TIBCO Substation ES™
Configuration and Resources

Software Release 2.7
November 2011
Important Information

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

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

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

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.

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

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

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

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

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

Copyright © 1999-2011 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information
# Contents

**Figures** .................................................................................................................. ix

**Tables** ........................................................................................................................ xi

**Preface** .......................................................................................................................... xiii

<table>
<thead>
<tr>
<th>Changes from the Previous Release of this Guide</th>
<th>xiv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related Documentation</td>
<td>xvi</td>
</tr>
<tr>
<td>TIBCO Substation ES</td>
<td>xvi</td>
</tr>
<tr>
<td>Related TIBCO Products</td>
<td>xvi</td>
</tr>
<tr>
<td>Third-Party Documentation</td>
<td>xvii</td>
</tr>
<tr>
<td>Typographical Conventions</td>
<td>xviii</td>
</tr>
</tbody>
</table>

| Connecting with TIBCO Resources                   | xxii |
| How to Join TIBCOMmunity                          | xxii |
| How to Access All TIBCO Documentation             | xxii |
| How to Contact TIBCO Support                      | xxii |

**Chapter 1  Configuration of Services** .................................................................... 1

| Overview of Data Relationships                    | 2 |
| Definitions of Configuration Metadata             | 3 |
| System                                           | 3 |
| Inbound or Outbound Message Data                  | 3 |
| Metadata                                         | 3 |
| Metadata Relationship                            | 4 |
| Allocation of Configuration File                  | 5 |

| Configuration Guidelines                          | 6 |
| Defining the Back-end System (BES)                | 7 |
| Defining Network Parameters (ESB Endpoint)        | 8 |
| Using the Cobol Copybook Converter                | 10 |
| Defining Messages                                 | 12 |
| Defining Buffers                                  | 13 |
| Defining Conversion Rules                         | 15 |
| Defining Recipes                                  | 16 |
| Defining Triggers                                 | 19 |
| Defining Groups                                   | 21 |

| Transformer DCUI Panels and Usage                 | 22 |
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
Select a Message Definition Panel .................................................... 38
Select a Buffer Definition Panel ......................................................... 39
Define Msg-Field or Buffer-Field Conversion Panel .............................. 40
Define Message Fields to Conversion Rule Panel .................................... 42
Recipes & Triggers using a Conversion Rule Panel ................................ 44
Define Processing System Resources Panel ........................................ 45
Define ESB Endpoints Panel ............................................................ 46
ESB Endpoint Extensions Panel ........................................................ 47
ESB Endpoint – Connection Factory Extensions Panel ............................. 50
ESB Endpoint SSL Extensions Panel ................................................. 52
Recipe List Panel ................................................................. 54
Select ESB Endpoint Panel (Recipe) .................................................... 56
Select Processing System Resources Panel (Recipe) ............................... 57
Select Conversion Rules Panel (Recipe) ................................................. 58
Recipe Details Panel ................................................................. 59
CICS Service Details Panel ............................................................. 65
IMS Service Details Panel .............................................................. 67
EMS Recipe Details Extension Panel .................................................... 68
EMS Properties (Recipe) Panel .......................................................... 72
EMS Selector Specification Panel ....................................................... 74
Define Triggers Panel ................................................................. 75
Select ESB Endpoint Panel (Triggers) .................................................. 76
Select Output Conversion Rule Panel (Triggers) ..................................... 77
Trigger Details Panel ................................................................. 78
EMS Trigger Details Extension Panel .................................................. 81
EMS Properties (Trigger) Panel ........................................................ 84
Group Definitions Panel ............................................................... 86
Select Triggers and Recipes for Group Panel ........................................ 87
Display Configuration File Detail Panel ................................................. 88
Select Desired Entry Panel ............................................................. 89
Summary of Definition Panel (Recipe) .................................................. 90
Summary of Definition Panel (Trigger) ................................................ 97
Summary of Definition Panel (Conversion) .......................................... 102
Summary of Definition Panel (Group) ............................................... 106
Copybook Converter Panel ............................................................ 107
Example of ESB Request – Reply and Trigger .................................................. 173
  Component Summary ................................................................. 173
  Execution Parameters of IVP Sender Listener ................................. 173
Example of IMS BMP Trigger ................................................................. 175
  Component Summary ................................................................. 175
  Execution Parameters of JCL ......................................................... 175
Example of IMS Conversational Functions ........................................... 176
  Component Summary ................................................................. 176
  Execution Parameters of IVP IMS Conversational Sender or Listener .. 177
Examples of Transform-Only Data ......................................................... 178
  Component Summary ................................................................. 178
  Execution of Data Forwarder and Receiver .................................... 179

Chapter 5 Communication of Application Messages ......................... 181
  Communication Fields .................................................................. 182
    Categories .............................................................................. 182
    Prefix ..................................................................................... 182
  System Fields ............................................................................ 183
    Definitions .............................................................................. 183
    Usage ................................................................................. 184
    Defining System Fields on an Inbound Conversion Rule ............... 186
    Dynamic Transaction IDs ......................................................... 187
  User Fields ............................................................................... 188
    Definitions .............................................................................. 188
    Attributes .............................................................................. 188
  Application Notifications ............................................................... 190
    Information Types .................................................................. 190
    Structure .............................................................................. 190

Appendix A Sample IVP Resources ...................................................... 193

Appendix B Summary of Transformer Configuration File Utilities .... 197

Appendix C Conversion Formats from Rendezvous ......................... 199

Appendix D Conversion Formats from TIBCO Enterprise Message Service .... 202

Appendix E BusinessWorks Sample Projects for Substation ES — IMS Project ..... 205

Appendix F BusinessWorks Sample Projects for Substation ES — CICS Project .... 207
## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Relationship of Conversion Rule With Message Definition and Buffer Definition.</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Recipe and Trigger Components</td>
<td>4</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Example of Substation ES Data Flow</td>
<td>179</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Structure of Application Notification Message</td>
<td>191</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Nested Structure of Substation ES Error Message</td>
<td>191</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Structure of Substation ES Error Messages as Properties Fields</td>
<td>191</td>
</tr>
</tbody>
</table>
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>General Typographical Conventions</td>
<td>xviii</td>
</tr>
<tr>
<td>Table 2</td>
<td>Syntax Typographical Conventions</td>
<td>xix</td>
</tr>
<tr>
<td>Table 3</td>
<td>Configuration File Conversion Utility DDNames.</td>
<td>130</td>
</tr>
<tr>
<td>Table 4</td>
<td>Transformer Conversion Utility Parameters</td>
<td>130</td>
</tr>
<tr>
<td>Table 5</td>
<td>DD Names of Configuration File Utility.</td>
<td>133</td>
</tr>
<tr>
<td>Table 6</td>
<td>Function Parameters of Configuration File Utility</td>
<td>133</td>
</tr>
<tr>
<td>Table 7</td>
<td>Configuration File Utility Parameters</td>
<td>134</td>
</tr>
<tr>
<td>Table 8</td>
<td>DD Names of Selective Export Utility</td>
<td>136</td>
</tr>
<tr>
<td>Table 9</td>
<td>Selective Export Utility Parameters</td>
<td>136</td>
</tr>
<tr>
<td>Table 10</td>
<td>Entity Type Values</td>
<td>137</td>
</tr>
<tr>
<td>Table 11</td>
<td>Definition Qualifier Values</td>
<td>138</td>
</tr>
<tr>
<td>Table 12</td>
<td>Rendezvous IVP Sender Listener Control Parameters</td>
<td>145</td>
</tr>
<tr>
<td>Table 13</td>
<td>Control Parameters for EMS IVP Sender Listener</td>
<td>147</td>
</tr>
<tr>
<td>Table 14</td>
<td>IVP Request or Reply EMS Consumer Producer Parameters</td>
<td>150</td>
</tr>
<tr>
<td>Table 15</td>
<td>Control Parameters for IVP Generic Error Listener</td>
<td>154</td>
</tr>
<tr>
<td>Table 16</td>
<td>IMS Conversational Sender or Listener Control Parameters</td>
<td>177</td>
</tr>
<tr>
<td>Table 17</td>
<td>Data Transform Only Control Parameters</td>
<td>180</td>
</tr>
<tr>
<td>Table 18</td>
<td>System Fields</td>
<td>183</td>
</tr>
<tr>
<td>Table 19</td>
<td>Definitions of System Fields</td>
<td>184</td>
</tr>
<tr>
<td>Table 20</td>
<td>User Field Attribute Summary</td>
<td>189</td>
</tr>
<tr>
<td>Table 21</td>
<td>Sample IVP Programs for Invoking Resources for the CICS Interface</td>
<td>193</td>
</tr>
<tr>
<td>Table 22</td>
<td>Sample Programs for CICS Outbound Requests or Replies</td>
<td>194</td>
</tr>
<tr>
<td>Table 23</td>
<td>Sample IVP Programs for the IMS Interface</td>
<td>195</td>
</tr>
<tr>
<td>Table 24</td>
<td>Summary of Configuration File Utility: DD Names</td>
<td>197</td>
</tr>
<tr>
<td>Table 25</td>
<td>Summary of Configuration File Utility: Parameters</td>
<td>197</td>
</tr>
<tr>
<td>Table 26</td>
<td>RV-to-COBOL Data Types</td>
<td>199</td>
</tr>
<tr>
<td>Table 27</td>
<td>RV-to-COBOL Data Types (Numeric Fields)</td>
<td>199</td>
</tr>
<tr>
<td>Table 28</td>
<td>EMS-to-COBOL Data Types</td>
<td>202</td>
</tr>
<tr>
<td>Table 29</td>
<td>EMS-to-COBOL Data Types (Numeric Fields)</td>
<td>202</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Table 30</td>
<td>TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on IMS.</td>
<td>205</td>
</tr>
<tr>
<td>Table 31</td>
<td>TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on CICS</td>
<td>207</td>
</tr>
</tbody>
</table>
Preface

TIBCO Substation ES™ subscribes to and publishes data to TIBCO applications and transaction processing systems, such as Customer Information Control System (CICS) and Information Management System (IMS), which run in the z/OS operating-system environment.

This manual shows you how to configure TIBCO Substation ES. Sample code accompanies the procedures as examples.

Topics

- Changes from the Previous Release of this Guide, page xiv
- Related Documentation, page xvi
- Typographical Conventions, page xviii
- Connecting with TIBCO Resources, page xxi
Changes from the Previous Release of this Guide

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

SSL Parameter Definitions

All parameters related to SSL must now be defined in the ESB Endpoint SSL Extensions panel. Previously, SSL was implemented at the interface level. All applications or recipes or triggers using SSL were required to add an Enterprise Service Bus (ESB) interface that had SSL enabled. See ESB Endpoint SSL Extensions Panel on page 52 for more information.

New Panels

These panels are new to this release:

- ESB Endpoint – Connection Factory Extensions Panel
- ESB Endpoint SSL Extensions Panel
- EMS Properties (Recipe) Panel
- EMS Properties (Trigger) Panel
- CICS Service Details Panel
- IMS Service Details Panel

Renamed Panels

These panels have been renamed:

<table>
<thead>
<tr>
<th>Former Panel Name</th>
<th>New Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Transport Entity Panel</td>
<td>Define ESB Endpoints Panel</td>
</tr>
<tr>
<td>Transport Entity Extensions Panel</td>
<td>ESB Endpoint Extensions Panel</td>
</tr>
<tr>
<td>Select Transport Panel (Recipe)</td>
<td>Select ESB Endpoint Panel (Recipe)</td>
</tr>
<tr>
<td>Select Transport Panel (Triggers)</td>
<td>Select ESB Endpoint Panel (Triggers)</td>
</tr>
<tr>
<td>EMS Properties Panel</td>
<td>This panel has been split into two separate panels:</td>
</tr>
<tr>
<td></td>
<td>- EMS Properties (Recipe) Panel</td>
</tr>
<tr>
<td></td>
<td>- EMS Properties (Trigger) Panel</td>
</tr>
</tbody>
</table>
New Topics

Two new topics are covered:

- **Using RVDQ with Multiple Substation ES Instances on page 121** documents how to configure multiple Substation ES instances to use RV distributed queue processing in order to communicate with a single CICS region.

