ES instances:

### Procedure

1. Create three CICS SIP members, specifying the same INTF-ID for each.  
This INTF-ID must match the Interface Id defined in the ESB configuration file (on the Define Processing System Resources panel, option 4).  
In this example, the new CICS SIP members are named SXCINTF1, SXCINTF2 and SXCINTF3.
2. Create three RV ESB SIP members, setting RVDQ-YN set to Y for all three. Each must also have a unique INTF-ID.  
In this example, the new RV ESB SIP members are named SXSIRV1, SXSIRV2 and SXSIRV3.
3. Optional: Create three copies of the Substation ES SIP for Admin interface members. Name the copies SXSIADM1, SXSIADM2 and SXSADM3. In each new member, specify the corresponding ESB INTF-ID from [Step 2](#) to each new ESB-INTF-ID keyword value.  
 This step is required only if Substation ES Heartbeat communications is used.
4. Create three Substation ES SIP members, specifying a unique SUBSTATION-ID for each, and entering INTF-MEMBER settings for the corresponding CICS SIP and RV ESB SIP members that were created in [Step 1](#) and [Step 2](#).  
In this example, the new Substation ES SIP members are named SXSSIP\$1, SXSSIP\$2 and SXSSIP\$3:
  - SXSSIP\$1 has the INTF-MEMBER for SXCINTF1, SXSIRV1, and SXSIADM1.
  - SXSSIP\$2 has the INTF-MEMBER for SXCINTF2, SXSIRV2, and SXSIADM2.
  - SXSSIP\$3 has the INTF-MEMBER for SXCINTF3, SXSIRV3, and SXSIADM3.
5. Create three SSP members, specifying the SIPMEM setting for the corresponding Substation ES SIP members that were created in [Step 4](#).

In this example, the SSP members are named SXSSSP\$1, SXSSSP\$2 and SXSSSP\$3:

- SXSSSP\$1 has the SIPMEM for SXSSIP\$1.
  - SXSSSP\$2 has the SIPMEM for SXSSIP\$2.
  - SXSSSP\$2 has the SIPMEM for SXSSIP\$3.
6. Create three Substation ES startup JCL members, using the same ESB configuration file for all three. In this example, the Substation ES JCL members are named TIBSSES1, TIBSSES2 and TIBSSES3:
    - In TIBSSES1 specify PARM='-SSPMEM SXSSSP\$1'.
    - In TIBSSES2 specify PARM='-SSPMEM SXSSSP\$2'.
    - In TIBSSES3 specify PARM='-SSPMEM SXSSSP\$3'.
  7. Start all Substation ES instances.
 

All instances use RV distributed queue processing. The first started Substation ES instances acts as the scheduler.

## Using RVDQ with One Substation ES Instance

This section describes the process taken to configure one Substation ES instance to use RV distributed queue processing to communicate with a single CICS region or multiple CICS regions.

The following steps configure three RV ESB SIP and CICS SIP members for one Substation ES instance:

### Procedure

1. Create three ESB configuration files.
 

Each file must have a CICS BES Interface ID unique but otherwise contain identical recipe settings. The CICS BES Interface ID is set on the Transformer Configuration panel, option 4.

In this example, the CICS Interface IDs for the new ESB configuration files are named CICS\_A, CICS\_B and CICS\_C.
2. Create three CICS SIP members.
 

Each member must have a unique INTF-ID. This INTF-ID must match the Interface Id defined in the corresponding ESB configuration file. If this is for multiple CICS regions, the APPLID must have the corresponding generic CICS application ID.

In this example, the new CICS SIP members are named SXCINTF1, SXCINTF2 and SXCINTF3.
3. Create three RV ESB SIP members, setting RVDQ-YN to Y for all three.
 

Each must also have a unique INTF-ID. In this example, the new RV ESB SIP members are named SXSIRV1, SXSIRV2 and SXSIRV3.
4. Create one Substation ES SIP member, entering all INTF-MEMBER settings for the corresponding CICS SIP and RV ESB SIP members.
 

In this example, the new Substation ES SIP member is named SXSSIP\$1.
5. Create one SSP member, specifying the SIPMEM setting for the corresponding Substation ES SIP member.
 

In this example, the SSP member is named SXSSSP\$1.
6. Create one Substation ES startup JCL member, using three ESB configuration files for the corresponding CICS SIP and RV ESB SIP members.
 

In this example, the Substation ES JCL member is named TIBSSES1, and uses three configure files named TIBCFG1, TIBCFG2 and TIBCFG3:

  - TIBCFG1 corresponds to CICS SIP SXCINTF1 and RV ESB SIP SXSIRV1.

- TIBCFG2 corresponds to CICS SIP SXCINTF2 and RV ESB SIP SXSIRV2.
  - TIBCFG3 corresponds to CICS SIP SXCINTF3 and RV ESB SIP SXSIRV3.
7. Start the Substation ES instance.
- All interfaces use RV distributed queue processing. The last startup interface acts as the scheduler.

## Using Opaque Fields and Message Sizes

When using RV ESB, you can configure and use Opaque field and message size settings. With the settings, Substation ES can calculate the size of data communication with the BES (CICS or IMS).

By default, the RV ESB uses the size defined in the message buffer definition to cater for outbound data. To configure Substation ES to use the message data size instead, you must define the message definition and the conversion rule for a recipe service in the appropriate manner for reply data to be sent efficiently over the network.

The conditions for this are listed as follows:

- Enter only one field in the output conversion rule.
- In the buffer definition, specify a buffer offset of zero.
- In the message definition:
  - In the output message, the **Receiving** field is the Opaque data type.
  - The **Decimal** indicator on the message has a value greater than zero.
- In the Conversion rule definition, the output conversion rule **Field SZ** is set to M.

The conversion then uses either the defined buffer size or the returned data size, based on the size field in the conversion rule, to create the outbound message.

# Utilities for Configuration File

This section describes the three utilities for the TIBCO Substation ES configuration file.

This section includes the following parts:

- [Overview](#)
- [Conversion Utility](#)
- [Configuration File Utility](#)
- [Selective Export Utility](#)

## Overview

Before using Substation ES, you must supply the information in the configuration file to Substation ES ESB Interface. Substation ES ESB Interface requires this configuration file information to initialize successfully. Information includes data about connecting to messaging daemons and servers used for message communication.

The structure of the configuration file has changed in this release and can change in future releases as new features and improvements are made. A configuration conversion utility is supplied to reformat your definitions when you upgrade from one release to another. See the *TIBCO Substation ES Release Notes* for the release of interest. You can then determine whether conversions are necessary and what to do to upgrade.

A Configuration File must be allocated with IDCAMS. See [Allocation of Configuration File](#) for details.

Substation ES contains configuration utilities to assist you to use, maintain, and promote definitions in configuration files.

## Conversion Utility

The Configuration File's Conversion Utility extracts the configuration entities from a Substation ES configuration file and creates a sequential EXPORT file. You can then import the EXPORT file to a Substation ES configuration file with the Configuration File Utility.

For details on how to create and allocate configuration files, see [Allocation of Configuration File](#).

## Function

The Configuration File Conversion Utility extracts the entities from a Substation ES configuration file created by a previous release of Substation ES.

These utility JCLs are included with Substation ES:

- SXSCFC26: convert a version 2.6.x configuration file to a version 2.7.0 configuration file.
- SXSCFC24: convert a version 2.1.x through V2.5.x configuration file to a version 2.6.0 configuration file.
- SXSCFC21: convert a version 2.0.0 configuration file to a version 2.1.0 configuration file.
- SXSCFC20: convert a version 1.x configuration file to a version 2.0.0 configuration file.

## JCL Sample (SXSCFC26)

Execution is controlled by parm values and JCL statements.

```
//EXPORT EXEC PGM=SXT3CV26,REGION=OM,PARM='parm values'
//STEPLIB DD DISP=SHR,DSN=< USERHLQ>.LOAD
//TIBLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

```
//CONFIG DD DISP=SHR,DSN=old.config.file
//EXPORT DD DISP=(,CATLG),DSN=new.export.file,
//          UNIT=SYSDA,SPACE=(TRK,(20,20)),
//          DCB=(BLKSIZE=14000.LRECL=350,RECFM=VB)
```

The following table shows Configuration File Conversion Utility DDNames:

Field	Description
<b>CONFIG</b>	Points to the old Transformer Configuration file that provides the input data to the conversion process.
<b>EXPORT</b>	Points to a new EXPORT file that is written during an export operation.

## Parameter Values

The following table describes the values for the **parm** field of the EXEC statement. All the values are optional; the conversion process can proceed with no parm data other than SXSCFC26.

Value	Meaning	Default	Explanation
-D <i>n</i>	Debug	2	Specifies the level of message display. The value for <i>n</i> can be between 0 and 5, with 0 providing very few messages and 5 showing the hex displays of all the records processed.
-Ixxx	Input DD name	CONFIG	Enables another DD name to be used in place of CONFIG.
-Nxxx	Export DD name	EXPORT	Writes out the entire contents of CONFIG in export format to the EXPORT file. No selection is provided; the entire file is processed. Specify xxx as the DD name to replace EXPORT.
-P	Print	None	Prints each recipe and trigger with the associated Backend, ESB Endpoint, Group membership, and Conversion rules.
-R	Read-only	Update	Opens the CONFIG file as read-only. If combined with Import, all the updates from the import operation are lost.
-T <i>n</i>	Table debug	2	Defines the diagnostic level for the internal TABLE processor, which ties together information from the various VSAM-keyed records.
-X	Hex dump	None	Sequentially reads and hex-displays the entire configuration file as a diagnostic aid.

## Configuration File Utility

The Configuration File utility performs numerous functions that relate to a configuration file.

### Functions

The functions which the Configuration File utility performs include the following items:

- Populates (imports) a configuration file from an **IMPORT** file previously exported by one of the configuration utilities.
- Copies all the entities from one configuration file to another.
- Extracts all the entities from a configuration file to a sequential **EXPORT** file.
- Prints the definitions for all the entities within a configuration file.
- Validates that configuration file entities are consistent and related.

The Configuration File Utility is usually used to take a sequential file created by one of the utilities and update the contents to a Substation ES configuration file. Configuration file entities, such as messages, conversion rules, recipes, trigger definitions, and so forth, are read from the file referenced on the **IMPORT DD Name** and inserted to the file referenced on the **CONFIG DD Name** file.

When importing configuration file entities that already exist in the destination file, the entities are updated and not duplicated.

### JCL Sample (SXSCFIMP and SXSCFEXP)

Execution is controlled by values on **PARM** statement and JCL parameters. See the following JCL sample.

```
//STEP0001 EXEC PGM=SXT3CFUT,REGION=0M,PARM='values '
//STEPLIB DD DISP=SHR,DSN= USERHLQ.LOAD
//TIBLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CONFIG DD DISP=SHR,DSN=config.file
//CONFIG2 DD DISP=SHR,DSN=output.config.file
//IMPORT DD DISP=SHR,DSN=import.file
//EXPORT DD DISP=(,CATLG),DSN=export.file,
// UNIT=SYSDA,SPACE=(TRK,(20,20)),
// DCB=(BLKSIZE=14000.LRECL=350,RECFM=VB)
//
```

The following table describes the DD names of the Configuration File Utility:

Field	Description
<b>CONFIG</b>	Points to the current Transformer configuration file that all operations are to be performed upon.
<b>CONFIG2</b>	Points to a new copy of the Transformer configuration file, which is updated during a copy operation.
<b>IMPORT</b>	Points to an existing <b>IMPORT</b> file, which is read during an import operation.
<b>EXPORT</b>	Points to a new <b>EXPORT</b> file, which is written during an export operation. This file can be a preallocated disk file.

For a summary of the usages of the DD names, see [Summary of Substation ES ESB Configuration File Utilities](#).



## Parameter Values

The following table describes the values you can enter in the **PARM** field of the EXEC statement. In the absence of function action requests, the default action is print.

Value	Meaning	Default	Explanation
-C	Copy	None	Copies all the definitions from one configuration file to another file defined by CONFIG2. The file referenced by CONFIG2 is initialized before copying starts.
-Mddn	Import	None, IMPORT	Reads the contents of the IMPORT file and apply to the CONFIG file. If you specify <i>ddn</i> , that DD name is used instead of IMPORT.
-Nddns	Export	None, EXPORT	Writes out the entire contents of CONFIG in export format to the EXPORT file. If you specify <i>ddn</i> , that DD name is used instead of EXPORT.

The following table describes the parameter values of the Configuration File Utility:

Value	Meaning	Default	Explanation
-Dn	Debug	2	Specifies the level of message display. The value for <i>n</i> can be between 0 and 5, with 0 providing very few messages and 5 showing the hex displays of all the records processed.
-Ixxx	Input DD name	CONFIG	Allows another DD name to be used in place of CONFIG.
-Oxxx	Output DD name	CONFIG2	Allows another DD name to be used in place of CONFIG2.
-P	Print	None	Prints out each recipe and trigger with the associated Backend, ESB Endpoint, Group membership, and Conversion rules.
-R	Read-only	Update	Opens the CONFIG file as read-only. If combined with Import, all the updates from the import operation are lost.
-T	Initialize	None	Allows the program to initialize the CONFIG file if the file does not contain the <code>Transformer_Anchor</code> structure at the beginning of the file. The Initialize request does not apply if you also specify read-only.
-V	Validate	None	Validates all the data structures for consistency and block identifiers (eyeball characters). If you specify -V -v, validation does not terminate after the first error occurs.

Value	Meaning	Default	Explanation
-X	Hex dump	None	Displays all the active definitions in the file in hex and does not show the relationships among the definitions.
-Z	Zero file	None	Resets the CONFIG file to empty before starting the requested processes. Note the warning as follows.



Do not specify the z option when importing data into a nonempty configuration file. Doing so destroys all the existing data in the configuration file.

For a summary of the usage of the parameter options, see [Summary of Substation ES ESB Configuration File Utilities](#).

## Selective Export Utility

The Configuration File's Selective Export Utility SXT3CSIE exports entities and definitions that relate to a configuration file.

### Functions

The Configuration File's Selective Export Utility SXT3CSIE exports entities and definitions that relate to a configuration file.

The SXT3CSIE utility performs the following tasks:

- Extracts all or the selected entities within a configuration file.
- Writes the selected entities to an sequential EXPORT file.

This utility exports “logical pieces” of a configuration file to a sequential file. The entities to export are selected by reading the specified control parameters. To import the entities in the exported sequential file, use the [Configuration File Utility](#).

### JCL Sample (SXSCFEXS)

Execution is controlled by PARM values, JCL statements, and parameters in the CONTROL DD name. See the following JCL sample:

