TIBCO Substation ES™

Configuration and Resources

Software Release 2.6
February 2010
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 ACCESSSED 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 LICENSE.PDF) 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.

TIB, TIBCO, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO Rendezvous, TIBCO Enterprise Message Service, 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.TXT 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 © 1999-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information
# Contents

## Tables

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>vii</td>
</tr>
</tbody>
</table>

## Preface

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ix</td>
</tr>
</tbody>
</table>

Changes from the Previous Release of this Guide ................................. x

Related Documentation

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xi</td>
</tr>
</tbody>
</table>

  * TIBCO Substation ES Documentation ........................................... xi
  * Other TIBCO Product Documentation ........................................... xi
  * Third Party Documentation .................................................. xii

How to Contact TIBCO Support ....................................................... xiii

## Chapter 1 Substation ES Transformer Configuration .............................. 1

Data Transformation Overview .......................................................... 2

Transformer Configuration Definitions ................................................. 3
  * Understanding Metadata Definitions ........................................ 4

Configuration File Allocation ............................................................ 5

Configuration Guidelines ...................................................................... 6
  * Defining Back-end Systems (BES) .................................................. 7
  * Defining Network Parameters (Transport for ESB) ......................... 8
  * Using the Cobol Copybook Converter ........................................... 10
  * Defining Messages ................................................................... 12
  * Defining Buffers .................................................................... 13
  * Defining Conversion Rules ....................................................... 15
  * Defining Recipes .................................................................... 17
  * Defining Triggers ................................................................... 19
  * Defining Groups .................................................................... 21

Transformer DCUI Panels and Usage .................................................... 22
  * Accessing the Transformer DCUI Panels ....................................... 22
  * Transformer Configuration Panel ............................................... 23
  * Message Definitions Panel ....................................................... 25
  * Define Fields in a Message Panel ............................................... 26
  * Conversion Rules using Message Definition Panel ...................... 29
  * Buffer Definitions Panel .......................................................... 30
  * Define Fields in a Buffer Panel ................................................ 31
  * Buffer Field Details Panel ........................................................ 33
  * Conversion Rules using Buffer Definition Panel ......................... 35
  * Conversion Rules Panel ............................................................ 36
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIBCO Substation ES Configuration and Resources</td>
<td></td>
</tr>
<tr>
<td>Chapter 3 Configuration File Utilities</td>
<td>109</td>
</tr>
<tr>
<td>Configuration File Overview</td>
<td>110</td>
</tr>
<tr>
<td>Chapter 2 Configuring Additional Interfaces</td>
<td>101</td>
</tr>
<tr>
<td>Adding an Additional Substation ES CICS Interface</td>
<td>102</td>
</tr>
<tr>
<td>Adding an Additional ESB Interface</td>
<td>103</td>
</tr>
<tr>
<td>Adding an Additional EMS ESB Interface With SSL</td>
<td>104</td>
</tr>
<tr>
<td>Adding an Additional Rendezvous ESB Interface With SSL</td>
<td>106</td>
</tr>
<tr>
<td>Adding an Additional Substation ES to a Single CICS Region</td>
<td>107</td>
</tr>
</tbody>
</table>
Chapter 4  Substation ES Examples ............................... 121

Overview ............................................................... 122

Substation ES Installation IVPs ................................... 123
  Request or Reply .................................................... 123
  Triggers ............................................................... 123

Substation ES Generic IVP Components ........................... 124
  Rendezvous IVP Sender Listener ................................ 124
  EMS IVP Sender Listener ......................................... 127
  IVP Generic Listener .............................................. 130
  IVP IMS Conversational Sender or Listener .................... 130
  IVP Generic Error Listener ...................................... 131

ESB Request – Reply DPL Example .................................. 132
  IVP Sender Listener Execution Parameters ................... 133

ESB Request – Reply DPL Container Example ..................... 134
  IVP Sender Listener Execution Parameters ................... 134

ESB Request – Reply DPL Storage Example ....................... 135
  IVP Sender Listener Execution Parameters ................... 135

ESB Request – Reply DPL TSQ Example ............................ 137
  IVP Sender Listener Execution Parameters ................... 137

CICS Trigger Example .................................................. 139
  CICS Execution Command ......................................... 139

CICS High Volume Trigger (HVT) Example ......................... 140
  CICS Execution Command ......................................... 140

ESB Request DPL XCTL Example ..................................... 141
  IVP Sender Listener Execution Parameters ................... 141

ESB Invoking a CICS Transaction Example ....................... 142
  IVP Sender Listener Execution Parameters ................... 142
ESB to TDQ Example ................................................................. 143
  IVP Sender Listener Execution Parameters. .......................... 143
CICS TSQ Entry Written Example ............................................. 144
  IVP Sender Listener Execution Parameters. .......................... 144
ESB Request – Reply Complex DPL Example .......................... 145
  IVP Sender Listener Execution Parameters. .......................... 145
CICS Initiated Request or Reply Example ............................... 146
  CICS Execution Command .................................................. 147
ESB Request – Reply IMS Example .......................................... 148
  IVP Sender Listener Execution Parameters. .......................... 148
ESB Request – Reply and Trigger Example ............................. 149
  IVP Sender Listener Execution Parameters. .......................... 149
IMS BMP Trigger Example ...................................................... 150
IMS Conversational Example .................................................. 151
  IVP IMS Conversational Sender or Listener Execution Parameters .............................................. 152
Transform Only Data Examples .............................................. 153
  Data Forwarder and Receiver Execution ............................... 154

Chapter 5  Application Message Communication .......................... 157
Communication Fields Overview .............................................. 158
  Communication Fields Prefix ............................................. 158
System Fields ................................................................. 159
  System Field Definition Summary ..................................... 160
  System Fields Definition .................................................. 161
  Using Dynamic Transaction IDs ........................................ 162
User Fields ................................................................. 163
  User Fields Definitions Summary ..................................... 164
Application Notifications ..................................................... 165
  TIBCO Applications Notifications Structure .......................... 165

Appendix A  Sample IVP Resources ............................................ 167

Appendix B  Transformer Configuration File Utilities Summary .............. 171

Appendix C  Conversion Formats from TIBCO Rendezvous ............. 173

Appendix D  Conversion Formats from TIBCO EMS .......................... 176

Index ................................................................. 179
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Related Documents</td>
<td>xii</td>
</tr>
<tr>
<td>Table 2</td>
<td>Configuration File Conversion Utility DDNames</td>
<td>111</td>
</tr>
<tr>
<td>Table 3</td>
<td>Transformer Conversion Utility Parameters</td>
<td>112</td>
</tr>
<tr>
<td>Table 4</td>
<td>Configuration File Utility DDNames</td>
<td>114</td>
</tr>
<tr>
<td>Table 5</td>
<td>Configuration File Utility Function Parameters</td>
<td>114</td>
</tr>
<tr>
<td>Table 6</td>
<td>Configuration File Utility Parameters</td>
<td>115</td>
</tr>
<tr>
<td>Table 7</td>
<td>Configuration File Selective Export Utility DDNames</td>
<td>118</td>
</tr>
<tr>
<td>Table 8</td>
<td>Configuration File Selective Export Utility Parameters</td>
<td>118</td>
</tr>
<tr>
<td>Table 9</td>
<td>Entity Type Values</td>
<td>119</td>
</tr>
<tr>
<td>Table 10</td>
<td>Definition Qualifier Values</td>
<td>120</td>
</tr>
<tr>
<td>Table 11</td>
<td>Rendezvous IVP Sender Listener Control Parameters</td>
<td>125</td>
</tr>
<tr>
<td>Table 12</td>
<td>EMS IVP Sender Listener Control Parameters</td>
<td>128</td>
</tr>
<tr>
<td>Table 13</td>
<td>IVP Generic Error Listener Control Parameters</td>
<td>131</td>
</tr>
<tr>
<td>Table 14</td>
<td>IMS Conversational Sender or Listener Control Parameters</td>
<td>152</td>
</tr>
<tr>
<td>Table 15</td>
<td>Data Transform Only Control Parameters</td>
<td>155</td>
</tr>
<tr>
<td>Table 16</td>
<td>System Field Names</td>
<td>159</td>
</tr>
<tr>
<td>Table 17</td>
<td>System Field Definitions</td>
<td>160</td>
</tr>
<tr>
<td>Table 18</td>
<td>User Field Attribute Summary</td>
<td>164</td>
</tr>
<tr>
<td>Table 19</td>
<td>CICS Interface Sample IVP Invocations of Resources</td>
<td>167</td>
</tr>
<tr>
<td>Table 20</td>
<td>CICS Outbound Request or Reply Sample Programs</td>
<td>168</td>
</tr>
<tr>
<td>Table 21</td>
<td>IMS Interface Sample IVP Programs</td>
<td>169</td>
</tr>
<tr>
<td>Table 22</td>
<td>Configuration File Utilities DD Names Summary</td>
<td>171</td>
</tr>
<tr>
<td>Table 23</td>
<td>Configuration File Utilities Parameters Summary</td>
<td>171</td>
</tr>
<tr>
<td>Table 24</td>
<td>TIBCO Rendezvous to Cobol Data Types</td>
<td>173</td>
</tr>
<tr>
<td>Table 25</td>
<td>TIBCO Rendezvous to Cobol Data Types (numeric fields)</td>
<td>174</td>
</tr>
<tr>
<td>Table 26</td>
<td>TIBCO EMS to Cobol Data Types</td>
<td>176</td>
</tr>
<tr>
<td>Table 27</td>
<td>TIBCO EMS to Cobol Data Types (numeric fields)</td>
<td>176</td>
</tr>
</tbody>
</table>
Preface

This manual describes the configuration of the TIBCO Substation ES product, and also includes sample code and examples.

Topics

- Changes from the Previous Release of this Guide, page x
- Related Documentation, page xi
- How to Contact TIBCO Support, page xiii
Changes from the Previous Release of this Guide

This section itemizes the major changes from the previous release of this guide.

New Content

The following have been added:

- The panel used to create or modify transport entity extensions was reorganized. For details, see Transport Entity Extensions Panel on page 46.

- Configuration content was moved from TIBCO Substation ES Operations and Administration and added to this document as Chapter 2. For details, see Chapter 2, Configuring Additional Interfaces, on page 101.

- New sections on configuring additional EMS/Rendezvous ESB interfaces with SSL. For details, see Chapter 2, Configuring Additional Interfaces, on page 101.

- New and revised control parameters for IVP components. For details, see Rendezvous IVP Sender Listener on page 124 and EMS IVP Sender Listener on page 127.
Related Documentation

This section lists documentation resources you may find useful.

**TIBCO Substation ES Documentation**

The following documents form the Substation ES documentation set:

- *TIBCO Substation ES Concepts*: Read this manual for an overview of Substation ES.
- *TIBCO Substation ES Installation*: Read this manual for instructions on site preparation and installation.
- *TIBCO Substation ES Operations and Administration*: Read this manual for details on operations and administrative tasks.
- *TIBCO Substation ES Configuration and Resources*: Read this manual for instructions on configuring communications and data conversions, and descriptions of the sample programs.
- *TIBCO Substation ES Messages and Codes*: Use this manual as a reference to error and information messages and codes.
- *TIBCO Substation ES Release Notes*: Read this document for information about new features, deprecated features, and open and closed issues.

**Other TIBCO Product Documentation**

You can find it useful to read the documentation for the following TIBCO products:

- TIBCO Rendezvous™ and TIBCO Enterprise Message Service™ software: These are TIBCO Software’s real-time transport layers that are used by the Substation ES software.
- *TIBCO Rendezvous for z/OS Installation and Configuration*  
  Read this manual for instructions on installing and operating TIBCO Rendezvous on IBM z/OS systems.
- *TIBCO Rendezvous for z/OS COBOL Reference* and *TIBCO Rendezvous C Reference*  
  Read these manuals for instructions on the TIBCO Rendezvous for z/OS COBOL and C APIs.
- *TIBCO Enterprise Message Service User’s Guide*  
  Read this manual for instructions on TIBCO EMS functionality.
- *TIBCO Enterprise Message Service C & COBOL API Reference*  
  Read this manual for instructions on the TIBCO EMS C and COBOL APIs.
Third Party Documentation

You may also find the following IBM documents useful:

Table 1  Related Documents

<table>
<thead>
<tr>
<th>Publication Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM CICS External Interfaces Guide</td>
</tr>
<tr>
<td>IBM CICS System Definition Guide</td>
</tr>
<tr>
<td>IBM CICS Transaction Server for z/OS CICS RACF Security Guide</td>
</tr>
<tr>
<td>IBM IMS/ESA Administration Guide: System</td>
</tr>
<tr>
<td>IBM IMS/ESA Customization Guide</td>
</tr>
<tr>
<td>IBM IMS/ESA Installation Volume 1: Installation and Verification</td>
</tr>
<tr>
<td>IBM IMS/ESA Installation Volume 2: System Definition and Tailoring</td>
</tr>
<tr>
<td>IBM ISPF Dialog Developers Guide and Reference</td>
</tr>
<tr>
<td>IBM RACF User’s Guide</td>
</tr>
<tr>
<td>IBM SecureWay Security Server RACF Security Administration Guide</td>
</tr>
</tbody>
</table>

You can find many of the IBM documents in these CD-ROM sets:

- Online Library Omnibus Edition MVS Collection
- Online Library Omnibus Edition z/OS Collection
How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- 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.
Chapter 1 Substation ES Transformer Configuration

This chapter describes how to configure the Substation ES to exchange data between systems running in the z/OS environment and external TIBCO messaging applications.

Topics

- Data Transformation Overview, page 2
- Transformer Configuration Definitions, page 3
- Configuration File Allocation, page 5
- Configuration Guidelines, page 6
- Transformer DCUI Panels and Usage, page 22
Data Transformation Overview

Substation ES subscribes to and publishes TIBCO messages on behalf of CICS and IMS transaction processing applications running on the z/OS system. 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 the Substation ES Transformer that you configure via the Transformer Data Configuration User Interface (DCUI) panels.

Transforming Inbound Messages

You configure Substation ES to subscribe to and process requests generated by a TIBCO messaging application (for example, inbound messages) by specifying several items, including the following:

- The message subject to listen for
- How data is transformed for delivery to a transaction processing application
- The method of invocation
- 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 this data or acknowledgement is transformed for the calling application.

For information about messaging flow and different types of Methods of Invocations (MOIs), refer to *TIBCO Substation ES Concepts*.

Transforming Outbound Messages

You configure Substation ES to publish data generated by transaction processing applications (for example, outbound messages) by using the trigger method of invocation. In this case, you assign a subject to the data that allows the data to be delivered to TIBCO messaging applications subscribing to this subject.

You 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.
Transformer Configuration Definitions

The items that are configured for the Substation ES Transformer can be organized into the following categories:

- System
- Inbound or Outbound Message Data
- Metadata

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

System

System definitions consist of the following:

- Back-end system (BES) – defines the interfaces to mainframe systems where transaction processing applications reside.
- Network – defines transport entities, meaning, ports, daemons or EMS servers, IP networks.

Inbound or Outbound Message Data

Inbound or Outbound Message Data definitions consist of the following:

- Message – defines inbound TIBCO Rendezvous or EMS 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

Metadata definitions consist of the following:

- 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 are used to achieve customized routing goals.
Understanding Metadata Definitions

The following diagram illustrates a conversion rule definition and its relationship with a message definition and a buffer definition:

*Figure 1  Conversion Rule Overview*

The following diagram illustrates a recipe or trigger definition and its components:

*Figure 2  Recipe or Trigger Overview.*
Configuration File Allocation

The File Creation JCL is used to create a new version of the Substation ES Transformer Configuration file for usage with Substation ES.

A VSAM linear file is defined and allocated. This file can be used as the repository for all the transformation configuration settings that are read by Substation ES.

Sharing the Configuration File

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

- Transformer Configuration file is a catalogued data set shared between LPARs.
- VSAM SHAREOPTIONS (2 3) must be used to limit updates to one user.
- First user who opens the configuration file for edit has update authority.
- Second and subsequent users have READONLY access.

The JCL and control statements needed to define and allocate a Transformer Configuration file as follows:

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

The data set name, DASD volume name and DASD volume device must be changed to conform to the user’s environment.

<USERHLQ> is substituted with the High Level Qualifier for your Substation ES libraries.
Configuration Guidelines

The following examples can be used as guidelines for configuring the Substation ES Transformer.

- Defining Back-end Systems (BES)
- Defining Network Parameters (Transport for ESB)
- Using the Cobol Copybook Converter
- Defining Messages
- Defining Buffers
- Defining Conversion Rules
- Defining Recipes and Defining Triggers
- Defining Groups
Defining Back-end Systems (BES)

Complete the following steps to define or modify the processing system resources or back-end system (BES) to be used with the recipe definition or trigger definition.

There are no dependencies to consider before beginning these steps.

1. Select option 4 —Define Processing System Resources panel.
   For further information about this panel, refer to Define Processing System Resources Panel on page 44.
2. Type ADD on the command line and press the ENTER key to create a new entry.
3. Select C(ics), I(ms) or A(dmin) for the Type.
4. Enter up to sixteen characters to name the processing system resource Identifier.
5. Enter up to fifteen characters to name the assigned transaction processing system interface in the Interface ID.
   The Interface ID must match the ID defined in the Interface SIP member.
6. Press the ENTER key before leaving the panel
7. Exit the Define Processing System Resources panel:
   — by pressing PFK3
     or
   — entering END on the command line and pressing the ENTER key
8. Review the upper right hand corner to check that your changes were saved.

After you have exited back to the Transformer Configuration panel, the screen displays the message Changes Saved. If you cancelled using the CAN command, it displays the message Changes Cancelled. If you made a wrong selection during the panel editing, it displays a related error message in the upper right hand corner. If the file was opened read-only and changes were made, the message is Change and read-only.
Defining Network Parameters (Transport for ESB)

Complete the following steps to define or modify a transport definition to be used with a recipe definition or trigger definition.

Refer to the TIBCO Rendezvous Administration Guide or the TIBCO Enterprise Message Service documentation for detailed information and default settings for the parameters used in this procedure.

There are no dependencies to consider before beginning these steps.

1. Select option 5 — Define Transport - Parameters from the Transformer Configuration panel.
   
   For further information about this panel, refer to Define Transport Entity Panel on page 45.

2. Type ADD on command line and press the ENTER key to create a new entry.

3. Set the field values for defining the transport.

   For EMS:
   — In the Type field, specify E for EMS.
   — In the Startup field, specify (Y/N) whether this transport starts during Substation ES initialization.
   — In the URL field, specify the host name or host IP address used by Substation ES to find the EMS server and establish communication (the TCP port can be defined here or in the Port field).
   — In the Port field, specify (when applicable) the value of the TCP port used by the Substation ES transport.
   — In the User Id field, specify a user ID if EMS requires a specific ID.
   — In the Password field, specify a password if EMS requires a specific password.
   — In the Alt. Conn and Alt. Port fields, specify the alternate URL and port for fault tolerant EMS servers.
For Rendezvous:

— In the **Type** field, specify R for Rendezvous.

— In the **Startup** field, specify (Y/N) whether this transport starts during Substation ES initialization.

— In the **URL** field, specify (when applicable) the host name or host IP address that Substation ES uses to find the Rendezvous daemon and establish communication.

— In the **Port** field, specify the daemon port if needed (default is 7500).

— In the **Service** field, specify (when applicable) the value of the UDP service port number.

— In the **Network** field, specify (when applicable) the network IP address.

— In the **Multicast** field, specify (Y/N) whether this transport is to use multicast addressing.

— In the **Multicast Address** field, specify (when applicable) the multicast addresses. This field must be specified if the **Multicast** field is set to Y.

4. Press the **ENTER** key before you leave the panel.

5. Exit the Transport Entity Extensions panel

   — by pressing **PFK3**

   or

   — entering **END** on the command line and pressing the **ENTER** key.

6. Review the upper right hand corner to check that your changes were saved.

7. When you return to the “Define Transport Entity” panel, you can change the identifier for this new transport by over typing the value in the **Identifier** field.
Using the Cobol Copybook Converter

The Copybook Converter can read a COBOL copybook and create Substation ES transformer configuration definitions.

With the selection 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 is specified, you can also define a conversion rule that creates a one to one mapping. This mapping is used in data conversion.

To create buffer, message and convert rule names identical to the copybook name, insert an asterisk (*) in the panel field corresponding to the selection.

Before beginning these steps, check the dependencies on the COBOL copybook members.

1. Select option 10 - COBOL Copybook Converter.
   For further information about this panel, refer to Copybook Converter Panel on page 93.

2. Enter the COBOL copybook member name to use for conversion in Copy Member.

3. Enter up to sixteen characters as a name that identifies the buffer definition in Buffer Identifier.

4. Enter up to sixteen characters as a name that identifies the message definition in Message Identifier.

5. Enter up to sixteen characters as a name that identifies the conversion rule definition in Convert Rule Id.

6. Enter the DSN of the partitioned data set containing the copybook member in Library containing Copy Member.

7. Enter DSNs of the partitioned data set that can contain other copybooks that are included within the copybook member in Additional libraries for COPY.

8. Press the ENTER key before you leave the panel.

9. Exit the Copybook Converter panel
   — by pressing PFK3
   — entering END on the command line and pressing the ENTER key.
10. Review the upper right hand corner to check that your changes were saved.

In some cases, features not supported by the converter can be required for the definitions. In this case, you can need to edit the definitions to change the default attributes, deselect mapping selections and apply changes.

Features currently not supported:

- Level 66, 77 and 88 items are ignored.
- Picture with BLANK WHEN ZERO, DATE FORMAT, EXTERNAL, GLOBAL are ignored.
- OCCURS clauses only generates the first occurring field items.
- REDEFINE clauses selects 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

Complete the following steps to define a new message definition or select an existing message definition for a conversion rule for a modification.

There are no dependencies to consider before beginning these steps.

1. Select option 1 — Define Message Contents panel.
   For further information about this panel, refer to Message Definitions Panel on page 25.

2. Type:
   — ADD on command line and press the ENTER key to create a new entry.
   or
   — Enter S for Sel to choose an existing definition to modify, and press the ENTER key.

3. Enter up to sixteen characters to name the message definition in Message Id.

4. Enter a valid data type for Type.

5. Enter (when applicable) the implied number of digits after a decimal point for Dec.

6. Enter up to fifty-nine characters to name the message field in Name.

7. Press the ENTER key before you exit the panel.

8. Exit the Define Message Content panels
   — by pressing PFK3
   or
   — entering END on the command line and press the ENTER key.

9. Review the upper right hand corner to check that your changes were saved.
Defining Buffers

Complete the following steps to define a new buffer definition or select an existing buffer definition for a modification.

There are no dependencies to consider before beginning these steps.

1. Select option 2 — Define Buffer Areas panel
   For further information about this panel, refer to Buffer Definitions Panel on page 30.

2. Type:
   — ADD on command line and press the ENTER key to create a new entry.
   or
   — Enter S for Sel to choose an existing definition to modify, and press the ENTER key.

3. Enter up to sixteen characters to name the buffer definition in Buffer ID.

4. Enter (optional) the maximum size of the buffer in Maximum Size.

5. Enter the mainframe data type that the invoked application expects in Type.

6. Enter the value to use for the field starting byte location offset in the buffer in Start.

7. Enter the value to use for the length of the field in Len. The length is in bytes, not decimal characters retained. Thus a COBOL PIC S9(9) COMP retains 9 characters, but is 4 bytes in length.

8. Enter (when applicable) the implied number of digits after a decimal point in Dec.

9. Enter (when applicable) the character to use for padding purposes in Pad.

10. Enter up to forty-five characters to name the buffer field in Name.

11. Enter S under Sel, and press the ENTER key.

   You see the Buffer Fields Definition panel. For further information about this panel, refer to Define Fields in a Buffer Panel on page 31.

12. Enter (when applicable) N, L or T for the sign selection in Separate Sign.

13. Enter (when applicable) Y to Justify Right.
14. 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 Initial Value.

15. Enter (when applicable) the value for the length of the Initial Value field in Initial Value Length.

16. Press the ENTER key before you exit the panel.

17. Exit the Define Buffer Areas panels
   — by pressing PFK3
   or
   — entering END on the command line and pressing the ENTER key.

18. Review the upper right hand corner to check that your changes were saved.
Defining Conversion Rules

Complete the following steps to define a new conversion rule definition or select an existing conversion rule definition for modification.

Successfully completing these steps depends on first completing:

- Message Definition
- Buffer Definition

1. Select option 3 — Define Conversion Relationships panel.
   For further information about this panel, refer to Conversion Rules Panel on page 36.

2. Type:
   — ADD on command line and press the ENTER key to create a new entry.
   or
   — Enter S for Sel to choose an existing definition to modify, and press the ENTER key.

3. Under Sel, enter S next to the message definition you want to use and press the ENTER key.
   For further information about this panel, refer to Select a Message Definition Panel on page 37.