- **Using Opaque Field and Message Size on page 125** documents the process and conditions for using Opaque field and message size settings with the RV ESB.

- **Appendix E, BusinessWorks Sample Projects for Substation ES — IMS Project, on page 205**, describes these sample projects that are included with your Substation ES installation.
Related Documentation

This section lists documentation resources you may find useful.

TIBCO Substation ES

The following documents form the Substation ES documentation set:

- **TIBCO Substation ES Concepts**: Read this manual for the background concepts.
- **TIBCO Substation ES Installation**: Read this manual for the procedures on site preparation and installation.
- **TIBCO Substation ES Operations and Administration**: Read this manual for details on the operations and administrative tasks.
- **TIBCO Substation ES Configuration and Resources**: Read this manual for the procedures on configuring communications and data conversions along with sample programs.
- **TIBCO Substation ES Messages and Codes**: Refer to this manual for the error and information messages, listed by code.
- **TIBCO Substation ES Release Notes**: Read this document for information on the new features, deprecated features, and closed and known issues.

Related TIBCO Products

You might find the documentation for the following related TIBCO products helpful:

- **TIBCO Rendezvous™ and TIBCO Enterprise Message Service™ software**: These are TIBCO Software’s real-time transport layers used by Substation ES.
- **TIBCO Rendezvous for z/OS Installation and Configuration**: Read this manual for the procedures 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 to learn how to use TIBCO Rendezvous for z/OS COBOL and C APIs.
- **TIBCO Enterprise Message Service User’s Guide**: Read this manual for the TIBCO Enterprise Message Service capabilities.
- **TIBCO Enterprise Message Service C & COBOL API Reference**: Read this manual to learn how to use Enterprise Message Service C and COBOL APIs.
Third-Party Documentation

You might also find the following IBM documents useful:

- IBM CICS External Interfaces Guide
- IBM CICS System Definition Guide
- IBM CICS Transaction Server for z/OS CICS RACF Security Guide
- IBM IMS/ESA Administration Guide: System
- IBM IMS/ESA Customization Guide
- IBM IMS/ESA Installation Volume 1: Installation and Verification
- IBM IMS/ESA Installation Volume 2: System Definition and Tailoring
- IBM ISPF Dialog Developers Guide and Reference
- IBM RACF User’s Guide
- IBM SecureWay Security Server RACF Security Administration Guide

Many of the IBM documents are in these CD-ROM sets:

- Online Library Omnibus Edition MVS Collection
- Online Library Omnibus Edition z/OS Collection
Typographical Conventions

The following typographical conventions are used in this manual.

Table 1  General Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ENV_NAME</code></td>
<td>TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments. An installation environment consists of the following properties:</td>
</tr>
<tr>
<td><code>TIBCO_HOME</code></td>
<td>• <strong>Name</strong> Identifies the installation environment. This name is referenced in documentation as <code>ENV_NAME</code>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start &gt; All Programs menu.</td>
</tr>
<tr>
<td><code>TIBCO_HOME</code></td>
<td>• <strong>Path</strong> The folder into which the product is installed. This folder is referenced in documentation as <code>TIBCO_HOME</code>.</td>
</tr>
</tbody>
</table>

- **code font**
  - Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:
    - Use `MyCommand` to start the foo process.

- **bold code font**
  - Bold code font is used in the following ways:
    - In procedures, to indicate what a user types. For example: Type `admin`.
    - In large code samples, to indicate the parts of the sample that are of particular interest.
    - In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled:
      - `MyCommand [enable | disable]`
### Table 1  General Typographical Conventions (Cont’d)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>italic font</em></td>
<td>Italic font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• To indicate a document title. For example: See <em>TIBCO ActiveMatrix BusinessWorks Concepts</em>.</td>
</tr>
<tr>
<td></td>
<td>• To introduce new terms For example: A portal page may contain several portlets. <em>Portlets</em> are mini-applications that run in a portal.</td>
</tr>
<tr>
<td></td>
<td>• To indicate a variable in a command or code syntax that you must replace. For example: <em>MyCommand PathName</em></td>
</tr>
</tbody>
</table>

| Key combinations | Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. |
|                 | Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q. |

- The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.

- The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.

- The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

### Table 2  Syntax Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
<td>An optional item in a command or code syntax. For example:</td>
</tr>
<tr>
<td></td>
<td><em>MyCommand [optional_parameter] required_parameter</em></td>
</tr>
</tbody>
</table>

| | A logical OR that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: |
| | *MyCommand para1 | param2 | param3* |
Table 2  Syntax Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ }</td>
<td>A logical group of items in a command. Other syntax notations may appear within each logical group. For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4. MyCommand {param1 param2}</td>
</tr>
</tbody>
</table>
Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access All TIBCO Documentation

After you join TIBCOmmunity, you can access the documentation for all supported product versions here:

http://docs.tibco.com/TibcoDoc

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  

**Configuration of Services**

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

**Topics**

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

Substation ES subscribes to and publishes TIBCO messages on behalf of CICS and IMS transaction processing applications that run 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.

Transformation of 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 the *TIBCO Substation ES Concepts* manual.

Transformation of 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 enables 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.
Definitions of Configuration Metadata

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, which are stored in a file that is read by Substation ES, 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 ESB endpoints, ports, daemons, or TIBCO Enterprise Message Service servers and IP networks.

**Inbound or Outbound Message Data**

Inbound or Outbound Message Data definitions consist of the following:

- **Message** – Defines inbound Rendezvous or TIBCO Enterprise Message Service messages and their data fields (name and value pairs).
- **Buffer** – Defines outbound transaction processing application buffers and their data fields (name and value pairs).

**Metadata**

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 achieve customized routing goals.

**Metadata Relationship**

Figure 1 illustrates a conversion rule definition and its relationship with a message definition and a buffer definition:

*Figure 1  Relationship of Conversion Rule With Message Definition and Buffer Definition*

![Diagram showing the relationship between a conversion rule and message and buffer definitions]

Figure 2 illustrates a recipe and trigger and their components.

*Figure 2  Recipe and Trigger Components*
Allocation of Configuration File

The File Creation JCL creates a new version of the Substation ES Transformer Configuration file for use 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.

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.
- The first user who opens the configuration file for edit has update authority.
- The second and subsequent users have READONLY access.

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

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

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

The following examples serve as guidelines for configuring the Substation ES Transformer.

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

Complete the following steps to define or modify the processing system resources or BES to be used with the recipe definition or trigger definition.

You need not consider any dependencies before performing 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 45.

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 16 characters to name the processing system resource Identifier.

5. Enter up to 8 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 exiting the panel.

7. Exit the Define Processing System Resources Panel by doing either of the following:
   — Pressing PFK3.
   — Entering END on the command line and pressing the ENTER key.

8. Review the upper right hand corner to verify that your changes have been saved.

After you have exited back to the Transformer Configuration Panel, the screen displays the message Changes Saved. If you canceled with the CAN command, the screen displays the message Changes Cancelled. If you made a wrong selection while editing the panel, the screen displays a related error message in the upper right hand corner. If the file was opened read-only and you edited it, the message is Change and Readonly.
Defining Network Parameters (ESB Endpoint)

Complete the following steps to define or modify an ESB definition for a recipe or trigger definition. See the *TIBCO Rendezvous Administration Guide* or the TIBCO Enterprise Message Service documentation for details and the default settings for the parameters in this procedure.

You need not consider any dependencies before performing these steps.

1. Select option 5 —Define ESB Endpoints and Parameters from the Transformer Configuration Panel.
   
   For details on this panel, see Define ESB Endpoints Panel on page 46.

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

3. Set the field values for defining the ESB.

   For TIBCO Enterprise Message Service:

   — In the **Type** field, specify `E` for TIBCO Enterprise Message Service.

   — In the **Startup** field, specify (`Y` or `N`) whether this ESB starts during the initialization of Substation ES.

   — In the **URL** field, specify the host name or host IP address used by Substation ES to find the TIBCO Enterprise Message Service server and establish communication (the TCP port can be defined here or in the **Port** field).

   — In the **Port** field, specify (if applicable) the value of the TCP port used by the Substation ES ESB.

   — In the **Conn User** field, specify (if applicable) a user ID if TIBCO Enterprise Message Service requires a specific ID.

   — In the **Conn Pswd** field, specify (if applicable) a password if TIBCO Enterprise Message Service requires a specific password.

   — In the **Alt. Conn** and **Alt. Port** fields, specify (if applicable) the alternate URL and port for fault-tolerant TIBCO Enterprise Message Service servers.
For Rendezvous:
— In the Type field, specify R for Rendezvous.
— In the Startup field, specify (Y or N) whether this ESB starts during the initialization of Substation ES.
— In the URL field, specify (if applicable) the host name or host IP address that Substation ES uses to find the Rendezvous daemon and establish communication.
— In the Port field, specify the daemon port if needed. The default is 7500.
— In the Service field, specify the UDP service port.
— In the Network field, specify (if applicable) the network IP address.
— In the Multicast field, specify (Y or N) whether this ESB is to use multicast addressing.
— In the Multicast Address field, specify (if applicable) the multicast addresses. This field must be specified if the Multicast field is set to Y.

4. Press the ENTER key before exiting the panel.

5. Exit the ESB Endpoint Extensions Panel by doing either of the following:
— Pressing PFK3.
— Entering END on the command line and pressing the ENTER key.

6. Review the display on the upper right corner to verify that your changes have been saved.

When you return to the Define ESB Endpoint Panel, you can change the identifier for this new ESB by overtyping the value in the Identifier field.
Using the Cobol Copybook Converter

The Copybook Converter can read a COBOL copybook and create configuration definitions for the Substation ES Transformer.

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, you can also define a conversion rule that creates a one-to-one mapping for data conversion.

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

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

1. Select option 10 - COBOL Copybook Converter.
   For details on this panel, see Copybook Converter Panel on page 107.
2. Enter the COBOL copybook member name to use for conversion in Copy Member.
3. Enter up to 16 characters as a name that identifies the buffer definition in Buffer Identifier.
4. Enter up to 16 characters as a name that identifies the message definition in Message Identifier.
5. Enter up to 16 characters as a name that identifies the conversion rule definition in Convert Rule Id.
6. Enter the data source name (DSN) of the partitioned data set (PDS) that contains the copybook member in Library containing Copy Member.
7. Enter the DSNs of the PDS that can contain other copybooks that are included within the copybook member in Additional libraries for COPY.
8. Press the ENTER key before leaving the panel.
9. Exit the Copybook Converter Panel by doing either of the following:
   — Pressing PFK3.
   — Entering END on the command line and pressing the ENTER key.
10. Review the display on the upper right corner to verify that your changes have been saved.
In cases where features not supported by the converter are required for the definitions, you must edit the definitions to change the default attributes, deselect the mapping selections, and apply the changes.

Note the following behavior of the converter:

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

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

You need not consider any dependencies before performing these steps.