```
//STEP0002 EXEC PGM=SXT3CSIE,REGION=0M,PARM='values'
//STEPLIB DD DISP=SHR,DSN=USERHLQ.LOAD
//TIBLOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CONFIG DD DISP=SHR,DSN=config.file
//EXPORT DD DISP=(,CATLG),DSN=export.file,
// UNIT=SYSDA,SPACE=(TRK,(20,20)),
// DCB=(BLKSIZE=14000.LRECL=350,RECFM=VB)
//CONTROL DD *
(export control parameters)
//
```

The following table describes the DD names of the Selective Export Utility:

Field	Description
CONFIG	Points to a current linear Transformer Configuration file.

Field	Description
<b>EXPORT</b>	Points to a new EXPORT file that is written during an export operation. This file can be a preallocated disk file.

For a summary of the usages of DD names, see [Summary of Substation ES ESB Configuration File Utilities](#).

## Parameter Values

The following table describes the values you can enter possible in the **PARM** field of the EXEC statement.

Value	Meaning	Default	Default Explanation
-Cxxx	Control DD name	CONTROL	Allows another DD name to be used in place of CONTROL.  Supports FB PDS data sets only.
-Dn	Debug	2	Specifies the level of message display. The value for <i>n</i> can be between 0 and 5, with 0 providing very few messages and 5 showing the hex displays of all the records processed.
-Ixxx	Input DD name	CONFIG	Allows another DD name to be used in place of CONFIG.
-Nxxx	Export DD name	EXPORT	Allows another DD name to be used in place of EXPORT.
-P	Print	None	Prints out each export recipe and/or trigger with the associated Backend, ESB Endpoint, Group membership, and Conversion rules.
-R	Read-only	Update	Opens the CONFIG file as read-only mode.

For the parameter options, see [Summary of Substation ES ESB Configuration File Utilities](#).

## Control Parameters

The processing of the utility is specified by a series of export or delete control statements defined in the CONTROL DD statement.

The format of each statement is as follows:

- EXPORT *type identifier qualifiers*
- REP *field Replacement-Value*
- DEL *type identifier*

where the *identifier* is the entity name. Note that *identifier* and *Replacement-Value* are case sensitive.

You must specify the identifier name for DEL. The value cannot be an asterisk (\*). The same deletion rules apply as in the ISPF deletions: The entry to be deleted cannot be used by any other entity.

## Entity Type Values

You must specify an entity *type* value in an export control statement. The following table shows the entity type values:

Value	Description
TRIGGER	For trigger definitions only.
RECIPE	For recipe definitions only.
BACKEND	For BES definitions only.
ESB	For ESB definitions only.
BUFFER	For buffer definitions only.
MSG	For message definitions only.
CONVERTER	For conversion-role definitions only.
GROUP	For group definitions only.
ALL	For all definitions.

## Identifier Values

You must specify a value for *identifier* to identify an entry. The identifier is the ID value of a specific entry:

- An identifier must be entered.
- If the type is `all`, the identifier must be an asterisk (\*).

## Qualifier Values

A *qualifier* can be entered depending on the *type* specified. If no qualifiers are entered, the default is `ALL`.

If *type* is `all`, the default *qualifier* value is `ONLY`.

A *qualifier* value that does not apply to *type* is ignored.

The following table shows the definition *qualifier* values:

Value	Description
ALL	Exports all the related definitions of the selected <i>type</i> .
ONLY	Exports only the selected <i>type</i> definition and does not export the related structure definitions.
BUF	Exports the related buffer definitions for the selected <i>type</i> .
MSG	Exports the related message definitions for the selected <i>type</i> .
CVR	Exports the related Conversion Rule definitions for the selected <i>type</i> .

Value	Description
BES	Exports the related BES definitions for the selected <i>type</i> .
TRN	Exports the related ESB definitions for the selected <i>type</i> .
GRP	Exports the related Group definitions for the selected <i>type</i> .

### REP Control Parameter

The **REP** control parameter provides flexibility to administrators and allows you to manage environmental specific information when moving configuration entities from TEST to QA to Production. The following are the only entity field names allowed to be specified when creating an export file for migration. The REP statement must follow an EXPORT statement and applies only to the previously specified EXPORT ENTITY type.

The following table shows the replaceable fields:

ENTITY Type	Field Name	Description
BACKEND	<b>Intf-Id</b>	Interface Id field
ESB	<b>URL</b>	Primary URL definition
	<b>URL-ECF</b>	EMS Connection Factory URL definition
RECIPE	<b>BES-Name</b>	Back-end system name specified for endpoints
	<b>BES-ALT-Name</b>	Back-end system name specified for endpoints
	<b>CICS-System-Name</b>	CICS AOR routing system identifier
	<b>ESB</b>	ESB name specified for endpoints
	<b>Listen-To</b>	The subject or destination name of the recipe to listen to
	<b>Publish-To</b>	The subject or destination name of the recipe to publish to.
	<b>Error-To</b>	The error subject or destination name of the recipe to publish to.
TRIGGER	<b>Trace-Lvl</b>	The user trace debug level for an individual recipe. Valid range: 0 through 5.
	<b>ESB</b>	ESB name specified for endpoints.
	<b>Publish-To</b>	The subject or destination name of the trigger to publish to.

ENTITY Type	Field Name	Description
	<b>Publish-To-Sec</b>	The second subject or destination name of the trigger to publish to.
	<b>Reply-To</b>	The reply subject or destination name for the trigger service.
	<b>Error-To</b>	The error subject or destination name of the trigger to publish to.
	<b>Trace-Lvl</b>	The user trace debug level for an individual trigger. Valid range: 0 through 5.

### Examples

Here are a few examples:

- To export a recipe definition named `rcp1` and all its related structures:  
`EXPORT RECIPE rcp1`
- To export a recipe definition named `rcp1` and all its related but completely qualified structures instead of accepting the default qualifier `ALL`:  
`EXPORT RECIPE rcp1 BUF MSG CVR BES TRN GRP`
- To export an entire file:  
`EXPORT ALL * ONLY`  
`RVY3CFUT` with the parameter `-N` has the same capability.
- To export all ESB definitions only:  
`EXPORT ESB * ONLY`
- To delete a recipe definition named `rcp1` and all its related structures:  
`DEL RECIPE rcp1`
- To delete a buffer definition named `buf2` and all its related structures:  
`DEL BUFFER buf2`

# Examples

---

This section contains examples that demonstrate how to use Substation ES.

This section include the following topics:

- [Overview](#)
- [Substation ES Installation IVPs](#)
- [Substation ES Communication Programs](#)
- [Example of ESB Request – Reply DPL](#)
- [Example of CICS Trigger](#)
- [Example of CICS High Volume Trigger](#)
- [Example of ESB Request DPL XCTL](#)
- [Example of ESB Invoking a CICS Transaction](#)
- [Example of ESB to TDQ](#)
- [Example of CICS TSQ Entry](#)
- [Example of ESB Request – Reply Complex DPL](#)
- [Example of CICS-Initiated Request or Reply](#)
- [Example of ESB Request – Reply IMS](#)
- [Example of ESB Request – Reply and Trigger](#)
- [Example of IMS BMP Trigger](#)
- [Example of IMS Synchronous Callout — Request or Reply](#)
- [Example of IMS Conversational Functions](#)

## Overview

Substation ES receives messages on a specified subject located in a recipe defined in the Transformer's configuration file. A sample host-side application that performs the required function is initiated.

Sample code has been included to help you understand how each host-side application is initiated and what happens as a result of the function that is performed. The request or reply applications are written in C and run on the z/OS host.

The IVP Sender and IVP Generic Listener programs in the previous versions have been combined into a single program called IVP Sender Listener in Substation ES version 2.3. The modules are SXJ3ISL1 for TIBCO Enterprise Message Service and SXR3ISL1 for TIBCO Rendezvous.

Also included is an Error listener source SXR3IELS, which listens for errors published by instances of TIBCO Substation ES. For summary usage of sample IVPs resources, see [Sample IVP Resources](#).

Before executing IVPs, it is good practice to start the Generic Listener and the Generic Error Listener so that you can see the printed output, including errors, while executing.



For more information about Generic Listener, see Generic Sender or Listener Utility in the *TIBCO Substation ES Operations and Administration* manual. For more information about Generic Error Listener, see Generic Error Listener Utility in the same manual.

## Substation ES Installation IVPs

During the installation of Substation ES, IVP programs are supplied by default. To ensure that the network, Substation ES, and the back-end system (BES) interfaces (CICS and IMS) are correctly executing bidirectional communication with Substation ES, execute those IVP programs.

The installation default transformer configuration file contains all the entries required for executing Substation ES IVP programs.

### Request or Reply

To demonstrate request or reply message processing, the Substation ES IVP Sender Listener publishes a message (using a selected messaging application) to Substation ES. Substation ES receives the message, transforms the data contents, and forwards the information to the appropriate BES.

Upon completion, an acknowledgment or result set is returned to the Substation ES IVP Sender Listener through Substation ES.

### Triggers

Triggers are BES (CICS or IMS) applications that initiate published data from their respected environments. Substation ES can publish data on behalf of these BES (CICS or IMS) applications.

In this case, the data is translated from the z/OS data format and packaged as a TIBCO Rendezvous message for delivery to an external application that is subscribing to this data.

Upon completion, an acknowledgment or result set is returned to Substation ES IVP listener through Substation ES.

## Substation ES Communication Programs

This section describes the generic IVP components in Substation ES.

### Rendezvous IVP Sender Listener

The program `SXR3ISL1` works as a TIBCO messaging requestor application. It publishes Substation ES IVP requesting messages based on a set of parameters passed in through `SYSIN DD` statements in the run JCL and then waits for a reply from Substation ES.

If an error occurs, the process stops; otherwise, the process loops on the number specified in the **REPEAT** parameter.

When the BES application executes, it returns a reply; the messages are consumed and reported by this Substation ES IVP Rendezvous sender listener. Errors are also reported. The following two run JCLs are available:

Data Set: `USERHLQ.JCL`

Member:

- `SXCRIRSL` for CICS Interface IVPs
- `SXIRIRSL` for IMS Interface IVPs



The `EXEC PARM - IVPNETW` is substituted with a set of TIBCO Rendezvous transport parameters. It is recommended that you use the provided default settings for all IVP runs. To use the other settings, see the *TIBCO Rendezvous Administration* manual for information about the parameters.

### SYSIN Parameters

The **SYSIN** parameter members are in the following member:

Data Set: `USERHLQ.CNTL`

Member:



- SXCRIP01 for CICS Interface IVPs
- SXIRIP01 and SXIRIP02 for IMS Interface IVPs

The following example shows the input control statements in the **SYSDIN** parameter member:

```
DEBUG ON
*
SUBJ-REQ tibss.IMS.Tran.Request
WAIT 0
*
```

The requesting messages that the IVP Sender Listener publishes are determined by the keywords and operands in the SYSDIN DD Name file. See the following table for the control parameters.

The following table shows the Rendezvous IVP Sender Listener control parameters:

Keyword – Operand	Description
BES <i>name</i>	The BES interface ID for the tibss-BES system field. The default is a blank.
BES-DELAY <i>timevalue</i>	The length of time (in seconds) to wait between message replies on the CICS side. The value must fall between 0 and 60. The default is 0.
CLIENT-ID <i>name</i>	The identifier to use with the Rendezvous daemon connection.
DAEMON <i>URL:port</i>	The Rendezvous daemon URL and TCP port.
DEBUG ON OFF	On or Off. When On, the debug mode is enabled.
DESCR <i>name</i>	The 56 characters for the <b>IVP-DESCR</b> field.
IGNORE-TIMEOUT	When specified, the application continues to send the next request message even if the previous request message reached its timeout before receiving a reply.
INPUT-DDN <i>name</i>	The input data DDNAME to use for test data. The default is TIBDATA.
INPUT-FLDNAME <i>name</i>	The field name to be used for data. The default is DATA.
INPUT-FLDTYPE <i>name</i>	The field type to use for data. May be OPA or STR. The default is OPA.
INPUT-FLDRECORD <i>number</i>	Indicates the position of the record to be used in the INPUT file. The default is 1, indicating the first record of the file.
IVP-ID <i>name</i>	The 8 characters for the <b>IVP-ID</b> field.
NETWORK <i>address</i>	The Rendezvous network parameter.
PATTERN-LEN <i>number</i>	The creation of the variable field <b>IVP-STRING</b> of length <i>number</i> , populated with 1-----10-----.... The default is 0.

Keyword – Operand	Description
PSWD <i>password</i>	The password for the user ID previously specified and authorized to execute this process. This password depends on the Security Interface and is for the <b>tibss-password</b> system field. The default is a blank.
PUBLISH-ONLY	When included, the application sends messages but does not listen for replies.
REPEAT <i>number</i>	The number of times to repeat the send process. The default is 1.
RESOURCE <i>name</i>	<p>The name of the <b>tibss-resource</b> system field, up to a maximum of eight characters. The default is a blank.</p> <p>If present, this value overrides the recipe <b>Resource Name</b> field, which depends on the Method on Invocation. For details, see Resource Name in <a href="#">Recipe Details Panel</a>.</p>
SERVICE <i>port</i>	The port number of the Rendezvous UDP service.
SUBJ-REP <i>name</i>	The name of the Rendezvous reply subject to which the reply message is sent. If not PUBLISH-ONLY and SUBJ-REP are absent, a private inbox subject name is used.
SUBJ-REQ <i>name</i>	The Rendezvous request subject name to which the message is sent.
SUBJ-REQ-DYNAMIC	<p>When specified, directs the application to send each request message with dynamic subject name. The subject name is based on the input value of REPEAT.</p> <p>For example, if SUBJ-REQ is <code>tibss.CICS.C.Request</code> and REPEAT is <code>xx</code>, then the first message is sent to <code>tibss.CICS.C.Request.N1</code> the <code>xx</code>th message is sent to <code>tibss.CICS.C.Request.Nxx</code>.</p>
TIMEOUT <i>timevalue</i>	The length of time (in seconds) for the response timeout. The default is 5.
USER <i>userid</i>	The RACF or back-end identifier for the user who is authorized to execute this process. This Id depends on Security Interface. For the <b>tibss-userid</b> system field; the default is a blank.
WAIT <i>timevalue</i>	<p>The length of time (in seconds with decimals) to wait between message sends. The default is 0.</p> <p>For example: 1.5</p>
* <i>text</i>	If the line begins with an asterisk (*), the following text becomes a comment.

## Enterprise Message Service IVP Sender Listener

The program SXJ3ISL1 works as a TIBCO messaging requestor application. It publishes Substation ES IVP requesting messages according to a set of parameters passed in through `SYSIN DD` statements in the run JCL, then waits for a reply from Substation ES.

If an error occurs, the process stops; otherwise, the process loops on the number specified in the **REPEAT** parameter.

When the BES application executes, it returns a reply; the messages are consumed and reported by this Substation ES IVP TIBCO Enterprise Message Service Sender Listener. Errors are also reported. The following two run JCLs are available:

Data Set: `USERHLQ.JCL`

Member:

- `SXCRIJSL` for CICS Interface IVPs
- `SXIRIJSL` for IMS Interface IVPs



The `EXEC PARM - IVPNETW` is substituted with a set of TIBCO Enterprise Message Service ESB parameters. It is recommended that you use the provided default settings for all IVP runs. To use other settings, see the *TIBCO Enterprise Message Service User's Guide* for information about the parameters.

### SYSIN Parameters

The `SYSIN` parameter members are in the following:

Data Set: `USERHLQ.CNTL`

Member:

- `SXCJIP01`, `SXCJIP02`, and `SXCJIP03` for CICS Interface IVPs
- `SXIJIP01` and `SXIJIP02` for IMS Interface IVPs

Following is a sample of the input control statements in the `SYSIN` parameter member:

```
DEBUG OFF
*
DEST-REQ tibss.IMS.Tran.Request
DEST-REP tibss.IMS.Tran.Reply
TYPE QUEUE
WAIT 0
*
```

The requesting messages that the IVP Consumer Producer publishes are determined by the keywords and operands in the `SYSIN DD` Name file. See the following table for the control parameters.

The following table shows the EMS IVP Sender Listener control parameters:

Keyword – Operand	Description
ACKNOWLEDGE {N A}	Specifies either no acknowledgment or automatic acknowledgment to EMS server for this message. Values are: <ul style="list-style-type: none"> <li>• N: Non acknowledgment</li> <li>• A: Acknowledgment</li> </ul>
BES <i>name</i>	The BES interface ID for the <b>tibss-BES</b> system field. That ID can be sent as a user property field or MAP message. The default is a blank.

Keyword – Operand	Description
BES-DELAY <i>timevalue</i>	The length of time (in seconds) to wait between message replies on the CICS side. The value must fall between 0 and 60. The default is 0.
CLIENT-ID <i>name</i>	The identifier to use with the EMS connection.
DEBUG ON OFF	On or Off. When On, the debug mode is enabled.
DELIVERY {R N P}	The message delivery mode. Specify one of: <ul style="list-style-type: none"> <li>• R: Reliable</li> <li>• N: Non-Persistent</li> <li>• P: Persistent</li> </ul>
DEST-REP <i>name</i>	The name of the EMS destination to which the reply is to be sent. If not PUBLISH-ONLY and DEST-REP are absent, a temporary topic or queue is used.
DEST-REQ <i>name</i>	The name of the EMS subject destination to which the message is sent.
DESCR <i>name</i>	The 56 characters for the <b>IVP-DESCR</b> field.
EMS-PSWD <i>password</i>	The password to connect to the EMS server.
EMS-USER <i>userid</i>	The user ID to connect to the EMS server.
HOSTPAGE <i>code page</i>	The value for setting the host code page. The default is IBM-1047.
IGNORE-TIMEOUT	When specified, the application continues to send the next request message even if the previous request message reached its timeout before receiving a reply.
INPUT-DDN <i>name</i>	The input data DDNAME to use for test data. The default is TIBDATA.
INPUT-FLDNAME <i>name</i>	The field name to be used for data. The default is DATA.
INPUT-FLDTYPE <i>name</i>	The field type to use for data. May be OPA or STR. The default is OPA.
INPUT-FLDRECORD <i>number</i>	Indicates the position of the record to be used in the INPUT file. The default is 1, indicating the first record of the file.
IVP-ID	The maximum eight characters for the <b>IVP-ID</b> field.
IVP-WRITES <i>number</i>	The response size of the IVP sample (SXCCI015) for each container. The total size of the IVP sample is the IVP default of 384 plus the size specified with this parameter.
MSGTYPE MAP BYTES	The type of message. The default is MAP.
NETPAGE <i>code page</i>	The value for setting the network code page. The default is ISO8859-1.

Keyword – Operand	Description
PATTERN-LEN <i>number</i>	<p>The creation of the MAP message field <b>IVP-STRING</b> of length <i>number</i>, populated with 1-----10-----.... The default is 0.</p> <p>For MSGTYPE=BYTES, the pattern message replaces the standard IVP structure. The length of the bytes message is PATTERN-LEN.</p>
PROPERTIES YES NO	A stipulation of whether the Substation ES system fields ( <b>tibss-userid</b> , <b>tibss-password</b> , <b>tibss-resource</b> , and <b>tibss-BES</b> ) are to be added to a message as user property fields. The default is YES.
PROPERTY <i>Name Value</i>	The property field <i>Name</i> with <i>Value</i> that is to be added to a message as user property field, or a mapped message field. The field type depends on the PROPERTIES setting.
PSWD <i>password</i>	The password for the <b>tibss-password</b> user property field or <b>MAP</b> message field. The default is a blank.
PUBLISH-ONLY	A directive for the program to only send a message and not listen for a reply.
REPEAT <i>number</i>	The number of times to repeat the send process. The default is 1.
RESOURCE <i>name</i>	<p>The name of the <b>tibss-resource</b> user property or <b>MAP</b> message field, up to a maximum of eight characters. The default is a blank.</p> <p>If present, this value overrides the recipe <b>Resource Name</b> field. The use of the <b>Resource Name</b> field depends on the Method on Invocation. For details, see Resource Name in <a href="#">Recipe Details Panel</a>.</p>
SEG-LEN <i>number</i>	This is the segment length value to be segmented the request message.
SEG-BREAK <i>HexValue</i>	<p>The segment break in hex value to be insert to each segmented request message. The value specified must match the value of the field Segment Break that is defined under the IMS Service Details panel.</p> <p>Valid values are 4 bytes in hex value. The default is FAFAFafa.</p>
SERVER <i>URL:port</i>	The URL and TCP port number of the EMS server.
TIMEOUT <i>timevalue</i>	The length of time (in seconds) for the response timeout. The default is 5.
TYPE TOPIC QUEUE	A stipulation of whether the program is working with topics or queues. The default is TOPIC.
USER <i>userid</i>	The user ID for the <b>tibss-userid</b> system field. The ID can be sent as a user property or <b>MAP</b> message field. The default is a blank.
WAIT <i>timevalue</i>	<p>The length of time (in seconds) to wait between message sends. The default is 0.</p> <p>The length of time (in seconds with decimals) to wait between message sends. The default is 0. For example, to wait one and a half seconds between sends, specify: 1.5</p>

Keyword – Operand	Description
* <i>text</i>	A comment.

## IVP Request or Reply EMS Consumer Producer

The program SXJ3RRCP works as a TIBCO messaging transactional server application.

The Consumer Producer receives request message published by a Substation ES CICS or IMS initiated request trigger process. The Consumer Producer then sends back a response message to the reply destination which that either came with the request message, or the destination that was provided in the **SYSIN** parameter **RESPONSE-DEST**. The response message type is determined by the inbound request message type.

The sample JCL stream for Consumer Producer can be found in:

Data Set: *USERHLQ.JCL*

Member: *SXJRRRCPP*



The EXEC PARM - IVPNETW is substituted with a set of TIBCO Enterprise Message Service ESB parameters. It is recommended that you use the provided default settings for all IVP runs. To use other settings, see the *TIBCO Enterprise Message Service User's Guide* for information about the parameters.

### SYSIN Parameters

The response messages sent by the utility are determined by the keywords and operands specified in the SYSIN DD Name file. The following table shows IVP Request or Reply EMS Consumer Producer parameters:

Keyword	Operand	Description
CLIENT-ID	<i>name</i>	The client identifier to use with the EMS connection.
DEBUG   DBG	OFF   ON	Enables debug messages. Retain the default (OFF) unless you are requested to enable debugging facilities by TIBCO Support.
DELAY	<i>timevalue</i>	The length of time (in seconds) to wait between receive and reply processes. The value must fall between 0 and 60. The default is 0, indicating no wait.
HOSTPAGE	<i>code page</i>	The value for setting the host code page.
NETPAGE	<i>code page</i>	The value for setting the network code page.
PSWD	<i>password</i>	The password associated with the user ID that is used to connect to the EMS server.
REQUEST-DEST	<i>destination</i>	Specifies the EMS destination name on which messages are received.

Keyword	Operand	Description
RESPONSE-DEST	<i>destination</i>	Specifies the EMS destination name to which reply messages are sent. Replies are also sent to the destination specified in the request message <b>ReplyTo</b> field.
RESPONSE-INFO	I   U	Determines which information is contained in the response messages. When <b>I</b> is specified, the message contains the IVP message structure. When <b>U</b> is specified, the message contains data from the user input for field <b>RESULT</b> .  The default is <b>I</b> . The message type is determined by the inbound request message type.
RESULT	<i>data</i>	The 63 characters data to be used in field <b>IVP-RESULT</b> . This field is blank delimited.
SERVER	<i>URL:port</i>	The EMS server URL and TCP port.
TYPE	TOPIC   QUEUE	Specifies whether the destination used is a topic or queue.
USER	<i>userid</i>	The user ID used to connect to the EMS server.
WORKERS	<i>number</i>	The concurrent producer to be run. The value must be between 1 and 100. The default is 1.

## IVP Generic Listener

The program `SXR3ILS1` works as a TIBCO messaging subscriber application. It receives all IVP messages published from the Substation ES region. The messages originate from the TIBCO messaging requestor applications.

The sample JCL stream is in the following:

Data Set: `USERHLQ.JCL`

Member: `SXRGILS`



The `EXEC PARM - IVPNETW` is substituted with a set of TIBCO Rendezvous transport parameters. It is recommended that you use the provided default settings for all IVP runs. For details about the other settings, see the *TIBCO Rendezvous Administration* manual.

The message subjects to which the IVP Generic Listener subscribes are determined by the `SYSIN DD` statements, which start with the keyword `SUBJECT` followed by the subject name. There are up to three statement entries for the IVP Generic Listener.

### SYSIN Parameters

Following is a sample of the input control statements in the `SYSIN` parameter member:

```
DEBUG      OFF
subject    tibss.ivp.>
subject    tibss.ibm.>
UFLD-PFX   tibss-
```

## IVP IMS Conversational Sender or Listener

The program SXR3ICV1 works as a TIBCO messaging transactional client application. The Conversational Sender or Listener publishes Substation ES IMS Conversational IVPs requesting messages. Which messages are requested depends on a set of parameters passed in through SYSIN DD statements in the run JCL.

Each output resulting from the IMS application is written to the log output medium. The sample JCL stream is in the following:

Data Set: *USERHLQ.JCL*

Member: SXIRICVS



The EXEC PARM - IVPNETW is substituted with a set of TIBCO Rendezvous transport parameters. It is recommended that you use the provided default settings for all IVP runs. To use the other settings, see the *TIBCO Rendezvous Administration* manual for information on the parameters.

## IVP Generic Error Listener

The programs SXR3IELS and SXJ3IELS initiate a TIBCO messaging subscriber that receives error messages published by Substation about data transformation errors, warnings, and error conditions encountered during Substation ES IVP's runtime processing.

The sample JCL stream for error listeners is in the following:

Data Set: *USERHLQ.JCL*

Member:

- SXRRIELS for TIBCO Rendezvous
- SXJRIELS for TIBCO Enterprise Message Service



The EXEC PARM - IVPNETW is substituted with a set of TIBCO Rendezvous or TIBCO Enterprise Message Service ESB endpoint parameters. It is recommended that you use the provided default settings for all IVP runs.

### SYSIN Parameters

The error messages subjects to which the IVP Generic Error Listener subscribes are determined by the keywords and operands in the SYSIN file in the following:

Data Set: *USERHLQ.CNTL*

Member:

- SXRPERLS for TIBCO Rendezvous
- SXJPERLS for TIBCO Enterprise Message Service

The following table describes the control parameters for IVP Generic Error Listener:

Keyword – Operand	Description
SUBJECT {tibss.error.>}	The IVP error subject names to be listened to.
QUEUE or TOPIC {tibss.error.out}	TIBCO Enterprise Message Service only. The subject destination on which an error message is published if Transformer conversion or BES encounters errors.



Keyword – Operand	Description
PSWD	TIBCO Enterprise Message Service only. The password associated with the user ID if used in the session.
UFLD-PFX {tibss-}	The prefix of the Substation ES system field name for the processing status. We recommend that you do not change this value for IVP runs.
USER	TIBCO Enterprise Message Service only. The user-specified identifier if used in the session.
TIMEOUT <number> (in seconds)	A program is terminated after <number> seconds. The number range is 0 to 86400, and 0 means no time limit and 86400 means 24 hours.

For details about the Generic Error Listener, see "Generic Error Listener Utility" in the *TIBCO Substation ES Operations and Administration* manual.

## Example of ESB Request – Reply DPL

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application with the CICS DPL method through Substation ES.

The CICS application replies by putting information into the COMMAREA; the data in this area is picked up by Substation ES. The server delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

For the C language:

- SXCRIJSL or SXCIRSL: the publisher that sends the request message
- SXC-DPL-C: the Substation ES recipe identification
- SXC-DPL-MRO: the Substation ES recipe identification for the DPL MRO Method of Invocation
- SXC3I001: the CICS C application program

For the COBOL language:

- SXCRIJSL or SXCIRSL: the publisher that sends the request message
- SXC-DPL-COBOL: the Substation ES recipe identification
- SXC-DPL-MRO: the Substation ES recipe identification for the DPL MRO Method of Invocation
- SXCCI001: the CICS COBOL application program

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages.

### Rendezvous – DPL Request/Reply (LANGUAGE C)