4. Under Sel, enter S next to the buffer definition you want to use and press the ENTER key.
   For further information about this panel, refer to Select a Buffer Definition Panel on page 38.

5. Enter up to sixteen characters to name the conversion rule definition in Conversion ID.

6. Enter (when applicable) a value to use for the maximum buffer size in Max Buffer Size.

7. Enter Y or N to truncate data that extends beyond the specified buffer length in Trunc.

8. Enter Y or N to indicate if the field is required in Reqd.

9. Enter the sequence number of the buffer field that maps to the message field for the transformation in BFFld.

10. Press the ENTER key before you exit the panel.
11. Exit the Define Conversion Rules panels
   — by pressing PFK3
   or
   — entering END on the command line and pressing the ENTER key.
12. Review the upper right hand corner to check that your changes were saved
    and there are no error messages.
Defining Recipes

Complete the following steps to define a new recipe definition or modify an existing recipe definition.

Successfully completing these steps depends on first completing:

- Transport (Network definition)
- BES definition
- Conversion rule definitions

1. Select option 6 — Define Recipes -Inbound or Outbound /Subjects/BES.
   For further information about this panel, refer to Recipe List Panel on page 48.

2. Type:
   — ADD on command line and press the ENTER key to create a new entry.
   or
   — Enter S for Sel to choose an existing definition to modify, and press the ENTER key. Enter E for Sel to set EMS-specific settings.

3. Under Sel, enter S to select the transport you want to use and press the ENTER key to continue the process.
   For further information about this panel, refer to Select Transport Panel (Recipe) on page 50

4. Under Sel, enter S to select a processing system definition and press the ENTER key to continue the process.
   For further information about this panel, refer to Define Processing System Resources Panel on page 44

5. Under Sel, enter I, O or B next to a conversion rule definition and press the ENTER key to continue the process
   For further information about this panel, refer to Select Conversion Rules Panel (Recipe) on page 52.

If you enter I (input) the first time, this step is repeated one more time to enter O (output) selection.

6. Enter up to thirty-one characters to name the Recipe Identifier.

7. Enter the invoke subject name that Substation ES listens for in order to invoke a given transformation in Subject.
8. Enter (when applicable) the outbound subject name that Substation ES uses to publish the reply message for this process in **Publish Subject**.

9. Enter (when applicable) the number of bytes in the buffer to be used for a dynamic reply subject name under **Dynamic Subject Leng**.

10. Enter (when applicable) the starting offset in the buffer that is to be used for the dynamic reply subject name in **Start**.

11. Enter a subject name for the error message subject in **Error Subject**.

12. Enter the name of the BES resource to be invoked upon receipt of a request in **Resource Name**.

13. Select the process method that Substation ES invokes for this transformation in **Method of Invocation**.

14. Select the option of outbound data that is expected for this process in **Reply Method**.

15. Press the **ENTER** key before you exit the panel.

16. Enter (when applicable) **PROP** on command line and press the **ENTER** key to set EMS Properties. For information about this panel, refer to **EMS Properties Panel on page 61**.

   Press the **ENTER** key before pressing the **PFK3** to exit the EMS Properties panel.

17. Enter (when applicable) **SEL** on command line and press the **ENTER** key to set the EMS Selector Specification. For information about this panel, refer to **EMS Selector Specification Panel on page 62**.

   Press the **ENTER** key before pressing the **PFK3** to exit the EMS Selector Specification panel.

18. Exit the Define Recipes panel
   
   — by pressing **PFK3**
   
   or
   
   — entering **END** on the command line and pressing the **ENTER** key.

19. Review the upper right hand corner to check that your changes were saved.
Defining Triggers

Complete the following steps to define a new trigger definition or modify an existing trigger definition.

Successfully completing these steps depends on first completing:

- Transport (Network definition)
- Conversion rule definitions

1. Select option 7 — Define Triggers - Outbound Convert or Subjects.
   For further information about this panel, refer to Define Triggers Panel on page 63.

2. Type:
   — ADD on command line and press the ENTER key to create a new entry.
   or
   — Enter S for Sel to choose an existing definition to modify, and press the ENTER key. Enter E for Sel to set EMS-specific settings.

3. Under Sel, enter S next to the transport definition you want to use and press the ENTER key to continue the process.
   For further information about this panel, refer to Select Transport Panel (Triggers) on page 64.

4. Under Sel, enter S or O next to a conversion rule definition and press the ENTER key to continue the process.
   For further information about this panel, refer to Select Output Conversion Rule Panel (Triggers) on page 65.

5. Enter up to thirty-one characters to name the Recipe Identifier.

6. Enter the name of the outbound subject that a message is published on for a given trigger transaction in Publish Subject.

7. Enter (when applicable) the number of bytes in the buffer to be used for a dynamic reply subject name under Dynamic Subject Leng.

8. Enter the starting offset in the buffer that is to be used for the dynamic reply subject name (when applicable) in Start.

9. Enter a subject name for error messages in Error Subject.

10. Select the option of outbound data that is expected for this process in Reply Method.
11. Enter the starting offset in the buffer that contains the text to be used for the 
    search criteria in a Trigger process in **Start**.

12. Enter the number of bytes of the text to be used for the search criteria in a 
    Trigger process in **Length**.

13. Enter the text to be used for the search criteria in a Trigger process in **Value**.

14. Press the **ENTER** key before you exit the panel.

15. Enter (when applicable) **PROP** on command line and press the **ENTER** key to set 
    EMS Properties. For information about this panel, refer to **EMS Properties 
    Panel on page 61**.

    Press the **ENTER** key before pressing the **PFK3** to exit the EMS Properties panel.

16. Exit the Define Triggers panels

    — by pressing **PFK3**

    or

    — entering **END** on the command line and pressing the **ENTER** key.

17. Review the upper right hand corner to check that your changes were saved.
Defining Groups

The Group definition combines recipe and trigger definitions to be used by Substation ES. The user can provide the group names in the transformer SIP member that tells Substation ES which groups of recipe and trigger definitions are to be loaded.

Grouping enables one configuration file to be used by multiple instances of Substation ES at the same time. Therefore, grouping reduces the requirement for defining duplicate recipe and trigger definitions.

Complete the following steps to define a new group definition or modify an existing group definition consisting of previously defined recipe and trigger definitions.

1. Select option 8 — Define Groups.
   For further information about this panel, refer to Group Definitions Panel on page 72.
2. Type:
   — ADD on command line and press the ENTER key to create a new entry.
   or
   — Enter S for Sel to choose an existing definition to modify, and press the ENTER key.
3. Under Sel, press S next to each trigger and recipe that you want to select into the group. Press the ENTER key to continue the process.
   For further information about this panel, refer to Select Triggers and Recipes for Group Panel on page 73.
4. Press the ENTER key before you exit the panel.
5. Exit the Define Groups panel
   — by pressing PFK3
   or
   — entering END on the command line and pressing the ENTER key.
6. Review the upper right hand corner to check that your changes were saved.

Successfully completing these steps depends on first completing:

- Recipes definitions
- Trigger definitions
Transformer DCUI Panels and Usage

Accessing the Transformer DCUI Panels

You access the Transformer DCUI panels as follows:

1. Go to the ISPF Command Shell (Option 6)
2. Run the following command:
   
   ```
   ex '<USERHLQ>.CLIST(SXTSSPNL)'
   ```

   where `<USERHLQ>` is substituted with the High Level Qualifier for your Substation ES installed libraries. The Substation ES Main Entry Panel appears.
3. Select option 1, Configuration.

   **NOTE:** To see a panel ID, issue "PANELID" on the command line. This toggles 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  nn
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 Descriptions

**Select Option**

The selections are: 1 - Configuration (DCUI Panels) and 2 – LTA Search and Display Facility
Transformer Configuration Panel

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

Panel Layout

SXTPMAIN ------------------- TIBCO Substation (ES) ----------------------------
Transformer Configuration

SELECT OPTION ===> 

File Name ===> <USERHLQ.CFGIVP>
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 Transport - Parameters
6 - Define Recipes - Inbound or Outbound / Destinations / BES
7 - Define Triggers - Outbound Convert or Destinations
8 - Define Groups
9 - Show Related Structures
10 - COBOL Copybook Converter

Enter END command to terminate.

Field Descriptions

Select Option

Selects the panels and functions to perform through the Transformer DCUI. The valid selections are:

1 - Define Message Contents
2 - Define Buffer Areas
3 - Define Conversion Relationships
4 - Define Processing System Resources
5 - Define Transport - Parameters
6 - Define Recipes - Inbound or Outbound / Destinations / BES
7 - Define Triggers - Outbound Convert or Destinations
8 - Define Groups
9  Show Related Structures

10  COBOL Copybook Converter

**File Name**  The name of the Substation ES Transformer configuration VSAM file. You must specify the filename before selecting an option.

**Read Only**  Selecting N specifies updates to the Transformer configuration VSAM file are to be saved. Y specifies updates are not saved.

Valid selections: Y(es) or N(o). Default: N.

**Debug**  The level of informational messages recorded to the screen. Should always be 0 unless TIBCO Software Inc. support personnel require the change.  Required.

Valid range: 0 through 5. Default: 0.
Message Definitions Panel

This panel is used to create or modify a message definition.

Panel Layout

SXTP01 ----------------------- Message Definitions ----------- Row 1 to 1 of 1
Command ===>                                                   Scroll ==> CSR

(Use ADD command to create a new message)

Sel Flds   Message Id
-   ----   ----------------
0   New-Msg

Field Descriptions

Command

The ADD command creates a new message definition. The CAN command cancels all modifications just done to the existing message definition.

Valid commands: ADD and CAN

Sel

Selects an existing message definition to modify.

Valid selections: S(elect), D(lete), R(epro) or U(sedon).

If you select U, you enter the panel described in Conversion Rules using Message Definition Panel on page 29.

Flds

The total number of fields in the message definition.

Message Id

The user-defined name of the message definition. Required, cannot be blank.

Default: New-Msg (if not specified).
Define Fields in a Message Panel

This panel is used to modify the fields of a message definition.

Panel Layout

SXTP01F ------------------ 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-Msg

<table>
<thead>
<tr>
<th>Sel</th>
<th>Type</th>
<th>Dec</th>
<th>Id</th>
<th>Seq</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

Message Id   The name of the message definition.
Sel   Modifies a previously defined field in a message definition.

Valid selections: S(elect), D(lete) or R(epro).
Type  The data type for the field.

Valid types for TIBCO EMS and TIBCO Rendezvous:

- **OPA** - RV: represents opaque byte sequence
  EMS: represents bytes
- **STR** - RV: represents character string
  EMS: represents EMS/JMS UTF8 data type
- **BOL** - represents Boolean
- **I8** - RV: represents 8-bit integer
  EMS/JMS: represents a byte
- **I16** - RV: represents 16-bit integer
  EMS: represents EMS/JMS short data type
- **U16** - RV: represents unsigned 16-bit integer
  EMS: wchar data type
- **I32** - RV: represents 32-bit integer
  EMS: represents EMS/JMS int data type
- **U64** - RV: represents unsigned 64-bit integer
  EMS: represents EMS/JMS long data type
- **F32** - represents 32-bit floating point
- **F64** - RV: represents 64-bit floating point
  EMS: represents EMS/JMS double data type
- **MSG** - RV: represents message
  EMS: represents EMS mapped message

Valid types for TIBCO Rendezvous only:

- **DAT** - represents TIBCO Rendezvous datetime
- **U8** - represents unsigned 8-bit integer
- **U32** - represents unsigned 32-bit integer
- **I64** - represents 64-bit integer
- **POR** - represents 2-byte IP port
- **IPA** - represents 4-byte IP address

Dec  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.
Id  A message field identifier. Zero (0) is a special value that signifies no field identifier. All non-zero field identifiers must be unique within each message. It is illegal to add a field that has both a NULL field name, and a non-zero field identifier. This field is not implemented in the current release.

Seq  Defines the sequence in which each field appears in the Conversion Rules panel. Numeric only.

Valid range: 0 - 9999. Default: 0.

Name  The user-defined name of the field. Required, cannot be blank.
Conversion Rules using Message Definition Panel

This panel displays the conversion rules in which a message definition is used.

You access this panel by selecting U in the Sel field of the Message Definitions panel. For details, see Message Definitions Panel on page 25.

Panel Layout

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

Message Id ===> New-Msg

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

Field Descriptions

Message Id  The name of the message definition used in the conversion rule. Required, cannot be blank.

Default: New-Msg (if not specified).

Used in these Conversion Rules  The name of the conversion rule definitions that uses the message definition.
Buffer Definitions Panel

This panel is used to create or modify a buffer definition.

Panel Layout

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

(Use ADD command to create a new buffer definition)

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

Field Descriptions

Command
The ADD command creates a new buffer definition. The CAN command cancels all modifications just done to the existing message definition.
Valid commands: ADD and CAN

Sel
Selects an existing buffer definition to modify.
Valid selections: S(elect), D(lete), R(epro) or U(sedon).
If you select U, you enter the panel described in Conversion Rules using Buffer Definition Panel on page 35.

Flds
The total number of fields in the buffer definition.

Buffer Id
The user-defined name of the buffer definition. Required, cannot be blank.
Default: New-Buffer (if not specified).
Define Fields in a Buffer Panel

This panel is used to modify the fields of a buffer definition.

Panel Layout

SXT002F ---------------- 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-Buffer
Maximum Size ===> 0

<table>
<thead>
<tr>
<th>Sel</th>
<th>Type</th>
<th>Start</th>
<th>Len</th>
<th>Dec</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

Buffer Id  The name of the buffer definition.

Maximum Size  The maximum size of the buffer (data area) that is used for communication.

Default: 0. Valid range is 0 to 4194304. When 0, the size of the buffer is determined by the field definitions. When not 0, this overrides the determination from the fields. If specified, it must be greater than or equal to the field determination.

Sel  Modifies a previously-defined field in a buffer definition.

Valid selections: S(elect) D(elete) or R(epro).
**Type** The type of mainframe data expected by the invoked application for the field. Required, cannot be blank.

Valid types:

- **Str** - represents variable length CHARACTER data string. Always contain a null terminator at the end (as ALPHANUMERIC, NUMERIC and ALPHABETIC).
- **Text** - represents fixed length CHARACTER data string (as ALPHANUMERIC, NUMERIC and ALPHABETIC).
- **Pdec** - represents PACKED-DECIMAL or COMP-3 data format
- **Updec** - represents UNSIGNED PACKED-DECIMAL or COMP-3 data format
- **Zdec** - represents EXTERNAL DECIMAL data format (as USAGE or DISPLAY)
- **Uzdec** - represents UNSIGNED ZONED DECIMAL data format
- **Ubin** - represents UNSIGNED BINARY, COMP or COMP-5 data format
- **Bin** - represents SIGNED BINARY or COMP data format
- **Float** - represents Floating-Point data format (as COMP-1 and COMP-2)
- **Opa** - a block of unchanged characters (no ASCII / EBCDIC conversion)

**Start** The number of starting bytes that the field is offset in the buffer. The buffer offset is used to position the data at a given byte position.

**Len** The maximum size of the field represented in storage or memory. Required, must be greater than 0. Valid range is 0 to 4194304.

**Dec** 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.

**Name** The user-defined name for the field. Required, cannot be blank.
Buffer Field Details Panel

This panel is used to modify the details of a field in a buffer definition.

Panel Layout

SXTP02D ---------------------- Buffer Field Details ----------------------

Command ===> 

Field Name ===> string field

Field Type ===> STR  (STR, ...)
Buffer Start ===> 0
Buffer Length ===> 8
Decimal Places ===> 0  (0-6)
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 Descriptions

Field Name  The name of the field.

Field Type  The type of data expected by the invoked application for the field. Required, cannot be blank.

Valid types: Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float and Opa

For type definitions, see Type on page 32.

Buffer Start  The number of starting bytes that the field is offset in the buffer. The buffer offset is used to position the data at a given byte position.

Buffer Length  The expected length of the data for the field. If the data exceeds this length, and is not truncated as specified in the Trunc field of the Define Msg-Field or Buffer-Field Conversion panel, the transformation fails.
### Decimal Places
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.

### Pad Character
The character to use for padding. Keyboard-accessible characters can be used and are placed at the beginning or the end based on the justify configuration.
Default: space.

### or hex( )
The fill character defined as a hex value instead of an alpha value.

### Separate Sign
Indicates whether a byte is reserved for the sign or the position of the sign.
Valid selections: N(one) or Y(es). Default: N.

### Leading Sign
The field contains a leading sign (as opposed to a trailing sign).
Valid selections: N(one) or Y(es). Default: N.

### Justify Right
Indicates the positioning of the string. Padding characters can be used to fill in the data on either side as applicable.
Valid selections: Y(es) or N(o). Default: N.

### Initial Value
The value or characters used to fill in the field, if the data is not available in the buffer.

### Initial Value Length
The length of the Initial Value field when used.

#### Defining Initial Values
Setting initial values is only for mapped buffer fields, for example, fields that have corresponding message field names selected on a conversion rule.

When a message field is not present in the input message, the following applies:

- 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 HEX input field of the Buffer Field Details panel. Do not specify an Initial Value Length.

- If the BES application expects the buffer field to be filled with an array of characters, specify the Initial Value together with the Initial Value Length. When Initial Value Length is set to the value 99, Substation uses the first character of Initial Value to initialize the entire buffer field (the default value for Initial Value is a low value).
Conversion Rules using Buffer Definition Panel

This 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 details, see Buffer Definitions Panel on page 30.

Panel Layout

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

       Buffer Id ==> New-Buffer

       Used in these Conversion Rules:

*******************************************************************************

Field Descriptions

**Buffer Id**

The name of the buffer definition used in the conversion rule. Required, cannot be blank.

Default: New-Buffer (if not specified).

**Used in these Conversion Rules**

The name of the conversion rules that uses the buffer definition.
**Conversion Rules Panel**

This panel is used to create or modify a conversion rule.

**Panel Layout**

<table>
<thead>
<tr>
<th>Sel</th>
<th>Flds</th>
<th>Bufsz</th>
<th>Type</th>
<th>Conversion Id</th>
<th>Message Id</th>
<th>Buffer Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>M</td>
<td>CVR-I-ADMIN</td>
<td>MSG-IN-ADMIN</td>
<td>BUF-I0-ADMIN</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td></td>
<td>M</td>
<td>CVR-I-IMSCVS-01</td>
<td>MSG-IN-IMSCVS</td>
<td>BUF-I-IMSCVS-01</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td></td>
<td>M</td>
<td>CVR-I-TSRR-01</td>
<td>MSG-IO-RR-01</td>
<td>BUF-I0-BF01</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td></td>
<td>M</td>
<td>CVR-I-01</td>
<td>MSG-IN-01</td>
<td>BUF-I0-BF01</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td></td>
<td>M</td>
<td>CVR-I-02</td>
<td>MSG-IN-02</td>
<td>BUF-I0-BF02</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td>B</td>
<td>CVR-IO-BYTES</td>
<td>MSG-JMS-BYTES</td>
<td>BUF-BYTES</td>
</tr>
</tbody>
</table>

**Field Descriptions**

**Command**
The `ADD` command creates a new conversion rule. The `CAN` command cancels all modifications just done to the existing conversion rule.

Valid commands: `ADD` and `CAN`.

**Sel**
Selects an existing conversion rule to modify.

Valid selections: `S`elect, `M`essages, `D`elete, `R`epro, or `U`sedon.

If you select `M`, you enter the panel described in Define Message Fields to Conversion Rule Panel on page 41. If you select `U`, you enter the panel described in Recipes & Triggers using a Conversion Rule Panel on page 43.

**Flds**
The number of fields that have been mapped between messages and buffers.

**Max Bufsz**
The maximum size of the buffer (data area) that is used for communication. Optional, numeric only.

Default: 0.

**EMS Type**
The message type of a nested message.

Valid selections: `M`ap, `S`tream, `T`ext, and `B`ytes. Default: `M`.

**SZ**
Used only with EMS messages and when the message type is either BYTES or TEXT.
Valid selections: \texttt{M} and \texttt{B}. Default: \texttt{B}.

Determines how Substation ES calculates the size for data communication with the back-end system (CICS or IMS). If \texttt{M}, the payload (EMS message) size is used to determine the data communication area size; note that both the input and output data communication size is 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). If \texttt{B}, the data communication size is determined by the area and/or fields defined in the conversion rules buffer definition (the existing method prior to version 2.5).

**Conversion Id**

The user-defined name of the conversion rule. Required, cannot be blank.

Default: \texttt{New\_Rule} (if not specified).

**Message Id**

The name of the message definition used in the conversion rule.

**Buffer Id**

The name of the buffer definition used in the conversion rule.

### Select a Message Definition Panel

This panel is used to select a message definition for a new conversion rule.

**Panel Layout**

```
SXTP03M ----------------- Select a Message Definition -------- Row 1 to 1 of 1
Command ===>                                                   Scroll ==> CSR
Sel Flds   Message Id
-    ----   ----------------
0   New-Msg
********************************** Bottom of data **********************************
```

### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sel</td>
<td>Selects a message definition for a Msg-Field or Buffer-Field Conversion.</td>
</tr>
<tr>
<td></td>
<td>Valid selection: \texttt{S}(elect).</td>
</tr>
<tr>
<td>Flds</td>
<td>The number of fields in the message definition.</td>
</tr>
<tr>
<td>Message Id</td>
<td>The name of the message definition.</td>
</tr>
</tbody>
</table>
Select a Buffer Definition Panel

This panel is used to select a buffer definition for a new conversion rule.

Panel Layout

SXTP03B ----------------- Select a Buffer Definition --------- Row 1 to 1 of 1
Command ===>                                                   Scroll ===> CSR

Sel Flds  Buffer Id
-  ----  ----------------
  1  New-Buffer

****************************************************************************** Bottom of data *********************
Define Msg-Field or Buffer-Field Conversion Panel

This panel is used to map the fields of a message definition with the fields of a buffer definition for a conversion rule.