1. Select option 1 — Define Message Contents Panel.
   For details on this panel, see Message Definitions Panel on page 25.
2. Type either of the following:
   — ADD on the command line and press the ENTER key to create a new entry.
   — S for Sel to choose an existing definition to modify, and press the ENTER key.
3. Enter up to 16 characters to name the message definition in Message Id.
4. Enter a valid data type for Type.
5. Enter (if applicable) the implied number of digits after a decimal point for Dec.
6. Enter up to 54 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 doing either of the following:
   — Pressing PFK3.
   — Entering END on the command line and press the ENTER key.
9. Review the display on the upper right corner to verify that your changes have been saved.
Defining Buffers

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

You need not consider any dependencies before starting these steps.

1. Select option 2 — Define Buffer Areas Panel. For details on this panel, see Buffer Definitions Panel on page 30.
2. Type either of the following:
   — ADD on the command line and press the ENTER key to create a new entry.
   — S for Sel to choose an existing definition to modify. Press the ENTER key.
3. Enter up to 16 characters to name the buffer definition in Buffer ID.
4. Optionally, enter 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 nine characters but is 4 bytes in length.
8. Enter (if applicable) the implied number of digits after a decimal point in Dec.
9. Enter up to 45 characters to name the buffer field in Name.
10. Enter S under Sel, and press the ENTER key. You see the Buffer Field Details Panel. For details on this panel, see Buffer Field Details Panel on page 33.
11. Enter (if applicable) the character to use for padding purposes in Pad Character.
12. Enter (if applicable) Y for the sign selection in Separate Sign.
13. Enter (if 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 (if applicable) the value for the length of the Initial Value field in Initial Value Length.
16. Press the **ENTER** key before exiting the panel.

17. Exit the Define Buffer Areas Panels by doing either of the following:
   - Pressing **PFK3**.
   - Entering **END** on the command line and pressing the **ENTER** key.

18. Review the display on the upper right corner to verify that your changes have been 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 your first completing the message definition and the buffer definition.

1. Select option 3 — Define Conversion Relationships Panel.
   For details on this panel, see Conversion Rules Panel on page 36.

2. Type either of the following:
   — ADD on the command line and press the ENTER key to create a new entry.
   — 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 details on this panel, see Select a Message Definition Panel on page 38.

4. Under Sel, enter S next to the buffer definition you want to use and press the ENTER key.
   For details on this panel, see Select a Buffer Definition Panel on page 39.

5. Enter up to 16 characters to name the conversion rule definition in Conversion ID.

6. Enter (if 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 exiting the panel.

11. Exit the Define Conversion Rules Panels by doing either of the following:
   — Pressing PFK3.
   — Entering END on the command line and pressing the ENTER key.

12. Review the display on the upper right corner to verify that your changes have been saved and that no error messages are displayed.
Defining Recipes

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

Be sure to define the following before starting the steps:
- ESB (network)
- BES
- Conversion rule

1. Select option 6 — Define Recipes – Inbound/Outbound/Destinations/BES.
   For details on this panel, see Recipe List Panel on page 54.

2. Type either of the following:
   - ADD on the command line and press the ENTER key to create a new entry.
   - S for Sel to choose an existing definition to modify, and press the ENTER key. Enter E for Sel to set TIBCO Enterprise Message Service-specific settings.

3. Under Sel, enter S to select the ESB you want to use and press the ENTER key to continue the process.
   For details on this panel, see Select ESB Endpoint Panel (Recipe) on page 56

4. Under Sel, enter S to select a processing system definition and press the ENTER key to continue the process.
   For details on this panel, see Select Processing System Resources Panel (Recipe) on page 57

5. Under Sel, enter I, O, or B next to a conversion-rule definition and press the ENTER key to continue the process.
   - Select B to use the same Rule for both input and output.
   - Select I to specify a Rule for input, then repeat the process selecting O to specify a different rule for output.
   For details on this panel, see Select Conversion Rules Panel (Recipe) on page 58.

6. Enter up to 31 characters to name the Recipe Identifier.

7. Enter the invoke subject name for which Substation ES listens to invoke a given transformation in Subject.

8. Enter (if applicable) the outbound subject name that Substation ES uses to publish the reply message for this process in Publish Subject.
9. Enter (if applicable) the number of bytes in the buffer to be used for a dynamic reply subject name under **Dynamic Subject Leng**.

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

11. Enter (if applicable) the maximum number of concurrent messages to be processed by an ESB for this recipe in **Usage Limit**. To specify no limit, enter 0.

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

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

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

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

16. Press the **ENTER** key before exiting the panel.

17. Enter (if applicable) **CICS** on the command line and press the **ENTER** key to set CICS Service Details.

   For details on this panel, see **CICS Service Details Panel on page 65**.

18. Press the **ENTER** key before pressing **PFK3** to exit the CICS Service Details Panel.

19. Enter (if applicable) **IMS** on the command line and press the **ENTER** key to set IMS Service Details.

   For details on this panel, see **IMS Service Details Panel on page 67**.

20. Press the **ENTER** key before pressing **PFK3** to exit the IMS Service Details Panel.

21. Enter (if applicable) **EMS** on the command line and press the **ENTER** key to set EMS Recipe Details Extension.

   For details on this panel, see **EMS Recipe Details Extension Panel on page 68**.

22. Press the **ENTER** key before pressing **PFK3** to exit the EMS Recipe Details Extension Panel.

23. Enter (if applicable) **PROP** on the command line and press the **ENTER** key to set EMS Properties.

   For details on this panel, see **EMS Properties (Recipe) Panel on page 72**.

24. Press the **ENTER** key before pressing **PFK3** to exit the EMS Properties Panel.
25. Enter (if applicable) SEL on the command line and press the ENTER key to set the EMS Selector Specification.

For details on this panel, see EMS Selector Specification Panel on page 74.

26. Press the ENTER key before pressing PFK3 to exit the EMS Selector Specification Panel.

27. Exit the Define Recipes Panel by doing either of the following:
   — Pressing PFK3.
   — Entering END on the command line and pressing the ENTER key.

28. Review the display on the upper right corner to verify that your changes have been saved.
**Defining Triggers**

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

Be sure to define the ESB (network) and conversion rules before starting the steps.

1. Select option 7 — Define Triggers – Outbound Convert/Destinations.
   For details on this panel, see Define Triggers Panel on page 75.

2. Type either of the following:
   — ADD on the command line and press the ENTER key to create a new entry.
   — S for Sel to choose an existing definition to modify, and press the ENTER key. Enter E for Sel to set TIBCO Enterprise Message Service-specific settings.

3. Under Sel, enter S next to the ESB definition you want to use and press the ENTER key to continue the process.
   For details on this panel, see Select ESB Endpoint Panel (Triggers) on page 76.

4. Under Sel, enter S or O next to a conversion rule definition and press the ENTER key to continue the process.
   For details on this panel, see Select Output Conversion Rule Panel (Triggers) on page 77.

5. Enter up to 31 characters to name the Trigger Identifier.

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

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

8. Enter (if applicable) the starting offset in the buffer for the dynamic reply subject name 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 Output 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 exiting the panel.

15. Enter (if applicable) **EMS** on the command line and press the **ENTER** key to set EMS Trigger Details Extension.
   
   For details on this panel, see **EMS Trigger Details Extension Panel on page 81**.

16. Press the **ENTER** key before pressing the **PFK3** to exit the EMS Trigger Details Extension Panel.

17. Exit the Define Triggers Panels by doing either of the following:
   
   — Pressing **PFK3**.
   
   — Entering **END** on the command line and pressing the **ENTER** key.

18. Review the display on the upper right corner to verify that your changes have been saved.
Defining Groups

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

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.

Be sure to first define the recipes and triggers before starting the steps.

1. Select option 8 — Define Groups.
   For details on this panel, see Group Definitions Panel on page 86.
2. Type either of the following:
   — **ADD** on the command line and press the **ENTER** key to create a new entry.
   — **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 details on this panel, see Select Triggers and Recipes for Group Panel on page 87.
4. Press the **ENTER** key before exiting the panel.
5. Exit the Define Groups Panel by doing either of the following:
   — Pressing **PFK3**.
   — Entering **END** on the command line and pressing the **ENTER** key.
6. Review the display on the upper right corner to verify that your changes have been saved.
Transformer DCUI Panels and Usage

This section describes how to access and use the Transformer DCUI Panels.

Transformer DCUI Panels

To access the Transformer DCUI Panels, do the following:

1. Go to the ISPF Command Shell (Option 6).
2. Run the following command:
   ```
   ex 'USERHLQ.CLIST(SXGSSPNL)' 
   ```
   where `USERHLQ` is 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. `PANELID` 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  n n n
S    u  u  bbbb   s     tt   aaaa   tt    ii   o  o  n n n
SS  u  u  b  b    ss   tt   a  a   tt    ii   o  o  nn
SSSS  uuuu  bbbb  ssss   tt   a  a   tt   iiii  oooo  n  nn
((   EEEE  SSSS   ))
((    EE     SSSSSSS      ))
((    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 2.7

SELECT OPTION ===>=

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

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

Enter END command to terminate.

Field Descriptions

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

1  Define Message Contents
2  Define Buffer Areas
3  Define Conversion Relationships
4  Define Processing System Resources
5  Define ESB Endpoints and Parameters
6 Define Recipes – Inbound 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 file name 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) and N(o). Default: N.

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

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

You define a message or revise the definition in the Message Definitions Panel.

Panel Layout

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

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

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

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

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), and 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. This definition is required and the field must not be blank.

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

You modify the fields of a message’s definition in the Define Fields in a Message Panel.

### 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>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

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

### 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**elete, and **R**epro.

- **Type**  
The data type for the field.
The valid types for TIBCO Enterprise Message Service and Rendezvous are as follows:

- **OPA** – RV: Represents an opaque byte sequence.
  EMS: Represents bytes.
- **STR** – RV: Represents a character string.
  EMS: Represents an EMS/JMS UTF8 data type.
- **BOL** – Represents Boolean.
- **I8** – RV: Represents an 8-bit integer.
  EMS/JMS: Represents a byte.
- **I16** – RV: Represents a 16-bit integer.
  EMS: Represents an EMS/JMS short data type.
- **U16** – RV: Represents an unsigned 16-bit integer.
  EMS: Represents the wchar data type.
- **I32** – RV: Represents a 32-bit integer.
  EMS: Represents an EMS/JMS integer data type.
- **U64** – RV: Represents an unsigned 64-bit integer.
  EMS: Represents an EMS/JMS long data type.
- **F32** – Represents a 32-bit floating point.
- **F64** – RV: Represents a 64-bit floating point.
  EMS: Represents an EMS/JMS double data type.
- **MSG** – RV: Represents a message.
  EMS: Represents an EMS-mapped message.
- **DAT** – Represents the Rendezvous date-time.
- **U8** – Represents an unsigned 8-bit integer.
- **U32** – Represents an unsigned 32-bit integer.
- **I64** – Represents a 64-bit integer.
- **POR** – Represents a 2-byte IP port.
- **IPA** – Represents a 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 nonzero field identifiers must be unique within each message. You must *not* add a field that contains both a NULL field name and a nonzero field identifier. This field is not implemented in the current release.
**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. This definition is required and the field must not be blank.
Conversion Rules using Message Definition Panel

The Conversion Rules using Message Definition 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. This definition is required and the field must not 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**