```
SUBJ-REQ  tibss.CICS.C.Request
RESOURCE  SXC3I001
REPEAT    1
```

TIMEOUT	15
WAIT	0
BES-DELAY	5

### Rendezvous – DPL MRO Request/Reply (LANGUAGE C)

SUBJ-REQ	tibss.CICS.MRO.Request
RESOURCE	SXC3I001
REPEAT	1
TIMEOUT	15
WAIT	0
BES-DELAY	5

### Enterprise Message Service – DPL Request/Reply (LANGUAGE C)

DEST-REQ	tibss.CICS.C.Request
DEST-REP	tibss.CICS.C.Reply
TYPE	QUEUE
RESOURCE	SXC3I001
REPEAT	1
TIMEOUT	15
WAIT	0
BES-DELAY	5

### Enterprise Message Service – DPL Request/Reply (LANGUAGE COBOL)

DEST-REQ	tibss.CICS.Cobol.Request
DEST-REP	tibss.CICS.Cobol.Reply
TYPE	QUEUE
RESOURCE	SXCCI001

### Enterprise Message Service – DPL MRO Request/Reply (LANGUAGE COBOL)

DEST-REQ	tibss.CICS.MRO.Request
DEST-REP	tibss.CICS.MRO.Reply
TYPE	QUEUE
RESOURCE	SXCCI001

## Example of ESB Request – Reply DPL Container

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application by using the CICS DPL method with Container through Substation ES. This method is MRO capable.

The CICS application finds its input in a container with the same name as the program and replies by putting information into the container. The data in this area is picked up by Substation ES. The server delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

You can use the Request components to run the ESB Request - Reply DPL Container example.

- SXCRIJSL or SXCRIJSL: the publisher that sends the request message
- SXC-DPL-COBOL-Container: the Substation ES recipe identification
- SXCCI014: the CICS COBOL application program

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages.

### Rendezvous – DPL Container Request/Reply (LANGUAGE COBOL)

```
SUBJ-REQ  tibss.CICS.Cobol.Container.Request
RESOURCE  SXCCI014
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
```

### Enterprise Message Service – DPL Container Request/Reply (LANGUAGE COBOL)

```
DEST-REQ  tibss.CICS.Cobol.Container.Request
DEST-REP  tibss.CICS.Cobol.Container.Reply
TYPE      QUEUE
RESOURCE  SXCCI014
```

## Example of ESB Request – Reply DPL Storage

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application by using the CICS DPL method with Storage Pointer through Substation ES. This method is not MRO capable.

The CICS application finds its input by using a Storage Pointer in the COMMAREA and replies by putting information into the same area. The data in this area is picked up by Substation ES. The server delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

You can use the Request components to run the ESB Request - Reply DPL Storage.

For the C language:

- SXCRIJSL or SXCRIJSL: the publisher that sends the request message
- SXC-DPL-C-Storage: the Substation ES recipe identification
- SXC3I013: the CICS C application program

For the COBOL language:

- SXCRIJSL or SXCRIJSL: the publisher that sends the request message
- SXC-DPL-COBOL-Storage: the Substation ES recipe identification
- SXCCI013: the CICS COBOL application program

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages.

### Rendezvous – DPL Storage Request/Reply (LANGUAGE C)

```
SUBJ-REQ  tibss.CICS.C.STG.Request
RESOURCE  SXC3I013
REPEAT    1
TIMEOUT   15
```

WAIT	0
BES-DELAY	5

### Rendezvous – DPL Storage Request/Reply (LANGUAGE COBOL)

SUBJ-REQ	tibss.CICS.Cobol.STG.Request
RESOURCE	SXCCI013
REPEAT	1
TIMEOUT	15
WAIT	0

### Enterprise Message Service – DPL Storage Request/Reply (LANGUAGE C)

DEST-REQ	tibss.CICS.C.STG.Request
DEST-REP	tibss.CICS.C.STG.Reply
TYPE	QUEUE
RESOURCE	SXC3I013

### Enterprise Message Service – DPL Storage Request/Reply (LANGUAGE COBOL)

DEST-REQ	tibss.CICS.Cobol.STG.Request
DEST-REP	tibss.CICS.Cobol.STG.Reply
TYPE	QUEUE
RESOURCE	SXCCI013

## Example of ESB Request – Reply DPL TSQ

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application by using the CICS DPL method with an eight-character TSQ name in the COMMAREA through Substation ES. This method is MRO capable.

The CICS application finds its input in a TSQ record and replies by putting information into the TSQ. The data in this area is picked up by Substation ES. The server delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

This section lists the components that involve for this sample.

For the C language:

- SXCRIJSL or SXCRIRSL: the publisher that sends the request message
- SXC-DPL-C-TSQ: the Substation ES recipe identification
- SXC3I012: the CICS C application program

For the COBOL language:

- SXCRIJSL or SXCRIRSL: the publisher that sends the request message
- SXC-DPL-COBOL-TSQ: the Substation ES recipe identification
- SXCCI012: the CICS COBOL application program

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages.

### Rendezvous – DPL TSQ Request/Reply (LANGUAGE C)

SUBJ-REQ	tibss.CICS.C.TSQ.Request
RESOURCE	SXC3I012

REPEAT	1
TIMEOUT	15
WAIT	0
BES-DELAY	5

### Rendezvous – DPL TSQ Request/Reply (LANGUAGE COBOL)

SUBJ-REQ	tibss.CICS.Cobol.TSQ.Request
RESOURCE	SXCCI012
REPEAT	1
TIMEOUT	15
WAIT	0

### Enterprise Message Service – DPL TSQ Request/Reply (LANGUAGE C)

DEST-REQ	tibss.CICS.C.TSQ.Request
DEST-REP	tibss.CICS.C.TSQ.Reply
TYPE	QUEUE
RESOURCE	SXC3I012

### Enterprise Message Service – DPL TSQ Request/Reply (LANGUAGE COBOL)

DEST-REQ	tibss.CICS.Cobol.TSQ.Request
DEST-REP	tibss.CICS.Cobol.TSQ.Reply
TYPE	QUEUE
RESOURCE	SXCCI012

## Example of CICS Trigger

This IVP example illustrates how a trigger process delivers data from a CICS application to a TIBCO messaging application through Substation ES.

The CICS application initiates the request by writing the output data to the Substation ES trigger transient data queue (TDQ). The Substation ES gets the trigger based on the defined search criteria for this trigger process. The Substation ES Transformer then processes the data conversions according to the selected conversion rule in the trigger recipe.

After the data has been converted, Substation ES publishes the output message to the IVP Generic Listener.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXTT: the CICS transaction ID, the request initiator
- SXCCI002: the CICS COBOL application program
- TRIGGER-FLDS: the Substation ES trigger identification

## CICS Execution Command

The command for executing transactions entered from a CICS 3270 screen is SXTT.

## Example of CICS High Volume Trigger

These IVP examples illustrate the high-speed throughput of this trigger process in delivering data from a CICS application to a TIBCO messaging application through Substation ES. The High Volume Trigger (HVT) can handle record sizes greater than 32,000 bytes. These samples support guaranteed, reliable, and ordered delivery.

A CICS application initiates a request by linking to the HVT module to write output data to the Substation ES trigger TDQ (depending on the parameters passed in). Substation ES receives the trigger

based on the defined search criteria for this trigger process. The Substation ES Transformer then processes the data conversion according to the selected conversion rule in the trigger.

After the data has been converted, Substation ES publishes the output message to the IVP Generic Listener. For more information about the process flow, see "High Volume Trigger" in the *TIBCO Substation ES Operations and Administration* manual.

## Component Summary

The section lists the components that involve for this sample.

For the C language:

- SXT6: the CICS transaction Id, the request initiator
- SXC3I032: the CICS application program
- TRIGGER-FLDS: the Substation ES trigger identification
- TRIGGER-FLDS-GUR: the Substation ES trigger identification
- TRIGGER-TEXT: the Substation ES trigger identification

For the COBOL language:

- SXTF: the CICS transaction Id, the request initiator
- SXCCI032: the CICS application program
- TRIGGER-FLDS: the Substation ES trigger identification
- TRIGGER-FLDS-GUR: the Substation ES trigger identification
- TRIGGER-TEXT: the Substation ES trigger identification

## CICS Execution Command

The command for executing transactions entered from a CICS 3270 screen is SXTF or SXT6.

For details, see the "High Volume Trigger" section of the *TIBCO Substation ES Operations and Administration* manual.

For example, the following sample commands run the High Volume Triggers:

```
SXTF
SXTF,R,1,2000,TRIGGER-BYTES
SXTF,G,1,2000,TRIGGER-TEXT
SXT6
SXT6,R,1,2000,TRIGGER-BYTES
SXT6,G,1,2000,TRIGGER-TEXT
```

## Example of RED CICS Trigger

These IVP examples illustrate the RED CICS trigger process in delivering data from a CICS application to a TIBCO messaging application through Substation ES. The RED CICS Trigger can handle record sizes greater than 32,000 bytes up to maximum 4096000 bytes. These samples support guaranteed, reliable, and fire and forget delivery.

A CICS application initiates a request by linking to the SXCOTRED RED CICS Trigger module to write output data to the Substation ES COMMAREA or Containers (depending on the parameters passed in). Substation ES receives the trigger based on the Service Name parameter for this trigger process. The Substation ES Transformer then processes the data conversion according to the selected conversion rule in the trigger.

After the data has been converted, Substation ES publishes the output message to IVP Generic Listener.

For more information about the process flow, see "RED CICS Triggers" in the *TIBCO Substation ES Operations and Administration* manual.

## Component Summary

The section lists the components that involve for this sample.

For the C language:

- XTC2: the CICS transaction Id, the request initiator
- SXX3CI32: the CICS application program
- RED-TRG-NO-RULES: the Substation ES trigger service name
- RED-TRG-BYTES: the Substation ES trigger service name
- RED-TRG-TEXT: the Substation ES trigger service name

For the COBOL language:

- XTC2: the CICS transaction Id, the request initiator
- SXXCCI32: the CICS application program
- RED-TRG-NO-RULES: the Substation ES trigger service name
- RED-TRG-BYTES: the Substation ES trigger service name
- RED-TRG-TEXT: the Substation ES trigger service name

## CICS Execution Command

The command for executing transactions entered from a CICS 3270 screen is XTC2 or XT32.

Their formats are listed as follow:

- XTC2, <Type(G/R/F)>, <#-Msg>, <Msg-Len(1-4096000)>, <Service Name>, <#-CTN>, <Delay>
- XT32, <Type(G/R/F)>, <#-Msg>, <Msg-Len(1-4096000)>, <Service Name>, <#-CTN>, <Delay Sec>

The details for each input parameter are listed as follows:

- **G**: guaranteed using RED Send with Reply API
- **R**: reliable using RED Send with Confirm API
- **F**: fire and forget using RED Send API
- **#-Msg**: the number of Trigger message to be sent
- **Msg-Len(1-4096000)**: the size of the Trigger message to be sent; valid values for this parameter are in the range 1 - 4096 KB
- **Service Name**: the Trigger service name to be used
- **#-CTN**: the number of containers number to be used; valid values for this parameter are in the range 1 - 20. If this parameter is not present, the COMMAREA is used.
- **Delay**: the delay time in seconds between calls to SXCOTRED

For example, the following sample commands run the RED CICS Triggers:

```
XTC2, ?
XTC2, , , , RED-TRG-NO-RULES
XTC2, R, 1, 2000, RED-TRG-NO-RULES, 3
XTC2, G, 1, 2000, RED-TRG-TEXT

XT32, ?
XT32, , , , RED-TRG-NO-RULES
XT32, R, 1, 2000, RED-TRG-NO-RULES, 5, 10
XT32, G, 1, 2000, RED-TRG-TEXT
```

## Example of ESB Request DPL XCTL

This IVP example illustrates how a TIBCO messaging client application initiates a request message to start a CICS program with the CICS XCTL method through Substation ES.

The CICS program replies by writing the reply information into the Substation ES trigger TDQ. Substation ES gets the trigger based on the defined search criteria for this trigger process and then delivers the reply and an acknowledgment to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to the selected conversion rules in the recipe and trigger.

### Component Summary

The section lists the components that involve for this sample.

- SXCRIJSL or SXCIRSL: the publisher that sends the request message
- SXC-DPL-COBOL-START-PGM: the Substation ES recipe identification
- SXCCI003: the CICS COBOL application program
- TRIGGER-FLDS: the Substation ES trigger identification

### Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the reply.

#### Rendezvous