Panel Layout

```
SXTP03F -------- Define Msg-Field / Buffer-Field Conversion - Row 1 to 1 of 1
Command ===>                                                   Scroll ==> CSR
  Conversion Id   ===> New_Rule
  Using Message   ...: New-Msg
  Using Buffer    ...: New-Buffer
  Max Buffer Size ===> 0

Trunc Req’d BfFld Msg Field | Num   Buffer Field
  -    -   ----- ------------------------ | ---- -----------------------------
  Y    Y   0                              | 1    string field

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

Field Descriptions

- **Conversion Id**: The name of the conversion rule.
- **Using Message**: The name of the message definition.
- **Using Buffer**: The name of the buffer definition.
- **Max Buffer Size**: The maximum size of the buffer (data area) that is used for communication. Optional, numeric only.
  - Default: 0. When 0, the size of the buffer is determined by the field definitions. When not 0, this overrides the determination from the fields. If specified, it must be greater than or equal to the field determination.
- **Trunc**: Specifies 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 truncate 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.
  - Default: Y.
  - S—Specifies the field is used as a system field, and when used as such no mapping to buffer field is required.
**Reqd**  Specifies the field is required for a conversion. Required, cannot be blank.

Valid selections: Y(es) or N(o). Default: Y.

For an inbound process: If you specify Y, the transaction is not completed successfully unless a field with this name is encountered in the inbound TIBCO Rendezvous message. If you specify N, no specific check is made as to the availability of that message field in the TIBCO Rendezvous message.

For an outbound process: If this field is set to N, and the Data Type is numeric, and the contents of the field have not been initialized by the host-side application, the field is initialized by the Transformer and no editing errors occurs.

**BfFld**  The sequence number of the buffer field that is to be mapped to the message field.

**Msg Field**  The name of the field in the message definition.

**Num**  The sequence number of the field in the buffer definition.

**Buffer Field**  The name of the field in the buffer definition.
Define Message Fields to Conversion Rule Panel

This panel is used to define a nested message field (RVMSG) to a conversion rule.

You reach this panel by selecting M under Sel in the Conversion Rule panel, selecting a Conversion Rule that contains Nested Message Fields. For details, see Conversion Rules Panel on page 36.

Panel Layout

<table>
<thead>
<tr>
<th>SXTP03G</th>
<th>Define Message Fields to Conversion Rule -- Row 1 to 1 of 1</th>
<th>Scroll ==&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>===&gt;</td>
<td></td>
</tr>
<tr>
<td>Conversion Id</td>
<td>...: New_Rule</td>
<td></td>
</tr>
<tr>
<td>Using Message</td>
<td>...: New-Msg</td>
<td></td>
</tr>
<tr>
<td>Sel</td>
<td>Trunc</td>
<td>Reqd</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

**Conversion Id**
The name of the conversion rule to which the nested message field is defined.

**Using Message**
The name of the message definition containing the nested message field.

**Sel**
Selects a conversion rule to which the nested message field can be defined.

Valid selection: S(elect).

**Trunc**
Not applicable for nested messages; can be ignored.

**Reqd**
Specifies the field is required for a conversion. Required, cannot be blank.

Valid selections: Y or N. Default: Y.

For an inbound process: If you specify Y, the transaction is not completed successfully unless a field with this name is encountered in the inbound message. If you specify N, no specific check is made as to the availability of that message field in the message.

For an outbound process: If this field is set to N, and the Data Type is numeric, and the contents of the field have not been initialized by the host-side application, the field is initialized by the Transformer and no editing errors occur.
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CvRule</td>
<td>The sequence number of the conversion rule to which the nested message field is defined.</td>
</tr>
<tr>
<td>Msg Field</td>
<td>The name of the nested message field.</td>
</tr>
<tr>
<td>Num</td>
<td>The sequence number of the conversion rule.</td>
</tr>
<tr>
<td>Conversion Rule</td>
<td>The name of the conversion rule.</td>
</tr>
</tbody>
</table>
Recipes & Triggers using a Conversion Rule Panel

This panel displays the recipe and trigger definitions in which a conversion rule is used.

You access this panel by selecting U in the Sel field of the Conversion Rule panel, selecting a Conversion Rule that contains Nested Message Fields. For details, see Conversion Rules Panel on page 36.

Panel Layout

SXTP03U ---------- Recipes & Triggers using a Conversion Rule------------------
Command ===>                                                   Scroll ==> CSR

Conversion Rule ===> New_Rule

Used in these Recipes and Triggers:
****************************************************************************** Bottom of data ******************************************************************************

Field Descriptions

Conversion Rule  The name of the conversion rule used in the recipe and trigger definition.

Used in these Recipes and Triggers  The name of the recipe and trigger definitions that uses the conversion rule.
Define Processing System Resources Panel

This panel is used to create or modify a Substation ES Interface.

Panel Layout

SXTP04 ------------ Define Processing System Resources ----- Row 1 to 1 of 1
Command ===> Scroll ==> CSR
Type: C(ics) I(ms)
(Use ADD command to create a new Processing System)

Sel Type Identifier Interface Id.
- - ---------------------- ----------------------
  I New-BES ITIMS

Field Descriptions

**Command**

The **ADD** command creates a new Substation ES interface. The **CAN** command cancels all modifications just done to the existing Substation ES interface.

Valid commands: **ADD** and **CAN**

**Sel**

Selects an existing Substation ES interface to modify.

Valid selections: **S**elect, **D**elete or **R**epro.

**Type**

The type of interface.

Valid types: **A**(dmin), **C**(ics) and **I**(ms)

**Identifier**

The user-defined description of the Substation ES interface. Required, cannot be blank. Must be unique.

Default: **New-BES** (if not specified).

**Interface Id**

Name (up to eight characters) assigned to Substation ES interface to communicate with the transaction processing systems (such as CICS and IMS) running in the z/OS operating system environment. Required, cannot be blank.

This identification must match the interface Id that is configured through the **INTF-ID** parameter in the Interface System Initialization parameter (SIP) member. The SIP member is located in the `<USERHLQ>.CNTL` library.
Define Transport Entity Panel

This panel is used to create or modify a transport entity.

Panel Layout

SXTP07  ------------------------  Define Transport Entity  ------------------  Row 1 to 1 of 1
Command ====> Scroll ===> CSR

(Use ADD command to create a new Transport Definition)

Sel  Identifier   Service  Port  Daemon
-   -------------------  -----   -------- ----------------------------------------
  New-Transport  0        0

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

Field Descriptions

Command  The **ADD** command creates a new transport entity. The **CAN** command cancels all modifications just done to the existing transport entity.

Valid commands: **ADD** and **CAN**

Sel  Selects an existing transport entity to modify.

Valid selections: **S**(elect), **D**(elete) or **R**(epro).

If you select **S**, you enter the panel described in Transport Entity Extensions Panel on page 46.

Identifier  The user-defined name of the transport entity. Required, cannot be blank.

Default: **New-Transport** (if not specified).

Service  The value for the UDP service port number that defines the service group.

Default: 0.

Port  The TCP port for the transport entity that distinguishes both Substation ES and the role of the transport entity within it.

Default: 0.

Daemon  The host name or host IP address that Substation ES uses to find the RV daemon or EMS server and establish communication.

Default: blanks.
Transport Entity Extensions Panel

This panel is used to create or modify transport entity extensions.

Panel Layout

<table>
<thead>
<tr>
<th>Command</th>
<th>Id</th>
<th>Type</th>
<th>Startup</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New-Transport</td>
<td>B (E/R/B EMS, RV, Both)</td>
<td>Y (Y/N Start during Substation Initialization)</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rendezvous Values:

| Service | 0 |
| Network | 0 0 0 0 |
| Multicast | N |
| Multicast Address |  |

Substation EMS Values:

| User Id |  |
| Password |  |
| Alt. Conn | 0 0 0 0 |
| Alt. Port | 0 |

Field Descriptions

Id
The name of a transport entity.

Type
The type(s) of ESB to be defined.

Valid selections: E(MS), R(endezvous) and B(oth). Default: B.

Startup
Specifies whether to start this transport during Substation ES initialization.

Valid selections: Y(es) or N(o). Default: Y.

URL
For Rendezvous, the daemon URL. For EMS, the URL for the EMS server.

Port
For Rendezvous, the daemon port; for EMS, the port portion of the EMS server URL.

Default: 0.
**Service**  The value for the UDP service port number that defines the service group.  
Default: 0.

**Network**  Specifies the service group by selecting a local network by network IP address.  
Default: 0.0.0.0.

**Multicast**  Specifies the address is multicast enabled.  
Valid selections: Y(es) or N(o). Default: N.

**Multicast Address**  Specifies the multicast addresses (when the host computer has multiple network interfaces).

**User Id**  User Id if a special identification is to be used in the session. Maximum character length is 59.

**Password**  Password associated with the User Id if a special identification is used. Maximum character length is 59.

**Alt. Conn.**  The host IP address that Substation ES uses as an alternate connection (EMS supports an alternate connection; for example, a fault tolerance server pair).

**Alt. Port.**  The value for the alternate port when an alternate connection is used.

---

When defining fields related to TIBCO Rendezvous, the following conditions are enforced:

- The **Service** value is used if non-zero.
- The **Network** value is used if non-zero. 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 non-zero, a colon symbol and the port number are appended to the **URL** value.

When defining fields related to TIBCO EMS, the following conditions are enforced:

- The **URL** value is used as specified. If the value specified for **Port** is non-zero, a colon symbol and the port number are appended to the **URL** value.
Recipe List Panel

This panel is used to create or modify a recipe.

Panel Layout

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

(Use ADD command to create a new Recipe)

Sel  Resourc  MOI  Reply  Processor     Identification
     -------  ----  --------------  -------------------------------
        2     2       New-BES      New-Recipe

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

Field Descriptions

**Command**  The **ADD** command creates a new recipe. The **CAN** command cancels all modifications just done to the existing recipe.

Valid commands: **ADD** and **CAN**

**Sel**  Selects an existing recipe to modify.

Valid selections: **S**elect, **D**elete, **R**epro or **E**(MS).

If you select **E**, you enter the panel described in **EMS Recipe Details Extension Panel on page 58**.

**Resourc**  The name of the resource. For details, see the **Resource Name** field on the **Recipe Details Panel on page 53**.

**MOI**  The method of invocation used to initiate the configured transformation.

Valid selections:

- 2 **DPL**
- 3 **Program**
- 4 **Task/IMS Tran**
- 5 **TDQ Write**
- 6 **TSQ Write**
- 7 **Conversational Tran** (IMS only)
- 8 **Command**
9 DPL MRO
10 Container
11 Storage
12 DPL TSQ

**Reply**  
The type of data expected as output from the transaction processing application. For details, see the Reply Method field on the Recipe Details Panel on page 53.

**Processor**  
The description of the Substation ES interface specified in the Identifier field of the Define Processing System Resources Panel on page 44.

**Identification**  
The user-defined name of the recipe. Required, cannot be blank. Default: New-Recipe (if not specified).
Select Transport Panel (Recipe)

This panel is used to select a transport entity for a recipe.

Panel Layout

<table>
<thead>
<tr>
<th>SXTP05T</th>
<th>Select Transport</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Command ===&gt;</td>
<td>Scroll ==&gt; CSR</td>
</tr>
</tbody>
</table>

(Use S to select a Transport.)

<table>
<thead>
<tr>
<th>Sel</th>
<th>Service Port</th>
<th>Identifier</th>
<th>Daemon</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>New-Transport</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sel</td>
<td>Selects the transport entity for a recipe. Select one transport for each recipe. Valid selection: S(elect).</td>
</tr>
<tr>
<td>Service</td>
<td>The value for the UDP service port number that defines the service group.</td>
</tr>
<tr>
<td>Port</td>
<td>For Rendezvous, the daemon port; for EMS, the port portion of the EMS server URL.</td>
</tr>
<tr>
<td>Identifier</td>
<td>The name of a transport entity to be used in the recipe.</td>
</tr>
<tr>
<td>Daemon</td>
<td>The host name or host IP address used by Substation ES to find the Rendezvous daemon or EMS server and establish communication.</td>
</tr>
<tr>
<td>Network</td>
<td>The service group selected by network IP address or multicast addresses.</td>
</tr>
</tbody>
</table>
Select Processing System Resources Panel (Recipe)

This panel is used to select a processing system resource for a recipe.

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-BES          IIIMS

Field Descriptions

- **Sel**  Selects the processing system resource for a recipe.
  Valid selection: S(elect).

- **Type**  The type of interface.
  Valid types: A(dmin), C(ics) and I(ms)

- **Identifier**  The description of the Substation ES interface.

- **Interface Id.**  The name assigned to Substation ES interface to communicate with the transaction processing systems.
Select Conversion Rules Panel (Recipe)

This panel is used to select a conversion rule for a recipe.

Panel Layout

```
SXTP05C ------------------- Select Conversion Rules -------- Row 1 to 13 of 13
Command ===>
(I for input converter, O for output, B for both)

Max Sel Fldsz Bufsz Conversion Id    Message Id       Buffer Id
-     ---- ----- ---------------- ---------------- ----------------
 10    CVR-I-ADMIN      MSG-IN-ADMIN     BUF-IO-ADMIN
 13    CVR-I-01         MSG-IN-01        BUF-IO-BF01
 12    CVR-I-02         MSG-IN-02        BUF-IO-BF01
 20    CVR-IO-BYTE      MSG-JMS-BYTE     BUF-BYTES
  4    CVR-IO-STRING    MSG-IO-TEXT      BUF-STRING

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

Field Descriptions

- **Sel**: Selects the conversion rule for a recipe.
  - Valid selections: I(nput), O(utput) and B(oth)

- **Flds**: The number of fields in the conversion rule.

- **Max Bufsz**: The maximum buffer size to be used for the conversion rule.

- **Conversion Id**: The name of the conversion rule to be used in the recipe.

- **Message Id**: The name of the message definition used in the conversion rule.

- **Buffer Id**: The name of the buffer definition used in the conversion rule.
Recipe Details Panel

This panel is used to create or modify the details of a recipe.

Panel Layout

SXT05N---------------------------- Recipe Details ----------------------------

Command ===>

Recipe Identifier ===> New-Recipe
Subject ===> Publish Subject ===>
Publish Always ===> Y
Error Subject ===> Error Threshold ===> 0 (0 - 99999)
Dynamic Subject Leng ===> 0 Start ===> 0
Input Conversion (?) ===> Output Conversion(?) ===> New-Rule
New-BES
Back End System (?) ===> New-Transport
User Tran ===> N
Resource Name ===> Mirror Tran ===> New-Transport
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 ===> 2 0 None 2 Message
1 Status 3 Both Status and Message

Field Descriptions

Command  The EMS command accesses the panel described in EMS Recipe Details Extension Panel on page 58. The PROP command accesses the panel described in EMS Properties Panel on page 61. The SEL command accesses the panel described in EMS Selector Specification Panel on page 62.

Valid commands: EMS, PROP or SEL.

Recipe Identifier  The name of the recipe.

Subject  When configuring Substation ES to subscribe to data from a TIBCO messaging application, this field specifies the subject that Substation ES listens for in order to invoke a given transformation. Required, cannot be blank.

Publish Subject  The outbound subject on which a message is published by a transaction process. See the Publish Always field description for information on the rules regarding published messages.
**Publish Always**

Determines 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 above Publish Subject field (in EMS this is the outbound destination name), or both.

The reply Id in a message is set by the application issuing 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 Publish Always field.

Valid selections: \texttt{Y(es)} or \texttt{N(o)}. Default: \texttt{Y}.

- When set to \texttt{Y}, the response is sent as follows:
  - To the reply Id in the message – when the reply Id is specified and the Publish Subject or Destination is not specified.
  - To the Publish Subject or Destination – when the Publish Subject or Destination is specified and the reply Id is not specified.
  - To both the reply Id in the message and Publish Subject or Destination – when both are specified.

- When set to \texttt{N}, the response is sent as follows:
  - To ONLY the reply Id in the message – when the reply Id is specified, and the Publish Subject or Destination is or is not specified.
  - To the Publish Subject or Destination – when the Publish Subject or Destination is specified and the reply Id is not specified.
  - Not sent when both have not been specified.

**Error Subject**

The subject on which an error message is published when Transformer conversion or BES encounters errors.

When using EMS as your transport, you can now select the EMS message type of the error message. See the \texttt{ERROR-MSG-TYPE} keyword in \textit{TIBCO Substation ES Installation}.

**Error Threshold**

The number of errors allowed before the recipe is disabled.

**Error Percent**

The percentage of errors allowed before the recipe is disabled.

**Dynamic Subject Leng**

The outbound reply subject can be allocated dynamically within the buffer.

Specifies the number of bytes to be used for the Dynamic Subject length. Optional.

Default value is 0.

**Start**

The starting offset in the buffer to be used for the dynamic subject name.
The name of the conversion rule used for input conversion. Specifying ? displays a list of all conversion rules from which you can select. Required, cannot be blank.

The name of the conversion rule used for output conversion. Specifying ? displays a list of all conversion rules from which you can select.

The description of the Substation ES interface. Specifying ? displays a list of all interfaces from which you can select. Required, cannot be blank.

The name of the transport entity. Specifying ? displays a list of all transport entities from which you can select. Required, cannot be blank.

Specifies the destination of the transaction process is a user transaction. Valid selections: Y(es) or N(o). Default: N.

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.

If your MOI is TDQ Write or TSQ Write, this field specifies the name of the Transient Data Queue or Temporary Storage Queue. The length of the TDQ name can be 4 characters or less. The maximum length of the TSQ name is 8 characters.

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.

- Substation ES uses transaction SXEX by default that passes data via the COMMAREA to programs compiled with AMODE(ANY). This enables data that is passed to CICS programs to use CICS extended storage that is the desired method.

- Substation ES uses transaction SXBX by default that passes data via 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 16M storage line.

- If different security options for transactions are required, specifying a Mirror Transaction per invocation of the CICS resource allows flexibility without complicating RACF/ACF2 definitions.

The method used to initiate the transformation that is being configured. If 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. If configuring Substation ES to subscribe to data generated by a TIBCO messaging application, you can choose one or more of the other methods of invocation.
Valid selections:

2 **DPL** – transaction invoked using the DPL (Dynamic Program Link) method of invocation that allows a response or return data to be sent back to the calling TIBCO messaging application.

3 **Program** – transaction invoked using the XCTL (Transfer Control) mechanism of CICS. Only a status return is available with this option.

4 **Task/IMS Tran** – invokes a CICS task or transaction (4 character resource name) as opposed to a CICS program. Can be a status return when using this method. Alternatively, invokes an IMS transaction to be processed by the BES that allows a response or return data to be sent back to the calling TIBCO messaging application.

5 **TDQ Write** – transaction invoked by a write to the TDQ the transaction is *already* polling from. The length of the TDQ name can be 4 characters or less.

6 **TSQ Write** – transaction invoked by a write to TSQ. The maximum length of the TSQ name is 8 characters.

7 **Conversational Tran** – an IMS conversational transaction is invoked for processing.

8 **Command** – invokes a Substation ES admin task for processing operation commands, Substation ES Console commands, or IMS commands that form a TIBCO message.

9 **DPL MRO** – transaction invoked via CICS LINK with a CommArea. Useful for MRO routing.

10 **Container** – transaction invoked via 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 32K input and output.

11 **Storage** – transaction invoked via CICS LINK with a ComArea. The ComArea contains a pointer to a storage area and its length. Can be used for greater than 32K input and output.

12 **DPL TSQ** – transaction invoked via CICS LINK with a ComArea. The ComArea contains an 8-character TSQ Id that contains the input data and can also be used for output.
**MOI Usage**

<table>
<thead>
<tr>
<th>MOI</th>
<th>Requirements</th>
<th>Interface</th>
<th>Inbound Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Resource Name; 1 – 32000 bytes in size</td>
<td>EXCI</td>
<td>User Program</td>
</tr>
<tr>
<td>3, 4, 5, 6, 9</td>
<td>Resource Name; 1 – 32000 bytes in size</td>
<td>EXCI</td>
<td>DPL server (SXCOSRVD)</td>
</tr>
<tr>
<td>10, 11, 12</td>
<td>Resource Name; 1 – 4 MB in size</td>
<td>EXCI</td>
<td>DPL server for &gt; 32000 (SXCOSRVG)</td>
</tr>
<tr>
<td>2</td>
<td>User Tran = Y; Mirror Tran present; 1 – 32000 bytes in size</td>
<td>CSMI</td>
<td>User Transaction</td>
</tr>
</tbody>
</table>

**CICS**

**IMS**

<table>
<thead>
<tr>
<th>MOI</th>
<th>Requirements</th>
<th>Interface</th>
<th>Inbound Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>4, 7, 8</td>
<td>Resource Name; 1 – 4 MB in size</td>
<td>OMTA</td>
<td>User Transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XCF</td>
<td></td>
</tr>
</tbody>
</table>

**Trace Level**

The user trace debug level for an individual recipe. Should 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.

Valid range: 0 through 5. Default: 0 (if not specified).

**Reply Method**

Specifies 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 method of invocation, 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.

Valid selections:

0 None – No reply method for this recipe.

1 Status – A status message is sent to the requesting application via the reply subject name, and the subject specified in the Publish Subject field.

2 Message – Return data is to be transformed and published as a message to the TIBCO messaging application.

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.
EMS Recipe Details Extension Panel

This panel is used to create or modify Enterprise Message Service (EMS) values for a recipe.

You access this panel by selecting E in the Sel field in of the Recipe List panel. For details, see Recipe List Panel on page 48. Alternatively, this panel is accessed using the EMS command on the Recipe Details panel. For details, see Recipe Details Panel on page 53.

Panel Layout

```
SXTPO5E----------------- EMS Recipe Details Extension -------------------------
Command ===> 
Recipe Identifier ===> New-Recipe
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)
SMSG Mode ===> RA (RA Auto-Ack, RN No-Ack, GA All, GS SS Recv)
Delivery Mode ===> R (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 (L,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 Correlation ===> N (N/Y)
```

Field Descriptions

**Recipe Identifier**
The name of the recipe.

**Input Dest Type**
The input destination type.
Valid selections: T(topic) or Q(queue). Default: T.

**Input Dest Name**
The destination for which Substation ES listens to invoke a given transformation.

**Output Dest Type**
The output destination type.
Valid selections: T(topic) or Q(queue). Default: T.
**Output Dest Name**  
The outbound destination a message is published on for a given transaction process. See the **Publish Always** field description on the **Recipe Details** panel for information on the rules regarding published messages.

**Error Dest Type**  
The error destination type.
Valid selections: T(topic) or Q(queue). Default value is Q.

**Error Dest Name**  
The destination an error message is published on when Transformer conversion or BES encounter errors.

**Durable Subscriber**  
Connect the session as a durable subscriber.
Valid selections: N(no) or Y(es). Default: N.

**Durable Client Id**  
The client Id used in a durable connection. If not specified, the Recipe name is used.

**SMSL Mode**  
The SMSL mode for EMS.
Valid selections:
- RA Auto-Ack (Reliable Auto Acknowledge) – EMS client library automatically confirms the receipt of a message.
- RN No-Ack (Reliable No Acknowledge) – EMS client library does not send a receipt of the message that was received. The server destroys the message after it has been sent.
- GA All (Guaranteed All) – Substation ES calls the EMS client library and explicitly confirms receipt of a message only once after all Substation ES processes and executions controlled by Substation ES for a given message successfully complete.
- GS SS Recv (Guaranteed Substation Processing) – Substation ES calls the EMS client library and explicitly confirms receipt of a message when Substation ES processes complete execution 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.
Default: RA.

**Delivery Mode**  
The delivery mode for EMS.
Valid modes: P Persistent, N Non-Persistent and R Reliable. Default: P.

**Output Message properties**  
For additional information on Output Message properties, see the *TIBCO Enterprise Message Service User’s Guide.*
Priority | Set EMS message priority to the value specified in the **Priority Value** field.  
| Valid selections: N(o) or Y(es). Default: N.  
Priority Value | The EMS message priority value to be used.  
Expiration | Set EMS message expiration to the value specified in the **Expiration Value** field.  
| Valid selections: N(o) or Y(es). Default: N.  
Expiration Value | The EMS message expiration value to be used.  
| Valid range: 0 to 99999. Default: 0.  
Expiration Unit | The EMS message expiration unit to be used.  
| Valid selections: L = millisecond, S(ec)ond, M(inute), H(our) or D(ay). Default: S.  
Compression | Compress the EMS message for storage on the server.  
| Valid selections: N(o) or Y(es). Default: N.  
Preserve | If a message is removed from its queue by the EMS server for reasons other than being consumed, the server checks this message property. If set to Y, the message is placed on the server’s the undelivered message queue.  
| Valid selections: N(o) or Y(es). Default: N.  
Body Trace | Tracing for this destination generates trace messages that include the message body.  
| Valid selections: N(o) or Y(es). Default: N.  
Message Type | The type of EMS output message.  
| Valid selections: M(ap), S(ream), T(Ext), and B(ytes). Default: M.  
Copy Correlation | Copy the JMSCorrelationID from an inbound message and put it in the output message. If Y is selected, Substation checks if the JMSCorrelationID header field is defined under EMS Properties (for details, see EMS Properties Panel on page 61). If so, it replaces the original value with the value specified under EMS Properties; if not, it uses the original value from the inbound message.  
| If N is selected, Substation again checks if the JMSCorrelationID header field is defined under EMS Properties. If so, replaces the original value with the value specified under EMS Properties; if not, no action is performed.  
| Valid selections: N(o) or Y(es). Default: N.
EMS Properties Panel

This panel is used to specify EMS properties to be inserted in output messages. Up to fifteen properties can be specified. Note: the JMSCorrelationID header field can also be specified using this panel.

You access this panel by issuing the `PROP` command on either the Recipe Details panel or Trigger Details panel. For details, see Recipe Details Panel on page 53 or Trigger Details Panel on page 66.