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

**Panel Layout**

```
SXTP02 ----------------------- Buffer Definitions ------------ Row 1 to 1 of 1
Command ===>
(ADD=Create new Buffer, CAN=Cancel updates)

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), and 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. This definition is required and the field must not be blank.

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

You modify the fields of a buffer’s definition in the Define Fields in a Buffer Panel.

Panel Layout

| SXTP02F ------------------ 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>
<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: 0 to 4194304.
  If the value is 0, the size of the buffer is determined by the field definitions. Otherwise, this value overrides the determination from the fields. If specified, the value must be greater than or equal to the field determination.

**Sel**
Modifies a previously-defined field in a buffer definition.
Valid selections: S(elect) D(lete), and R(epro).

**Type**

The type of mainframe data expected by the invoked application for the field. This definition is required and the field must not be blank.

The valid types are as follows:

- **Str** – Represents the variable-length CHARACTER data string. Always contains a null terminator at the end (as ALPHANUMERIC, NUMERIC and ALPHABETIC).
- **Text** – Represents the fixed-length CHARACTER data string (as ALPHANUMERIC, NUMERIC, and ALPHABETIC).
- **Pdec** – Represents the PACKED-DECIMAL or COMP-3 data format.
- **Updec** – Represents the UNSIGNED PACKED-DECIMAL or COMP-3 data format.
- **Zdec** – Represents the EXTERNAL DECIMAL data format (as USAGE or DISPLAY).
- **Uzdec** – Represents the UNSIGNED ZONED DECIMAL data format.
- **Ubin** – Represents the UNSIGNED BINARY, COMP or COMP-5 data format.
- **Bin** – Represents the SIGNED BINARY or COMP data format.
- **Float** – Represents the FLOATING-POINT data format (as COMP-1 and COMP-2).
- **Opa** – Represents 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 positions the data at a given byte position.

**Len**

The maximum size of the field represented in storage or memory. This size is required and must be greater than 0. The valid range is 0 to 4194304.

**Dec**

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

**Name**

The user-defined name for the field. This definition is required and the field must not be blank.
Buffer Field Details Panel

You modify the details of a field in a buffer’s definition in the Buffer Field Details Panel.

Panel Layout

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

      Field Name ===> string field

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

Press END to save

Field Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>The name of the field.</td>
</tr>
<tr>
<td>Field Type</td>
<td>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.</td>
</tr>
<tr>
<td>Buffer Start</td>
<td>The number of starting bytes that the field is offset in the buffer. The buffer offset positions the data at a given byte position.</td>
</tr>
<tr>
<td>Buffer Length</td>
<td>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 / Buffer-Field Conversion Panel, the transformation fails.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Decimal Places</strong></td>
<td>The implied number of digits after the decimal point. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td><strong>Pad Character</strong></td>
<td>The character to use for padding. You can use keyboard-accessible characters, which are at the beginning or the end according to the justify configuration. Default: space.</td>
</tr>
<tr>
<td><strong>or hex( )</strong></td>
<td>The fill character defined as a hex value instead of an alpha value.</td>
</tr>
<tr>
<td><strong>Separate Sign</strong></td>
<td>Indicates whether a byte is reserved for the sign or the position of the sign. Valid selections: N(one) and Y(es). Default: N.</td>
</tr>
<tr>
<td><strong>Leading Sign</strong></td>
<td>The field contains a leading sign (as opposed to a trailing sign). Valid selections: N(one) and Y(es). Default: N.</td>
</tr>
<tr>
<td><strong>Justify Right</strong></td>
<td>The positioning of the string. You can fill in the data on either side with padding characters, as appropriate. Valid selections: Y(es) and N(o). Default: N.</td>
</tr>
<tr>
<td><strong>Initial Value</strong></td>
<td>The value or characters that fill in the field if the data is not available in the buffer.</td>
</tr>
<tr>
<td><strong>Initial Value Length</strong></td>
<td>The length of the Initial Value field when used. Setting initial values is only for mapped buffer fields, for example, fields that have corresponding message field names selected on a conversion rule. If a message field is not present in the input message, the following applies:</td>
</tr>
<tr>
<td></td>
<td>- 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.</td>
</tr>
<tr>
<td></td>
<td>- 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. If you set Initial Value Length to 99, Substation ES uses the first character of Initial Value to initialize the entire buffer field. The default value for Initial Value is a low value.</td>
</tr>
</tbody>
</table>

*Tip*: Setting initial values is only for mapped buffer fields, for example, fields that have corresponding message field names selected on a conversion rule. If 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. If you set Initial Value Length to 99, Substation ES 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

The Conversion Rules using Buffer Definition Panel displays the conversion rules for a buffer definition.

You access this panel by selecting \textbf{v} in the Sel field of the Buffer Definitions Panel. For details, see Buffer Definitions Panel on page 30.

Panel Layout

\begin{verbatim}
SXTP02U -------- Conversion Rules using Buffer Definition-------------------------
Command ===>                                                   Scroll ==> CSR
Buffer Id ===> New-Buffer
Used in these Conversion Rules:
************************************************************************ Bottom of data ******************************************************
\end{verbatim}

Field Descriptions

- **Buffer Id**: The name of the buffer definition used in the conversion rule. This definition is required and the field must not be blank.
  
  Default: \textit{New-Buffer} (if not specified).

- **Used in these Conversion Rules**: The name of the conversion rules that uses the buffer definition.
Conversion Rules Panel

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

Panel Layout

SXTPO3 ----------------------- Conversion Rules -------------- Row 1 to 6 of 6
Command ===>                                                   Scroll ==> CSR
(ADD=Create new Conversion Rule, CAN=Cancel updates)
(Use line cmd 'M' to relate Message fields to conversion rules)

Max    EMS  S
Sel Flds Bufsz  Type  Z  Conversion Id    Message Id       Buffer Id
-   ---- ------- ---- -  ---------------- ---------------- ---------------
 1 0         M  B  CVR-I-ADMIN      MSG-IN-ADMIN     BUF-IO-ADMIN
 9 0         M  B  CVR-I-IMSCVS-01  MSG-IN-IMSCVS    BUF-I-IMSCVS-01
13 0         M  B  CVR-I-TSRR-01    MSG-IO-RR-01     BUF-IO-BF01
12 0         M  B  CVR-I-01         MSG-IO-01        BUF-I-0-BF01
20 0         M  B  CVR-I-02         MSG-IO-02        BUF-I-0-BF02
 1 0         B  M  CVR-IO-BYTES     MSG-JMS-BYTES    BUF-BYTES

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(lete), R(epro), and U(sedon).

If you select M, you enter the panel described in Define Message Fields to Conversion Rule Panel on page 42. If you select U, you enter the panel described in Recipes & Triggers using a Conversion Rule Panel on page 44.

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(ynst). Default: M.
SZ Used only with TIBCO Enterprise Message Service messages and if the message type is either BYTES or TEXT.

Valid selections: M and B. Default: B.

Determines how Substation ES calculates the size for data communication with the BES (CICS or IMS), as follows:

- If you select M, the payload (TIBCO Enterprise Message Service message) size determines the data communication area size. Note: 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 you select B, the data communication size is determined by the area or fields (or both) in the conversion rules buffer definition. This is the existing method prior to Substation ES Release 2.5.

Conversion Id The user-defined name of the conversion rule. This definition is required and the field must not be blank.

Default: 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

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

Panel Layout

<table>
<thead>
<tr>
<th>Sel Flds</th>
<th>Message Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>New-Msg</td>
</tr>
</tbody>
</table>

Field Descriptions

Sel  Selects a message definition for a Msg-Field or Buffer-Field Conversion.  
     Valid selection: S(elect).

Flds The number of fields in the message definition.

Message Id The name of the message definition.
Select a Buffer Definition Panel

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

Panel Layout

<table>
<thead>
<tr>
<th>SXTP03B</th>
<th>Select a Buffer Definition</th>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sel Flds Buffer Id</td>
<td>1 New-Buffer</td>
<td>CSR</td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

- **Sel**: Selects a buffer definition for a Msg-Field or Buffer-Field Conversion. Valid selection: S(elect).
- **Flds**: The number of fields in the buffer definition.
- **Buffer Id**: The name of the buffer definition.
Define Msg-Field or Buffer-Field Conversion Panel

You map the fields of a message’s definition with the fields of a buffer’s definition for a conversion rule in the Define Msg-Field or Buffer-Field Conversion Panel.

Panel Layout

| SXTP03F --------- Define Msg-Field / Buffer-Field Conversion - Row 1 to 1 of 1 |
| Command ===>                                                   Scroll ==> CSR |
| Conversion Id   ===> New_Rule |
| Using Message   ...: New-Msg |
| Using Buffer    ...: New-Buffer |
| Max Buffer Size ===> 0 |
| Trunc          | Bffld | Msg Field | Num | Buffer Field |
| Y   Y   1      | STRING-FIELD |

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) 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, the size 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 the truncate option is set to Y, the string is limited to the length specified in the Buffer Length field on Buffer Field Details Panel. If the data exceeds the specified length and the truncate option is set to N, the conversion fails.
  
  Default: Y.

  S specifies that the field is a system field, in which case no mapping to the buffer field is required.
<table>
<thead>
<tr>
<th><strong>Reqd</strong></th>
<th>Specifies the field is required for a conversion. This definition is required and the field must not be blank.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Valid selections: \textit{Y(es)} and \textit{N(o)}. Default: \textit{Y}.</td>
</tr>
<tr>
<td></td>
<td>For an inbound process, if you specify \textit{Y}, the transaction is not completed successfully unless a field with this name is encountered in the inbound Rendezvous message. If you specify \textit{N}, no specific check is made as to the availability of that message field in the Rendezvous message.</td>
</tr>
<tr>
<td></td>
<td>For an outbound process, if this field is set to \textit{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.</td>
</tr>
<tr>
<td><strong>BfFld</strong></td>
<td>The sequence number of the buffer field that is to be mapped to the message field.</td>
</tr>
<tr>
<td><strong>Msg Field</strong></td>
<td>The name of the field in the message definition.</td>
</tr>
<tr>
<td><strong>Num</strong></td>
<td>The sequence number of the field in the buffer definition.</td>
</tr>
<tr>
<td><strong>Buffer Field</strong></td>
<td>The name of the field in the buffer definition.</td>
</tr>
</tbody>
</table>
Define Message Fields to Conversion Rule Panel

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

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

| SXTP03G ---------- Define Message Fields to Conversion Rule -- Row 1 to 1 of 1 |
| Command ===> | Scroll ==> CSR |

<table>
<thead>
<tr>
<th>Conversion Id</th>
<th>Using Message</th>
<th>Sel</th>
<th>Trunc</th>
<th>Reqd</th>
<th>CvRul</th>
<th>Msg Field (type MSG)</th>
<th>Num</th>
<th>Conversion Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td>1</td>
<td>New_Rule</td>
</tr>
</tbody>
</table>

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

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. This definition is required and the field must not be blank.

Valid selections: Y and N. Default: Y.

For an inbound process, if you set this field to Y, the transaction is not completed successfully unless a field with this name is encountered in the inbound 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 you set this field to N, 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.
| **CvRul** | The sequence number of the conversion rule to which the nested message field is defined. |
| **Msg Field** | The name of the nested message field. |
| **Num** | The sequence number of the conversion rule. |
| **Conversion Rule** | The name of the conversion rule. |
Recipes & Triggers using a Conversion Rule Panel

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

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

SXTPO3U  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

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

Panel Layout

SXTP04 -------------- Define Processing System Resources ----- Row 1 to 1 of 1
Command ===>                                                   Scroll ==> CSR
    Type:  C(ics) I(ms) A(dmin)
    (ADD=Create new Processing System, CAN=Cancel updates)

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

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

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(lete), and 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. This definition is required, the field must not be blank, and the value must be unique.
Default: New-BES (if not specified).

Interface Id
Name (up to eight characters) assigned to the Substation ES interface for communication with the transaction-processing systems, such as CICS and IMS, which run in the z/OS environment. This definition is required and the field must not 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 ESB Endpoints Panel

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

Panel Layout

<table>
<thead>
<tr>
<th>SXTP07</th>
<th>Define ESB Endpoints</th>
<th>Row 1 to 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>ADD=Create new ESB Endpoint, CAN=Cancel updates</td>
<td>Scroll ==&gt; CSR</td>
</tr>
</tbody>
</table>

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

Field Descriptions

Command
The ADD command creates a new ESB endpoint. The CAN command cancels all the modifications last made to the existing ESB endpoint.
Valid commands: ADD and CAN.

Sel
Selects an existing ESB endpoint to modify.
Valid selections: S(elect), D(lete), and R(epro).
If you select S, you enter the panel described in ESB Endpoint Extensions Panel.

Identifier
The user-defined name of the ESB endpoint. This definition is required and the field must not be blank.
Default: New-ESB (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 ESB endpoint that distinguishes both Substation ES and the role of the ESB endpoint within it.
Default: 0.

Daemon
The host name or IP address that Substation ES uses to find the Rendezvous daemon or TIBCO Enterprise Message Service server and to establish communication.
Default: a blank space.
ESB Endpoint Extensions Panel

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

Panel Layout