```
SUBJ-REQ  tibss.CICS.Cobol.Program.Request
RESOURCE  SXCCI003
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV03
```

#### Enterprise Message Service

```
DEST-REQ  tibss.CICS.Cobol.Program.Request
DEST-REP  tibss.CICS.Cobol.Program.Status
TYPE      QUEUE
RESOURCE  SXCCI003
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV03
```

## Example of ESB Invoking a CICS Transaction

This IVP example illustrates how a TIBCO messaging client application initiates a request message to start a CICS transaction task with the CICS transaction invoked method through Substation ES.

The CICS program replies by writing the reply information into the Substation ES trigger TDQ. Substation ES gets the trigger based on the defined search criteria for this trigger process and then delivers the reply and an acknowledgment to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe and trigger.

### Component Summary

The section lists the components that involve for this sample.

- SXCRIJSL or SXCIRSL: the publisher that sends the request message



- SXC-DPL-COBOL-START-TRAN: the Substation ES recipe identification
- SXCCI004: the CICS COBOL application program
- TRIGGER-FLDS: the Substation ES trigger identification

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the reply.

### Rendezvous

```
SUBJ-REQ  tibss.CICS.Task.Request
RESOURCE  SXTI
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV04
```

### Enterprise Message Service

```
DEST-REQ  tibss.CICS.Task.Request
DEST-REP  tibss.CICS.Task.Status
TYPE      QUEUE
RESOURCE  SXTI
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV04
```

## Example of ESB to TDQ

This IVP example illustrates how a TIBCO messaging client application initiates a request message that forces Substation ES to write the data into a TDQ with the CICS WRITEQ TD method.

The Substation ES delivers an acknowledgment to the IVP Sender Listener. In this example, the data is written to the Substation ES trigger TDQ. Therefore, Substation ES gets a trigger based on the defined search criteria for this trigger process and then delivers a reply to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe and trigger.

## Component Summary

The section lists the components that involve for this sample.

- SXCRIJSL or SXCIRSL: the publisher that sends the request message
- SXC-DPL-WRITE-TDQ: the Substation ES recipe identification
- SXQT: the Substation TDQ name
- TRIGGER-FLDS: the Substation ES trigger identification

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the status reply.

### Rendezvous

```
SUBJ-REQ  tibss.CICS.TDQ.Request
RESOURCE  SXQT
REPEAT    1
TIMEOUT   15
```

WAIT	0
BES-DELAY	5
IVP-ID	SXCIV05

### Enterprise Message Service

DEST-REQ	tibss.CICS.TDQ.Request
DEST-REP	tibss.CICS.TDQ.Status
TYPE	QUEUE
RESOURCE	SXQT
REPEAT	1
TIMEOUT	15
WAIT	0
BES-DELAY	5
IVP-ID	SXCIV05

## Example of CICS TSQ Entry

This IVP example illustrates how a TIBCO messaging client application initiates a request message that forces Substation ES to write the data into a temporary storage queue (TSQ) with the CICS WRITEQ TS method.

Substation ES delivers an acknowledgment to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXCRIJSL or SXCIRSL: the publisher that sends the request message
- SXC-DPL-WRITE-TSQ: the Substation ES recipe identification
- SXCTSQT1: the name of a CICS TSQ Substation ES temporary storage queue

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the status reply.

### Rendezvous

SUBJ-REQ	tibss.CICS.TSQ.Request
RESOURCE	SXCTSQT1
REPEAT	1
TIMEOUT	15
WAIT	0
BES-DELAY	5
IVP-ID	SXCIV06

### Enterprise Message Service

DEST-REQ	tibss.CICS.TSQ.Request
DEST-REP	tibss.CICS.TSQ.Status
TYPE	QUEUE
RESOURCE	SXCTSQT1
REPEAT	1
TIMEOUT	15
WAIT	0
BES-DELAY	5
IVP-ID	SXCIV06

## Example of ESB Request – Reply Complex DPL

This IVP example, which works in the same manner as the CICS request or reply IVP, illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application by using the CICS DPL method through Substation ES.

The CICS application replies by putting information into the COMMAREA, from which the data is retrieved by Substation ES. Substation ES then delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipe. This example demonstrates most of the supported data type usages.

### Component Summary

The section lists the components that involve for this sample.

- SXCRIJSL or SXCRIISL: the publisher that sends the request message
- SXC-DPL-COBOL-COMPLEX: the Substation ES recipe identification
- SXCCI010: the CICS COBOL application program

### Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the status reply.

#### Rendezvous

```
SUBJ-REQ  tibss.CICS.Cobol.Complex.Request
RESOURCE  SXCCI010
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV010
```

#### Enterprise Message Service

```
DEST-REQ  tibss.CICS.Cobol.Complex.Request
DEST-REP  tibss.CICS.Cobol.Complex.Reply
TYPE      QUEUE
RESOURCE  SXCCI010
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV010
```

## Example of CICS-Initiated Request or Reply

This IVP example illustrates how a request is initiated via a CICS transaction on a CICS 3270 screen terminal.

The invoked CICS program writes the request information to the Substation ES RR TDQ and waits a set time for the reply. The Substation ES gets a trigger based on the defined search criteria for this CICS initiated request trigger process. In addition, a DPL process is invoked by the requesting message. The process works as a TIBCO client application to produce the reply to answer the request.

The Substation ES reply program is invoked by the reply message. The reply data is delivered to the originating CICS program. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipes and trigger.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

For the C language:

- SXR3: CICS transaction ID of the request initiator
- SXC31RR1: the CICS C application program
- CICS-RR-REQUEST: the Substation ES trigger identification for initiating requests

The same process can be completed in the following two ways:

- SXJRRRC: the JCL member that submits the EMS Consumer/Producer
- SXJ3RRCP: the COBOL Consumer/Producer application program which receives a request message from EMS server and produces a reply message

Or

- SXC-RR-EXTRN-PROCESS: the Substation ES recipe identification for generating replies
- SXCCIRR3: the CICS COBOL application program, which receives a request from an application through DPL and replies through the COMMAREA
- SXC-RR-ANSWER: the Substation ES recipe identification for receiving replies
- SXCREPLY: the Substation ES Request or Reply module.

For the COBOL language:

- SXRC: the CICS transaction ID of the request initiator use of COMMAREA
- SXRT: the CICS transaction ID of the request initiator use of TSQ
- SXRS: the CICS transaction ID of the request initiator use of Storage Pointer
- SXCCIRR1: the CICS COBOL application program
- CICS-RR-REQUEST: Substation ES trigger identification for initiating requests

The same process can be completed in the following two ways:

- SXJRRRC: the JCL member that submits the EMS Consumer/Producer
- SXJ3RRCP: the COBOL Consumer/Producer application program which receives a request message from EMS server and produces a reply message

Or

- SXC-RR-EXTRN-PROCESS: the Substation ES recipe identification for generating replies
- SXCCIRR3: the CICS COBOL application program, which receives a request from an application through DPL and replies through the COMMAREA
- SXC-RR-ANSWER: the Substation ES recipe identification for receiving replies
- SXCREPLY: the Substation ES Request or Reply module.

## CICS Execution Command

This section lists the CICS transactions to initiate the request.

COBOL entered from a CICS 3270 screen: SXRC, SXRT or SXRS

C entered from a CICS 3270 screen: SXR3

## Example of RED CICS-Initiated Request or Reply

The IVP example illustrates how a request is initiated through a CICS transaction on a CICS 3270 screen terminal.

The invoked RED CICS program writes the request information to the COMMAREA, CONTAINER, TSQ or Storage Pointer; depends on the initiated parameter, and then waits a set time for reply. The Substation ES gets a trigger based on the service name passed in from the parameter for this RED CICS initiated request trigger process. Substation ES Transformer then processes the data conversions according to the selected conversion rule in the trigger service.

After the data has been converted, Substation ES publishes the output message to the IVP awaiting EMS Consumer/Producer. The process works as a TIBCO client application to produce the reply to answer the request.

The Substation ES reply program is invoked by the reply message. The reply data is delivered to the originating RED CICS program. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipes and trigger.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- XTR3: CICS transaction ID of the request initiator
- SXX31RR1: the CICS RED C application program
- RED-RR-REQUEST: the Substation ES trigger identification for initiating requests
- SXJRRRC: the JCL member that submits EMS Consumer/Producer
- SXJ3RRCP: the C Consumer/Producer application program which receives a request message from EMS server and produces a reply message
- RED-RR-ANSWER: the Substation ES recipe identification for receiving replies
- SXCREPLY: the Substation ES Request or Reply module

## CICS Execution Command

C entered from a CICS 3270 screen: XTR3

The input parameters format is listed as follows:

```
XTR3,<C|N|T|S>,<size number>,<timeout number>,<service name>
```

where:

- ?: means Help.
- S: means Storage Pointer.
- C: implies use of common area (Default).
- N: implies use of Container.
- T: implies use of TSQ.
- S: implies use of Storage Pointer.
- *size number*: indicates the request message size to be sent. The minimum size is 308 by default, and any number over this size will increase the length of the data sent. This allows testing of sizes that are greater than 32K.
- *timeout number*: indicates the time in seconds that the application to be wait for the reply message. The default value is 20.

- *service name*: indicates the Trigger service name to be used in publishing the initial message. The default value is RED-RR-REQUEST.

## Example of ESB Request – Reply IMS

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke an IMS Message Processing Program (MPP). The client uses an IMS message queue via Substation ES.

The IMS MPP replies by putting information into the IO-AREA and inserts it on the IMS message queue. Substation ES picks up the data from there and delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer processes the data conversions based on the selected conversion rules in the recipe.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXIRIJSL or SXIRIRSL: the JCL member that submits the request initiator
- SXR3ISL1 or SXJ3ISL1: the IVP Sender Listener that sends the request message
- SXI-TRAN: the Substation ES recipe identification
- SXICIT01: the IMS transaction identification
- SXICIP01: the IMS COBOL message processing program

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the status reply.

### Rendezvous

```
SUBJ-REQ  tibss.IMS.Tran.Request
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
```

### Enterprise Message Service

```
DEST-REQ  tibss.IMS.Tran.Request
DEST-REP  tibss.IMS.Tran.Reply
TYPE      QUEUE
REPEAT    1
TIMEOUT   15
WAIT      0
```

## Example of ESB Request – Reply and Trigger

This example illustrates how a TIBCO messaging client application initiates a request message to invoke an IMS Message Processing Program (MPP) by using the IMS message queue through Substation ES.

The IMS MPP replies by putting information into the IO-AREA and inserting it to the IMS message queue. In addition, the MPP uses ALT-PCB to insert the output data to the destined Substation ES trigger TPipe. Substation ES picks up that data and delivers the reply and trigger messages to the IVP Sender Listener. In both directions, the Substation ES Transformer processes the data conversion according to the selected conversion rules in the recipe and trigger.

For more information about the process flow, see the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXIRIJSL or SXIRIRSL: the JCL member for that submits the request initiator
- SXR3ISL1 or SXJ3ISL1: the IVP Sender Listener sending of that sends the request message
- SXICIT02: the IMS transaction identification
- SXICIP02: the IMS COBOL message processing program
- TRIGGER-FLDS: the Substation ES trigger identification.

A version of this program, SXICIP03, demonstrates the use of variable-length IMS transaction codes (not using the first 8 bytes of the message).

## Execution Parameters of IVP Sender Listener

The following input control statement sets instruct the IVP Sender Listener to publish the corresponding request messages and wait for the status reply.

### Rendezvous

```
SUBJ-REQ  tibss.IMS.Tran-Trig.Request
REPEAT    1
TIMEOUT   15
WAIT      0
BES-DELAY 5
```

### Enterprise Message Service

```
DEST-REQ  tibss.IMS.Tran-Trig.Request
DEST-REP  tibss.IMS.Tran-Trig.Reply
TYPE      QUEUE
REPEAT    1
TIMEOUT   15
WAIT      0
```

## Example of IMS BMP Trigger

This example illustrates how a trigger process delivers data from an IMS Batch Message Program (BMP) or a 3270 device to a TIBCO messaging application through Substation ES.

The JCL is supplied with the installation of Substation ES. The IMS BMP Trigger example uses this JCL to invoke the IMS transaction.

The IMS BMP initiates the request by inserting the output data to the IMS message queue (TPipe). The TPIPE delivers the IMS outbound events to Substation ES, which gets the trigger based on the defined search criteria for this trigger process. Substation ES Transformer then processes the data conversions according to the selected conversion rule in the trigger recipe.

After the data has been converted, Substation ES publishes the output message to the IVP Generic Listener.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXIIVPBT: the JCL member that submits the batch request initiator
- SXICITB1: the IMS transaction identification

- SXICIPB1: the IMS COBOL batch message program
- TRIGGER-BYTES: the Substation ES IMS interface trigger identification

## Execution Parameters of JCL

```
//STEP1BT EXEC IMSBATCH,MBR=SXICIPB1,PSB=SXICIPB1,IMSID=&IMSID,
//          IN=SXICITB1,APARM='000001SXITRG$RYNSXIIVB'
```

Alternatively:

```
//STEP1BT EXEC IMSBATCH,MBR=SXICIPB1,PSB=SXICIPB1,IMSID=&IMSID,
//          IN=SXICITB1,APARM='000001SXITRG$GYNSXIIVB'
```

## Example of IMS Synchronous Callout — Request or Reply

This IVP example illustrates how a request is initiated using an IMS Batch Message Program (BMP) or a 3270 device to a TIBCO messaging application through Substation ES.

The JCL is supplied with the installation of Substation ES. The IMS Synchronous Callout Request or Reply example uses this JCL to invoke the IMS transaction.

The IMS BMP initiates the request by inserting the output data to the IMS message queue (TPipe). The TPipe delivers the IMS outbound events to Substation ES, which gets the trigger based on the defined search criteria for this IMS-initiated request trigger process. Substation ES Transformer then processes the data conversions according to the selected conversion rule in the trigger service.

After the data has been converted, Substation ES publishes the output message to the IVP awaiting EMS Consumer/Producer. The process works as a TIBCO client application to produce the reply to answer the request.

The Substation ES reply program is invoked by the reply message. The reply data is delivered to the originating IMS program. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipes and trigger.

For more information about the process flow, see the "Messages Flow" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXIIIVPRR: the JCL member that submits the request initiator
- SXICITRR : the IMS transaction ID of the request initiator
- SXICIRR1: the IMS COBOL application program
- CALLOUT-REQUEST: Substation ES trigger service identification for initiating requests
- SXJRRRCF: the JCL member that submits the EMS Consumer/Producer
- SXJ3RRCP: the C Consumer/Producer application program, which receives a request message from EMS server and produces a reply message

## Example of IMS Conversational Functions

This example illustrates how a TIBCO messaging transactional client application initiates request messages to perform the IMS conversational functions through Substation ES. This example uses the IBM IMS INSTALL/IVP sample of a simple IBM phone-book application. All functions performed by that application can be executed with Substation ES IMS Conversational IVP.

You must install the phone-book application before running this IMS Conversational example. See *IMS Installation* for the installation requirements for all programs, PSBs, DBDs, MFSs, and other supporting materials used by the application.



You can invoke the IMS Conversational Sender or Listener by submitting JCL with a set of function operand parameters.

The IMS Conversational Sender or Listener initiates each function, sending the request message through Substation ES to instruct the IMS application. The IMS application sends a reply message back to confirm the process status of each function. Substation ES then delivers the reply to the IMS conversational Sender or Listener and continues the conversation with the IMS application until it receives the END confirmation message from IMS.

In both directions, the Substation ES Transformer converts the data according to the conversion rules in the recipe and then publishes the output message to the IMS Conversational Listener.

For more information about the process flow, see the "Messages Flow for IMS" section of the *TIBCO Substation ES Concepts* manual.

## Component Summary

The section lists the components that involve for this sample.

- SXIRICVS or SXIJICVS: JCL member that submits the IMS Conversational Sender or Listener
- SXR3ICV1 or SXJ3ICV1: the C TIBCO transactional client application program that initiates and receives conversational request and reply messages
- SXI-CONVERSATIONAL: the Substation ES recipe identification
- IVTCB: the IBM phone book sample IMS transaction code
- DFSIVP34: the COBOL, IBM phone book application program



When you run these IVP components, the IVP Generic Error Listener is present and listening to the subject `tibss.ibm.ims.cvr.s.reply`.

## Execution Parameters of IVP IMS Conversational Sender or Listener

The following table describes IMS Conversational Sender or Listener control parameters:

Keyword	Operand	Description
-ENTRY	<i>Operand,Fields</i>	One message entry for a specific IMS program function in IBM sample phone book application.
	<i>ADD,last name,first name,ext #,ZIP code</i>	Performs the ADD function to add this new record to the IMS database.
	<i>DEL,last name</i>	Performs the DEL function of deleting this record from the IMS database.
	<i>DIS,last name</i>	Performs the DIS function of displaying this record from the IMS database.
	END	Stipulates the sending of the message to terminate the current transactional process.
	<i>UPD,last name,first name,ext #,ZIP code</i>	Performs the UPD function for updating changes of this record to the IMS database.

The following values are entered through the SYSIN DD statements. These control statements instruct the Sender to publish the correspond requesting message.

```
-ENTRY ADD,PEPPER,JOHN,EXT77,Z94304
-ENTRY ADD,SALTY,SANDRA,EXT44,Z94001
```

```
-ENTRY DIS,CINAMON  
-ENTRY DIS,PEPPER  
-ENTRY DEL,SALTY  
-ENTRY ADD,CINAMON,CINDY,EXT88,Z94001  
-ENTRY UPD,CINAMON,CINDY,EXT11,Z94999  
-ENTRY DEL,SALTY  
-ENTRY DIS,CINAMON  
-ENTRY DEL,CINAMON  
-ENTRY DEL,PEPPER  
-ENTRY END
```

The complete output from the IMS Conversational IVP is in the following:

Data Set: *USERHLQ.DATA*

Member: *SXIIOCVS*

# Communication of Application Messages

---

This section describes how Substation ES communicates with applications that use communication fields and notifications.

The section includes the following topics:

- [Communication Fields](#)
- [System Fields](#)
- [User Fields](#)
- [Application Notifications](#)

## Communication Fields

Communication fields are named fields within a user message that assist in bidirectional Substation ES communication. These fields enhance the ability of an application to effectively forward additional information and receive notifications when conversing with Substation ES.

### Categories

Substation ES communication fields are divided into the following categories:

- **Substation ES system fields**, which are defined in a user message and are used by applications and Substation ES during communication. System fields can be bidirectional. System fields are not stored within the Substation ES communication buffer passed to a BES application. Therefore, Substation ES processes these fields differently than normal message fields.
- **Substation ES user fields**, which are part of the Application Notification and are Substation ES output-only fields. These fields enable applications to identify information regarding the application's communication with Substation ES.

All communication field names consist of a variable prefix and a suffix constant. The prefix length is a maximum of 51 characters.

### Prefix

All Substation ES communication fields use a default prefix of **tibss-**. When explicitly referenced, the prefix together with the name must be specified either in Substation configuration UI or by applications communicating with Substation, such as TIBCO ActiveMatrix BusinessWorks processes or network programs.

## System Fields

System fields are not always used. When they are used, they are either in the user message or as message properties. To ensure that the contents of the fields are correctly processed and relayed to the participating interfaces by Substation ES, depending on the direction of the conversation, specify certain system fields in the definition of the configuration message.

## Definitions

The definitions describe the system fields, which do not include the default prefix **tibss-**.

### *System Fields*

Field Name	Description
<b>envelope</b>	The Substation ES context information that is used during a CICS and IMS request or reply application communication or during IMS application conversational interaction.
<b>userid</b>	The user identifier required by Substation ES and the BES.
<b>password</b>	The user password required by Substation ES and the BES.
<b>resource</b>	The dynamically supplied name of the resource to be executed in the BES in the payload or message.
<b>BES</b>	The dynamically supplied name assigned to the Substation ES interface for communicating with the transaction processing systems (such as CICS and IMS) in the payload or message.
<b>MFS-Name</b>	The dynamically-supplied MFS Map Name to be used in IMS BES.
<b>tran-code</b>	The dynamically supplied user transaction name to be executed in the BES in the payload or message (CICS usage only).
<b>rro-tran-code</b>	The dynamically supplied a four-character identifier for the transaction responsible for running the program on the remote CICS system.
<b>rro-sysid</b>	The dynamically supplied a four-character system identifier for the CICS region where the program ultimately is run.
<b>service-id</b>	Substation ES adds this field to the user message. The content is this service's identifier.
<b>ss-id</b>	Substation ES adds this field to the users message. The content is this Substation's identifier.

## Usage

The Definitions of System Fields table defines the system fields, which do not include the default prefix **tibss-**.

Field Name	Substation Message Direction	Definition and Usage
<b>envelope</b>	Output	Substation ES adds this field to the users message. If the user's application issues a reply, this field and its contents must be returned unchanged in the application's reply message. No configuration definition is necessary.

Field Name	Substation Message Direction	Definition and Usage
	Input	<p>If a user application receives this field and a reply is required by the originating application, the contents must be returned unchanged in the reply message of the application.</p> <p>Define the input message with the Opaque data type. In the field conversion rule panel, specify S in the <b>Trunc</b> field and Y in the <b>Reqd</b> field.</p>
<b>userid</b>	Input	<p>The user's application sends this field.</p> <p>Define the input message with the data type String. In the field conversion rule panel, specify S in the <b>Trunc</b> field and Y or N in the <b>Reqd</b> field.</p>
<b>password</b>	Input	<p>The user's application sends this field.</p> <p>Define the input message with the data type String. In the field conversion rule panel, specify S in the <b>Trunc</b> field and Y or N in the <b>Reqd</b> field.</p>
<b>resource</b>	Input	<p>The user's application sends this field.</p> <p>Define the input message with the data type String. In the field conversion rule panel, specify S in the <b>Trunc</b> field and Y or N in the <b>Reqd</b> field.</p>
<b>tran-code</b>	Input	<p>The user's application sends this field. This field is used to replace the Mirror Transaction defined in the recipe.</p> <p>Define the input message with the String data type. In the fieldconversion rule panel, specify S in the <b>Trunc</b> field and Y or N in the <b>Reqd</b> field.</p>

Field Name	Substation Message Direction	Definition and Usage
<b>BES</b>	Input	<p>The user's application sends this field.</p> <p>Define the input message with the data type String. In the field conversion rule panel, specify <b>S</b> in the <b>Trunc</b> field and <b>Y</b> or <b>N</b> in the <b>Reqd</b> field.</p> <p>This field is used for both the primary or secondary BES. This can be specified on the Service Recipe or dynamically as input.</p> <p>Valid input values are listed as follows:</p> <ul style="list-style-type: none"> <li>• <b>BES</b>: override the BES value and skip the Alt-BES value defined on the Service.</li> <li>• <b>BES ;</b>: override the BES value and keep the Alt-BES value defined on the Service.</li> <li>• <b>BES ;Alt-BES</b>: override both BES values defined on the Service.</li> <li>• <b> ;Alt-BES</b>: keep the BES value and override the Alt-BES value defined on the Service.</li> </ul>
<b>MFS-Name</b>	Input	<p>The dynamically supplied MFS Map Name to be used in IMS BES.</p> <p>The user application sends this field, which overrides the MFS Map Name value defined on the service.</p> <p>Define the input message with the data type String. In the field conversion rule panel, specify <b>S</b> in the <b>Trunc</b> field and <b>Y</b> or <b>N</b> in the <b>Reqd</b> field.</p>
<b>ReplyTo</b>	Output	<p>The destination name to which Substation ES could not successfully write. Substation ES adds this field to the TIBCO Enterprise Message Service message as a user property when writing a TIBCO Enterprise Message Service dead message.</p>
<b>ReplyTo-Type</b>	Output	<p>The destination type of <b>ReplyTo-Name</b>. Substation ES adds this field to the EMS message as a user property when writing an EMS dead message.</p>
<b>service-id</b>	Output	<p>Substation ES adds this field to the user's message. The content is the service identifier.</p>
<b>ss-id</b>	Output	<p>Substation ES adds this field to the user's message. The content is the Substation identifier.</p>

## Defining System Fields on an Inbound Conversion Rule

An input user application message can contain Substation ES system fields by defining system fields on an inbound conversion rule.

You can perform the following steps to finish the task:

## Procedure

1. Define the required system fields to the message definition.
2. Define the attributes on the conversion rule for the appropriate recipe definition.

For details on the definitions, see the descriptions of DCUI panels in [Configuration of Services](#).

```

----- Define Msg-Field / Buffer-Field Conversion Row 1 to 16 of 16
Command ==>
                                Scroll ==> CSR

Conversion Id   ==> CVR-I-TSRR-01
Using Message   ...: MSG-IO-RR-01
Using Buffer     ...: BUF-IO-BF01
Max Buffer Size ==> 0

Trunc  Req'd  BfFld  Msg  Field                               | Num  Buffer Field
-----|-----
Y      Y      1      IVP-ID                               | 1    IVP-ID
Y      Y      2      IVP-DESCR                             | 2    IVP-DESCR
Y      N      3      IVP-RESULT                             | 3    IVP-RESULT
Y      N      4      IVP-BES-NAME                           | 4    IVP-BES-NAME
Y      N      5      IVP-BES-TRAN-ID                         | 5    IVP-BES-TRAN-ID
Y      N      6      IVP-BES-PGM-NAME                       | 6    IVP-BES-PGM-NAME
Y      N      7      IVP-USERID                             | 7    IVP-USERID
Y      N      9      IVP-REPLY-LILSECS                      | 8    IVP-START-LILSECS
Y      Y      8      IVP-START-LILSECS                      | 9    IVP-REPLY-LILSECS
Y      Y      0      IVP-SUBJ-SFX                           | 10   IVP-SUBJ-SFX
Y      N     12      IVP-BTCH-SEQ-NO                        | 11   IVP-BES-DELAY
S      Y      0      tibss-envelope                         | 12   IVP-BTCH-SEQ-NO
S      N      0      tibss-password                         | 13   IVP-START-DATE-CHAR
S      N      0      tibss-userid                          | 14   IVP-START-TIME-CHAR
Y      Y      0      IVP-REPLY-DATE-CHAR                    | 15   IVP-REPLY-DATE-CHAR
Y      Y      0      IVP-REPLY-TIME-CHAR                    | 16   IVP-REPLY-TIME-CHAR
***** Bottom of data *****

```



System fields have an **S** defined in the Trunc column. Substation ES does not pass the contents to the back-end application buffer because they have a **0** defined in the BfFld column. Instead, Substation ES passes the contents to the required BES process for usage or validation.

## System Fields Used as Property Fields

An output or input user application message can contain Substation ES information in system fields. You can configure Substation ES to pass system fields for use as property fields.

### System Fields in Properties

Property Name	B/E/V	Value	Function
jmscorrelationid	B	Buffer Field Name	Puts the buffer field of the Trigger outgoing message in the correlation ID.
jmscorrelationid	V	\$\$envelope	Puts the envelope token in the JMS Correlation ID of the Trigger outgoing message, rather than in the property field called <b>tibss-envelope</b> .
jmscorrelationid	V	Text	Puts the text value specified by the user into the correlation ID.
jmscorrelationid	E	Buffer Field Name	Puts the correlation ID of the incoming message in this buffer field location.

Property Name	B/E/V	Value	Function
jmsmessageid	E	Buffer Field Name	Puts the message ID of the incoming message in this buffer field location.
tibss-BES	E	\$\$SYSTEM	Sets the name of the back-end system to be used. The name entered must match one of the CICS or IMS names, as specified with the <b>INTF_ID</b> parameter.
tibss-BES	V	\$\$SYSTEM	Puts the name of the back end system on the outgoing message as a property, <b>tibss-BES</b> .
tibss-envelope	E	\$\$SYSTEM	Looks for the property <b>tibss-envelope</b> and places it in the message token area, which is used in BES for Request/Reply events.  The recipe response message automatically adds the property, <b>tibss-envelope</b> .
tibss-envelope	V	\$\$SYSTEM	Puts the Substation ES back-end system context information on the trigger's outgoing message as a property, <b>tibss-envelope</b> .
tibss-MFS-Name	E	\$\$SYSTEM	Sets the IMS MFS map name to be used. Specify 1 to 8 characters.
tibss-password	E	\$\$SYSTEM	Sets a password from the message to the back end system.
tibss-resource	E	\$\$SYSTEM	Sets the resource name for the back end system.
tibss-rro-sysid	E	\$\$SYSTEM	For the CICS back end. Sets the remote system ID when using CICS MRO.
tibss-rro-tran-code	E	\$\$SYSTEM	For the CICS back end. Sets the remote transaction ID when using CICS MRO.
tibss-service-id	V	\$\$SYSTEM	Puts the service ID of the trigger or recipe on the outgoing message as a property, <b>tibss-service-id</b> .
tibss-ss-id	V	\$\$SYSTEM	Puts the Substation ES ID on the outgoing message as a property, <b>tibss-ss-id</b> .
tibss-userid	E	\$\$SYSTEM	Sets the user ID of the CICS transaction.



## Trigger Service Example

The following one is an example setup for a Trigger service that uses Substation ES system fields as property fields.