Panel Layout

```
SXTP05P------------------------ EMS Properties ------------- Row 1 to 3 of 15
Command ===>

Buffer Id: New-Buffer

<table>
<thead>
<tr>
<th>Property Name</th>
<th>B/V Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property-1</td>
<td>B    IVP-BES-PGM-NAME</td>
</tr>
<tr>
<td>Property-2</td>
<td>V    This is the property field</td>
</tr>
</tbody>
</table>

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

Field Descriptions

- **Buffer Id**: If selecting property values from a buffer, this is the buffer name associated with an output conversion rule from which a selection can be made.

- **Property Name**: The name of the property.

- **B/V**
  - **B**: specifies the property value is selected from a buffer field.
  - **V**: specifies the property value is defined by the user in the Value field.

  Valid selections: **B(uffer)** or **V(alue)**.

- **Value**: If selected from the buffer, type the question mark (?) symbol and press the **ENTER** key. This displays a list of buffer field names in the buffer from which to choose. If user-defined, type the desired value.
EMS Selector Specification Panel

This panel is used to specify EMS message selector information.

You access this panel by issuing the SEL command on the Recipe Details panel. For details, see Recipe Details Panel on page 53.

Panel Layout

SXTP05S------------------ EMS Selector Specification --------------------------
Command ===> 
Selector ===> 
(End of Selector)

Field Descriptions

**Command**
Currently not used.

**Selector**
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.

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.

For information about message selector syntax, see the documentation for the Message class in the relevant TIBCO EMS API reference document.
Define Triggers Panel

This panel is used to create or modify a trigger.

Panel Layout

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

(Use ADD command to create a new Trigger)

Sel Output Identification                   At   Value
-   -      -------------------------------- ---- ------------------------------
  2      New-Trigger                      0    MATCH

Field Descriptions

**Command**

The *ADD* command creates a new trigger. The *CAN* command cancels all modifications just done to the existing trigger.

Valid commands: *ADD* and *CAN*

**Sel**

Selects an existing trigger to modify.

Valid selections: *S*(elect), *D*(elete), *R*(epro) or *E*(MS).

If you select *E*, you enter the panel described in EMS Trigger Details Extension Panel on page 69.

**Output**

The type of data expected as output from the transaction processing application.

Valid selections: *1* Status, *2* Message or *3* Both Status and Message.

For details, see the Reply Method field in the Recipe Details Panel on page 53.

**Identification**

The user-defined name for the trigger. Required, cannot be blank.

**At**

The starting offset in the buffer that contains the text to be used for the search criteria in a trigger process.

**Value**

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 Transport Panel (Triggers)

This panel is used to select a transport entity for a trigger.

Panel Layout

```
SXTP06T------------------------ Select Transport -------------- Row 1 to 1 of 1
Command ===>
(Use S to select a Transport)
Sel Service Port Identifier Daemon Network
- ---- ----- ---------------- --------------------------- ---------------
0 0 New-Transport 0 0 0 0
```

Field Descriptions

- **Sel**: Selects the transport entity for a trigger. Select one transport for each trigger. Valid selection: S(elect).
- **Service**: The value provided for the UDP service port number that defines the service group.
- **Port**: The TCP port given to a transport entity that distinguishes both Substation ES and the role of the transport entity within it.
- **Identifier**: The name of a transport entity to be used in the trigger.
- **Daemon**: The host name or host IP address used by Substation ES to find the RV daemon or EMS server and establish communication.
- **Network**: The service group selected by IP network number or multicast addresses.
Select Output Conversion Rule Panel (Triggers)

This panel is used to select a conversion rule for a trigger.

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_Rule         New-Msg          New-Buffer
******************************* Bottom of data *******************************
```

Field Descriptions

- **Sel** Selects the conversion rule for a trigger. Valid selections: $\text{S(elect)}$ or $\text{O(utput)}$.
- **Flds** The number of fields in the conversion rule.
- **Max Bufsz** The maximum buffer size to be used for the conversion rule.
- **Conversion Id** The name of the conversion rule to be used in the trigger.
- **Message Id** The name of the message definition used in the conversion rule.
- **Buffer Id** The name of the buffer definition used in the conversion rule.
**Trigger Details Panel**

This panel is used to create or modify the details of a trigger.

**Panel Layout**

| SXTP06N------------------------ Trigger Details ------------------------------ |
| Command ====> |
| Trigger Identifier ===> New-Trigger |
| Publish Subject ===> |
| Reply Subject ===> |
| Error Subject ===> |
| Error Threshold ===> 0 (0 - 99999) Error Percent ===> 0 (0 - 99) |
| Dynamic Subject Leng ===> 0 Start ===> 0 |
| Output Conversion(?) ===> New_Rule |
| Transport (?) ===> New-Transport |
| Trace Level ===> 0 0-5 |
| Output Method ===> 2 1 Status 2 Message 3 Both Status and Message |

**Field Descriptions**

| Command | The EMS command accesses the panel described in EMS Trigger Details Extension Panel on page 69. The PROP command accesses the panel described in EMS Properties Panel on page 61. Valid commands: EMS or PROP. |
| Trigger Identifier | The name of the trigger. |
| Publish Subject | The outbound subject on which a message is published for a given trigger transaction. Necessary when using the trigger method of invocation. Required, cannot be blank. |
| Reply Subject | The reply subject on which an external messaging application returns a response to Substation ES (for a CICS-initiated request reply process). Optional. |
| Error Subject | The subject on which an error message is published when Transformer conversion or BES encounters errors. |
| Error Threshold | The number of errors allowed before the trigger is disabled. |
### Error Percent
The percentage of errors allowed before the trigger is disabled.

### Dynamic Subject Length
The outbound subject name can be specified dynamically within the buffer. Specifies the number of bytes to be used for the Dynamic Subject name. Optional.
Default: 0.

### Start
The starting offset in the buffer to be used for the Dynamic Subject Name.

### Output Conversion (?)
The name of the conversion rule used for output conversion. Specifying ? displays a list of all conversion rules from which you can select. Required, cannot be blank.

### Transport (?)
The name of a transport entity. Specifying ? displays a list of all transport entities from which you can select. Required, cannot be blank.

### Trace Level
The user trace debug level for an individual trigger process. Should always be 0 or 1, unless TIBCO Software Inc. support personnel require the output. Note that a certain amount of overhead and many lines of output are produced when this value is greater than 1.
Valid range: 0 through 5. Default: 0 (if not specified).

### Output Method
Specifies whether data is expected to be output from the transaction processing application. If you are configuring Substation ES to subscribe to data generated by a TIBCO messaging application using the DPL method of invocation, you can specify whether return data and a status is expected back. 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.
Valid selections:

1. **Status** – a status message is sent to the awaiting TIBCO messaging application via the value specified in the Publish Subject field.

2. **Message** – outbound data is to be transformed and published as a message to the TIBCO messaging application.

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.
Compare Text Contain in Buffer

**Start**  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).

**Length**  The size of the search text in the buffer of a TDQ record. Specifying 0 means there is no matching on buffer content. When 0, the only way to select this trigger is via the "Service Name" parameter on a programs call to SXCQWRIT. For details on implementing HVT and SXCQWRIT, see *TIBCO Substation ES Operations and Administration*.

**Value**  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.
EMS Trigger Details Extension Panel

This panel is used to create or modify Enterprise Message Service (EMS) values for a trigger.

You access this panel by selecting **E** in the Sel field of the Define Triggers panel. For details, see Define Triggers Panel on page 63. Alternatively, this panel is accessed using the **EMS** command on the Trigger Details panel. For details, see Trigger Details Panel on page 66.

Panel Layout

| SXTP06E-------------- EMS Trigger Details Extension ------------------------ |
|----------------------|------------------------|
| Command ====>       | Command ====>          |
| Trigger Identifier  | New-Trigger            |
| Output Dest Type    | T Name:                |
| Reply Dest Type     | T Name:                |
| Error Dest Type     | Q Name:                |
| 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 (L,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(ys) |

Field Descriptions

**Trigger Identifier**  The name of the trigger.

**Output Dest Type**  The output destination type.

Valid selections: T(opic) or Queue (queue). Default: T.

**Output Dest Name**  The outbound destination on which a message is published for a given trigger transaction. Necessary when using the trigger method of invocation.

**Reply Dest Type**  The reply destination type.

Valid selections: T(opic) or Q(ueue). Default: T.
<table>
<thead>
<tr>
<th><strong>Reply Dest Name</strong></th>
<th>The reply destination on which an external messaging application returns a response to Substation ES (for a CICS-initiated request reply process).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error Dest Type</strong></td>
<td>The error destination type.</td>
</tr>
<tr>
<td><strong>Error Dest Name</strong></td>
<td>The destination an error message is published on when Transformer conversion or BES encounter errors.</td>
</tr>
<tr>
<td><strong>SMSL Mode</strong></td>
<td>The SMSL mode for EMS.</td>
</tr>
<tr>
<td><strong>Validity</strong></td>
<td>The delivery mode for EMS.</td>
</tr>
<tr>
<td><strong>Delivery Mode</strong></td>
<td>Valid modes: P Persistent, N Non-Persistent and R Reliable. Default: P.</td>
</tr>
</tbody>
</table>

**Output Message properties**

For additional information on Output Message properties, see the *TIBCO Enterprise Message Service User’s Guide*.

<table>
<thead>
<tr>
<th><strong>Priority</strong></th>
<th>Set EMS message priority to the value specified in the <strong>Priority Value</strong> field.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Validity</strong></td>
<td>Valid selections: N(no) or Y(es). Default: N.</td>
</tr>
<tr>
<td><strong>Priority Value</strong></td>
<td>The EMS message priority value to be used.</td>
</tr>
<tr>
<td><strong>Validity</strong></td>
<td>Valid range: 0 to 9. Default value: 4.</td>
</tr>
<tr>
<td><strong>Expiration</strong></td>
<td>Set EMS message expiration to the value specified in the <strong>Expiration Value</strong> field.</td>
</tr>
<tr>
<td><strong>Validity</strong></td>
<td>Valid selections: N(no) or Y(es). Default: N.</td>
</tr>
<tr>
<td><strong>Expiration Value</strong></td>
<td>The EMS message expiration value to be used.</td>
</tr>
<tr>
<td><strong>Validity</strong></td>
<td>Valid range: 0 to 99999. Default: 0.</td>
</tr>
</tbody>
</table>
**Expiration Unit**  
The EMS message expiration unit to be used.  
Valid selections: **L** = millisecond, **S**(econd), **M**(inute), **H**(our) or **D**(ay). Default: **S**.

**Compression**  
Compress the EMS message for storage on the server.  
Valid selections: **N**(o) or **Y**(es). Default: **N**.

**Preserve**  
If a message is removed from its queue by the EMS server for reasons other than being consumed, the server checks this message property. If set to **Y**, the message is placed on the server’s the undelivered message queue.  
Valid selections: **N**(o) or **Y**(es). Default: **N**.

**Body Trace**  
Tracing for this destination generates trace messages that include the message body.  
Valid selections: **N**(o) or **Y**(es). Default: **N**.

**Message Type**  
The type of EMS output message.  
Valid selections: **M**ap, **S**(team), **T**(ext), and **B**(ytes). Default: **M**.
Group Definitions Panel

This panel is used to create or modify a group definition.

Panel Layout

SXTP08------------------------ Group Definitions ------------- Row 1 to 1 of 1
Command ===>

(Use ADD command to create a new group)

Sel Group Id
- ----------------
  New_Group

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

Field Descriptions

Command  The ADD command creates a group definition. The CAN command cancels all
          modifications just done to the existing group definition.

          Valid commands: ADD and CAN

Sel       Selects an existing group definition to modify.

          Valid selections: S(elect), D(lete) or R(epro).

Group Id  The user-defined name of the group definition. Required, cannot be blank.

          Default: New_Group (if not specified).
Select Triggers and Recipes for Group Panel

This panel is used to select the recipes and triggers to be included in a group.

Panel Layout

SXTPO8G---------- Select Triggers and Recipes for Group --- Row 1 to 2 of 2
Command ====>                                                   Scroll ==> CSR

   Group Name ==> New_Group

   Use S to select each Trigger and/or Recipe into the Group.
   Use D to delete a selection.

Sel Type Identifier
- ------ --------------
Recipe New-Recipe
Trigger New-Trigger

Field Descriptions

Group Name  The user-defined name of the group definition.

Sel  Selects each trigger and recipe for the group.
    Valid selections: S(elect) or D(eselect)

Type  Trigger or recipe.

Identifier  The name of the trigger or recipe.
Display Configuration File Detail Panel

This panel is used to select the Summary of Definition for a recipe, trigger, conversion or group definition.

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 Descriptions

Command
The desired Summary of Definition.
Valid values: 1 Recipe, 2 Trigger, 3 Conversion and 4 Group Definition.

Identification (?)
The name of the recipe, trigger, conversion or group definition. Entering ? displays a list of definitions of the type specified under Command.

Display in hex
Displays the Summary of Definition in hexadecimal format.
Valid selections: Y(es) or N(o). Default: N.
Select Desired Entry Panel

This panel is used to select an existing recipe, trigger, conversion rule or group definition for detailed display.

Panel Layout

SXTP09S-------------- Select Desired Entry ----------- Row 1 to 1 of 1
Command ===>                       Scroll ===> CSR

Use S to select the proper Entry

Sel    Identifier
-       --------------------------------
New-Recipe
**************************************** Bottom of data ****************************************

Field Descriptions

Sel    Selects a specific entry from a list of recipes, triggers, conversion rules or group definitions.
        Valid selection: S.

Identifier    The name of the recipe, trigger, conversion rule or group definition entry.
Summary of Definition Panel (Recipe)

This panel provides a detailed display of a recipe.

Panel Layout

<table>
<thead>
<tr>
<th>Recipe name   : Substation-Admin-IMS</th>
<th>Command ==&gt;</th>
<th>Scroll ==&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject       : tibss.admin.sxi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response      : , Always: Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reply Sub     :</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Err Subj      : tibss.error.ivp.out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error Thresholds: Count: 0, Percent: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transcode     : , MOI:8, Trace: 0, Reply: 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dynamic Subject Leng:0, Start: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expire: 0, Units: S, Priority: 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Durable Subscriber: N, ClientId:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMSL Mode: RA, Delivery Mode: P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Back End System: IVP Intf - IMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: I, Target: IIIMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport id: SS IVP Default, Daemon: , Network: 127.0.0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service: 7555, Port: 0, Multicast: N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Userid: TIBUSER, Password: ******</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alt Server: 0.0.0.0, Port: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Conversion: SXI-CMD-I-CR01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max buffer size: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiving Buffer max: 0, id: SXI-G-BUF01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Message id: SXS-ADM-I-MSG01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From RV id: 0, name: SXS-COMMAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>To offset: 0000, length: 2048, name: COMMAND-TXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In decimal: 0, Out decimal: 0, Buf-type: TEXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Truncate: Y, Required: Y, Lead-sign: N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sep-sign: N, Right-just: N, Pad: 40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Init len: 0, Init value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From RV id: 0, name: tibss-userid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Truncate: S, Required: N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>From RV id: 0, name: tibss-password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Truncate: S, Required: N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Conversion: SXI-CMD-O-CR01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max buffer size: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sending Buffer max: 0, id: SXI-G-BUF01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Message id: SXS-ADM-O-MSG01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recipe name</strong></td>
<td>The name of the recipe.</td>
</tr>
<tr>
<td><strong>Subject</strong></td>
<td>The subject Substation ES listens for to invoke a given transformation.</td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td>The subject on which a message is published for a given transaction.</td>
</tr>
<tr>
<td><strong>Always</strong></td>
<td>Determines where the response from a transaction process is sent. For details, see the Publish Always field of the Recipe Details Panel on page 53.</td>
</tr>
<tr>
<td><strong>Reply Subj</strong></td>
<td>Subject that a status message is published on for the requesting application.</td>
</tr>
<tr>
<td><strong>Err Subj</strong></td>
<td>Subject that an error message is published on when Transformer conversion or BES encounters errors.</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>The number of errors allowed before the recipe is disabled.</td>
</tr>
<tr>
<td><strong>Percent</strong></td>
<td>The percentage of errors allowed before the recipe is disabled.</td>
</tr>
<tr>
<td><strong>Transcode</strong></td>
<td>Name of the resource.</td>
</tr>
<tr>
<td><strong>MOI</strong></td>
<td>Method of invocation used to initiate the configured transformation. For details, see the Method of Invocation field of the Recipe Details Panel on page 53.</td>
</tr>
<tr>
<td><strong>Trace</strong></td>
<td>The user trace debug level.</td>
</tr>
<tr>
<td><strong>Reply</strong></td>
<td>The type of data expected as output from the transaction processing application. For details, see the Reply Method field of the Recipe Details Panel on page 53.</td>
</tr>
<tr>
<td><strong>Dynamic Subject Leng</strong></td>
<td>The number of bytes to be used for the Dynamic Subject name.</td>
</tr>
<tr>
<td><strong>Start</strong></td>
<td>The starting offset in the buffer to be used for the dynamic subject name.</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>The destination of the subject Substation ES listens for to invoke a given transformation: T(opic) or Q(ueue).</td>
</tr>
</tbody>
</table>
### Output
The destination of an outbound subject a message is published on for a given transaction process: T(opic) or Q(ueue).

### Error
The subject destination an error message is published on when Transformer conversion or BES encounters errors: T(opic) or Q(ueue).

### Prty
The priority has been set for this message: N(o) or Y(es).

### Expir
The expiration has been set for this message: N(o) or Y(es).

### Compr
Compression has been set for this message: N(o) or Y(es).

### Presv
The message is to be preserved on the server: N(o) or Y(es).

### Trace
Trace generates messages that include the message body: N(o) or Y(es).

### Expire
The EMS expiration value to be used for message.

### Units
The EMS legal unit value used for the message.

### Priority
The EMS priority value used for the message.

### Durable Subscriber
Connect the session as a durable subscriber: N(o) or Y(es).

### ClientId
The client Id used in a durable connection.

### SMSL Mode
The SMSL mode used for this message by Substation ES.

### Delivery Mode
The delivery mode used for this message by Substation ES.

### Back End System
The description of the Substation ES interface.

### Type
The type of interface: A(dmin), C(ICS) and I(MS).

### Target
The identification of the interface that Substation ES uses to communicate with the transaction processing application. Must match the interface ID that is configured through the INTF-ID parameter in the system initialization member.

### Transport id
The name of the transport entity.

### Daemon
The host name or host IP address used by Substation ES to find the RV daemon or EMS server and establish communication.
**Network**
The service group selected by network IP network number or multicast addresses.

**Service**
The value provided for the UDP service port number that defines the service group.

**Port**
The TCP port given to a transport entity that distinguishes both Substation ES and the role of the transport entity within it.

**Multicast**
The address is multicast enabled: \( N(o) \) or \( Y(es) \).

**Userid**
User Id if a special identification is used in the session.

**Password**
Password associated with the User Id if a special identification is used.

**Alt Server**
The host IP address that Substation ES uses as an alternate connection (EMS supports an alternate connection).

**Port**
The value for the alternate port when an alternate connection is used.

**Input Conversion**
The name of the conversion rule used for the input conversion.

**Max buffer size**
The maximum input buffer size defined for the conversion rule.

**Receiving Buffer**
The maximum size of the inbound buffer.

**id**
The name of the buffer associated with the transformation used for the receiving buffer.

**Input Message id**
The name of the message.

**From RV id**
A message field identifier. Zero (0) is a special value that signifies no message field identifier.

**name**
The user-defined field name in the message definition associated with the transformation from which data is to be extracted.

**To offset**
The number of bytes for the field that the starting offset in the inbound buffer. The buffer offset is used to position the data at a given byte position.

**length**
The length of the data type field in the buffer definition associated with the transformation.
name  The user-defined name for the field in the buffer definition associated with the transformation.

In decimal  The implied number of digits after the decimal point in the input. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.

Out decimal  The implied number of digits after the decimal point in the output. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.

Buf-type  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 Type on page 32.

Truncate  Whether data that extends beyond the specified buffer length allocated for a message field was truncated. By default, the truncate option is set to Yes.

s-The field is used as a system field. When used as a system field, no mapping to buffer field is required.

Required  Whether the field is required for a Conversion. Default value is Y.

If Y is specified, the transaction is not completed successfully unless a field with this name is encountered in the inbound TIBCO Rendezvous message. If N is specified, no specific check is made as to the availability of that message field in the TIBCO Rendezvous message.

Lead-sign  The field contains a leading sign (as opposed to a trailing sign).

Sep-sign  Whether a byte is reserved for the sign. Default is N. Valid selections are: N(one), L(eading) and T(railing).

Right-just  The positioning of the string. Padding characters can be used to fill in the data on either side as applicable. Default is NO and strings are left justified.

Pad  The character to use for padding purposes. Keyboard accessible characters can be used as a pad and are placed either at the beginning, at the end or in both positions depending on the configuration. The default pad character is a space.

Init len  The length of the Init Value field when used.

Init value  Value or characters to be used, to fill in the field, if the data is not available in the buffer.

Output Conversion  The user-defined name for the output conversion associated with the transformation.
Max buffer size | The maximum outbound buffer size of all fields defined in the Msg-Field or Buffer-Field conversion rule.

Sending Buffer max | The maximum size of the outbound buffer for this recipe process to be sent.

id | The user-defined name of the buffer definition associated with the transformation used for the sending buffer.

Output Message id | The name of the message definition associated with the transformation that was entered on the Message Definitions panel.

From offset | The number of bytes for the field that the starting offset in the outbound buffer. The buffer offset is used to position the data at a given byte position.

length | The length of the data type field in the buffer definition associated with the transformation.

name | The user-defined name for the field in the buffer definition associated with the transformation.

Buffer decimal | The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.

RV Msg decimal | The implied number of digits after the decimal point in the message field. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.

Buf-type | 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 a definition of each buffer type, refer to Type on page 32.

To RV id | A message field identifier. Zero (0) is a special value that signifies no field identifier.

Type | 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, IPA

For definitions of each message type, refer to Type on page 27.

Reqd | Whether the field is required for a Conversion. Default value is Y.
If this field is set to \texttt{N}, and the Data Type is numeric, and the contents of the field have not been initialized by the host-side application, the Transformer initializes the field and no editing errors occurs.

\textbf{name} \hspace{1em} The user-defined name of the field in the message definition associated with the transformation that was entered on the Define Fields in a message panel.
Summary of Definition Panel (Trigger)

This panel provides a detailed display of a trigger definition.

Panel Layout

SXTP09D--------------------- Summary of Definition --------- Row 1 to 38 of 38
Command ===>                                                   Scroll ==> CSR

| Trigger name  : SXC-RR-REQUEST
|   Reply: 2, Trace: 2, Response : tibss.ivp.20.request
|   Reply Sub:
|   Err Subj : tibss.error.ivp.out
|   Error Thresholds: Count: 0, Percent: 0
|   Dynamic Subject Leng:0, Start: 0
|   Match at: 0, Length: 6, Value:SXCIRR<
| EMS Extensions:
|   Destination Types: Output: T, Reply: T, Error: Q
|   Expire: 0, Units: S, Priority: 4
|   SMSL Mode: RN, Delivery Mode: P
| Transport id: SS IVP Default, Daemon: , Network: 127.0.0.1
|   Service: 7555, Port: 0, Multicast: N
|   Userid: , Password: Null
| Alt Server: 0.0.0.0, Port: 0
| Output Conversion: SXC-RR-O-CR01-T
|   Max buffer size: 0
|   Sending Buffer max: 0, id: SXG-IO-BF01
|   Output Message id: SXC-IO-RR-MS01
| From offset: 0000, length: 008, name: IVP-ID
|   Buffer decimal: 0, RV Msg decimal: 0, Buf-type: TEXT
|   To RV id: 0, Type: STR, Reqd: Y, name: IVP-ID
| From offset: 0008, length: 056, name: IVP-DESCR
|   Buffer decimal: 0, RV Msg decimal: 0, Buf-type: TEXT
|   To RV id: 0, Type: STR, Reqd: Y, name: IVP-DESCR
| From offset: 0176, length: 008, name: IVP-START-LILSECS
|   Buffer decimal: 0, RV Msg decimal: 0, Buf-type: FLOAT
|   To RV id: 0, Type: F64, Reqd: Y, name: IVP-START-LILSECS
| From offset: 0236, length: 004, name: IVP-BTCH-SEQ-NO
|   Buffer decimal: 0, RV Msg decimal: 0, Buf-type: BIN
|   To RV id: 0, Type: I32, Reqd: N, name: IVP-BTCH-SEQ-NO

******************************************************************************

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

TIBCO Substation ES Configuration and Resources
Field Descriptions

**Trigger name**  
The user-defined trigger name associated with a particular transformer configuration.

**Reply**  
The data expected to be output from the transaction processing application. For further clarification, see *Output Method* on the Trigger Details Panel on page 66.

**Trace**  
The user trace debug level assigned on the Trigger Details panel. The valid range is 0 through 5.

**Response**  
A subject that a message is published on for a given trigger transaction.

**Reply Sub**  
The reply subject or destination message property for the published message.

**Err Subj**  
A subject that an error message is published on when Transformer conversion or BES encounters errors.

**Count**  
The number of errors allowed before the recipe is disabled.

**Percent**  
The percentage of errors allowed before the recipe is disabled.

**Dynamic Subject Leng**  
Optionally, the outbound subject name can be specified dynamically within the buffer. Specifies the number of bytes to be used for the Dynamic Subject name. The default is 0.

**Start**  
The starting offset in the buffer to be used for the dynamic subject name.

**Match at**  
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 what kind of transformation to perform.

**Length**  
The size of the search string in the buffer of a TDQ.

**Value**  
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.

**Output**  
The output destination type: T(opic) or Q(ueue).

**Reply**  
The reply destination type: T(opic) or Q(ueue).

**Error**  
Specifies the subject destination an error message is published on when Transformer conversion or BES encounters errors: T(opic) or Q(ueue).