```
SXTP07D  ---------------------- ESB Endpoint Extensions  ----------------------
Command ===>                                         (ECF,SSL)

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

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

EMS Values:
Conn User ===> 
Conn Pswd  ===> 
Alt. Conn  ===> 0 0 0 0
Alt. Port  ===> 0
```

Field Descriptions

**Command**  The ECF command accesses the panel described in ESB Endpoint – Connection Factory Extensions Panel on page 50. The SSL command accesses the panel described in ESB Endpoint SSL Extensions Panel on page 52.

These two commands are not supported when the ESB endpoint type is RV.

Valid commands: ECF or SSL.

**Id**  The name of an ESB endpoint.

**Type**  One or more types of Enterprise Service Bus (ESB) to be defined.

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

**Startup**  Specifies whether to start this ESB during Substation ES startup.

Valid selections: Y(es) and N(o). Default: Y.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>For Rendezvous, the daemon URL; for TIBCO Enterprise Message Service, the server URL.</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>For Rendezvous, the daemon port; for TIBCO Enterprise Message Service, the port portion of the TIBCO Enterprise Message Service server URL. Default: 0.</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>The value for the UDP service port number that defines the service group. Default: 0.</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>The service group. You select a local network by network IP address. Default: 0.0.0.0.</td>
<td></td>
</tr>
<tr>
<td>Multicast</td>
<td>Specifies that the address is multicast enabled. Valid selections: Y(es) and N(o). Default: N.</td>
<td></td>
</tr>
<tr>
<td>Multicast Address</td>
<td>The multicast addresses if the host computer has multiple network interfaces.</td>
<td></td>
</tr>
<tr>
<td>Conn User</td>
<td>The user identifier if a special identification is to be used in the session. The maximum character length is 59.</td>
<td></td>
</tr>
<tr>
<td>Conn Pswd</td>
<td>The password associated with Conn User if a special identification is used. The maximum character length is 59.</td>
<td></td>
</tr>
<tr>
<td>Alt. Conn.</td>
<td>The host IP address that Substation ES uses as an alternate connection. TIBCO Enterprise Message Service supports an alternate connection; for example, a fault tolerance server pair.</td>
<td></td>
</tr>
<tr>
<td>Alt. Port.</td>
<td>The value for the alternate port if an alternate connection is in use.</td>
<td></td>
</tr>
</tbody>
</table>
When defining the fields related to Rendezvous, note the following conditions:

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

When defining fields related to TIBCO Enterprise Message Service, note that the URL value is used as specified. If the value specified for Port is nonzero, a colon symbol and the port number are appended to the URL value.
ESB Endpoint – Connection Factory Extensions Panel

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

Panel Layout

SXTP07F -------- ESB Endpoint - Connection Factory Extensions -----------------
Command ====>

ESB Id    ===> New-ESB

ECF User ===> ECF Pswd ===> ECF Name ===> ECF URL ===>

Conn User ===> Conn Pswd ===> ECF PK Password ===> ECF Metric ===> 0 (0=None, 1=Connections, 2=Byte Rate)

Connect                       Reconnect
Attempts ===> 9999999         ===> 9999999 (0=Use Default)
Delay (ms) ===> 500           ===> 500
Timeout (ms) ===> 15000       ===> 15000

Field Descriptions

ESB Id  The name of an ESB endpoint.
ECF User The connection object presents this user identity to the server. Maximum character length is 60.
ECF Pswd The connection object authenticates the user identity using this password. Maximum character length is 40.
ECF Name The name of the connection factory. Maximum character length is 60.
ECF URL The url-string to which connections created by this factory will be connected.
Conn User User ID if a special identification is to be used in the session. Maximum character length is 59.
Conn Pswd
Password associated with the User ID if a special identification is used. Maximum character length is 59.

ECF PK Password
When establishing SSL communication, connections created by the connection factory decode their SSL private key using this password.

ECF Metric
The load balancing metric of this connection factory.
Valid selections:
- 0 – Indicates absence of any load balancing metric.
- 1 – The connection factory balances the connection load among several servers by creating a connection to the server with the fewest number of connections.
- 2 – The connection factory balances the connection load among several servers by creating a connection to the server with the lowest total byte rate (input and output).

Attempts
Connect Attempts – Limits the number of times that Substation ES attempts to establish a connection to the EMS server when the EMS server is not present at Substation ES started time. Default value is 9999999. If 0 is specified, the value is determined by the definition in the Connection Factory, or the default setting in EMS.

Reconnect Attempts – Limits the number of times that Substation ES attempts to reestablish the connection to the EMS server. Default value is 9999999. If 0 is specified, the value is determined by the definition in the Connection Factory, or the default setting in EMS.

Delay (ms)
Connect Delay – Determines the time (in milliseconds) between connection attempts. Valid range: 250 through 20000. Default: 500 (if not specified).

Reconnect Delay – Determines the time (in milliseconds) between reconnection attempts. Valid range: 0, and 250 through 20000. Default: 500 (if not specified).

If 0 is specified, the value is determined by the Connection Factory definition, or the default setting in EMS.

Timeout (ms)
Connect Timeout – Determines the maximum time (in milliseconds) the Substation ES will wait for a connection to the server to be established. Valid range: 15000 through 30000. Default: 15000.

Reconnect Timeout – Determines the maximum time (in milliseconds) the Substation ES will wait for a reconnection to the server to be established. Valid range: 0, and 15000 through 30000. Default: 15000.

If 0 is specified, the value is determined by the Connection Factory definition, or the default setting in EMS.
ESB Endpoint SSL Extensions Panel

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

Panel Layout

SXTP07S ----------------- ESB Endpoint SSL Extensions -------------------------

Command ===>

ESB Id             ===> New-ESB

Use SSL            ===> N     (N/Y - Use SSL)
Authorization Only ===> N     (N/Y - SSL Authorization only)

Key DDNAME         ===>
Identity DDNAME    ===> 
Password           ===> 

SSL ECF Proxy Host ===>
SSL ECF Proxy Port ===> 0

Field Descriptions

ESB Id
The name of an ESB endpoint.

Use SSL
Indicates whether this ESB uses an SSL connection to server. The default value is N.

Authorization Only
Indicates whether this ESB connection use SSL only for authentication. The default value is N.

Key DDNAME
The DD Name of the file where the SSL server's private key digital certificate is stored. This value should matched the one provided in Substation ES startup JCL or PROC.

Set to the DDName for the SSL 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
- PKCS#8
- PKCS#12

Identity DDNAME

The DD Name for the file where the SSL server's digital certificate is stored. This value should match the one provided in the Substation ES startup JCL or PROC. Set to the DDName for the SSL Certificate ID. The DDName must be in the JCL for Substation ES. For digital certificates, the TIBCO EMS Client for z/OS (MVS) supports the following file formats:

- Privacy Enhanced Mail (PEM)
- PKCS#12

Password

The SSL server password. The maximum is 64 characters.

SSL ECF Proxy Host

The SSL proxy host of the connection factory.

SSL ECF Proxy Port

The SSL proxy port of the connection factory.
Recipe List Panel

You create or modify a recipe in the Recipe List Panel.

Panel Layout

<table>
<thead>
<tr>
<th>Command</th>
<th>Row 1 to 1 of 1</th>
<th>Scroll  ==&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXTP05</td>
<td>Recipe List</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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, and **E**(MS).

If you select **E**, you enter the panel described in **EMS Recipe Details Extension Panel** on page 68.

**Resourc**
The name of the resource. For details, see the **Resource Name** field on the **Recipe Details Panel** on page 59.

**MOI**
The method of invocation that initiates the configured transformation.

The valid selections are as follows:

2 DPL
3 Program
4 Task/IMS Tran
5 TDQ Write
6 TSQ Write
7 IMS Conversational Tran
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 59.

**Processor**  The description of the Substation ES interface specified in the Identifier field of the **Define Processing System Resources Panel** on page 45.

**Identification**  The user-defined name of the recipe. This definition is required and the field must not be blank.

Default: **New-Recipe** (if not specified).
Select ESB Endpoint Panel (Recipe)

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

Panel Layout

sxtp05t --------------- select esb endpoint ---------------- row 1 to 1 of 1
command ===>
scroll ==> csr

(use s to select an esb endpoint)

<table>
<thead>
<tr>
<th>sel</th>
<th>service</th>
<th>port</th>
<th>identifier</th>
<th>daemon</th>
<th>network</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>new-esb</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

************** bottom of data **************

Field Descriptions

Sel  Selects the ESB endpoint for a recipe, one ESB per recipe.
  Valid selection: $select$.

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

Port  For Rendezvous, the daemon port; for TIBCO Enterprise Message Service, the
  port portion of the EMS server URL.

Identifier  The name of an ESB endpoint to be used in the recipe.

Daemon  The host name or IP address used by Substation ES to find the Rendezvous
  daemon or TIBCO Enterprise Message Service server and to establish
  communication.

Network  The service group selected by network IP address or multicast addresses.
Select Processing System Resources Panel (Recipe)

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

Panel Layout

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

Use S to select a Processing System

<table>
<thead>
<tr>
<th>Sel</th>
<th>Type</th>
<th>Identifier</th>
<th>Interface Id.</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>I</td>
<td>New-BES</td>
<td>IIIMS</td>
</tr>
</tbody>
</table>

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)

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

Panel Layout

SXTP05C ------------------- Select Conversion Rules -------- Row 1 to 6 of 6
Command ===>                                                   Scroll ==> CSR
(I for input converter, O for output, B for both)

<table>
<thead>
<tr>
<th>Max</th>
<th>Sel</th>
<th>Fldsz</th>
<th>Buksz</th>
<th>Conversion Id</th>
<th>Message Id</th>
<th>Buffer Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CVR-I-ADMIN</td>
<td>MSG-IN-ADMIN</td>
<td>BUF-I0-ADMIN</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CVR-I-TSRR-01</td>
<td>MSG-I0-RR-01</td>
<td>BUF-I0-BF01</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CVR-I-01</td>
<td>MSG-I0-01</td>
<td>BUF-I0-BF01</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CVR-I-02</td>
<td>MSG-I0-02</td>
<td>BUF-I0-BF02</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CVR-I0-BYTE</td>
<td>MSG-JMS-BYTE</td>
<td>BUF-BYTES</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CVR-I0-STRING</td>
<td>MSG-I0-TEXT</td>
<td>BUF-STRING</td>
</tr>
</tbody>
</table>

Field Descriptions

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

Fldsz The number of fields in the conversion rule.

Max Buksz 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

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

Panel Layout

SXTPO5N ------------------------ Recipe Details ------------------------
Command ===>                         (CICS, IMS, EMS, PROP, SEL)

Recipe Identifier ===> SXC-DPL-C
Subject ===> tibss.CICS.C.Request
Publish Subject ===> 
    Publish Always ===> Y
Error Subject ===> tibss.error.out
Error Threshold ===> 0 (0 - 99999) Error Percentage ===> 0
Dynamic Subject Leng ===> 0 Start ===> 0 Usage Limit ===> 0
Input Conversion (?) ===> CVR-I-01
Output Conversion (?) ===> CVR-O-01
Back End System (?) ===> Interface-CICS1 Alt BES(?) : Interface-CICS2
ESB Endpoint (?) ===> EMS-SERVER
Resource Name ===> SXC3I001
Method of Invocation ===> 2
    2 DPL 3 PGM 9 DPL MRO 12 DPL TSQ
    10 Container 11 Storage
    4 Task/IMS Tran 6 TSQ Write 8 Command
    5 TDQ Write 7 IMS Conversational Tran
DPL > 32K Invocations ===> 1
Trace Level ===> 0 0-5 (Debugging)
Reply Method ===> 3
    0 None 2 Message
    1 Status 3 Both Status and Message

Field Descriptions

Command
The CICS command accesses the panel described in CICS Service Details Panel on page 65.
The IMS command accesses the panel described in IMS Service Details Panel on page 67.
The EMS command accesses the panel described in EMS Recipe Details Extension Panel on page 68.
The PROP command accesses the panel described in EMS Properties (Recipe) Panel on page 72.
The SEL command accesses the panel described in EMS Selector Specification Panel on page 74.
Valid commands: CICS, IMS, EMS, PROP and SEL.

Recipe Identifier
The name of the recipe.
Subject

During the configuration of Substation ES to subscribe to data from a TIBCO messaging application, this field specifies the subject for which Substation ES listens to invoke a given transformation. This definition is required and the field must not 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 (the outbound destination name in TIBCO Enterprise Message Service), or to both.

The reply ID in a message is set by the application that issues the request. The reply ID is extracted from the message by Substation ES and the response relating to that request could be delivered to the reply ID, depending on the setting of the Publish Always field.

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

- If you select 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.

- If you select N, the response is sent as follows:
  - To only the reply ID in the message – If the reply ID is specified and the Publish Subject or Destination is or is not specified.
  - To the Publish Subject or Destination – If 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 if Transformer conversion or BES encounters errors.

If you use TIBCO Enterprise Message Service as your ESB, you can now select the EMS message type of the error message. See the ERROR-MSG-TYPE keyword in TIBCO Substation ES Installation.
Error Threshold

The number of errors allowed before the recipe is disabled.

Error Percentage

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.

Usage Limit

The maximum number of concurrent messages to be processed by an ESB endpoint for this recipe. Default value is 0.

Input Conversion (?)

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.

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.

Back End System (?)

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

Alt BES(?)

Optional. The description of an alternate Substation ES interface. Specifying ? displays a list of all interfaces from which you can select. If the specified Back End System is not available during the process, then the messages attempts to process on this Back End System.

ESB Endpoint (?)

The name of the ESB endpoint. Specifying ? displays a list of all ESB endpoints from which you can select. Required, cannot be blank.

Resource Name

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 four characters or less. The maximum length of the TSQ name is eight characters.

Method of Invocation

The method that initiates the transformation that is being configured.
If you are configuring Substation ES to publish data on behalf of a CICS or IMS transaction-processing application, the sole choice is the trigger method of invocation.

If you are configuring Substation ES to subscribe to data generated by a TIBCO messaging application, you can choose one or more of the other methods of invocation.

Valid selections:

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

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

4 Task/IMS Tran – Invokes a CICS task or transaction (four-character resource name) as opposed to a CICS program. Can be a status return when this method is in use.

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 – The transaction invoked by a write to the TDQ from which the transaction is already polling. The length of the TDQ name can be four characters or less.

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

7 IMS 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 – The transaction invoked via CICS LINK with a COMMAREA. Useful for MRO routing.

10 Container – The 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 32 KB input and output.

11 Storage – The transaction invoked via CICS LINK with a COMMAREA. The COMMAREA contains a pointer to a storage area and its length. Can be used for greater than 32K input and output.
**12 DPL TSQ** – The transaction invoked via CICS LINK with a COMMAREA, which contains an eight-character TSQ ID, which in turn contains the input data and which can also be used for output.
MOI Usage

<table>
<thead>
<tr>
<th>MOI</th>
<th>Requirements</th>
<th>Interface</th>
<th>Inbound Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CICS</td>
<td></td>
</tr>
<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>

**IMS**

|     | Resource Name; 1 – 4 MB in size | OMTA | XCF | User transaction |

**Trace Level**

The user trace debug level for an individual recipe. Must always be 0 or 1, unless TIBCO Support requires the output. A certain amount of overhead and many lines of output are produced when this value is greater than 1.

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 applies for this recipe.

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

2 **Message** – The 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.
CICS Service Details Panel

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

Access this panel by entering the **CICS** command on the Recipe Details panel. For details, see Recipe Details Panel on page 59.

Panel Layout

<table>
<thead>
<tr>
<th>SXTP05I</th>
<th>------------------------------</th>
<th>------------------------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ====&gt;</td>
<td>CICS Service Details</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Recipe Identifier ====&gt; New-Recipe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Name ======&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method of Invocation ====&gt; 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mirror Transaction ====&gt; (Default SXEX)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSMI User Transaction ====&gt; N (N/Y)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Field Descriptions

**Recipe Identifier**

The name of the recipe.

**Resource Name**

The name of the resource.

**Method of Invocation**

The method used to initiate the transformation that is being configured.

**Mirror Transaction**

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 by way of 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.

**CSMI User Transaction**
Indicates whether this is a CSMI user transaction. Default is N (no).

**Transaction ID**
A four character transaction identifier responsible for running the program on the remote CICS system.

**System Name**
A four character system identifier for the CICS region where the program will ultimately be run.
IMS Service Details Panel

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

Access this panel by entering the **IMS** command on the Recipe Details panel. For details, see Recipe Details Panel on page 59.

Panel Layout