```
SXTP05PT ----- EMS Properties (Trigger) ----- Row 1 to 11 of 15
Command ==>                                     Scroll ==> CSR
```

```
Output Buffer Id: BUF-IO-BF01
```

Property Name	B/V	Value
tibss-BES	V	\$\$SYSTEM
tibss-service-id	V	\$\$SYSTEM
tibss-ss-id	V	\$\$SYSTEM
jmscorrelationid	V	\$\$envelope
tibss-envelope	V	\$\$SYSTEM

## Recipe Service Example

The following ones are examples of a Recipe service set up to use Substation ES system fields as property fields for inbound and outbound message.

```
SXTP05PR ----- EMS Properties (Recipe) ----- Row 1 to 11 of 15
Command ==>                                     Scroll ==> CSR
```

```
Output Buffer Id: <None>
```

```
Input Buffer Id: <None>
```

Property Name	B/E/V	Req	Value (100 char)
tibss-BES	V	Y	\$\$SYSTEM
tibss-service-id	V	Y	\$\$SYSTEM
tibss-ss-id	V	Y	\$\$SYSTEM

```
----- EMS Properties (Recipe) ----- Row 1 to 11 of 15
Command ==>                                     Scroll ==> CSR
```

```
Output Buffer Id: BUF-IO-BF01
```

```
Input Buffer Id: BUF-IO-BF01
```

Property Name	B/E/V	Req	Value (100 char)
jmsmessageid	E	_	IVP-DESCR
tibss-resource	E	_	\$\$SYSTEM
tibss-service-id	V	_	\$\$SYSTEM

## Dynamic Transaction IDs

You can dynamically specify a user transaction that a resource is to execute under in CICS by supplying Substation ES with an input communication field, the message property `tibss-tran-code`, in the TIBCO message.



This section applies to Substation ES 2.5 and later releases only.

When Substation ES identifies the message property `tibss-tran-code` in an incoming message, it internally requests CICS to run the recipe under the transaction ID in `tibss-tran-code`. The following process takes place in CICS:

1. Substation ES starts the transaction CSMI to handle EXCI communication.
2. CSMI switches the Transaction Id to the value of `tibss-tran-code`.
3. Security verification is done on the new transaction ID by CICS.
4. For DB2, DB2CONN is set up based on the new transaction ID by DB2.

Note the following items when using dynamic transaction IDs:

- If `tibss-tran-code` is used, the value known as **User Tran** on the run-time recipe is set to Y for that instance. The **Mirror Tran** field on the recipe is ignored.
- The use of `tibss-tran-code` is only available for CICS transactions (not IMS).
- If the message property field `tibss-tran-code` and the map message field `tibss-tran-code` are both in a message, the property value is used.
- For MOIs other than 2, the programs SXCOSRVD and SXCOSRVG must be allowed to run under the transaction ID specified in `tibss-tran-code`.
- The transaction ID specified by `tibss-tran-code` does not have to be defined in the CICS PCT.
- CSMI normally is defined by IBM as “below the line” (TASKDATALOC = Below). To prevent “short on storage” situations, you might need to redefine CSMI as “above the line” (TASKDATALOC = Any). Since this is CSMI and is used by other processes, be sure to test this type of change to ensure that it does not affect other critical processes.



## User Fields

User fields are only provided if you select the appropriate Reply Method on the Recipe or Trigger Definition panel. When in use, the fields are in the user message and the Substation ES error message `tibss-error-msg`.

If the EMS-USE-PROP flag in the ESB SIP member is enabled, the fields are included as property fields of the user message. User fields are only provided by the Substation ES on output.

For more details, see [Application Notifications](#).

## Definitions

The following table shows a list of the user fields, whose names do not include the default prefix `tibss-`.

Field Name	Description
<b>status</b>	<p>If a transformation, Substation ES, or BES error occurs, Substation ES adds the <b>status</b> field to the output message.</p> <ul style="list-style-type: none"> <li>• Only one <b>status</b> field is added to a message. The value of the <b>status</b> field is incremented for each notification written to the <code>tibss-error-msg</code> message.</li> <li>• In case of no errors, the status value is zero (0).</li> </ul>
<b>error-msg</b>	This field is created if the appropriate Reply Method indicator has been set and if a transformation, Substation ES, or BES error is detected.
<i>user field</i>	This is the name of the field of a message defined to the Substation ES configuration that caused the transformation error. The content is a Substation ES descriptive and documented message.
<b>error-info</b>	This is a Substation ES warning or error informational field that is a Substation ES descriptive and documented message or a BES message.

Field Name	Description
<b>ss-cfg-id</b>	This is the current recipe or trigger identifier field and is always in a <code>tibss-error-msg</code> message.
<b>ss-id</b>	This is a Substation ES informational field that is always in a <code>tibss-error-msg</code> message.

## Attributes

The following table lists the attributes for user fields, which do not include the default prefix **tibss-**.

### *User Field Attribute Summary*

Field Name	Attribute	Location and Description
<b>status</b>	Int32	This field, located in the user's message segment, specifies whether the interaction with the Substation ES has succeeded.
<b>error-msg</b>	Rendezvous nested message EMS map message	This field, located in the user's message segment, is the Substation ES error message field.
<b>error-info</b>	String256 char maximum	This field, located in the nested error message, contains Information on an error condition detected by Substation ES.
<i>user-field</i>	String	This field, located in the nested error message, contains the name on the user's input message as defined in the configuration. If a data transformation error occurs, this field also contains information on the error.
<b>cfg-id</b>	String32 char maximum	This field, located in the nested error message, identifies the Substation ES transformer recipe or trigger.
<b>ss-id</b>	String17 char maximum	This field, located in the nested error message, is the identifier of the Substation ES instance that reported the error.

## Application Notifications

Substation ES Application Notification provides information to applications regarding possible abnormal conditions, such as data transformation errors, possible Substation ES runtime failures and errors or failures that occur during the processing of applications requests.

An Application Notification consists of a **status** field and a nested or sub-message, or the fields are included as property fields of the user's message if the `EMS-USE-PROP` flag in the ESB SIP member is enabled. While communicating with Substation ES, if an abnormality occurs, the TIBCO messaging application receives an Application Notification. The application can log and trace abnormalities and possibly correct them.

Depending on the direction in which the failure of execution was detected, Substation ES sends back the information relayed by Application Notification to the TIBCO messaging application along with the application's request or reply and error message.

An Application Notification is included in messages that are delivered to the publish and error subjects as defined in the recipe and trigger definition panel.

## Information Types

By specifying the appropriate Reply Method on the recipe or trigger definition, you can choose to have your external applications receive no status input or either of the following items:

- The status field in all circumstances
- The status field only when an error condition exists

The field names that identify errors in the messages sent to user applications are defined in the user field. For details, see [Attributes](#).

## Structure

If a user application must receive notifications, you must specify the appropriate Reply Method for the recipe or trigger definition. The communication field **tibss-status** validates the delivery of notifications.

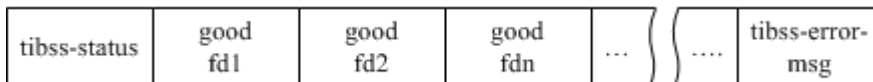
If an application receives a **tibss-status** field and the contents are nonzero, the user's message receives an associated nested error message with the error information. When the ESB interface parameter **EMS-USE-PROP** is set to Y, or if the message is a BYTES or TEXT message, the nested error message fields are sent as properties fields.

The format of the notification is as follows:

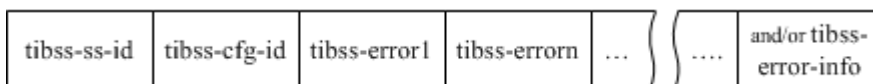
- Notification Control Fields within the user's received message.
- Notification Message Fields within a nested message.

The following figures illustrate the structures of those fields.

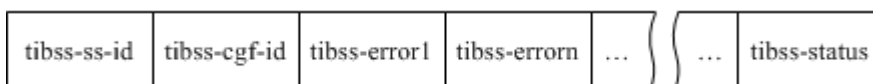
### *Structure of Application Notification Message*



### *Nested Structure of Substation ES Error Message*



### *Structure of Substation ES Error Messages as Properties Fields*



# IVP Resources and Configuration File Utilities

The section describes the Substation ES sample programs and the usage summary of the DD names and parameter combinations for each Transformer Configuration File utility.

- [Sample IVP Resources](#)
- [Summary of Substation ES ESB Configuration File Utilities](#)

## Sample IVP Resources

The following tables list the Substation ES sample programs, written in COBOL or C. The source for the COBOL samples is located in the *USERHLQ.COB* library and, for the C samples, in the *USERHLQ.C* library.

### Sample IVP Programs for Invoking Resources for CICS Interface

Description	Resource Name and Type	Program Source and Language
Request or reply application requesting an invocation of a CICS program through the DPL method	SXCCI001 (program)	SXCCI001 COBOL
	SXC3I001 (program)	SXC3I001 C
Request or reply application requesting an invocation of a CICS program through the DPL TSQ method	SXCCI012 (program)	SXCCI012 COBOL
	SXC3I012 (program)	SXC3I012 C
Request or reply application requesting an invocation of a CICS program through the DPL Storage Pointer method	SXCCI013 (program)	SXCCI013 COBOL
	SXC3I013 (program)	SXC3I013 C
Request or reply application requesting an invocation of a CICS program through the DPL Container method	SXCCI014 (program)	SXCCI014 COBOL
Request or reply application requesting an invocation of a CICS program through the DPL multiple containers method.	SXCCI015 (program)	SXCCI015 COBOL
Substation ES HVT trigger message initiated from within CICS	SXTF (transaction)	SXCCI032 COBOL
	SXT6 (transaction)	SXC3I032 C

Description	Resource Name and Type	Program Source and Language
Substation ES trigger message initiated from within CICS	SXTT (transaction)	SXCCI002 COBOL
Sender application requesting an invocation of a CICS program through the XCTL method	SXCCI003 (program)	SXCCI003 COBOL
Sender application requesting that a transaction in the CICS region be initiated	SXTI (transaction)	SXCCI004 COBOL
Application publishing a message to the Substation ES requesting that a TDQ entry be written with the message content	SXQT (TDQ name)	Substation ES internal function
Application publishing a message to the Substation ES requesting that a TSQ entry be written with the message content	SXCTSQI1 (TSQ name)	Substation ES internal function
Application displaying the records on the Dead Message Queue with the ability to delete and resend records in the CICS region	SXTD (transaction)	SXCCODMQ COBOL
Application publishing a message and expecting a reply from Substation ES that invokes a CICS program using DPL  Most z/OS COBOL data types are used in this example.	SXCCI010 (program)	SXCCI010 COBOL

#### Sample IVP Programs for Invoking Resources for RED Interface

Description	Resource Name and Type	Program Source and Language
Request or reply application requesting an invocation of a CICS program through the DPL method	SXCCI001 (program)	SXCCI001 COBOL
	SXC3I001 (program)	SXC3I001 C
Request or reply application requesting an invocation of a CICS program through the DPL Container method	SXCCI014 (program)	SXCCI014 COBOL
Request or reply application requesting an invocation of a CICS program through the DPL multiple containers method	SXCCI015 (program)	SXCCI015 COBOL