**Prty**  
Specifies whether the priority has been set for this message.
<table>
<thead>
<tr>
<th>Expir</th>
<th>Specifies whether the expiration has been set for this message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compr</td>
<td>Specifies whether compression has been set for this message.</td>
</tr>
<tr>
<td>Presv</td>
<td>Specifies whether the message is to be preserved on the server.</td>
</tr>
<tr>
<td>Trace</td>
<td>Specifies whether trace generates messages that include the message body.</td>
</tr>
<tr>
<td>Expire</td>
<td>Specifies the EMS expiration value to be used for message.</td>
</tr>
<tr>
<td>Units</td>
<td>Specifies the EMS legal unit value used for the message.</td>
</tr>
<tr>
<td>Priority</td>
<td>Specifies the EMS priority value used for the message.</td>
</tr>
<tr>
<td>SMSL Mode</td>
<td>Specifies the Substation Messaging Services Levels mode used for this message by Substation ES.</td>
</tr>
<tr>
<td>Delivery Mode</td>
<td>Specifies the delivery mode used for this message by Substation ES.</td>
</tr>
<tr>
<td>Transport id</td>
<td>The user-defined name of a transport definition for a particular transformer configuration entered on the Define Transport Entity panel.</td>
</tr>
<tr>
<td>Daemon</td>
<td>The host name or host IP address used by Substation ES to find the RV daemon or EMS server and establish communication.</td>
</tr>
<tr>
<td>Network</td>
<td>The service group selected by IP network number or multicast addresses.</td>
</tr>
<tr>
<td>Service</td>
<td>The value provided for the UDP service port number that defines the service group.</td>
</tr>
<tr>
<td>Port</td>
<td>The TCP port given to a Transport that distinguishes both Substation ES and the role of the transport within it.</td>
</tr>
<tr>
<td>Multicast</td>
<td>Specifies whether the address is multicast enabled.</td>
</tr>
<tr>
<td>Userid</td>
<td>User-specified Id if a special identification is used in the session.</td>
</tr>
<tr>
<td>Password</td>
<td>Password associated with the User Id if a special identification is used in the session.</td>
</tr>
<tr>
<td>Alt Server</td>
<td>The value for an alternate connection (EMS supports an alternate connection).</td>
</tr>
<tr>
<td>Port</td>
<td>The value for the alternate port when an alternate connection is used.</td>
</tr>
<tr>
<td>Output Conversion</td>
<td>The user-defined name for the output conversion associated with the transformation.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Max buffer size</td>
<td>The maximum outbound buffer size defined for the Msg-Field or Buffer-Field conversion rule.</td>
</tr>
<tr>
<td>Sending Buffer max</td>
<td>The maximum size of the outbound buffer for this trigger process to be sent.</td>
</tr>
<tr>
<td>id</td>
<td>The user-defined name of the buffer definition associated with the transformation used for the sending buffer.</td>
</tr>
<tr>
<td>Output Message id</td>
<td>The name of the message definition associated with the transformation that was entered on the Message Definitions panel.</td>
</tr>
<tr>
<td>From offset</td>
<td>The number of bytes by which the response message is offset in the buffer.</td>
</tr>
<tr>
<td>length</td>
<td>The length of the data type field in the buffer definition associated with the transformation.</td>
</tr>
<tr>
<td>name</td>
<td>The user-defined name for the field in the buffer definition associated with the transformation.</td>
</tr>
<tr>
<td>Buffer decimal</td>
<td>The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td>RV Msg decimal</td>
<td>The implied number of digits after the decimal point in the message field. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td>Buf-type</td>
<td>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 a definition of each buffer type, refer to Type on page 32.</td>
</tr>
<tr>
<td>To RV id</td>
<td>A message field identifier. Zero (0) is a special value that signifies no field identifier.</td>
</tr>
<tr>
<td>Type</td>
<td>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, IPA. For definitions of each message type, refer to Type on page 27.</td>
</tr>
<tr>
<td>Req</td>
<td>Whether the field is required for a Conversion. Default value is Y.</td>
</tr>
</tbody>
</table>
If this field is set to N, and the Data Type is numeric, and the contents of the field have not been initialized by the host-side application, the Transformer initializes the field and no editing errors occurs.

**name**  
The user-defined field name in the message definition associated with the transformation from which data is to be extracted.

The display **No buffer field** indicates this is a system field.
Summary of Definition Panel (Conversion)

This panel provides a detailed display of a conversion rule definition.

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 RV Message id: SXC-IO-RR-MS01                            |
| RV Field id: 0, type: OPA, name: tibss-envelope                 |
| Truncate: S, Required: Y                                         |
| RV Field id: 0, type: STR, name: IVP-ID                           |
| Buffer offset: 0000, length: 008, name: IVP-ID                  |
| RV Field 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:                                         |
| RV Field id: 0, type: STR, name: IVP-DESCR                       |
| Buffer offset: 0008, length: 056, name: IVP-DESCR               |
| RV Field 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:                                         |
| RV Field id: 0, type: STR, name: IVP-RESULT                      |
| Buffer offset: 0064, length: 080, name: IVP-RESULT              |
| RV Field 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:                                         |
| RV Field id: 0, type: STR, name: IVP-BES-NAME                    |
| Buffer offset: 0144, length: 008, name: IVP-BES-NAME            |
| RV Field 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:                                         |
| RV Field id: 0, type: STR, name: IVP-BES-TRAN-ID                 |
| Buffer offset: 0152, length: 008, name: IVP-BES-TRAN-ID         |
| RV Field 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:                                         |
| RV Field id: 0, type: STR, name: IVP-BES-NAME-NAME               |
| Buffer offset: 0144, length: 008, name: IVP-BES-NAME-NAME       |
| RV Field 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:                                         |
Field Descriptions

**Conversion Identification**

The user-defined name of the conversion definition associated with the transformation.

**Max buffer size**

The maximum buffer size defined for the Msg-Field or Buffer-Field conversion rule.

**Related buffer max**

The maximum size of the buffer definition associated with the transformation that was entered on the Define Fields in a Buffer panel.
<table>
<thead>
<tr>
<th>id</th>
<th>The name of the buffer definition associated with the transformation that was entered on the Buffer Definitions panel.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related RV Message id</td>
<td>The name of the message definition associated with the transformation that was entered on the Message Definitions panel.</td>
</tr>
<tr>
<td>RV Field id</td>
<td>A message field identifier. Zero (0) is a special value that signifies no field identifier.</td>
</tr>
<tr>
<td>type</td>
<td>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, IPA</td>
</tr>
<tr>
<td></td>
<td>For definitions of each message type, refer to Type on page 27.</td>
</tr>
<tr>
<td>name</td>
<td>The user-defined field name in the message definition associated with the transformation from which data is to be extracted.</td>
</tr>
<tr>
<td>Buffer offset</td>
<td>The number of bytes for the field that the starting offset in the buffer. The buffer offset is used to position the data at a given byte position.</td>
</tr>
<tr>
<td>length</td>
<td>The length of the data type field in the buffer definition associated with the transformation.</td>
</tr>
<tr>
<td>name</td>
<td>The user-defined field name in the message definition associated with the transformation from which data is to be extracted.</td>
</tr>
<tr>
<td>RV Field decimal</td>
<td>The implied number of digits after the decimal point in the message field. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td>Buffer decimal</td>
<td>The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td>Buf-type</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>For a definition of each buffer type, refer to Type on page 32.</td>
</tr>
<tr>
<td>Truncate</td>
<td>Whether data that extends beyond the specified buffer length allocated for a message field was truncated. By default, the truncate option is set to Yes.</td>
</tr>
<tr>
<td>s</td>
<td>The field is used as a system field. When used as a system field, nomapping to buffer field is required.</td>
</tr>
</tbody>
</table>
**Required**
Specifies whether the field is required for a Conversion. The valid selections are Y(es) or N(o). Default value is Y.

For an inbound process: If you specify Y, the transaction is not completed successfully unless a field with this name is encountered in the inbound TIBCO Rendezvous message. If you specify N, no specific check is made as to the availability of that message field in the TIBCO Rendezvous message.

For an outbound process: If this field is set to N, and the Data Type is numeric, and the contents of the field have not been initialized by the host-side application, the field is initialized by the Transformer and no editing errors occurs.

**Lead-sign**
The field contains a leading sign (as opposed to a trailing sign).

**Sep-sign**
Whether a byte is reserved for the sign. Default is N. Valid selections are: N(one), L(eading) and T(railing).

**Right-just**
The positioning of the string. Padding characters can be used to fill in the data on either side as applicable. Default is NO and strings are left justified.

**Pad**
The character to be used for padding purposes. Keyboard accessible characters can be used as a pad and are placed either at the beginning or the end, depending on the justify configuration. The default pad character is a space.

**Init len**
The length of the Init Value field when used.

**Init len**
The length of the Init Value field when used.

**Init value**
The value or characters to be used, to fill in the field, if the data is not available in the buffer.
Summary of Definition Panel (Group)

This 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: SXC-RR-REQUEST
******************************* Bottom of data *******************************

Field Descriptions

- **Group name**  The name of the group definition.
- **Recipe id** The name of the recipe identifier associated with the group.
- **Trigger id** The name of the trigger identifier associated with the group.
Copybook Converter Panel

This panel is used to select and convert a COBOL copybook member from a partitioned data set into a buffer definition within the configuration file. Optionally, a Message and a Conversion Rule can be created 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 Descriptions

**Copy Member**  
The name of the copybook member to be selected from a partitioned data set (PDS). Required, cannot be blank.

**Buffer Identifier**  
The sixteen character name to be used for the buffer identifier. If "*" is entered, the Copy Member name is used. Optional.

**Message Identifier**  
The sixteen character name to be used for the message identifier. If "*" is entered, the Copy Member name is used. Optional.

**Convert Rule Id**  
The sixteen character name to be used for the convert rule identifier. If "*" is entered, the Copy Member name is used. Optional.
A convert rule identifier can only be specified when both a buffer identifier and RV message identifier exist.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library containing Copy Member</td>
<td>The primary partitioned data set (PDS) that contains the selected copybook member. Required, cannot be blank.</td>
</tr>
<tr>
<td>Additional libraries for Copy</td>
<td>The DSNs of the partitioned data set that can contain other copybooks that are included within the copybook member.</td>
</tr>
<tr>
<td>Parser Debug level</td>
<td>The parser debug level. This should always be left to the default value zero (0), unless TIBCO Software Inc. support personnel request otherwise.</td>
</tr>
</tbody>
</table>
Log Viewer Panel

This panel is used to select the recorded log and trace information written to a pre-allocated disk file.

Panel Layout

SXLP001-------------------------- 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)

Enter END command to terminate.

Field Descriptions

**ISPF Log**
Specify a log to view by either providing values for Project, Group and Type, or specifying a fully qualified cluster name in the field, Other VSAM Cluster Name. The value for field Other VSAM Cluster Name is used when non-blank.

**Display Format**
Format 1 is one recorded entry per line (truncated when necessary) Format 2 is a two-line display where the description is on the second line.

**Direction**
Forward or Backward. Backward starts with the most recently recorded entries and reads backward for the specified number of records, thus showing the most current activity.

**Active Log?**
If Y, the log is treated as active (currently being updated by Substation ES). When active, the index record is re-read for each screen refresh to obtain the most current records.
<table>
<thead>
<tr>
<th><strong>Number to Read</strong></th>
<th>The number of recorded entries to display. If not specified, a default value of 200 is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VSAM Debug</strong></td>
<td>Defines debugging levels for the VSAM access routines. Refrain from changing unless requested by TIBCO authorized personnel. Default is zero.</td>
</tr>
<tr>
<td><strong>Pgm Debug</strong></td>
<td>Defines debugging levels for the ISPF interface program. Refrain from changing unless requested by TIBCO authorized personnel. Default is zero.</td>
</tr>
</tbody>
</table>
Log Entries Panel

This panel is used to select an individual log or trace record written to the log or trace disk file.

Panel Layout

<table>
<thead>
<tr>
<th>Sel</th>
<th>Time</th>
<th>Type</th>
<th>Msg Id</th>
<th>RC</th>
<th>Rsn</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11:09:54.0919</td>
<td>154</td>
<td>SXG1600I</td>
<td>0</td>
<td>0</td>
<td>Log Agent Starting - Logging to TIBLOG</td>
</tr>
<tr>
<td></td>
<td>11:09:54.1126</td>
<td>154</td>
<td>SXG1800I</td>
<td>0</td>
<td>0</td>
<td>Parameter Log Agent Starting - Logging</td>
</tr>
<tr>
<td></td>
<td>11:09:54.4368</td>
<td>154</td>
<td>SXS1000I</td>
<td>0</td>
<td>0</td>
<td>Starting ~ TIBCO Substation (ES) for 0</td>
</tr>
<tr>
<td></td>
<td>11:09:54.4596</td>
<td>154</td>
<td>SXS1009I</td>
<td>0</td>
<td>0</td>
<td>Substation (ES) - Version 2.0.0</td>
</tr>
<tr>
<td></td>
<td>11:09:54.7894</td>
<td>154</td>
<td>SXG2801I</td>
<td>0</td>
<td>0</td>
<td>Initialized ~ Substation ES - Admin &amp; Op</td>
</tr>
<tr>
<td></td>
<td>11:09:55.0454</td>
<td>154</td>
<td>SXG2800I</td>
<td>0</td>
<td>0</td>
<td>Starting ~ Substation ES - Transformer</td>
</tr>
<tr>
<td></td>
<td>11:09:55.0479</td>
<td>0</td>
<td>SXT5717I</td>
<td>0</td>
<td>0</td>
<td>Trace level 1, 10 work threads; Cfg:TI</td>
</tr>
<tr>
<td></td>
<td>11:09:55.2871</td>
<td>154</td>
<td>SXG2801I</td>
<td>0</td>
<td>0</td>
<td>Initialized ~ Substation ES - Transformer</td>
</tr>
<tr>
<td></td>
<td>11:09:55.3874</td>
<td>154</td>
<td>SXG2800I</td>
<td>0</td>
<td>0</td>
<td>Starting ~ Substation ES - CICS Interf</td>
</tr>
<tr>
<td></td>
<td>11:09:55.4673</td>
<td>154</td>
<td>SXC3000I</td>
<td>0</td>
<td>0</td>
<td>CICSTS4G - Connect IICICS to CICS Regi</td>
</tr>
<tr>
<td></td>
<td>11:09:55.5539</td>
<td>154</td>
<td>SXC3001I</td>
<td>0</td>
<td>0</td>
<td>CICSTS4G - Connect IICICS to CICS Regi</td>
</tr>
<tr>
<td></td>
<td>11:09:55.5590</td>
<td>154</td>
<td>SXG2801I</td>
<td>0</td>
<td>0</td>
<td>Initialized ~ Substation ES - CICS Int</td>
</tr>
<tr>
<td></td>
<td>11:09:55.5950</td>
<td>156</td>
<td>SXC3400I</td>
<td>0</td>
<td>0</td>
<td>CICSTS4G - Started CICS EXCI Session (</td>
</tr>
<tr>
<td></td>
<td>11:09:55.6130</td>
<td>156</td>
<td>SXC3400I</td>
<td>0</td>
<td>0</td>
<td>CICSTS4G - Started CICS EXCI Session (</td>
</tr>
<tr>
<td></td>
<td>11:09:55.6352</td>
<td>154</td>
<td>SXG2800I</td>
<td>0</td>
<td>0</td>
<td>Starting ~ Substation ES - IMS Interf</td>
</tr>
<tr>
<td></td>
<td>11:09:55.6612</td>
<td>154</td>
<td>SXI4000I</td>
<td>0</td>
<td>0</td>
<td>A01IMSG - Connect IIIMS to IMS Region</td>
</tr>
<tr>
<td></td>
<td>11:09:55.7283</td>
<td>154</td>
<td>SXI4001I</td>
<td>0</td>
<td>0</td>
<td>A01IMSG - Connect IIIMS to IMS Region</td>
</tr>
<tr>
<td></td>
<td>11:09:55.7326</td>
<td>154</td>
<td>SXG2801I</td>
<td>0</td>
<td>0</td>
<td>Initialized ~ Substation ES - IMS Int</td>
</tr>
<tr>
<td></td>
<td>11:09:55.7336</td>
<td>154</td>
<td>SXS1001I</td>
<td>0</td>
<td>0</td>
<td>Initialized ~ Substation ES for IVP's</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8350</td>
<td>154</td>
<td>SXG1801I</td>
<td>0</td>
<td>0</td>
<td>Parameter Log closed</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8362</td>
<td>154</td>
<td>SXS2120I</td>
<td>0</td>
<td>0</td>
<td>(B) HiMark:0 Occ:1024 Incr:0 Curr:0</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8363</td>
<td>154</td>
<td>SXS2121I</td>
<td>0</td>
<td>0</td>
<td>(F) Stress Levels Strt:768 Slow:819 St</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8408</td>
<td>154</td>
<td>SXS2120I</td>
<td>0</td>
<td>0</td>
<td>(A) HiMark:410 Occ:1024 Incr:102 Curr:1</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8504</td>
<td>154</td>
<td>SXG2806I</td>
<td>0</td>
<td>0</td>
<td>Activated ~ Substation ES - CICS Interf</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8521</td>
<td>154</td>
<td>SXG2806I</td>
<td>0</td>
<td>0</td>
<td>Activated ~ Substation ES - Transforme</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8525</td>
<td>0</td>
<td>SXT5036I</td>
<td>0</td>
<td>0</td>
<td>All transformer tasks are Good To Go</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8525</td>
<td>154</td>
<td>SXG2806I</td>
<td>0</td>
<td>0</td>
<td>Activated ~ Substation ES - Admin &amp; Op</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8531</td>
<td>154</td>
<td>SXG2806I</td>
<td>0</td>
<td>0</td>
<td>Activated ~ Substation ES - IMS Interf</td>
</tr>
<tr>
<td></td>
<td>11:09:55.8610</td>
<td>154</td>
<td>SXG2806I</td>
<td>0</td>
<td>0</td>
<td>Activated ~ Substation ES for IVP's</td>
</tr>
</tbody>
</table>

Field Descriptions

Sel  If s or x is specified in the input column, a popup panel with all values related to the selected message is shown.

Valid selections: s and x
| **Time** | Time this message was recorded by Substation ES’s LTA Agent. |
| **Type** | Internal message type code. |
| **Msg Id** | Message identifier. |
| **RC** | Return code associated with the message. |
| **Rsn** | Reason code associated with the message. |
| **Message** | Initial part of the message description. The entire message is available when the popup panel is selected. |
Single Message Values Panel

This panel displays the detail information 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 ===>

<table>
<thead>
<tr>
<th>Msg Id: SXG1600I</th>
<th>Union id: 0</th>
<th>Stck: B9EC624552B80481</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type: 154</td>
<td>GRIN: 0</td>
<td>RC: 0</td>
</tr>
<tr>
<td>Date: 2003/08/25</td>
<td>Proc Id: 67111444</td>
<td>Reason: 0</td>
</tr>
<tr>
<td>Time: 11:09:54.0919</td>
<td>Thread: 0</td>
<td></td>
</tr>
</tbody>
</table>

Message-Dependent Values:
Buffer length - 40

<table>
<thead>
<tr>
<th>No additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message:</td>
</tr>
<tr>
<td>Log Agent Starting - Logging to TIBLOGF1</td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td>****************************</td>
</tr>
</tbody>
</table>

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Msg Id</td>
<td>Eight-character message identifier.</td>
</tr>
<tr>
<td>Type</td>
<td>Internal message type.</td>
</tr>
<tr>
<td>Date</td>
<td>The date this message was recorded by the Substation ES LTA Agent.</td>
</tr>
<tr>
<td>Time</td>
<td>The time this message was recorded by the Substation ES LTA Agent.</td>
</tr>
<tr>
<td>Union id</td>
<td>Representation of what message fields were recorded for this entry. Refer to</td>
</tr>
<tr>
<td></td>
<td>TIBCO Substation ES Messages and Codes for a description.</td>
</tr>
<tr>
<td>GRIN</td>
<td>Internal Global Resource Identification Number.</td>
</tr>
<tr>
<td>Proc Id</td>
<td>Internal process identifier. Used to identify a Substation ES Task or sub-Task.</td>
</tr>
<tr>
<td>Thread</td>
<td>Identifies the Substation ES thread that issued the message</td>
</tr>
<tr>
<td>Stck</td>
<td>Internal 64-bit Store Clock value that uniquely identifies this message.</td>
</tr>
<tr>
<td>RC</td>
<td>Return code associated with the message.</td>
</tr>
<tr>
<td><strong>Reason</strong></td>
<td>Return code associated with the message.</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td><strong>Buffer length</strong></td>
<td>Complete message description. When necessary, multiple lines are used to display the message.</td>
</tr>
</tbody>
</table>
Chapter 2  Configuring Additional Interfaces

You can use Substation ES to specify different interface configurations to meet different business requirements. For example, a single Substation ES can be configured to communicate with multiple CICS regions at the same time. Alternatively, multiple Substation ES instances can be configured to communicate with the same CICS region. The requirements of your system determines how you utilize your Substation ES.

Topics

- Adding an Additional Substation ES CICS Interface, page 102
- Adding an Additional ESB Interface, page 103
- Adding an Additional EMS ESB Interface With SSL, page 104
- Adding an Additional Rendezvous ESB Interface With SSL, page 106
- Adding an Additional Substation ES to a Single CICS Region, page 107
Adding an Additional Substation ES CICS Interface

Perform the following steps to add an additional Substation ES CICS Interface connection to a CICS region.

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>.CNTL(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 being `<USERHLQ>.CNTL(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.

The current version of Substation ES supports about thirty interfaces per single started task.
Adding an Additional ESB Interface

To add an additional TIBCO ESB Interface (formerly referred to as the Transformer) connection to a CICS region, perform the following steps:

1. Add a Substation ES CICS Interface by performing the steps in Adding an Additional Substation ES CICS Interface on page 102.

2. Create a new ESB Interface member by copying the default ESB SIP member located in <USERHLQ>.CNTL(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 being <USERHLQ>.CNTL(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 use identical INTF-IDs in the ESB SIP members.
Adding an Additional EMS ESB Interface With SSL

In Substation ES, SSL is implemented at the interface level, not at the service level. All applications or recipes/triggers using SSL must use an ESB interface that has SSL enabled. To add an additional TIBCO ESB Interface that uses SSL, perform the following steps:

1. Add a Substation ES CICS Interface as described in Adding an Additional Substation ES CICS Interface on page 102.

2. Create a new ESB Interface member by copying the default ESB SIP member located in <USERHLQ>.CNTL(SXSIEMS1) to a new name. This new member must have a different INTF-ID keyword value. Change the New ESB interface member as follows:

   SSL-YN=Y
   Set to Y to indicate an SSL interface.

   SSL-PASSWORD=123#SSL
   Set to the SSL server password (from 1 to 64 characters maximum).

   SSL-IDENTITY-DDN=SSLCCERT
   Set to the DDName for the SSL Certificate Id. The DDName must be in the JCL for Substation ES.
   For digital certificates, TIBCO EMS Client for z/OS (MVS) supports the following file formats:
   - PEM (Privacy Enhanced Mail)
   - PKCS#12

   SSL-KEY-DDN=SSLKEY
   Set to the DDName for the SSK Key. The DDName must be in the JCL for Substation ES.
   For private keys, TIBCO EMS Client for z/OS (MVS) supports the following file formats:
   - PEM (Privacy Enhanced Mail)
   - PKCS#8
   - PKCS#12
   Please refer to TIBCO Enterprise Message Service User’s Guide for a complete description of SSL usage in TIBCO EMS.
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 being <USERHLQ>.CNTL(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.
Adding an Additional Rendezvous ESB Interface With SSL

Adding SSL for a Rendezvous interface is done in the Rendezvous daemon definition, not in Substation ES. Refer to the Rendezvous administrator to set up a Rendezvous SSL interface.
Adding an Additional Substation ES to a Single CICS Region

Perform the following steps to configure multiple Substations that communicate to a single CICS region. For each Substation ES:

1. Create a copy of the Substation ES SIP for CICS interface member `<USERHLQ>.CNTL(SXCINTF)`. In the new member specify a new INTF-ID keyword value.

2. Create a copy of Substation ES SIP Transformer interface member `<USERHLQ>.CNTL(SXSIEMS1 or SXSIRV1)`. Make parameter changes that are necessary.

3. Create a new copy of the Substation ES system initialization parameters SIP member `<USERHLQ>.CNTL(SXSSIP$1)`.

4. Change the INTF-MEMBER entries in the Substation ES SIP member for each of the newly-created CICS and Transformer Interface member names.

5. Change the Substation Id in the Substation ES SIP.

6. Create a new copy of the Substation ES system startup parameters (SSP) member `<USERHLQ>.CNTL(SXSSSP$1)`.

7. Change the SIPMEM keyword value and point it to the newly created SXSSIP$1 member

8. 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`

When sharing one Transformer configuration file with multiple instances of Substation ES, use groups to distinguish between recipes or triggers definitions for different Substations. For details, see Defining Groups on page 21.

The current version of Substation ES supports up to five Substation ES regions connected to a single CICS region.
Chapter 3  Configuration File Utilities

This chapter describes the TIBCO Substation ES configuration file utilities.

Topics

- Configuration File Overview, page 110
- Configuration File Conversion Utility, page 111
- Configuration File Utility, page 113
- Configuration File Selective Export Utility, page 117
Configuration File Overview

Before using Substation ES, you must supply information contained in the configuration file to the Substation ES Transformer Interface. The Substation ES Transformer Interface requires this configuration file information in order 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. Refer to the TIBCO Substation ES Release Notes for the release of interest. You are able to determine whether conversions are necessary and what should be done to successfully implement your upgrade.