```
SXTP05J --------------------- IMS Service Details -----------------------------
Command ===> 

Recipe Identifier ===> New-Recipe
Resource Name ===> 
Method of Invocation ===> 2
Variable Length Tran ===> N (N/Y)
```

Field Descriptions

- **Recipe Identifier**: The name of the recipe.
- **Resource Name**: The name of the resource.
- **Method of Invocation**: The method used to initiate the transformation that is being configured.
- **Variable Length Tran**: Indicates whether the IMS transaction ID provided in the Resource Name is less than eight characters.
  
  Valid selections: Y(es) or N(o). The default is N.
EMS Recipe Details Extension Panel

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

You access this panel by selecting \texttt{E} in the Sel field in of the Recipe List Panel. For details, see Recipe List Panel on page 54. Alternatively, you can access this panel with the \texttt{EMS} command on the Recipe Details Panel. For details, see Recipe Details Panel on page 59.

Panel Layout

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

- **Recipe Identifier**: \(\Rightarrow\) New-Recipe
- **Input Dest Type**: \(\Rightarrow T\) Name: EMS
- **Output Dest Type**: \(\Rightarrow T\) Name:
- **Error Dest Type**: \(\Rightarrow Q\) Name: tibss.error.ivp.out
- **Durable Subscriber**: \(\Rightarrow N\) (N/Y)
- **Durable Client Id**: \(\Rightarrow\) (Service Levels G=Guaranteed, R=Reliable)
- **SMSL Mode**: \(\Rightarrow RA\) (RA Auto-Ack, RN No-Ack, GA All, GS SS Recv)
- **Delivery Mode**: \(\Rightarrow P\) (P Persistent, N Non-Persistent, R Reliable)

Output Message properties:
- **Priority**: \(\Rightarrow N\) (N/Y) Value \(\Rightarrow 4\) (0 - 9)
- **Expiration**: \(\Rightarrow N\) (N/Y) Value \(\Rightarrow 0\) Unit \(\Rightarrow S\) (L,S,M,H,D)
- **Compression**: \(\Rightarrow N\) (N/Y)
- **Preserve**: \(\Rightarrow N\) (N/Y, undelivered messages)
- **Body Trace**: \(\Rightarrow N\) (N/Y)
- **Message Type**: \(\Rightarrow M\) M(ap)/S(tream)/T(Ext)/B(ytes)
- **Copy Correlation**: \(\Rightarrow N\) (N/Y)

Field Descriptions

- **Recipe Identifier**: The name of the recipe.
- **Input Dest Type**: The input destination type.
  Valid selections: \texttt{T(opic)} and \texttt{Queue}. Default: \texttt{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: \texttt{T(opic)} and \texttt{Queue}. Default: \texttt{T}.
<table>
<thead>
<tr>
<th><strong>Output Dest Name</strong></th>
<th>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.</th>
</tr>
</thead>
</table>
| **Error Dest Type**  | The error destination type.  
Valid selections: T(opic) and Q(ueue). Default value is Q. |
| **Error Dest Name**  | The destination in which an error message is published when Transformer conversion or BES encounter errors. |
| **Durable Subscriber** | Connect the session as a durable subscriber.  
Valid selections: N(o) and Y(es). Default: N. |
| **Durable Client Id** | The client identifier in a durable connection. If not specified, the recipe name is used. |
| **SMSL Mode**        | The SMSL mode for TIBCO Enterprise Message Service.  
Valid selections:  
RA Auto-Ack (Reliable Auto Acknowledge) – The EMS client library automatically confirms the receipt of a message.  
RN No-Ack (Reliable No Acknowledge) – The 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 the Substation ES processes and executions controlled by Substation ES for a given message have completed successfully.  
GS SS Recv (Guaranteed Substation Processing) – Substation ES calls the EMS client library and explicitly confirms receipt of a message after the Substation ES processes have executed successfully. Processing is considered complete when a response message is successfully delivered to its destination. This is regardless of types of message that are delivered back by Substation ES, error, or application type messages.  
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 TIBCO Enterprise Message Service message priority to the value specified in the Priority Value field.
Valid selections: N(o) and Y(es). Default: N.

Priority Value
The TIBCO Enterprise Message Service message priority value to be used.

Expiration
Set TIBCO Enterprise Message Service message expiration to the value specified in the Expiration Value field.
Valid selections: N(o) and Y(es). Default: N.

Expiration Value
The TIBCO Enterprise Message Service message expiration value to be used.
Valid range: 0 to 99999. Default: 0.

Expiration Unit
The TIBCO Enterprise Message Service message expiration unit to be used.
Valid selections: L = millisecond, S(cond), M(inute), H(our), and D(ay). Default: S.

Compression
Compress the TIBCO Enterprise Message Service message for storage on the server.
Valid selections: N(o) and Y(es). Default: N.

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

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

Message Type
The type of TIBCO Enterprise Message Service output message.
Valid selections: M(ap), S(tream), T(ext), and B(ys). Default: M.

Copy Correlation
Copy the JMSCorrelationID value from an inbound message and put it in the output message.
If **Y** is selected, Substation ES checks if the **JMSCorrelationID** header field is defined under EMS Properties (for details, see EMS Properties (Recipe) Panel on page 72). If so, Substation ES replaces the original value with the value specified under EMS Properties. Otherwise, Substation ES uses the original value from the inbound message.

If **N** is selected, Substation ES again checks if the **JMSCorrelationID** header field is defined under EMS Properties. If so, Substation ES replaces the original value with the value specified under EMS Properties. Otherwise, Substation ES performs no action.

Valid selections: **N**(o) and **Y** (es). Default: **N**.
EMS Properties (Recipe) Panel

You specify EMS properties to be inserted in output messages or map the EMS properties fields from the input message to the buffer fields in the EMS Properties (Recipe) Panel. You can specify up to fifteen properties, including the JMSCorrelationID header field.

Access this panel by issuing the PROP command on the Recipe Details panel. For details, see Recipe Details Panel on page 59.

Panel Layout

<table>
<thead>
<tr>
<th>Command ==&gt;</th>
<th>Scroll ==&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXTP05PR</td>
<td>EMS Properties (Recipe)</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Output Buffer Id: Out-Buffer</td>
<td></td>
</tr>
<tr>
<td>Input Buffer Id: In-Buffer</td>
<td></td>
</tr>
<tr>
<td>Property Name</td>
<td>B/E/V Req Value</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Property-Out-1</td>
<td>B  _  PGM-ID</td>
</tr>
<tr>
<td>Property-Out-2</td>
<td>V  _  This is the property field</td>
</tr>
<tr>
<td>Property-In</td>
<td>E  Y  HOST-NAME</td>
</tr>
</tbody>
</table>

Field Descriptions

Note that a maximum of 15 properties can be defined.

Output 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.

Input Buffer ID   If mapping a property value to a buffer, this is the buffer name associated with an input conversion rule from which a selection can be made.
<table>
<thead>
<tr>
<th><strong>Property Name</strong></th>
<th>The name of the property.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B/E/V</strong></td>
<td>B specifies that the property value is selected from a buffer field. E specifies that the property value is selected from an ESB input message field. V specifies that the property value is defined by the user in the Value field. Valid selections are B(uffer), E(SB) or V(alue).</td>
</tr>
<tr>
<td><strong>Value</strong></td>
<td>If the property value is to be selected from the buffer (B is specified in the B/E/V field), 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 (V is specified in the B/E/V field), type the desired value.</td>
</tr>
</tbody>
</table>
EMS Selector Specification Panel

You specify an TIBCO Enterprise Message Service message selector in the EMS Selector Specification Panel. Access this panel by issuing the SEL command on the Recipe Details Panel. For details, see Recipe Details Panel on page 59.

Panel Layout

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

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Currently not used.</td>
</tr>
</tbody>
</table>
| 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 on the message selector syntax, see the section on the Message class in the relevant TIBCO Enterprise Message Service API reference documentation. |
Define Triggers Panel

You create or modify a trigger in the Define Triggers Panel.

Panel Layout

<table>
<thead>
<tr>
<th>Command</th>
<th>SXTP06选择了 Define Triggers</th>
<th>Scroll =&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ADD=Create new Trigger, CAN=Cancel updates)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sel</td>
<td>Output</td>
<td>Identification</td>
</tr>
<tr>
<td>2</td>
<td>New-Trigger</td>
<td>MATCH</td>
</tr>
</tbody>
</table>

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

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, and **E**(MS).

If you select **E**, you enter the panel described in EMS Trigger Details Extension Panel on page 81.

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

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

For details, see the Reply Method field in the Recipe Details Panel on page 59.

**Identification**
The name you specify for the trigger. This field is required and 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 ESB Endpoint Panel (Triggers)

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

### Panel Layout

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Scroll ==&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXTP06T------------------------ Select ESB Endpoint -------------- Row 1 to 1 of 1</td>
<td>(Use S to select an ESB endpoint)</td>
</tr>
<tr>
<td>Sel Service Port Identifier Daemon Network</td>
<td>0 0 New-ESB 0 0 0 0</td>
</tr>
</tbody>
</table>

### Field Descriptions

- **Sel**: Selection of the ESB endpoint for a trigger. Select one ESB for each trigger.
  - Valid selection: $S$(elect).
- **Service**: The value for the UDP service port number that defines the service group.
- **Port**: The TCP port given to an ESB endpoint that distinguishes both Substation ES and the role of the ESB endpoint within it.
- **Identifier**: The name of an ESB endpoint in the trigger.
- **Daemon**: The host name or host IP address used by Substation ES to find the Rendezvous 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)

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

Panel Layout