Description	Resource Name and Type	Program Source and Language
Substation ES RED CICS trigger message initiated from within CICS	XTC2 (transaction)	SXXCCI32 COBOL
	XT32 (transaction)	SXX3CI32 C
Application publishing a message and expecting a reply from Substation ES that invokes a CICS program using DPL  Most z/OS COBOL data types are used in this example.	SXCCI010 (program)	SXCCI010 COBOL

### Sample Programs for CICS Outbound Requests or Replies

Description	Resource Name and Type	Program Source and Language
CICS-initiated application that sends a request through Substation ES and expects a reply within a period of time	SXRC (transaction)	SXCCIRR1 COBOL
	SXR3 (transaction)	SXC3IRR1 C
CICS application that processes the request and that replies to the CICS-initiated request or reply transaction	SXCCIRR3 (program)	SXCCIRR3 COBOL

### Sample Programs for RED CICS Outbound Requests or Replies

Description	Resource Name and Type	Program Source and Language
RED CICS-initiated application that sends a request through Substation ES and expects a reply within a period of time	XTR3 (transaction)	SXX3IRR1 C

### Sample IVP Programs for IMS Interface

Description	Resource Name and Type	Program Source and Language
Application publishing a message and expecting a reply from Substation ES that invokes an IMS Transaction (MPP) through OTMA	SXICIT01 (transaction)	SXICIP01 COBOL

Description	Resource Name and Type	Program Source and Language
Application publishing a message and expecting a reply and a trigger from Substation ES that invokes an IMS Transaction (MPP) through OTMA	SXICIT02 (transaction)	SXICIP02 COBOL
Version of SXICIT02 demonstrating the use of a variable-length transaction name (not the first eight characters of the message)	IT03 (transaction)	SXICIP03 COBOL
IMS BMP Program sending trigger messages through Substation ES to an awaiting subscriber	None	SXICIPB1 COBOL
IMS conversational	IVTCB (transaction)	SXR3ICV1 C
IMS BMP program using request/reply (synchronous callout or ICAL)	None	SXICIRR1 COBOL

## Summary of Substation ES ESB Configuration File Utilities

The following two tables list the usage summary of the DD names and parameter combinations for each Transformer Configuration File Utility.

### Summary of Configuration File Utility: DD Names

DD Name	Utility Type			
	Batch Copy	Batch Import	Batch Export	Selective Export
CONFIG	M	M	M	M
CONFIG2	M			
IMPORT		M		
EXPORT			M	M

### Summary of Configuration File Utility: Parameters

Parameter	Utility Type			
	Batch Copy	Batch Import	Batch Export	Selective Export
-c --> Compress	M			
-cxxx --> Control DD name				O



Parameter	Utility Type			
	Batch Copy	Batch Import	Batch Export	Selective Export
-Dn --> Debug	O	O	O	O
-Ixxx --> Input DD name	O	O	O	O
-Mxxx --> Import DD name		M		
-Nxxx --> Export DD name			M	O
-Oxxx --> Output DD name	O			
-P --> Print	O	O	O	
-R --> Read-only	O	O	O	O
-T --> Initialize		O		
-V --> Validate	O	O	O	
-X --> Hex dump	O	O	O	
-Z --> Zero file		O		

**Key:**

- M: the required parameter value for the process
- O: the optional parameter value for the process

For the Batch Import utility:

- Do not use the -v option to import data into an empty configuration file. If you use -v against a Rendezvous message with duplicate field names, the error SXT7075E occurs.
- Do not use the -z option to import data into a configuration file that is not empty. Otherwise, all the existing data in the configuration file is lost.



# Conversion Formats

The section describes the supported Rendezvous data types and TIBCO EMS data types during data transformation (mapping) performed by Substation ES.

- [Conversion Formats from Rendezvous](#)
- [Conversion Formats from TIBCO Enterprise Message Service](#)

## Conversion Formats from Rendezvous

The following two tables list the supported Rendezvous data types during data transformation (mapping) performed by Substation ES.

### RV-to-COBOL Data Types

RV Field	Usage	COBOL Picture	Maximum Value	Size in Bytes
STRING	Alphanumeric	PIC X(n)		n
OPAQUE	Alphanumeric	PIC X(n)		n
IPPORT16	BINARY	9(4) BINARY	+32767	2
		9(4) COMP-5	+65535	2
IPADDR32	Alphanumeric	PIC X(15)	255.255.255.255	15
	BINARY	9(9) COMP-5		4
		S9(10) BINARY		8
DATETIME	C runtime value = ctime	Field = 10 digits		
		9(9) COMP-5		4
BOOLEAN	Alphanumeric	PIC X(1)		1
	Numeric	Field => 2 bytes		

### RV-to-COBOL Data Types (Numeric Fields)

RV Field	Usage	COBOL Picture	Maximum Value	Size in Bytes
I8	BINARY	S9(3) BINARY	+127/-128	2
	ZONED	S9(3) DISPLAY		3
U8	BINARY	9(3) BINARY	+255	2
	ZONED	S9(3) DISPLAY		3
I16	BINARY	S9(4) BINARY	+32767/-32768	2

RV Field	Usage	COBOL Picture	Maximum Value	Size in Bytes
	ZONED	S9(5) DISPLAY		5
	PACKED	S9(5) COMP-3		3
U16	BINARY	9(4) COMP-5	+65535	2
	ZONED	9(5) DISPLAY		5
	PACKED	9(5) COMP-3		3
I32	BINARY	S9(9) BINARY	+2147483647/ -2147483648	4
	ZONED	S9(10) DISPLAY		10
	PACKED	S9(10) COMP-3		6
U32	BINARY	9(9) COMP-5	+4294967295	4
	ZONED	9(10) DISPLAY		10
	PACKED	9(10) COMP-3		6
I64	BINARY	S9(18) BINARY	+/- 999,999,999,999,999, 999	8
	ZONED	S9(18) DISPLAY		18
	PACKED	S9(18) COMP-3		10
U64	BINARY	9(18) BINARY	+999,999,999,999,999 ,999	8
	ZONED	9(18) DISPLAY		18
	PACKED	9(18) COMP-3		10
F32	FLOAT	COMP-1		4
	ZONED	S9(10)V999		13
	PACKED	S9(10)V999 COMP-3		7
F64	FLOAT	COMP-2		8
	ZONED	S9(15)V999 DISPLAY		18
	PACKED	S9(15)V999 COMP-3		10

## Conversion Formats from TIBCO Enterprise Message Service

The following tables list the supported TIBCO Enterprise Message Service data types during data transformation (mapping) performed by Substation ES.

### EMS-to-COBOL Data Types

Substation Message	EMS*	Usage	COBOL Picture	Maximum Value	Size in Bytes
STRING	UTF8	Alphanumeric	PIC X(n)		n
OPAQUE	BYTES	Alphanumeric	PIC X(n)		n
I8	BYTE	Alphanumeric	PIC X		1
		Numeric	Field => 2 bytes		
BOOLEAN	BOOL	Alphanumeric	PIC X		1
		Numeric	Field => 2 bytes		



EMS\* indicates TIBCO Enterprise Message Service representation for the field in the Substation ES Message Definition panel.

### EMS-to-COBOL Data Types (Numeric Fields)

Substation Message	EMS*	Usage	COBOL Picture	Maximum Value	Size in Bytes
I8	BYTE	BINARY	S9(3) BINARY	+127/-128	2
		ZONED	S9(3) DISPLAY		3
		BINARY	9(3) BINARY	+127	2
		ZONED	9(3) DISPLAY		3
I16	SHORT	BINARY	S9(4) BINARY	+32767/-32768	2
		ZONED	S9(5) DISPLAY		5
		PACKED	S9(5) COMP-3		3
		BINARY	9(4) COMP-5	+ 32767	2
		ZONED	9(5) DISPLAY		5
		PACKED	9(5) COMP-3		3

Substation Message	EMS*	Usage	COBOL Picture	Maximum Value	Size in Bytes
I32	INT	BINARY	S9(9) BINARY	+2147483647/-2147483648	4
		ZONED	S9(10) DISPLAY		10
		PACKED	S9(10) COMP-3		6
		BINARY	9(9) COMP-5	+2147483647	4
		ZONED	9(10) DISPLAY		10
		PACKED	9(10) COMP-3		6
U64	LONG	BINARY	S9(18) BINARY	+/-999,999,999,999,999,999	8
		ZONED	S9(18) DISPLAY		18
		PACKED	S9(18) COMP-3		10
		BINARY	9(18) BINARY	+ 999,999,999,999,999,999,999	8
		ZONED	9(18) DISPLAY		18
		PACKED	9(18) COMP-3		10
F32	FLOAT	FLOAT	COMP-1		4
		ZONED	S9(10)V999		13
		PACKED	S9(10)V999 COMP-3		7
F64	DOUBLE	FLOAT	COMP-2		8
		ZONED	S9(15)V999 DISPLAY		18
		PACKED	S9(15)V999 COMP-3		10



EMS\* indicates TIBCO Enterprise Message Service representation for the field in the Substation ES Message Definition panel.

# BusinessWorks Sample Projects for Substation ES

The section describes the sample projects available for TIBCO ActiveMatrix BusinessWorks on IMS and CICS.

- [IMS Project](#)
- [CICS Project](#)

## IMS Project

You can use sample projects available for TIBCO ActiveMatrix BusinessWorks on IMS. To use the IMS Project sample programs, you must set the following global variables:

Global Variable	Setting
<i>JmsProviderUrl</i>	The URL of the EMS Server
<i>JmsUserid</i>	The User ID to access the EMS server
<i>JmsPassword</i>	The password associated with the EMS server user ID

## TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on IMS

The following table describes the sample projects available for TIBCO ActiveMatrix BusinessWorks on IMS:

Description	Service Name	Program
<b>Substation HeartBeat event consumer</b>		
Listens to the topic <code>tibss.heartbeat</code> for a TEXT message.	HEARTBEAT (Trigger)	Substation Internal
<b>Listener for IMS Callout Process Definition</b>		
1. To start, execute the IMS BMP program. 2. TIBCO ActiveMatrix BusinessWorks listens to the queue <code>tibss.IMS.sync.Request</code> for a MAP message. 3. The message is sent by IMS by way of Substation. 4. The BusinessWorks process sends back an updated MAP message to <code>tibss.IMS.sync.Response</code> . 5. Substation ES reads the reply message and forwards it to the IMS program, issuing a RESPONSE transaction. Synchronization is done using the <code>tibss-envelope</code> field or property.	CALLOUT-REQUEST (Trigger)        SXI-CALLOUTRESPONSE (Recipe)	Program
<b>Listener for BYTES message - Queue</b>		

Description	Service Name	Program
Listens to the queue <code>tibss.trigger</code> for BYTES messages.	TRIGGER-BYTES (Trigger)	Substation Internal
<b>Listener for MAP message - Queue</b>		
1. Listens to the queue <code>tibss.trigger</code> for MAP messages. 2. Sent by Substation ES Trigger TRIGGER-FLDS using the IMS Trigger process from IMS BMP SXICIPB1 program.	TRIGGER-FLDS (Trigger)	IMS BMP SXICIRR1 COBOL
<b>Listener for TEXT message - Queue</b>		
Listens to the queue <code>tibss.trigger</code> for TEXT messages.	TRIGGER-TEXT (Trigger)	IMS BMP SXICIPB1 COBOL

## CICS Project

You can use sample projects available for TIBCO ActiveMatrix BusinessWorks on CICS. To use the CICS Project sample programs, you must set these global variables:

Global Variable	Setting
<i>JmsProviderUrl</i>	The URL of EMS Server
<i>JmsUserId</i>	The User ID to access EMS Server
<i>JmsPassword</i>	The password associated with the EMS Server user ID

### TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on CICS

The following table describes the sample projects available for TIBCO ActiveMatrix BusinessWorks on CICS:

Description	Service Name	Program
<b>Call CICS Pgm SXC3I001 - Bytes - Queue</b>		
1. Sends a message to the queue. 2. Receives a Bytes message response on <code>replyProcessId</code> .	DPL-BYTES (Recipe)	CICS SXC3I001 C
<b>Call CICS Pgm SXC3I001 - MAP - Queue</b>		



Description	Service Name	Program
<ol style="list-style-type: none"> <li>1. Sends a MAP message to the queue <code>tibss.CICS.C.Request</code>.</li> <li>2. Receives a MAP message response on <code>SXCCI001ProcessId</code>.</li> </ol>	SXC-DPL-C (Recipe)	CICS SXCI001 C
<b>Call CICS Pgm SXCCI001 - MAP - Queue</b>		
<ol style="list-style-type: none"> <li>1. Sends a MAP message to the queue <code>tibss.CICS.Cobol.Request</code>.</li> <li>2. Receives a MAP message response on <code>SXCCI001ProcessId</code>.</li> </ol>	SXC-DPL-COBOL (Recipe)	CICS SXCCI001 COBOL
<b>Listener CICS Bytes Messages - Queue</b>		
<ol style="list-style-type: none"> <li>1. Listens to the EMS queue <code>tibss.trigger</code> for BYTES messages.</li> <li>2. Sent by Substation ES using CICS transaction SXTF or SXT6 <code>SXTF,R,1,, TRIGGER-BYTES</code></li> </ol>	TRIGGER-BYTES (Trigger)	CICS Transaction SXTF or SXT6
<b>Listener CICS MAP Messages - Queue</b>		
<ol style="list-style-type: none"> <li>1. Listens to the EMS queue <code>tibss.trigger</code> for MAP messages.</li> <li>2. Sent by Substation ES Trigger TRIGGER-FLDS using CICS transaction SXTF or SXT6 <code>SXTF,R,1,, TRIGGER-FLDS</code></li> </ol>	TRIGGER-FLDS (Trigger)	CICS Transaction SXTF or SXT6
<b>Listener CICS Text Messages - Queue</b>		
<ol style="list-style-type: none"> <li>1. Listens to the EMS queue <code>tibss.trigger</code> for TEXT messages.</li> <li>2. Sent by Substation ES Trigger TRIGGER-TEXT using CICS transaction SXTF or SXT6 <code>SXTF,R,1,, TRIGGER-TEXT</code></li> </ol>	TRIGGER-TEXT (Trigger)	CICS Transaction SXTF or SXT6
<b>Listener for Substation Errors</b>		
<ol style="list-style-type: none"> <li>1. Listens to the EMS queue <code>tibss.error.out</code> for MAP Substation ES error messages.</li> <li>2. Sent by Substation ES error processing.</li> </ol>	Substation Internal	Substation Internal
<b>Listener for CICS Request-Reply Process</b>		