A Configuration File must be allocated with IDCAMS. Refer to Configuration File Allocation on page 5 for details.

Substation ES release contains configuration utilities to assist you to use, maintain and promote definitions in configuration files.
Configuration File Conversion Utility

The Configuration File Conversion Utility extracts the configuration entities from a Substation ES configuration file and creates a sequential EXPORT file. The EXPORT file can be imported to a Substation ES configuration file using the Configuration File Utility. To create and allocate a configuration file, refer to Configuration File Allocation on page 5.

Functions

- Extracts all entities from a Substation ES configuration file.

JCL Sample (SXSCFC21)

Execution is controlled by parm values and JCL statements.

```
//JS010   EXEC PGM=SXT3CV21,REGION=0M,PARM='parm values'
//STEPLIB   DD DISP=SHR,DSN=< USERHLQ>.LOAD
//TIBLOG DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//CEEDUMP 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)
```

Table 2 Configuration File Conversion Utility DDNames

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

Configuration File Conversion Utility Parameter Values

The possible values that can be entered in the parm field of the EXEC statement are described in the following table.
All the values in the list below are optional; the conversion process can proceed with no parm data.

**Table 3 Transformer Conversion Utility Parameters**

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Default</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dn</td>
<td>Debug</td>
<td>2</td>
<td>Values for n can be between 0 and 5, with 0 providing very few messages and 5 showing hex displays of all records processed.</td>
</tr>
<tr>
<td>-Ixxx</td>
<td>Input DD name</td>
<td>CONFIG</td>
<td>Allows another DD name to be used in place of CONFIG</td>
</tr>
<tr>
<td>-Nxxx</td>
<td>Export DD name</td>
<td>EXPORT</td>
<td>Write out the entire contents of CONFIG in export format to the EXPORT file. No selection is provided; the entire file is processed. If xxx is included, that is used as the DD name to replace EXPORT.</td>
</tr>
<tr>
<td>-Tn</td>
<td>Table Debug</td>
<td>2</td>
<td>Defines the diagnostic level for the internal TABLE processor, used to tie together information from the various VSAM keyed records.</td>
</tr>
<tr>
<td>-X</td>
<td>Hex Dump</td>
<td>None</td>
<td>Sequentially read and hex-display the entire Configuration file; this is used as a diagnostic aid.</td>
</tr>
</tbody>
</table>
The Configuration File Utility performs numerous functions relating to a configuration file.

**Functions**

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

Execution is controlled by values on PARM statement and JCL parameters.

```
//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)  
//
```
Chapter 3 Configuration File Utilities

For a summary of the usages of the DD names, refer to Appendix B, Transformer Configuration File Utilities Summary, page 171.

Table 4 Configuration File Utility DDNames

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFIG</td>
<td>Points to the current Transformer configuration file that all operations are to be performed upon.</td>
</tr>
<tr>
<td>CONFIG2</td>
<td>Points to a new copy of the Transformer configuration file. This file is updated during a copy operation.</td>
</tr>
<tr>
<td>IMPORT</td>
<td>Points to an existing IMPORT file. This file is read during an import operation.</td>
</tr>
<tr>
<td>EXPORT</td>
<td>Points to a new EXPORT file. This file is written during an export operation. This file can be a pre-allocated disk file.</td>
</tr>
</tbody>
</table>

Table 5 Configuration File Utility Function Parameters

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Default</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-C</td>
<td>Copy</td>
<td>None</td>
<td>Copy all definitions from one configuration file to another file defined by CONFIG2. The file referenced by CONFIG2 is initialized before the copy is started.</td>
</tr>
<tr>
<td>-M&lt;ddn&gt;</td>
<td>Import</td>
<td>None, IMPORT</td>
<td>Read the contents of the IMPORT file and apply to the CONFIG file. If ddn is specified, this DD Name is used instead of IMPORT.</td>
</tr>
<tr>
<td>-N&lt;ddn&gt;</td>
<td>Export</td>
<td>None, EXPORT</td>
<td>Write out the entire contents of CONFIG in export format to the EXPORT file. If ddn is specified, this DD Name is used instead of EXPORT.</td>
</tr>
</tbody>
</table>
### Table 6  Configuration File Utility Parameters

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Default</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dn</td>
<td>Debug</td>
<td>2</td>
<td>Values for n can be between 0 and 5, with 0 providing very few messages and 5 showing hex displays of all records processed.</td>
</tr>
<tr>
<td>-Ixxx</td>
<td>Input DD name</td>
<td>CONFIG</td>
<td>Allows another DD name to be used in place of CONFIG</td>
</tr>
<tr>
<td>-Oxxx</td>
<td>Output DD name</td>
<td>CONFIG2</td>
<td>Allows another DD name to be used in place of CONFIG2</td>
</tr>
<tr>
<td>-P</td>
<td>Print</td>
<td>None</td>
<td>Print out each recipe and trigger with the associated Backend, Transport, Group membership and Conversion rules.</td>
</tr>
<tr>
<td>-R</td>
<td>Read-only</td>
<td>Update</td>
<td>Open the CONFIG file as read-only. If combined with Import, all updates from the import operation are lost.</td>
</tr>
<tr>
<td>-T</td>
<td>Initialize</td>
<td>None</td>
<td>Allows the program to initialize the CONFIG file if the file does not contain the Transformer_Anchor structure at the beginning of the file. The Initialize request is not applicable if read-only is also specified.</td>
</tr>
<tr>
<td>-V</td>
<td>Validate</td>
<td>None</td>
<td>Validate all data structures for consistency and block identifiers (eyeball characters). If specified as -V -V, validation does not terminate after the first error is found.</td>
</tr>
<tr>
<td>-X</td>
<td>Hex Dump</td>
<td>None</td>
<td>All active definitions in the file are displayed in hex. Relationships among definitions are not shown.</td>
</tr>
<tr>
<td>-Z</td>
<td>Zero file</td>
<td>None</td>
<td>Reset the CONFIG file to empty before beginning the requested processes. See warning note below.</td>
</tr>
</tbody>
</table>
Do not use the Z option when importing data into a non-empty configuration file. Doing so destroys all existing data in the configuration file.

For a summary of the usage of the parameter options, refer to Appendix B, Transformer Configuration File Utilities Summary, on page 171.
Configuration File Selective Export Utility

The Configuration File Selective Export Utility SXT3CSIE exports entities and definitions relating to a configuration file.

Functions

The SXT3CSIE utility provides the following functions:

- Extract all or selected entities within a configuration file.
- Write selected entities to an sequential EXPORT file.

This utility is used to export “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 contained in the exported sequential file, use the Configuration File Utility.

In addition, this utility can be used to export “logical pieces” to a sequential file while also deleting those entities from the configuration file. In this case, although the delete request is entered during the export, it is not applied to the current file. The request is written to the export file, and processed during the IMPORT process against the updated file, using the Configuration File Utility.

JCL Sample (SXSCFEXS)

Execution is controlled by PARM values, JCL statements and parameters in the CONTROL DD Name.

```
//STEP0002 EXEC PGM=SXT3CSIE,REGION=0M,PARM='values'
//STEPLIB DD DISP=SHR,DSN=<USERHLQ>.LOAD
//TIBLOG DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSPRINT 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)
//
```
Table 7  Configuration File Selective Export Utility DDNames

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFIG</td>
<td>Points to a current linear Transformer Configuration file.</td>
</tr>
<tr>
<td>EXPORT</td>
<td>Points to a new EXPORT file that is written during an export operation. This file can be a pre-allocated disk file.</td>
</tr>
</tbody>
</table>

For a summary of the usages of DD names, refer to Appendix B, Transformer Configuration File Utilities Summary, page 171.

Configuration File Selective Export Utility PARM Field Values

The possible values that can be entered in the parm field of the EXEC statement are described in the following table.

Table 8  Configuration File Selective Export Utility Parameters

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Default</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Cxxx</td>
<td>Control DD name</td>
<td>CONTROL</td>
<td>Allows another DD name to be used in place of CONTROL</td>
</tr>
<tr>
<td>-Dn</td>
<td>Debug</td>
<td>2</td>
<td>Values for n can be between 0 and 5, with 0 providing very few messages and 5 showing hex displays of all records processed.</td>
</tr>
<tr>
<td>-Ixxx</td>
<td>Input DD name</td>
<td>CONFIG</td>
<td>Allows another DD name to be used in place of CONFIG</td>
</tr>
<tr>
<td>-Nxxx</td>
<td>Export DD name</td>
<td>EXPORT</td>
<td>Allows another DD name to be used in place of EXPORT</td>
</tr>
<tr>
<td>-R</td>
<td>Read-only</td>
<td>Update</td>
<td>Open the CONFIG file as read-only mode.</td>
</tr>
</tbody>
</table>

For a summary of the usage of the parameter options, refer to Appendix B, Transformer Configuration File Utilities Summary, on page 171.

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>

DELETE <type> <identifier>

For DELETE, the identifier name must be provided, it cannot be asterisk (*). The same deletion rules apply as in the ISPF deletes – the entry to be deleted cannot be used by any other entity.

Statement values are described below.

**Entity Type Values**

An entity type value must be entered in a export control statement.

*Table 9  Entity Type Values*

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER</td>
<td>For trigger definitions only</td>
</tr>
<tr>
<td>RECIPE</td>
<td>For recipe definitions only</td>
</tr>
<tr>
<td>BACKEND</td>
<td>For BES definitions only</td>
</tr>
<tr>
<td>TRANSSPORT</td>
<td>For transport definitions only</td>
</tr>
<tr>
<td>BUFFER</td>
<td>For buffer definitions only</td>
</tr>
<tr>
<td>RVMSG</td>
<td>For RV message definitions only</td>
</tr>
<tr>
<td>CONVERTER</td>
<td>For conversion role definitions only</td>
</tr>
<tr>
<td>GROUP</td>
<td>For group definitions only</td>
</tr>
<tr>
<td>ALL</td>
<td>For all definitions</td>
</tr>
</tbody>
</table>

**Identifier Values**

The identifier defines the id value of a specific entry.

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

**Qualifier Values**

A qualifier can be entered depending on the type specified.

If no qualifiers are entered, the default is all.
If the *type* is *all*, the default *qualifier* is *only*.

*Qualifiers* that are entered and do not apply to the type specified are ignored.

**Table 10  Definition Qualifier Values**

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Export all related definitions of the selected <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>ONLY</td>
<td>Export only the selected <code>&lt;type&gt;</code> definition, do not export related structure definitions</td>
</tr>
<tr>
<td>BUF</td>
<td>Export the related buffer definitions for the selected <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>MSG</td>
<td>Export the related message definitions for the selected <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>CVR</td>
<td>Export the related Conversion Rule definitions for the selected <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>BES</td>
<td>Export the related BES definitions for the selected <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>TRN</td>
<td>Export the related Transport definitions for the selected <code>&lt;type&gt;</code></td>
</tr>
<tr>
<td>GRP</td>
<td>Export the related Group definitions for the selected <code>&lt;type&gt;</code></td>
</tr>
</tbody>
</table>

**Examples**

- To export a recipe definition named `rcp1` and all its related structures:
  ```
  EXPORT RECIPE rcp1
  ```
- The same request, but completely qualified instead of accepting the default qualifier *all*:
  ```
  EXPORT RECIPE rcp1 BUF MSG CVR BES TRN GRP
  ```
- To export an entire file:
  ```
  EXPORT ALL * ONLY
  ```
  (this same functionality is provided by `RVY3CFUT` with parm `-N`)
- To delete a recipe definition named `rcp1` and all its related structures:
  ```
  DELETE RECIPE rcp1
  ```
- To delete a buffer definition named `buf2` and all its related structures:
  ```
  DELETE BUFFER buf2
  ```
Chapter 4  Substation ES Examples

This chapter contains examples of using Substation ES.

Topics

- Overview, page 122
- Substation ES Installation IVPs, page 123
- Substation ES Generic IVP Components, page 124
- ESB Request – Reply DPL Example, page 132
- CICS Trigger Example, page 139
- CICS High Volume Trigger (HVT) Example, page 140
- ESB Request DPL XCTL Example, page 141
- ESB Invoking a CICS Transaction Example, page 142
- ESB to TDQ Example, page 143
- CICS TSQ Entry Written Example, page 144
- ESB Request – Reply Complex DPL Example, page 145
- CICS Initiated Request or Reply Example, page 146
- ESB Request – Reply IMS Example, page 148
- ESB Request – Reply and Trigger Example, page 149
- IMS BMP Trigger Example, page 150
- IMS Conversational Example, page 151
- Transform Only Data Examples, page 153
Overview

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

Sample code has been included to help the user 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 & IVP Generic Listener programs in previous versions have been combined into a single program call IVP Sender Listener in Substation ES version 2.3. The modules are SXJ3ISL1 for EMS and SXR3ISL1 for Rendezvous.

An Error listener source SXR3IELS is also included; this program listens for errors published by instances of TIBCO Substation ES. For summary usage of sample IVPs resources, refer to Appendix A, Sample IVP Resources, on page 167.

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

For more information on the Generic Listener, see Generic Sender or Listener Utility in TIBCO Substation ES Operations and Administration. For more information on the Generic Error Listener, see Generic Error Listener Utility in TIBCO Substation ES Operations and Administration.
Substation ES Installation IVPs

During Substation ES installation, IVP programs are supplied by default. To ensure that the network, Substation ES and the back-end system interfaces (CICS and IMS) are executing bi-directional communication using Substation ES correctly, execute the IVP programs supplied.

The installation default transformer configuration file contains all the entries needed to execute Substation ES IVP programs.

Request or Reply

To demonstrate Request or Reply message processing, the Substation ES IVP Sender Listener is used to publish 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 back-end system.

Upon successful completion, an acknowledgement or result set is returned to the Substation ES IVP Sender Listener via 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 successful completion, an acknowledgement or result set is returned to Substation ES IVP listener via Substation ES.
Substation ES Generic IVP Components

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, 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 will return a reply; the messages will be consumed and reported by this Substation ES IVP RV sender listener. Errors are also reported. There are two run JCLs, that can be found in:

Data Set:  &lt;USERHLQ&gt;.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 other settings, refer to TIBCO Rendezvous Administration for information about the parameters.

SYSIN Parameters

The following SYSIN parameter members can be found in:

Data Set:  &lt;USERHLQ&gt;.CNTL
Member:  SXCRIP01  for CICS Interface IVPs
        SXIRIP01 and SXIRIP02  for IMS Interface IVPs

The following is a sample of the input control statements that can be found in the SYSIN 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 specified in the SYSIN DD Name file. Valid control parameters are described in Table 11.

**Table 11  Rendezvous IVP Sender Listener Control Parameters**

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BES &lt;name&gt;</td>
<td>BES interface ID for the tibss-BES system field; default = blank.</td>
</tr>
<tr>
<td>BES-DELAY &lt;timevalue&gt;</td>
<td>Time (seconds) to wait between message replies on the CICS side; default = 0.</td>
</tr>
<tr>
<td>CLIENT-ID &lt;name&gt;</td>
<td>ID to use with the Rendezvous daemon connection.</td>
</tr>
<tr>
<td>DAEMON &lt;url&gt;:&lt;port&gt;</td>
<td>Rendezvous daemon URL and TCP port.</td>
</tr>
<tr>
<td>DEBUG ON</td>
<td>OFF</td>
</tr>
<tr>
<td>DESCR &lt;name&gt;</td>
<td>The 56 characters to be used in field IVP-DESCR.</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>The eight characters to be used in field IVP-ID.</td>
</tr>
<tr>
<td>NETWORK &lt;address&gt;</td>
<td>Rendezvous network parameter.</td>
</tr>
<tr>
<td>PATTERN-LEN &lt;number&gt;</td>
<td>Create variable field IVP-STRING of length &lt;number&gt; and populate with 1--------10------…. default = 0.</td>
</tr>
<tr>
<td>PSWD &lt;password&gt;</td>
<td>The password for the User ID previously specified and authorized to execute this process. Security Interface dependent. For the tibss-password system field; default = blank.</td>
</tr>
<tr>
<td>PUBLISH-ONLY</td>
<td>Send message only; do not listen for reply message.</td>
</tr>
<tr>
<td>REPEAT &lt;number&gt;</td>
<td>Number of times to repeat the send process; default = 1.</td>
</tr>
</tbody>
</table>
### Table 11  Rendezvous IVP Sender Listener Control Parameters

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE &lt;name&gt;</td>
<td>Name for the tibss-resource system field; default = blank. Maximum of eight characters. When present, this value overrides the recipe Resource Name field. The use of the Resource Name field depends on the Method on Invocation. For details, see Resource Name in Recipe Details Panel on page 53.</td>
</tr>
<tr>
<td>SERVICE &lt;port&gt;</td>
<td>Rendezvous UDP service port number.</td>
</tr>
<tr>
<td>SUBJ-REP &lt;name&gt;</td>
<td>The Rendezvous reply subject name to which the reply message is sent. If not PUBLISH-ONLY and SUBJ-REP is not present, then a private inbox subject name will be used.</td>
</tr>
<tr>
<td>SUBJ-REQ &lt;name&gt;</td>
<td>The Rendezvous request subject name to which the message is sent.</td>
</tr>
<tr>
<td>TIMEOUT &lt;timevalue&gt;</td>
<td>Time (seconds) for response timeout; default = 5.</td>
</tr>
<tr>
<td>USER &lt;userid&gt;</td>
<td>The RACF or back-end User ID for the user who is authorized to execute this process. Security Interface dependent. For the tibss-userid system field; default = blank.</td>
</tr>
<tr>
<td>WAIT &lt;timevalue&gt;</td>
<td>Time (seconds) to wait between message sends; default = 0.</td>
</tr>
<tr>
<td>* &lt;text&gt;</td>
<td>Comment</td>
</tr>
</tbody>
</table>
EMS IVP Sender Listener

The program SXJ3ISL1 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, 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 will return a reply; the messages will be consumed and reported by this Substation ES IVP EMS Sender Listener. Errors are also reported. There are two run JCLs. They can be found in:

Data Set:  <USERHLQ>.JCL
Member:  SXCRJSL for CICS Interface IVPs
         SXIRJSL for IMS Interface IVPs

The EXEC PARM - IVPNETW is substituted with a set of TIBCO EMS transport parameters. It is recommended that you use the provided default settings for all IVP runs. To use other settings, refer to TIBCO EMS User’s Guide for information about the parameters.

SYSIN Parameters

The following SYSIN parameter members can be found in:

Data Set:  <USERHLQ>.CNTL
Member:  SXCJIP01, SXCJIP02 and SXCJIP03 for CICS Interface IVPs
         SXIJIP01 and SXIJIP02 for IMS Interface IVPs

The following is a sample of the input control statements that can be found 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 specified in the SYSIN DD Name file. Valid control parameters are described in Table 12.
### Table 12  EMS IVP Sender Listener Control Parameters

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BES <code>&lt;name&gt;</code></td>
<td>BES interface ID for the tibss-BES system field, it can be sent as a user property field or MAP message; default = blank.</td>
</tr>
<tr>
<td>BES-DELAY <code>&lt;timevalue&gt;</code></td>
<td>Time (seconds) to wait between message replies on the CICS side; default = 0.</td>
</tr>
<tr>
<td>CLIENT-ID <code>&lt;name&gt;</code></td>
<td>Id to use with the EMS connection</td>
</tr>
<tr>
<td>DEBUG ON</td>
<td>OFF</td>
</tr>
<tr>
<td>DEST-REP <code>&lt;name&gt;</code></td>
<td>The destination name, the EMS reply destination to which the reply is to be sent. If not PUBLISH-ONLY and DEST-REP is not present, then a temporary Topic or Queue will be used.</td>
</tr>
<tr>
<td>DEST-REQ <code>&lt;name&gt;</code></td>
<td>The destination name, the EMS subject destination to which the message is sent.</td>
</tr>
<tr>
<td>HOSTPAGE <code>&lt;code page&gt;</code></td>
<td>Value for setting host code page; default = IBM-1047.</td>
</tr>
<tr>
<td>DESCR <code>&lt;name&gt;</code></td>
<td>The 56 characters to be used in field IVP-DESCR.</td>
</tr>
<tr>
<td>EMS-PSWD <code>&lt;password&gt;</code></td>
<td>Password to connect to EMS server.</td>
</tr>
<tr>
<td>EMS-USER <code>&lt;userid&gt;</code></td>
<td>User ID to connect to EMS server.</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>The eight characters to be used in field IVP-ID.</td>
</tr>
<tr>
<td>MSGTYPE MAP</td>
<td>BYTES</td>
</tr>
<tr>
<td>NETPAGE <code>&lt;code page&gt;</code></td>
<td>Value for setting network code page; default = ISO8859-1.</td>
</tr>
<tr>
<td>PATTERN-LEN <code>&lt;number&gt;</code></td>
<td>Create MAP message field IVP-STRING of length <code>&lt;number&gt;</code> and populate with 1-------10------.... default = 0.</td>
</tr>
</tbody>
</table>
### Table 12  EMS IVP Sender Listener Control Parameters

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPERTIES YES</td>
<td>Indicates whether Substation ES system fields (tibss-userid, tibss-password, tibss-resource, tibss-BES) should be added to a message as user property fields; default = NO.</td>
</tr>
<tr>
<td>PSWD &lt;password&gt;</td>
<td>Password for the tibss-password user property field or MAP message field; default = blank.</td>
</tr>
<tr>
<td>PUBLISH-ONLY</td>
<td>Indicates program should not listen for replies.</td>
</tr>
<tr>
<td>REPEAT &lt;number&gt;</td>
<td>Number of times to repeat the send process; default = 1.</td>
</tr>
<tr>
<td>RESOURCE &lt;name&gt;</td>
<td>Name for the tibss-resource user property field or MAP message field; default = blank. Maximum of eight characters. When present, this value overrides the recipe Resource Name field. The use of the Resource Name field depends on the Method on Invocation. For details, see Resource Name in Recipe Details Panel on page 53.</td>
</tr>
<tr>
<td>SERVER &lt;url&gt;:&lt;port&gt;</td>
<td>EMS server URL and TCP port.</td>
</tr>
<tr>
<td>TIMEOUT &lt;timevalue&gt;</td>
<td>Time (seconds) for response timeout; default = 5.</td>
</tr>
<tr>
<td>TYPE TOPIC</td>
<td>QUEUE</td>
</tr>
<tr>
<td>USER &lt;userid&gt;</td>
<td>User ID for the tibss-userid system field; can be sent as a user property field or MAP message field; default = blank.</td>
</tr>
<tr>
<td>WAIT &lt;timevalue&gt;</td>
<td>Time (seconds) to wait between message sends; default = 0.</td>
</tr>
<tr>
<td>* &lt;text&gt;</td>
<td>Comment.</td>
</tr>
</tbody>
</table>
**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 can be found in:

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. To use other settings, refer to TIBCO Rendezvous Administration for detailed information about the parameters.

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

The following is a sample of the input control statements that can be found 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 can be found in:

Data Set: `<USERHLQ>.JCL`
Member: SXRIICVS

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 other settings, refer to TIBCO Rendezvous Administration for detailed information about the parameters.
IVP Generic Error Listener

The programs SXR3IELS and SXJ3IELS initiate a TIBCO messaging subscriber that receives error messages published by Substation ES about data transformation errors, and warning and error conditions encountered during Substation ES IVP’s run-time processing. The sample JCL stream for error listeners can be found in:

Data Set: `<USERHLQ>`.JCL
Member: SXRRIELS for Rendezvous
        SXJRIELS for EMS

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 other settings, refer to TIBCO Rendezvous Administration for detailed information about the parameters.

The error messages subjects that the IVP Generic Error Listener subscribes to are determined by the Keywords and Operands specified in the SYSIN file in:

Data Set: `<USERHLQ>`.CNTL
Member: SXRPERLS for Rendezvous
        SXJPERLS for EMS

Table 13 describes valid control parameters.

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJECT {tibss.error.ivp.&gt;}</td>
<td>IVP error subject names to be listened to.</td>
</tr>
<tr>
<td>QUEUE or TOPIC</td>
<td>EMS only. The subject destination that an error message is published on when Transformer conversion or BES encounters errors in a T(opic) or Q(ueue).</td>
</tr>
<tr>
<td>PSWD</td>
<td>EMS only. Password associated with the User Id if used in the session.</td>
</tr>
<tr>
<td>UFLD-PFX {tibss-}</td>
<td>Prefix of the Substation ES system field name for the processing status. We recommend that you do not change this value for IVP runs.</td>
</tr>
<tr>
<td>USER</td>
<td>EMS only. User supplied Id if used in the session.</td>
</tr>
</tbody>
</table>

For further details on the Generic Error Listener utility, see **Generic Error Listener Utility** in TIBCO Substation ES Operations and Administration.
ESB Request – Reply DPL Example

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

The CICS application replies by putting information into the COMMAREA, and 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 on the process flow, refer to the Messages Flow section of TIBCO Substation ES Concepts.