```
SXTP06C----------------- Select Output Conversion Rule ------- Row 1 to 1 of 1
Command ===>                                                   Scroll ==> CSR
Max
Sel  Flds  Buksz  Conversion Id  Message Id  Buffer Id
- ---- ----- --------------- --------------- ---------------
  0  0      New_Rule        New_Msg        New-Buffer
******************************* Bottom of data *******************************
```

Field Descriptions

- **Sel**: Selection of the conversion rule for a trigger.
  Valid selections: **S**(elect) and **O**(utput).

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

- **Max Buksz**: The maximum buffer size for the conversion rule.

- **Conversion Id**: The name of the conversion rule in the trigger.

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

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

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

Panel Layout

<table>
<thead>
<tr>
<th>SXTP06N</th>
<th>Trigger Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>===&gt; (EMS,PROP)</td>
</tr>
</tbody>
</table>

| 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 |
| ESB Endpoint (?) | ===> New-ESB |
| Trace Level | ===> 0 0-5 |
| Output Method | ===> 2 1 Status 2 Message 3 Both Status and Message |
| Compare Text Contain in Buffer |
| Start | ===> 0 |
| Length | ===> 5 |
| Value | ===> MATCH |

Field Descriptions

**Command**

The EMS command accesses the panel described in EMS Trigger Details Extension Panel on page 81. The PROP command accesses the panel described in EMS Properties (Trigger) Panel on page 84.

Valid commands: EMS and 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. This field is required and cannot be blank.

**Reply Subject**

Optional. The reply subject on which an external messaging application returns a response to Substation ES (for a CICS-initiated request reply process).

**Error Subject**

The subject on which an error message is published if 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 Leng** | **Optional.** The outbound subject name can be specified dynamically within the buffer. Specifies the number of bytes to be used for the Dynamic Subject name.  
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. |
| **ESB Endpoint (?)** | The name of an ESB endpoint. Specifying ? displays a list of all ESB endpoints from which you can select. Required, cannot be blank. |
| **Trace Level** | The user trace debug level for an individual trigger process. Must always be 0 or 1, unless TIBCO 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 with 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. |
| **Text in Buffer** | 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 program's 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

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

You access this panel by selecting E in the Sel field of the Define Triggers Panel. For details, see Define Triggers Panel on page 75. Alternatively, access this panel with the EMS command on the Trigger Details Panel. For details, see Trigger Details Panel on page 78.

Panel Layout

SXTP06E----------------- EMS Trigger Details Extension ------------------------
Command ===> 

   Trigger Identifier ===> New-Trigger
   Output Dest Type ===> T Name:
   Reply Dest Type ===> T Name:
   Error Dest Type ===> Q Name:

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

   Output Message properties:
   Priority ===> N (N/Y) Value ===> 4 (0 - 9)
   Expiration ===> N (N/Y) Value ===> 0 Unit ===> S (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(extend)/B(ypes)

Field Descriptions

**Trigger Identifier**

The name of the trigger.

**Output Dest Type**

The output destination type.

Valid selections: T(topic) and Q(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(topic) and Q(queue). Default: T.
The reply destination in which an external messaging application returns a response to Substation ES (for a CICS-initiated request reply process).

**Error Dest Type**
The error destination type.

**Error Dest Name**
The destination in which an error message is published if Transformer conversion or BES encounter errors.

**SMSL Mode**
The SMSL mode for TIBCO Enterprise Message Service.
Valid selections:
- **GA All** (Guaranteed All) – Substation ES specific BES Interface explicitly issues a commit type call of the transaction to the BES when all the Substation ES processes and executions for a given message have completed successfully. In case of an error, the message is moved to the Dead Message Queue.
- **RN No-Ack** (Reliable No Acknowledge) – The TIBCO Enterprise Message Service client library does not send a receipt of the message that was received. The server destroys the message after it has been sent.
Default: RN.

**Delivery Mode**
The delivery mode for TIBCO Enterprise Message Service.
Valid modes: Persistent, Non-persistent, and Reliable. Default: Persistent.
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: No and Yes. Default: No.

**Priority Value**
The EMS message priority value.

**Expiration**
Set EMS message expiration to the value specified in the Expiration Value field.
Valid selections: No and Yes. Default: No.

**Expiration Value**
The EMS message expiration value.
Valid range: 0 to 99999. Default: 0.

**Expiration Unit**
The message expiration unit.
Valid selections: L = millisecond, S(ekond), M(inute), H(our), and D(ay). Default: Second.
**Compression**  
Compress the EMS message for storage on the server.
Valid selections: N(o) and 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 value. If this value is set to Y, the message is placed on the server’s the undelivered message queue.
Valid selections: N(o) and Y(es). Default: N.

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

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

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

Note: You can also specify the JMSCorrelationID header field in this panel.

You access this panel by issuing the PROP command on the Trigger Details panel. For details, see Trigger Details Panel on page 78.

Panel Layout

SXTP05PT  ----------------- EMS Properties (Trigger) -------- Row 1 to 11 of 15
Command ==>                                                   Scroll ==> CSR

Output 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>

Field Descriptions

Output 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 that the property value is selected from a buffer field. V specifies that the property value is defined by the user in the Value field.

Valid selections: B(uffer) or V(alue).
**Value**  If you select this field from the buffer, type the question mark (?) symbol and press the **ENTER** key. The subsequent display is a list of field names in the buffer from which to choose.

If user-defined, type the desired value.
Group Definitions Panel

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

Panel Layout

```
SXTPO8 ----------------------- Group Definitions ------------- Row 1 to 1 of 1
Command ===>                                                   Scroll ==> CSR
          (ADD=Create new Group, CAN=Cancel updates)

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

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(elete), and R(epro).

**Group Id**  The name of the group specified by you. This field is required and cannot be blank.

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

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

Panel Layout

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

Group Name ==> New_Group

Use S to select each Trigger and/or Recipe into the Group.

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

Field Descriptions

Group Name  The name of the group specified by you.

Sel  Selection of each trigger and recipe for the group.

Valid selections: S(elect).

Type  Trigger or recipe.

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

You select the Summary of Definition for a recipe, trigger, conversion, or group definition in the Display Configuration File Detail Panel.

Panel Layout

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

Select one of the following:

1 Expand a Recipe
2 Expand a Trigger
3 Expand a Conversion
4 Expand a Group Definition

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

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

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

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

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 Selection of 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)

The Summary of Definition Panel displays a recipe in detail.