Component Summary

For the C language:

- SXCRJSL / SXCRIRSL – Publisher sending the request message
- SXC-DPL-C – Substation ES recipe identification
- SXC-DPL-MRO – Substation ES recipe identification for DPL MRO Method of Invocation
- SXC3I001 – CICS 'C' application program

For COBOL:

- SXCRJSL / SXCRIRSL – Publisher sending the request message
- SXC-DPL-COBOL – Substation ES recipe identification
- SXC-DPL-MRO – Substation ES recipe identification for DPL MRO Method of Invocation
- SXCCI001 – CICS COBOL application program
IVP Sender Listener Execution Parameters

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    300000
WAIT       0
BES-DELAY  5
```

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

```
SUBJ-REQ   tibss.CICS.MRO.Request
RESOURCE   SXC3I001
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
```

EMS – DPL Request/Reply (LANGUAGE 'C')

```
DEST-REQ   tibss.CICS.C.Request
DEST-REP   tibss.CICS.C.Reply
TYPE       QUEUE
RESOURCE   SXC3I001
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
```

EMS – DPL Request/Reply (LANGUAGE COBOL)

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

EMS – DPL MRO Request/Reply (LANGUAGE COBOL)

```
DEST-REQ   tibss.CICS.MRO.Request
DEST-REP   tibss.CICS.MRO.Reply
TYPE       QUEUE
RESOURCE   SXCCI001
```
ESB Request – Reply DPL Container Example

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application using CICS DPL method with Container via 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 on the process flow, refer to the Messages Flow section of *TIBCO Substation ES Concepts*.

### Component Summary

For COBOL:
- SXCRIJSL / SXCRIRSL – Publisher sending the request message
- SXC-DPL-COBOL-Container – Substation ES recipe identification
- SXCCI014 – CICS COBOL application program

### IVP Sender Listener Execution Parameters

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJ-REQ</td>
<td>tibss.CICS.Cobol.Container.Request</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXC3I014</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
</tbody>
</table>

**EMS – DPL Container Request/Reply (LANGUAGE COBOL)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REQ</td>
<td>tibss.CICS.Cobol.Container.Request</td>
</tr>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.Cobol.Container.Reply</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXCCI014</td>
</tr>
</tbody>
</table>
ESB Request – Reply DPL Storage Example

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application using CICS DPL method with Storage Pointer via 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 on the process flow, refer to the Messages Flow section of TIBCO Substation ES Concepts.

**Component Summary**

For the C language:
- SXCRJSL / SXCRIRSL – Publisher sending the request message
- SX-DPL-C-Storage – Substation ES recipe identification
- SX3I013 – CICS 'C' application program

For COBOL:
- SXCRJSL / SXCRIRSL – Publisher sending the request message
- SX-DPL-COBOL-Storage – Substation ES recipe identification
- SXCCI013 – CICS COBOL application program

**IVP Sender Listener Execution Parameters**

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

**Rendezvous – DPL Storage Request/Reply (LANGUAGE 'C')**

```plaintext
SUBJ-REQ         tibss.CICS.C.STG.Request
RESOURCE         SX3I013
REPEAT           1
TIMEOUT          300000
WAIT             0
BES-DELAY        5
```
Rendezvous – DPL Storage Request/Reply (LANGUAGE COBOL)

<table>
<thead>
<tr>
<th>SUBJ-REQ</th>
<th>tibss.CICS.Cobol.STG.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE</td>
<td>SXC3I013</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
</tbody>
</table>

EMS – DPL Storage Request/Reply (LANGUAGE 'C')

<table>
<thead>
<tr>
<th>DEST-REQ</th>
<th>tibss.CICS.C.STG.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.C.STG.Reply</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXC3I013</td>
</tr>
</tbody>
</table>

EMS – DPL Storage Request/Reply (LANGUAGE COBOL)

<table>
<thead>
<tr>
<th>DEST-REQ</th>
<th>tibss.CICS.Cobol.STG.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.Cobol.STG.Reply</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXCCI013</td>
</tr>
</tbody>
</table>
ESB Request – Reply DPL TSQ Example

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application using CICS DPL method with a 8-character TSQ Name in the CommArea via 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 on the process flow, refer to the Messages Flow section of TIBCO Substation ES Concepts.

Component Summary

For the C language:
- SXCRIJSL / SXCRIRSL – Publisher sending the request message
- SXC-DPL-C-TSQ – Substation ES recipe identification
- SXC3I012 – CICS 'C' application program

For COBOL:
- SXCRIJSL / SXCRIRSL – Publisher sending the request message
- SXC-DPL-COBOL-TSQ – Substation ES recipe identification
- SXCCI012 – CICS COBOL application program

IVP Sender Listener Execution Parameters

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJ-REQ</td>
<td>tibss.CICS.C.TSQ.Request</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXC3I012</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
</tbody>
</table>
Rendezvous – DPL TSQ Request/Reply (LANGUAGE COBOL)

SUBJ-REQ tibss.CICS.Cobol.TSQ.Request
RESOURCE SXC3I012
REPEAT 1
TIMEOUT 300000
WAIT 0

EMS – DPL TSQ Request/Reply (LANGUAGE 'C')

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

EMS – DPL TSQ Request/Reply (LANGUAGE COBOL)

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

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

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

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

For more information on the process flow, refer to the Messages Flow section of *TIBCO Substation ES Concepts*.

**Component Summary**
- SXTT — CICS transaction Id, the request initiator
- SXCCI002 — CICS COBOL application program
- TRIGGER-FLDS — Substation ES trigger identification

**CICS Execution Command**
Transaction entered from a CICS 3270 screen: SXTT
CICS High Volume Trigger (HVT) Example

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 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. Then the Substation ES Transformer processes the data conversion based on 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 on the process flow, refer to High Volume Trigger in *TIBCO Substation ES Operations and Administration*.

**Component Summary**

For the C language:
- SXT6 — CICS transaction Id, the request initiator
- SXC3I032 — CICS application program
- TRIGGER-FLDS — Substation ES trigger identification
- TRIGGER-GUARANTEED Substation ES trigger identification
- TRIGGER-TEXT — Substation ES trigger identification

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

**CICS Execution Command**

Transaction entered from a CICS 3270 screen: SXTF or SXT6. For details, see High Volume Trigger in *TIBCO Substation ES Operations and Administration*. 
ESB Request DPL XCTL Example

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

The CICS program replies by writing the reply information into the Substation ES trigger TDQ. The Substation ES gets the trigger based on the defined search criteria for this trigger process. The Substation ES delivers the reply message and an acknowledgement message 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
- SXCRIJSL / SXCRIRSL — Publisher sending the request message
- SXC-DPL-COBOL-START-PGM — Substation ES recipe identification
- SXCCI003 — CICS COBOL application program
- TRIGGER-FLDS — Substation ES trigger identification

IVP Sender Listener Execution Parameters

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       300000
WAIT          0
BES-DELAY     5
IVP-ID        SXCIV03

EMS

DEST-REQ       tibss.CICS.Cobol.Program.Request
DEST-REP       tibss.CICS.Cobol.Program.Status
TYPE           QUEUE
RESOURCE       SXCCI003
REPEAT          1
TIMEOUT       300000
WAIT          0
BES-DELAY     5
IVP-ID        SXCIV03
ESB Invoking a CICS Transaction Example

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

The CICS program replies by writing the reply information into the Substation ES trigger TDQ. The Substation ES gets the trigger based on the defined search criteria for this trigger process. The Substation ES delivers the reply message and an acknowledgement message 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
- SXCRIJSL / SXCRIRSL — Publisher sending the request message
- SXC-DPL-COBOL-START-TRAN — Substation ES recipe identification
- SXCCI004 — CICS COBOL application program
- TRIGGER-FLDS — Substation ES trigger identification

IVP Sender Listener Execution Parameters

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

Rendezvous

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJ-REQ</td>
<td>tibss.CICS.Task.Request</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXTI</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>SXCIV04</td>
</tr>
</tbody>
</table>

EMS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REQ</td>
<td>tibss.CICS.Task.Request</td>
</tr>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.Task.Status</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXTI</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>SXCIV04</td>
</tr>
</tbody>
</table>
ESB to TDQ Example

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

The Substation ES delivers an acknowledgement message 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. A reply message is also delivered 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

- SXCRRIJSL / SXCRIRSL — Publisher sending the request message
- SXC-DPL-WRITE-TDQ — Substation ES recipe identification
- SXQT — Substation transient data queue (TDQ) name
- TRIGGER-FLDS — Substation ES trigger identification

IVP Sender Listener Execution Parameters

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 300000
WAIT 0
BES-DELAY 5
IVP-ID SXCIV05

EMS

DEST-REQ tibss.CICS.TDQ.Request
DEST-REP tibss.CICS.TDQ.Status
TYPE QUEUE
RESOURCE SXQT
REPEAT 1
TIMEOUT 300000
WAIT 0
BES-DELAY 5
IVP-ID SXCIV05
CICS TSQ Entry Written Example

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

The Substation ES delivers an acknowledgement message 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 on the process flow, refer to the Messages Flow section of TIBCO Substation ES Concepts.

Component Summary
- SXRJISL / SXCRIRSL — Publisher sending the request message
- SXC-DPL-WRITE-TSQ — Substation ES recipe identification
- SXCTSQI1 — A CICS TSQ Substation ES temporary storage queue name

IVP Sender Listener Execution Parameters

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   SXCTSQI1
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
IVP-ID     SXCIV06
```

EMS

```
DEST-REQ   tibss.CICS.TSQ.Request
DEST-REP   tibss.CICS.TSQ.Status
TYPE       QUEUE
RESOURCE   SXCTSQI1
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
IVP-ID     SXCIV06
```
ESB Request – Reply Complex DPL Example

This IVP example (that works the same as the CICS Request or Reply IVP) illustrates how a TIBCO messaging client application initiates a request message to invoke a CICS application using CICS DPL method via Substation ES.

The CICS application replies by putting information into the COMMAREA, from which the data is retrieved by the Substation ES. The Substation ES 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

- SXCRIJSL / SXCRIRSL — Publisher sending the request message
- SXC-DPL-COBOL-COMPLEX - Substation ES recipe identification
- SXCCI010 - CICS COBOL application program

IVP Sender Listener Execution Parameters

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

Rendezvous

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJ-REQ</td>
<td>tibss.CICS.Cobol.Complex.Request</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXCCI010</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>SXCIV010</td>
</tr>
</tbody>
</table>

EMS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REQ</td>
<td>tibss.CICS.Cobol.Complex.Request</td>
</tr>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.Cobol.Complex.Reply</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXCCI010</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>SXCIV010</td>
</tr>
</tbody>
</table>
CICS Initiated Request or Reply Example

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 message 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 on the process flow, refer to the Messages Flow section of TIBCO Substation ES Concepts.

<table>
<thead>
<tr>
<th>Component Summary</th>
<th>For the C language</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SXR3 — CICS transaction Id, the request initiator</td>
</tr>
<tr>
<td></td>
<td>SXC31RR1 — CICS 'C' application program</td>
</tr>
<tr>
<td></td>
<td>SXC-RR-REQUEST — Substation ES trigger identification for initiate request</td>
</tr>
<tr>
<td></td>
<td>SXC-RR-EXTRN-PROCESS — Substation ES recipe identification for generate reply</td>
</tr>
<tr>
<td></td>
<td>SXCCIRR3 — CICS COBOL application program, receives a request from an application via DPL, and replies through the COMMAREA</td>
</tr>
<tr>
<td></td>
<td>SXC-RR-ANSWER — Substation ES recipe identification for receive reply</td>
</tr>
<tr>
<td></td>
<td>SXCREPLY — Substation ES Request or Reply reply module</td>
</tr>
</tbody>
</table>

For COBOL:

|                   | SXRC — CICS transaction Id, the request initiator |
|                   | SXCCIRR1 — CICS COBOL application program |
|                   | SXC-RR-REQUEST — Substation ES trigger identification for initiate request |
|                   | SXC-RR-EXTRN-PROCESS — Substation ES recipe identification for generate reply |
|                   | SXCCIRR3 — CICS COBOL application program, receives a request from an application via DPL, and replies through the COMMAREA |
- **SXCR-RR-ANSWER** — Substation ES recipe identification for receive reply
- **SXCREPLY** — Substation ES Request or Reply reply module

**CICS Execution Command**

COBOL entered from a CICS 3270 screen: SXRC
'C' entered from a CICS 3270 screen: SXR3
ESB Request – Reply IMS Example

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 a 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. The Substation ES picks up the data from there and deliver 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 on the process flow, refer to the Messages Flow section of TIBCO Substation ES Concepts.

Component Summary

- SXIRIJSL / SXIRIRSL - JCL member to submit the request initiator
- SXR3ISL1 / SXJ3ISL1 - IVP Sender Listener sending the request message
- SXI-TRAN - Substation ES recipe identification
- SXICITO1 - IMS transaction identification
- SXICIPO1 - IMS COBOL message processing program

IVP Sender Listener Execution Parameters

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    | 300000                 |
| WAIT       | 0                      |
| BES-DELAY  | 5                      |

EMS

| DEST-REQ   | tibss.IMS.Tran.Request |
| DEST-REP   | tibss.IMS.Tran.Reply   |
| TYPE       | QUEUE                  |
| REPEAT     | 1                      |
| TIMEOUT    | 300000                 |
| WAIT       | 0                      |
ESB Request – Reply and Trigger Example

This example illustrates how a TIBCO messaging client application initiates a request message to invoke an IMS Message Processing Program (MPP) using the IMS message queue via a Substation ES.

The IMS MPP replies by putting information into the IO_AREA and inserts it to the IMS message queue. In addition, the MPP uses the ALT-PCB to insert the output data to the destined Substation ES trigger TPipe. The Substation ES picks up these data and deliver the reply and trigger messages to the IVP Sender Listener. In both directions, the Substation ES Transformer processes the data conversion based on the selected conversion rules in the recipe.

For more information on the process flow, refer to TIBCO Substation ES Concepts.

Component Summary

- SXIRJSL / SXIRIRSL - JCL member to submit the request initiator
- SXR3ISL1 / SXJ3ISL1 - IVP Sender Listener sending of the request message
- SXI-TRAN-AND-TRG - Substation ES recipe identification
- SXICITO2 - IMS transaction identification
- SXICIP02 - IMS COBOL message processing program

IVP Sender Listener Execution Parameters

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

Rendezvous

<table>
<thead>
<tr>
<th>SUBJ-REQ</th>
<th>tibss.IMS.Tran-Trig.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIME-OUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
<tr>
<td>BES-DELAY</td>
<td>5</td>
</tr>
</tbody>
</table>

EMS

<table>
<thead>
<tr>
<th>DEST-REQ</th>
<th>tibss.IMS.Tran-Trig.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REP</td>
<td>tibss.IMS.Tran-Trig.Reply</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>REPEAT</td>
<td>1</td>
</tr>
<tr>
<td>TIME-OUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
</tbody>
</table>
**IMS BMP Trigger Example**

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 a 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. The Substation ES gets the trigger based on the defined search criteria for this Trigger process. Then the Substation ES Transformer processes the data conversions based on 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 on the process flow, refer to the Messages Flow section of *TIBCO Substation ES Concepts*.

<table>
<thead>
<tr>
<th>Component Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>• SXIIVPBT - JCL member to submit the batch request initiator</td>
</tr>
<tr>
<td>• SXICITB1 - IMS transaction identification</td>
</tr>
<tr>
<td>• SXICIPB1 - IMS COBOL batch message program</td>
</tr>
<tr>
<td>• SXI-TRIGGER - Substation ES IMS interface trigger identification</td>
</tr>
</tbody>
</table>

**JCL Execution Parameters**

```
//STEP1BT EXEC IMSBATCH,MBR=SXICIPB1,PSB=SXICIPB1,IMSID=&IMSID, 
//         IN=SXICITB1,APARM='000001SXITPTRGY'
```

or

```
//STEP1BT EXEC IMSBATCH,MBR=SXICIPB1,PSB=SXICIPB1,IMSID=&IMSID, 
//         IN=SXICITB1,APARM=''000001SXITRG$GY''
```
IMS Conversational Example

This example illustrates how a TIBCO messaging transactional client application initiates request messages to perform the IMS conversational functions via a Substation ES. This example uses the IBM IMS INSTALL/IVP sample of a simple phone book application. All functions performed by the simple phone book application can be executed with Substation ES IMS Conversational IVP.

The IBM phone book application must be installed before you can run this IMS Conversational example. Refer to *IMS Installation Vol 1: Installation Verification* for installation requirements for all programs, PSBs, DBDs, and MFSs, and other supporting materials used by the simple phone book application.

The IMS Conversational Sender or Listener is invoked 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. Then the Substation ES delivers the reply to the IMS conversational Sender or Listener. The Substation ES continues the conversation with 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. After the data has been converted, the Substation ES publishes the output message to the IMS Conversational Listener.

For more information on the process flow, refer to the Messages Flow for IMS section of *TIBCO Substation ES Concepts*.

Component Summary

- SXIRICVS / SXIJICVS – JCL member to submit the IMS Conversational Sender or Listener
- SXR3ICV1 / SXJ3ICV1 – ‘C’ TIBCO transactional client application program, initiates and receives conversational request and reply messages
- SXI-CONVERSATIONAL – Substation ES recipe identification
- IVTCB – IBM phone book sample IMS transaction code
- DFSIVP34 – COBOL, IBM phone book application program

When you run these IVP components, the IVP Generic Error Listener should be present and listening to the subject tibss.ibm.ims.cvrs.reply.
IVP IMS Conversational Sender or Listener Execution Parameters

Table 14 describes valid control parameters.

Table 14  IMS Conversational Sender or Listener Control Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-ENTRY</td>
<td>&lt;Operand&gt;,Fields</td>
<td>One message entry for a specify IMS program function in IBM sample phone book application.</td>
</tr>
<tr>
<td>ADD,last name,first name,ext #,zip code</td>
<td></td>
<td>Performs the ADD function to add this new record to the IMS database.</td>
</tr>
<tr>
<td>DEL,last name</td>
<td></td>
<td>Performs the DEL function to delete this record from the IMS database.</td>
</tr>
<tr>
<td>DIS,last name</td>
<td></td>
<td>Performs the DIS function to display this record from the IMS database.</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td>Indicates sending the message to terminate the current transactional process.</td>
</tr>
<tr>
<td>UPD,last name,first name,ext #,zip code</td>
<td></td>
<td>Performs the UPD function to update changes of this record to the IMS database.</td>
</tr>
</tbody>
</table>

The following values are entered through the SYSIN DD statements. These control statements instruct the Sender to publishing 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 expected output from the IMS Conversational IVP can be found in:

Data Set:  <USERHLQ>.DATA
Member:  SXII0CVS
Transform Only Data Examples

These examples illustrate how Substation ES can be used as a tool to transform data from different mainframe sources (databases or files) to TIBCO messaging applications via Substation ES.

These processes support transformations in both directions. Substation ES can additionally be used to easily transformation data from platforms that publishes a TIBCO Rendezvous or TIBCO Enterprise message to a mainframe format (databases or files). When the data has been transformed is can be stored to a permanent medium by a user procedure.

There are two examples:

- **Data Forwarder**: transforms data from the mainframe to a message.
- **Data Receiver**: transforms from a message to the mainframe format.

The **Data Forwarder** example uses a simple TIBCO messaging publisher to read records from a PDS file and publish each record to Substation ES as an *opaque* data type message. The structure of the record need not be known to the application at this point. The Substation ES receives the record, and transforms the data from the *opaque* record to a TIBCO formatted message. The message is published to the IVP Generic Listener. In the outbound direction, the Substation ES Transformer converts the data according to the conversion rule in the recipe.

The **Data Receiver** example receives the outbound data from the Substation ES and writes the input to a file. The data received would have been received by the Substation ES as a message, transformed to an *opaque* and published to the Data Receiver. In the inbound direction, the Substation ES Transformer converts the data according to conversion rule in the recipe.

For both examples, you must define the Substation ES Administrative Interface as the processing system resource (BES) in the recipes. For more information on the process flow, refer to the Messages Flow section of *TIBCO Substation ES Concepts*.

**Component Summary**

For the **Data Forwarder** example:

- SXRIGDF - JCL member to submit the Data Forwarder sender
- SXR3GDF - 'C' Publisher reads data record from a file and sends each record as a message in Opaque data type until end of the file
- <S>.DATA(SXGFL2RV) - contains the data to be forward
- SXS-TRANSFORM-ONLY-FORWARD - Substation ES recipe identification

For the **Data Receiver** example:

- Output messages from Data Receiver example
- SXS-TRANSFORM-ONLY-RECEIVE - Substation ES recipe identification
- SXRIGDR - JCL member to submit the Data Receiver listener
- SXR3GDR - 'C' Subscriber listening each message and write data record into a file.
- <USERHLQ>.DATA(SXGRV2M1) - contains the data received, after the listener has been stopped.

**Data Forwarder and Receiver Execution**

The Data Receiver must be started before the Data Forwarder, as the records that have been forwarded are written to another member when received by the Receiver

*Figure 3 Substation ES Data Only Example Flow.*
The valid control parameters are described in the table below.

**Table 15  Data Transform Only Control Parameters**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBG</td>
<td>{ON</td>
<td>OFF}</td>
</tr>
<tr>
<td>DDNAME</td>
<td>TIBINP01</td>
<td>Specifies the existing library that contains the data members to be use to retrieve or store data from or to.</td>
</tr>
<tr>
<td>FIELD-NAME</td>
<td>DATA</td>
<td>Message field name used by the Forwarder to send the record to the Substation and the name the Receiver gets the record content as.</td>
</tr>
<tr>
<td>MEMBER</td>
<td>{SXGFL2RV</td>
<td>SXGRV2M1}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SXGFL2RV is used as input to Forwarder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SXGRV2M1 is used as output of Receiver</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>See note</td>
<td>Subject name the Forwarder or Receiver uses</td>
</tr>
<tr>
<td>UFLD-PFX</td>
<td>tibss-</td>
<td>Substation ES communication field prefix</td>
</tr>
</tbody>
</table>

Note: These subjects are used by default. Don’t change them unless changing these corresponding recipe definitions in the configuration file.

Forwarder Subject:  tibss.data.input.user  
Receiver Subject:   tibss.data.output.user
Chapter 5  Application Message Communication

This chapter describes how Substation ES communicates with applications using communication fields and notifications.

Topics

- Communication Fields Overview, page 158
- System Fields, page 159
- User Fields, page 163
- Application Notifications, page 165
Communication Fields Overview

Communication fields are named fields contained within a user message that assist in bi-directional Substation ES communication. These fields enhance the ability of an application to effectively forward additional information and receive notifications when conversing with Substation ES. Substation ES communication fields are divided into the following categories:

**Substation ES System Fields**

Substation ES system fields are defined in a user’s message and are used by applications and Substation ES during communication. System fields can be bi-directional. System Fields are not stored within the Substation ES communication buffer passed to a back-end system application. Therefore, Substation ES processes these fields differently than normal message fields.

**Substation ES User Fields**

Substation ES user fields are part of the Application Notification and are Substation ES output-only fields. These fields enable applications to identify information regarding the applications 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.

**Communication Fields Prefix**

You can change the communication fields prefix before Substation ES startup. You change the prefix by specifying a value for the UFLDS-PREFIX keyword found in the SIP member \(<USERHLQ>.CNTL(SXSSIP$1)\).

If no keyword is specified, the default prefix of \(\text{tibss-}\) is used. If you specify a different keyword value, it is advised to specify a dash - after the name. Use the default prefix value unless it is absolutely necessary to enforce a change. When changing a prefix, perform the following tasks:

- Edit all user programs and Substation ES utilities using the communication fields and make the necessary changes. Recompile and or restart if necessary.
- Specify a different prefix in the Substation ES SIP member.
- Change Substation ES data configuration messages that use communication fields.
- Recycle Substation ES to activate fields containing the changed prefix.
System Fields

System fields are not always used. When they are used, they are always contained in the user’s message. Depending on the conversation direction, certain system fields must be specified in the configuration’s message definition in order to ensure that the contents of the fields are correctly processed and relayed to participating interfaces by Substation ES.

Table 16 lists the system fields. The names in the table do not include the default prefix `tibss-`.

Table 16  System Field Names

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>envelope</td>
<td>Substation ES context information. Used during a CICS Request or Reply application communication or during IMS application conversational interaction.</td>
</tr>
<tr>
<td>userid</td>
<td>User identification required by Substation ES and back-end system.</td>
</tr>
<tr>
<td>password</td>
<td>User’s password required by Substation ES and back-end system.</td>
</tr>
<tr>
<td>resource</td>
<td>The dynamically-supplied resource name to be executed in the back-end system in the payload/message.</td>
</tr>
<tr>
<td>BES</td>
<td>The dynamically-supplied name assigned to the Substation ES interface to communicate with the transaction processing systems (such as CICS and IMS) in the payload/message.</td>
</tr>
<tr>
<td>tran-code</td>
<td>The dynamically-supplied user transaction name to be executed in the back-end system in the payload/message (CICS usage only).</td>
</tr>
</tbody>
</table>
System Field Definition Summary

The following are a list of system fields. The field names in the table do not include the default prefix `tibss-`

Table 17 System Field Definitions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Substation Message Direction</th>
<th>Definition and Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>envelope</td>
<td>Output</td>
<td>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 applications reply message. No configuration definition is necessary.</td>
</tr>
<tr>
<td></td>
<td>Input</td>
<td>If a user application receives this field and a reply is required by the origination application, the contents must be returned unchanged in the applications reply message. <strong>Definition:</strong> Define on the input message with data-type <code>Opaque</code>. In the field conversion rule panel, specify <code>Trunc=S</code> and <code>Reqd=Y</code>.</td>
</tr>
<tr>
<td>userid</td>
<td>Input</td>
<td>The user’s application sends this field. <strong>Definition:</strong> Define on the input message with data-type <code>String</code>. In the field conversion rule panel, specify <code>Trunc=S</code> and <code>Reqd=Y</code> or <code>N</code>.</td>
</tr>
<tr>
<td>password</td>
<td>Input</td>
<td>The user’s application sends this field. <strong>Definition:</strong> Define on the input message with data-type <code>String</code>. In the field conversion rule panel, specify <code>Trunc=S</code> and <code>Reqd=Y</code> or <code>N</code>.</td>
</tr>
<tr>
<td>resource</td>
<td>Input</td>
<td>The user’s application sends this field. <strong>Definition:</strong> Define on the input message with data-type <code>String</code>. In the field conversion rule panel, specify <code>Trunc=S</code> and <code>Reqd=Y</code> or <code>N</code>.</td>
</tr>
<tr>
<td>BES</td>
<td>Input</td>
<td>The user’s application sends this field. <strong>Definition:</strong> Define on the input message with data-type <code>String</code>. In the field conversion rule panel, specify <code>Trunc=S</code> and <code>Reqd=Y</code> or <code>N</code>.</td>
</tr>
<tr>
<td>ReplyToName</td>
<td>Output</td>
<td>Specifies the destination name to which Substation ES could not successfully write. Substation ES adds this to the EMS message as a user property when it writes an EMS dead message.</td>
</tr>
<tr>
<td>ReplyToType</td>
<td>Output</td>
<td>Specifies the destination type of the ReplyToName. Substation ES adds this to the EMS message as a user property when it writes an EMS dead message.</td>
</tr>
</tbody>
</table>
System Fields Definition

Defining System Fields on an Inbound Conversion Rule

- An input user application message can contain Substation ES System Fields.
- First, define the required system fields to the message definition, SXC-I0-RR-MS01.
- Second, define the correct attributes on the conversion rule that is used for the appropriate recipe definition.

For further information on these definitions, refer to DCUI panels as described in Chapter 1, Substation ES Transformer Configuration, page 1.

---

--- Define Msg-Field / Buffer-Field Conversion Row 1 to 16 of 16

Command ===>                                                   Scroll ==> CSR

Conversion Id   ===> CVR-I-TSRR-01
Using Message   ...: MSG-I0-RR-01
Using Buffer    ...: BUF-I0-BF01
Max Buffer Size ===> 0

<table>
<thead>
<tr>
<th>Trunc</th>
<th>Req'd</th>
<th>Bffld</th>
<th>Msg Field</th>
<th>Num</th>
<th>Buffer Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y</td>
<td>1</td>
<td>IVP-ID</td>
<td>1</td>
<td>IVP-ID</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>2</td>
<td>IVP-DESCR</td>
<td>2</td>
<td>IVP-DESCR</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>3</td>
<td>IVP-RESULT</td>
<td>3</td>
<td>IVP-RESULT</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>4</td>
<td>IVP-BES-NAME</td>
<td>4</td>
<td>IVP-BES-NAME</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>5</td>
<td>IVP-BES-TRAN-ID</td>
<td>5</td>
<td>IVP-BES-TRAN-ID</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>6</td>
<td>IVP-BES-PGM-NAME</td>
<td>6</td>
<td>IVP-BES-PGM-NAME</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>7</td>
<td>IVP-USERID</td>
<td>7</td>
<td>IVP-USERID</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>9</td>
<td>IVP-REPLY-LILSECS</td>
<td>8</td>
<td>IVP-START-LILSECS</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>8</td>
<td>IVP-START-LILSECS</td>
<td>9</td>
<td>IVP-REPLY-LILSECS</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td>IVP-SUBJ-SFX</td>
<td>10</td>
<td>IVP-BES-DELAY</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>12</td>
<td>IVP-BTCH-SEQ-N0</td>
<td>11</td>
<td>IVP-BTCH-SEQ-N0</td>
</tr>
<tr>
<td>S</td>
<td>Y</td>
<td>0</td>
<td>tibss-envelope</td>
<td>12</td>
<td>IVP-START-DATE-CHAR</td>
</tr>
<tr>
<td>S</td>
<td>N</td>
<td>0</td>
<td>tibss-password</td>
<td>13</td>
<td>IVP-START-TIME-CHAR</td>
</tr>
<tr>
<td>S</td>
<td>N</td>
<td>0</td>
<td>tibss-userid</td>
<td>14</td>
<td>IVP-REPLY-DATE-CHAR</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td></td>
<td>15</td>
<td>IVP-REPLY-TIME-CHAR</td>
</tr>
</tbody>
</table>

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

System fields have an S defined in the Trunc column.
The contents are not passed to the back-end application buffer because they have a 0 defined in the Bffld column. Substation ES passes the contents to the required back-end system process for usage or validation.
**Using Dynamic Transaction IDs**

This section applies to Substation ES 2.5 and greater.

You can dynamically specify a user transaction that a resource is to execute under in CICS. This is done by supplying Substation ES with an input communication field, the message property `tibss-tran-code`, in the TIBCO message.

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 specified in `tibss-tran-code`. The following takes place in CICS:

1. Substation ES starts transaction CSMI to handle EXCI communication.
2. CSMI then switches the Transaction Id to the value of `tibss-tran-code`.
3. Security verification is done on the new Transaction Id.
4. For DB2 - DB2CONN is set up based on new Transaction Id.

Note the following when using dynamic Transaction Ids:

- When `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).

- When a message property field called `tibss-tran-code` and a map message field called `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 may have to redefine CSMI as "above the line" (TASKDATALoc = Any). Since this is CSMI and is used by other processes, you must test this type of change to make sure it does not affect other critical processes.
User Fields

User fields are only provided when the appropriate Reply Method is selected on the Recipe or Trigger Definition panel. When used, they are contained in the user’s message and the Substation ES error message tibss-error-msg. User fields are only provided by the Substation ES on output.

For more information, refer to Application Notifications on page 165.

The following is a list of user fields. The names do not include the default prefix tibss-

status  
When a transformation error, Substation ES or BES error is detected, the status field is added to the output message.
- Only one status field is added to a message. This value of the status field is incremented for each notification written to the tibss-error-msg message.
- When error free, status value is zero (0).

error-msg  
Created when appropriate Reply Method indicator has been set and when a transformation, Substation ES or BES error is detected.

<user field>  
Field name of a message defined to the Substation ES configuration that caused the transformation error. Content is a Substation ES descriptive and documented message.

error-info  
Substation ES warning or error informational field that is a Substation ES descriptive and documented message or a BES message.

ss-cfg-id  
Current recipe or trigger identifier field. This field is always contained in a tibss-error-msg.

ss-id  
Substation ES informational field. This field is always contained this field in a tibss-error-msg.
**User Fields Definitions Summary**

Table 18 lists attributes for user fields. The names in the table do not include the default prefix `tibss-`

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Attribute</th>
<th>Location and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>Int32</td>
<td>Found in the users message segment. Describes whether the interaction with the Substation ES has been successful.</td>
</tr>
<tr>
<td>error-msg</td>
<td>RV Nested Message EMS Map Message</td>
<td>Found in the users message segment. Substation ES Error Message field name</td>
</tr>
<tr>
<td>error-info</td>
<td>String256 char maximum</td>
<td>Found in the nested error message. Information regarding error condition detected by Substation ES.</td>
</tr>
<tr>
<td>&lt;user-field&gt;</td>
<td>String</td>
<td>Found in the nested error message. Field value is the name on the user’s input message defined in the configuration. Field contains information regarding a data transformation error. Only present if there is a data transformation errors.</td>
</tr>
<tr>
<td>cfg-id</td>
<td>String32 char maximum</td>
<td>Found in the nested error message. Substation ES Transformer Recipe or Trigger identification</td>
</tr>
<tr>
<td>ss-id</td>
<td>String17 char maximum</td>
<td>Found in the nested error message. Substation ES Identification used to determine which Substation ES reported the error</td>
</tr>
</tbody>
</table>
Application Notifications

Substation ES Application Notification provides information to applications regarding possible abnormal conditions, such as the following:

- Data transformation errors
- Possible Substation ES run-time failures
- Errors or failures that occur while processing the applications requests

An Application Notification consists of a status field and a nested or sub message. When a TIBCO messaging application is communicating with Substation ES, if there is an abnormality, the TIBCO messaging application receives an Application Notification. The application can log and trace abnormalities and possibly take corrective action.

The Application Notification information is relayed back to the TIBCO messaging application and included with the applications request or reply and error message, depending in which direction the failure of execution was detected.

An Application Notification is included in messages delivered to Publish and Error subjects defined on the recipe and trigger definition panel.

Users can choose whether their external applications receive:

- No status output
- The status field in all circumstances
- The status field only when an error condition exists

The user chooses appropriate Application Notification delivery by specifying the appropriate Reply Method on the Recipe or Trigger definition.

The field names used to identify errors in messages sent to user applications are always defined in the User Field. For attributes of the user fields described here, refer User Fields Definitions Summary on page 164.

TIBCO Applications Notifications Structure

If a user application must receive Notifications, the appropriate Reply Method must be specified for the Recipe or Trigger definition. The communication field tibss-status is used to validate delivery of Notifications.

When an application receives a tibss-status field and the contents are non-zero, the user's message receives an associated nested error message containing the error information.

The format of the notification is as follows:
- Notification Control Fields that are contained within the users received message.
- Notification Message Fields that are contained within a nested message.

Message structure for these fields is described below:

**Figure 4  Application Notification Message Structure**

<table>
<thead>
<tr>
<th>tibss-status</th>
<th>good fd1</th>
<th>good fd2</th>
<th>good fdn</th>
<th>...</th>
<th>tibss-error-msg</th>
</tr>
</thead>
</table>

**Figure 5  Substation ES Error Message Nested Structure**

<table>
<thead>
<tr>
<th>tibss-cfg-id</th>
<th>error fd1</th>
<th>error fdn</th>
<th>...</th>
<th>...</th>
<th>and/or tibss-error-info</th>
</tr>
</thead>
</table>
Appendix A  Sample IVP Resources

A reference to the sample application programs provided by the Substation ES is shown in following tables. Sample programs are written in COBOL and C languages. The source for COBOL language samples can be located in `<USERHLQ>.COB` library and for C language in the `<USERHLQ>.C` library.

Table 19  CICS Interface Sample IVP Invocations of Resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Resource Name and Type</th>
<th>Program Source and Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request or Reply application requesting an invocation of a CICS program via the DPL method.</td>
<td>SXCCI001 (program)</td>
<td>SXCCI001 COBOL</td>
</tr>
<tr>
<td></td>
<td>SXC3I001 (program)</td>
<td>SXC3I001 C</td>
</tr>
<tr>
<td>Request or Reply application requesting an invocation of a CICS program via the DPL TSQ method.</td>
<td>SXCCI012 (program)</td>
<td>SXCCI012 COBOL</td>
</tr>
<tr>
<td></td>
<td>SXC3I012 (program)</td>
<td>SXC3I012 C</td>
</tr>
<tr>
<td>Request or Reply application requesting an invocation of a CICS program via the DPL Storage Pointer method.</td>
<td>SXCCI013 (program)</td>
<td>SXCCI013 COBOL</td>
</tr>
<tr>
<td></td>
<td>SXC3I013 (program)</td>
<td>SXC3I013 C</td>
</tr>
<tr>
<td>Request or Reply application requesting an invocation of a CICS program via the DPL Container method.</td>
<td>SXCCI014 (program)</td>
<td>SXCCI014 COBOL</td>
</tr>
<tr>
<td>Substation ES HVT Trigger message initiated from within CICS.</td>
<td>SXTF (transaction)</td>
<td>SXCCI032 COBOL</td>
</tr>
<tr>
<td></td>
<td>SXT6 (transaction)</td>
<td>SXC3I032 C</td>
</tr>
<tr>
<td>Substation ES Trigger message initiated from within CICS.</td>
<td>SXTTT (transaction)</td>
<td>SXCCI002 COBOL</td>
</tr>
<tr>
<td>Sender application requesting an invocation of a CICS program via the XCTL method.</td>
<td>SXCCI003 (program)</td>
<td>SXCCI003 COBOL</td>
</tr>
</tbody>
</table>
### Table 19  CICS Interface Sample IVP Invocations of Resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Resource Name and Type</th>
<th>Program Source and Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender application requesting that a transaction in the CICS region be initiated.</td>
<td>SXTI (transaction)</td>
<td>SXCCI004 COBOL</td>
</tr>
<tr>
<td>Application publishing a message to the Substation ES requesting that a TDQ entry is written with the message contents.</td>
<td>SXQT (TDQ Name)</td>
<td>Substation ES internal function</td>
</tr>
<tr>
<td>Application publishing a message to the Substation ES requesting that a TSQ entry is written with the message contents.</td>
<td>SXCTSQI1 (TSQ Name)</td>
<td>Substation ES internal function</td>
</tr>
<tr>
<td>Displays the records on the Dead Message Queue, with the ability to delete and resend records in the CICS region.</td>
<td>SXTD (transaction)</td>
<td>SXCCODMQ COBOL</td>
</tr>
<tr>
<td>Application publishes a message and expects a reply from the Substation ES that invokes a CICS program using DPL. Most z/OS COBOL data types are used in this example.</td>
<td>SXCCI010 (program)</td>
<td>SXCCI010 COBOL</td>
</tr>
</tbody>
</table>

### Table 20  CICS Outbound Request or Reply Sample Programs

<table>
<thead>
<tr>
<th>Description</th>
<th>Resource Name and Type</th>
<th>Program Source and Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS initiated application that sends a Request out via Substation ES and expects a Reply within a period of time.</td>
<td>SXRC (transaction)</td>
<td>SXCCIRR1 COBOL</td>
</tr>
<tr>
<td></td>
<td>SXR3 (transaction)</td>
<td>SXC3IRR1 C</td>
</tr>
<tr>
<td>CICS application that processes the request and replies to the CICS initiated Request or Reply transaction.</td>
<td>SXCCIRR3 (program)</td>
<td>SXCCIRR3 COBOL</td>
</tr>
<tr>
<td>Description</td>
<td>Resource Name and Type</td>
<td>Program Source and Language</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Application publishes a message and expects a reply from Substation ES that invokes an IMS Transaction (MPP) via OTMA.</td>
<td>SXICITO1 (transaction)</td>
<td>SXICIPO1 COBOL</td>
</tr>
<tr>
<td>Application publishes a message and expects a reply and a Trigger from Substation ES that invokes an IMS Transaction (MPP) via OTMA</td>
<td>SXICITO2 (transaction)</td>
<td>SXICIPO2 COBOL</td>
</tr>
<tr>
<td>IMS BMP Program sends Trigger messages via Substation ES to an awaiting subscriber.</td>
<td>None</td>
<td>SXICIPB1 COBOL</td>
</tr>
<tr>
<td>IMS conversational</td>
<td>IVTCB (transaction)</td>
<td>SXR3ICV1 C</td>
</tr>
</tbody>
</table>
Appendix B  Transformer Configuration File Utilities

Summary

The following tables list the usage summary of the DD names and parameter combinations for each Transformer Configuration File Utility.

Table 22  Configuration File Utilities DD Names Summary

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Utility Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Batch Copy</td>
</tr>
<tr>
<td>CONFIG</td>
<td>M</td>
</tr>
<tr>
<td>CONFIG2</td>
<td>M</td>
</tr>
<tr>
<td>IMPORT</td>
<td></td>
</tr>
<tr>
<td>EXPORT</td>
<td>M</td>
</tr>
</tbody>
</table>

Table 23  Configuration File Utilities Parameters Summary

<table>
<thead>
<tr>
<th>Parm Value</th>
<th>Utility Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Batch Copy</td>
</tr>
<tr>
<td>-C --&gt; Compress</td>
<td>M</td>
</tr>
<tr>
<td>-Cxxx --&gt; Control DD Name</td>
<td></td>
</tr>
<tr>
<td>-Dn --&gt; Debug</td>
<td>O</td>
</tr>
<tr>
<td>-Ixxx --&gt; Input DD Name</td>
<td>O</td>
</tr>
<tr>
<td>-Mxxx --&gt; Import DD Name</td>
<td></td>
</tr>
<tr>
<td>-Nxxx --&gt; Export DD Name</td>
<td>M</td>
</tr>
</tbody>
</table>
### Parm Value | Utility Type
---|---
-Oxxx --> Output DD Name | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  | O  
-P --> Print | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  | O  | O  | O  
-R --> Read-only | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  | O  | O  | O  
-T --> Initialize | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  
-V --> Validate | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  | O  | O  
-X --> Hex Dump | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  | O  | O  
-Z --> Zero file | Batch Copy  | Batch Import  | Batch Export  | Selective Export  
-O  

**O** Optional parm value for the process  

**M** Required parm value for the process

⚠️ When using the Batch Import utility, do not use the **V** option to import data into an empty configuration file. If you use the **V** option against a Rendezvous message with duplicate field names, you get error SXT7075E.

⚠️ When using the Batch Import utility, do not use the **Z** option to import data into empty configuration that is not empty. Using the **Z** option in this case loses all the existing data in the configuration file.
Appendix C  Conversion Formats from TIBCO Rendezvous

The following tables list the supported TIBCO Rendezvous data types used during data transformation (mapping) performed by TIBCO Substation ES.

### Table 24  TIBCO Rendezvous to Cobol Data Types

<table>
<thead>
<tr>
<th>RV Field</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Max Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRING</td>
<td>Alphanumeric</td>
<td>PIC X(n)</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>OPAQUE</td>
<td>Alphanumeric</td>
<td>PIC X(n)</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>IPPORT16</td>
<td>BINARY</td>
<td>9(4) BINARY</td>
<td>+32767</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9(4) COMP-5</td>
<td>+65535</td>
<td>2</td>
</tr>
<tr>
<td>IPADDR32</td>
<td>Alphanumeric</td>
<td>PIC X(15)</td>
<td>255.255.255.255</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>BINARY</td>
<td>9(9) COMP-5</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S9(10) BINARY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATETIME</td>
<td>C runtime value</td>
<td>Field = 10 digits</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>= 'ctime'</td>
<td>9(9) COMP-5</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>Alphanumeric</td>
<td>PIC X(1)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Numeric</td>
<td>Field =&gt; 2 bytes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 25  TIBCO Rendezvous to Cobol Data Types (numeric fields)

<table>
<thead>
<tr>
<th>RV Field</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Max Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>I8</td>
<td>BINARY</td>
<td>S9(3) BINARY</td>
<td>+127 / -128</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(3) DISPLAY</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>U8</td>
<td>BINARY</td>
<td>9(3) BINARY</td>
<td>+255</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(3) DISPLAY</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>I16</td>
<td>BINARY</td>
<td>S9(4) BINARY</td>
<td>+32767 / -32768</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(5) DISPLAY</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(5) COMP-3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>U16</td>
<td>BINARY</td>
<td>9(4) COMP-5</td>
<td>+65535</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>9(5) DISPLAY</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>9(5) COMP-3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>I32</td>
<td>BINARY</td>
<td>S9(9) BINARY</td>
<td>+2147483647 / -2147483648</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(10) DISPLAY</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(10) COMP-3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>U32</td>
<td>BINARY</td>
<td>9(9) COMP-5</td>
<td>+4294967295</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>9(10) DISPLAY</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>9(10) COMP-3</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>I64</td>
<td>BINARY</td>
<td>S9(18) BINARY</td>
<td>+/- 999,999,999,999,999,999</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(18) DISPLAY</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(18) COMP-3</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>U64</td>
<td>BINARY</td>
<td>9(18) BINARY</td>
<td>+999,999,999,999,999,999</td>
<td>8</td>
</tr>
<tr>
<td>RV Field</td>
<td>Usage</td>
<td>COBOL Picture</td>
<td>Max Value</td>
<td>Size in Bytes</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>------------------</td>
<td>-----------</td>
<td>---------------</td>
</tr>
<tr>
<td>ZONED</td>
<td>9(18) DISPLAY</td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>PACKED</td>
<td>9(18) COMP-3</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>F32</td>
<td>FLOAT</td>
<td>COMP-1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>ZONED</td>
<td>S9(10)V999</td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>PACKED</td>
<td>S9(10)V999 COMP-3</td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>F64</td>
<td>FLOAT</td>
<td>COMP-2</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>ZONED</td>
<td>S9(15)V999 DISPLAY</td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>PACKED</td>
<td>S9(15)V999 COMP-3</td>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>
Appendix D  Conversion Formats from TIBCO EMS

The following tables list the supported TIBCO EMS data types used during data transformation (mapping) performed by Substation ES.

* EMS representation for the field used in the Substation ES message definition panel.

Table 26  TIBCO EMS to Cobol Data Types

<table>
<thead>
<tr>
<th>Substation Msg</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Max Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRING</td>
<td>UTF8</td>
<td>Alphanumeric</td>
<td>PIC X(n)</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>OPAQUE</td>
<td>BYTES</td>
<td>Alphanumeric</td>
<td>PIC X(n)</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>I8</td>
<td>BYTE</td>
<td>Alphanumeric</td>
<td>PIC X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Numeric</td>
<td>Field =&gt; 2 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>BOOL</td>
<td>Alphanumeric</td>
<td>PIC X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Numeric</td>
<td>Field =&gt; 2 bytes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 27  TIBCO EMS to Cobol Data Types (numeric fields)

<table>
<thead>
<tr>
<th>Substation Msg</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Max Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>I8</td>
<td>BYTE</td>
<td>BINARY</td>
<td>S9(3) BINARY</td>
<td>+127 / -128</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZONED</td>
<td>S9(3) DISPLAY</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>I16</td>
<td>SHORT</td>
<td>BINARY</td>
<td>S9(4) BINARY</td>
<td>+32767 / -32768</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZONED</td>
<td>S9(5) DISPLAY</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>
### Table 27  TIBCO EMS to Cobol Data Types (numeric fields)

<table>
<thead>
<tr>
<th>Substation Msg</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Max Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(5) COMP-3</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>BINARY</td>
<td>9(4) COMP-5</td>
<td>+ 32767</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>9(5) DISPLAY</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>9(5) COMP-3</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>I32</td>
<td>INT</td>
<td>BINARY</td>
<td>S9(9) BINARY</td>
<td>+2147483647 / -2147483648</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(10) DISPLAY</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(10) COMP-3</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>BINARY</td>
<td>9(9) COMP-5</td>
<td>+2147483647</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>9(10) DISPLAY</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>9(10) COMP-3</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>U64</td>
<td>LONG</td>
<td>BINARY</td>
<td>S9(18) BINARY</td>
<td>+/- 999,999,999,999,999,999,999</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(18) DISPLAY</td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(18) COMP-3</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>BINARY</td>
<td>9(18) BINARY</td>
<td>+ 999,999,999,999,999,999</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>9(18) DISPLAY</td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>9(18) COMP-3</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>F32</td>
<td>FLOAT</td>
<td>FLOAT</td>
<td>COMP-1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(10)V999</td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(10)V999 COMP-3</td>
<td></td>
<td></td>
<td>7</td>
</tr>
</tbody>
</table>
Table 27  TIBCO EMS to Cobol Data Types (numeric fields)

<table>
<thead>
<tr>
<th>Substation Msg</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Max Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F64</td>
<td>DOUBLE</td>
<td>FLOAT</td>
<td>COMP-2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZONED</td>
<td>S9(15)V999 DISPLAY</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PACKED</td>
<td>S9(15)V999 COMP-3</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

* EMS representation for the field used in the Substation ES message definition panel.
Index

C

changes from the previous release  x
CICS
   Recipe Details Overview  53
   System Resources Panel  44
customer support  xiii

I

IMS
   Processing System Resources Panel  44
   Recipe Details Panel  53
   Recipe List Panel  48
   Select Processing Panel  51
   Trigger Details Panel  66

R

Recipe Details Panel
   IMS  53
Recipe List Panel
   IMS  48

S

support, contacting  xiii

T

technical support  xiii