Panel Layout

| Recipe name  : Substation-Admin-IMS |
| Subject      : tibss.admin.sxi |
| Response     : , Always: Y |
| Reply Sub:    |
| Err Subj     : tibss.error.ivp.out |
| Error Thresholds: Count: 0, Percent: 0 |
| Transcode:   , MOI:8, Trace: 0, Reply: 3 |
| Dynamic Subject Leng:0, Start: 0, Limit: 0 |
| EMS Extensions: |
| Destination Types: Input: T, Output: T, Error: Q |
| Expire: 0, Units: S, Priority: 4 |
| Durable Subscriber: N, ClientId: |
| SMSL Mode: RA, Delivery Mode: P |
| Back End System: Interface-CICS1 |
| Type: C, Target: CICS1 |
| Alt BES System: Interface-CICS2 |
| Type: C, Target: CICS2 |
| ESB Endpoint id: SS IVP Default, Daemon: , Network: 127.0.0.1 |
| Service: 7555, Port: 0, Multicast: N |
| Userid: TIBUSER, Password: ******* |
| Alt Server: 0.0.0.0, Port: 0 |
| Input Conversion: SXI-CMD-I-CR01 |
| Max buffer size: 0 |
| Receiving Buffer max: 0, id: SXI-G-BUF01 |
| Input Message id: SXS-ADM-I-MSG01 |
| From RV id: 0, name: SXS-COMMAND |
| To offset: 0000, length: 2048, name: COMMAND-TXT |
| In decimal: 0, Out decimal: 0, Buf-type: TEXT |
| Truncate: Y, Required: Y, Lead-sign: N |
| Sep-sign: N, Right-just: N, Pad: 40 |
| Init len: 0, Init value: |
| From RV id: 0, name: tibss-userid |
| Truncate: S, Required: N |
| From RV id: 0, name: tibss-password |
| Truncate: S, Required: N |
| Output Conversion: SXI-CMD-O-CR01 |
| Max buffer size: 0 |
Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recipe name</td>
<td>The name of the recipe.</td>
</tr>
<tr>
<td>Subject</td>
<td>The subject Substation ES listens for to invoke a given transformation.</td>
</tr>
<tr>
<td>Response</td>
<td>The subject on which a message is published for a given transaction.</td>
</tr>
<tr>
<td>Always</td>
<td>Determines where the response from a transaction process is sent.</td>
</tr>
<tr>
<td></td>
<td>For details, see the Publish Always field of the Recipe Details Panel on page 59.</td>
</tr>
<tr>
<td>Reply Sub</td>
<td>The subject on which a status message is published for the requesting application.</td>
</tr>
<tr>
<td>Err Subj</td>
<td>The subject on which an error message is published if Transformer conversion or BES encounters errors.</td>
</tr>
<tr>
<td>Count</td>
<td>The number of errors allowed before the recipe is disabled.</td>
</tr>
<tr>
<td>Percent</td>
<td>The percentage of errors allowed before the recipe is disabled.</td>
</tr>
<tr>
<td>Transcode</td>
<td>The name of the resource.</td>
</tr>
<tr>
<td>MOI</td>
<td>The method of invocation that initiates the configured transformation.</td>
</tr>
<tr>
<td>Trace</td>
<td>The user trace debug level.</td>
</tr>
<tr>
<td>Reply</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 59.</td>
</tr>
<tr>
<td>Dynamic Subject Leng</td>
<td>The number of bytes to be used for the Dynamic Subject name.</td>
</tr>
<tr>
<td>Start</td>
<td>The starting offset in the buffer to be used for the dynamic subject name.</td>
</tr>
</tbody>
</table>
Limit  The maximum number of concurrent messages to be processed by an ESB endpoint for this recipe.

Input  The destination of the subject for which Substation ES listens to invoke a given transformation: T(opic) or Q(ueue).

Output The destination of an outbound subject on which a message is published for a given transaction process: T(opic) or Q(ueue).

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

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

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

Compr  Compression 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 Generate messages that include the message body? N(o) or Y(es).

Expire The EMS expiration value for the message.

Units  The EMS legal unit value for the message.

Priority The EMS priority value for the message.

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

ClientId The client identifier 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, both Primary and Alternate.

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

Target The identification of the interface through which Substation ES communicates with the transaction processing application. Must match the interface ID that is configured through the INTF-ID parameter in the system initialization member.
<table>
<thead>
<tr>
<th><strong>ESB Endpoint id</strong></th>
<th>The name of the ESB endpoint.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Daemon</strong></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><strong>Network</strong></td>
<td>The service group selected by network IP network number or multicast addresses.</td>
</tr>
<tr>
<td><strong>Service</strong></td>
<td>The value provided for the UDP service port number that defines the service group.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>The TCP port given to a ESB endpoint that distinguishes both Substation ES and the role of the ESB endpoint within it.</td>
</tr>
<tr>
<td><strong>Multicast</strong></td>
<td>Is the address is multicast enabled? N(0) or Y(es).</td>
</tr>
<tr>
<td><strong>Userid</strong></td>
<td>The user identifier you specify if a special identification is required in the session.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>The password associated with UserId, if applicable.</td>
</tr>
<tr>
<td><strong>Alt Server</strong></td>
<td>The host IP address that Substation ES uses as an alternate connection (TIBCO Enterprise Message Service supports an alternate connection).</td>
</tr>
<tr>
<td><strong>Alt Port</strong></td>
<td>The value for the alternate port if an alternate connection is in use.</td>
</tr>
<tr>
<td><strong>Input Conversion</strong></td>
<td>The name of the conversion rule for the input conversion.</td>
</tr>
<tr>
<td><strong>Max buffer size</strong></td>
<td>The maximum input buffer size defined for the conversion rule.</td>
</tr>
<tr>
<td><strong>Receiving Buffer</strong></td>
<td>The maximum size of the inbound buffer.</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td>The name of the buffer associated with the transformation for the receiving buffer.</td>
</tr>
<tr>
<td><strong>Input Message</strong></td>
<td>The name of the message.</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td>The name of the message.</td>
</tr>
<tr>
<td><strong>From Message</strong></td>
<td>A message field identifier. Zero (0) is a special value that signifies no message field identifier.</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td>The field name you specify in the message definition associated with the transformation from which data is to be extracted.</td>
</tr>
</tbody>
</table>
To offset  The number of bytes for the field that is the starting offset in the inbound buffer. The buffer offset positions the data at a given byte position.

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

name  The name you specify 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, the 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, the 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 Yes.

The S field serves as a system field, which means no mapping to the buffer field is required.

Required  Whether the field is required for a conversion. The default value is Y.

If you specify Y, the transaction is not considered complete unless a field with this name is encountered in the inbound Rendezvous message. If you specify N, no specific check is made as to the availability of that message field in the Rendezvous message.

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

Sep-sign  Whether a byte is reserved for the sign. The default is N. The valid selections are N(one), L(leading), and T(railing).

Right-just  The positioning of the string. You can use padding characters to fill in the data on either side, as applicable. The default is NO and strings are left justified.

Pad  The character for padding. You can use keyboard-accessible characters as a pad, which are placed at the beginning, at the end, or in both positions, depending on the configuration. The default pad character is a space.

Init len  The length of the Init Value field when used.
| **Init value** | The value or characters for filling in the field if the data is not available in the buffer. |
| **Output Conversion** | The name you specify 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 id** | The maximum size of the outbound buffer for this recipe process to be sent. The name you specify for the buffer associated with the transformation 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 is the starting offset in the outbound buffer. The buffer offset positions 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 name you specify 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, the implied decimal is 2. This field only applies to integers. |
| **Message decimal** | The implied number of digits after the decimal point in the message field. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers. |
| **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 the definitions of the buffer types, refer to Type on page 32. |
| **To Message 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, and IPA |
For the definitions of the message types, refer to Type on page 26.

**Reqd**  Whether the field is required for a conversion. The default is y.

If you set this field to n, if the Data Type is numeric, and if the content of the field has not been initialized by the host-side application, the Transformer initializes the field and no editing errors occurs.

**name**  The name you specify for 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)

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

Panel Layout

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

| Trigger name : CICS-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 |
| ESB Endpoint 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 ****************************************************************************
### Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger name</td>
<td>The user-defined trigger name associated with a particular transformer configuration.</td>
</tr>
<tr>
<td>Reply</td>
<td>The data expected to be output from the transaction processing application. For further clarification, see Output Method on the Trigger Details Panel on page 78.</td>
</tr>
<tr>
<td>Trace</td>
<td>The user trace debug level assigned on the Trigger Details Panel. The valid range is 0 through 5.</td>
</tr>
<tr>
<td>Response</td>
<td>A subject that a message is published on for a given trigger transaction.</td>
</tr>
<tr>
<td>Reply Sub</td>
<td>The reply subject or destination message property for the published message.</td>
</tr>
<tr>
<td>Err Subj</td>
<td>A subject on which an error message is published if Transformer conversion or BES encounters errors.</td>
</tr>
<tr>
<td>Count</td>
<td>The number of errors allowed before the recipe is disabled.</td>
</tr>
<tr>
<td>Percent</td>
<td>The percentage of errors allowed before the recipe is disabled.</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Optionally, you can specify the outbound subject name dynamically within the buffer. This field specifies the number of bytes for the Dynamic Subject name. The default is 0.</td>
</tr>
<tr>
<td>Subject Leng</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>The starting offset in the buffer for the dynamic subject name.</td>
</tr>
<tr>
<td>Match at</td>
<td>The number of character positions that a string is offset in the buffer to indicate that it is a TDQ search string. The Transformer searches for a string with this offset and evaluates the string to determine the kind of transformation to perform.</td>
</tr>
<tr>
<td>Length</td>
<td>The size of the search string in the buffer of a TDQ.</td>
</tr>
<tr>
<td>Value</td>
<td>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.</td>
</tr>
<tr>
<td>Output</td>
<td>The output destination type: T(opic) or Q(ueue).</td>
</tr>
<tr>
<td>Reply</td>
<td>The reply destination type: T(opic) or Q(ueue).</td>
</tr>
<tr>
<td>Error</td>
<td>The subject destination in which an error message is published if Transformer conversion or BES encounters errors: T(opic) or Q(ueue).</td>
</tr>
<tr>
<td>Prty</td>
<td>Specifies whether the priority has been set for this message.</td>
</tr>
<tr>
<td><strong>Expir</strong></td>
<td>Specifies whether the expiration has been set for this message.</td>
</tr>
<tr>
<td><strong>Compr</strong></td>
<td>Specifies whether compression has been set for this message.</td>
</tr>
<tr>
<td><strong>Presv</strong></td>
<td>Specifies whether the message is to be preserved on the server.</td>
</tr>
<tr>
<td><strong>Trace</strong></td>
<td>Specifies whether trace generates messages that include the message body.</td>
</tr>
<tr>
<td><strong>Expire</strong></td>
<td>Specifies the EMS expiration value to for the message.</td>
</tr>
<tr>
<td><strong>Units</strong></td>
<td>Specifies the EMS legal unit value for the message.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Specifies the EMS priority value for the message.</td>
</tr>
<tr>
<td><strong>SMSL Mode</strong></td>
<td>Specifies the Substation Messaging Services Levels mode used for this message by Substation ES.</td>
</tr>
<tr>
<td><strong>Delivery Mode</strong></td>
<td>Specifies the delivery mode used for this message by Substation ES.</td>
</tr>
<tr>
<td><strong>ESB Endpoint id</strong></td>
<td>The name you specify for a ESB definition for a particular transformer configuration entered on the Define ESB Endpoint Panel.</td>
</tr>
<tr>
<td><strong>Daemon</strong></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><strong>Network</strong></td>
<td>The service group selected by IP network number or multicast addresses.</td>
</tr>
<tr>
<td><strong>Service</strong></td>
<td>The value for the UDP service port number that defines the service group.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>The TCP port given to an ESB that distinguishes both Substation ES and the role of the ESB within it.</td>
</tr>
<tr>
<td><strong>Multicast</strong></td>
<td>Specifies whether the address is multicast enabled.</td>
</tr>
<tr>
<td><strong>Userid</strong></td>
<td>The user identifier you specify if a special identification is required in the session.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>The password associated with UserId, if applicable.</td>
</tr>
<tr>
<td><strong>Alt Server</strong></td>
<td>The value for an alternate connection (TIBCO Enterprise Message Service supports an alternate connection).</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>The value for the alternate port if an alternate connection is used.</td>
</tr>
<tr>
<td><strong>Output Conversion</strong></td>
<td>The name you specify for the output conversion associated with the transformation.</td>
</tr>
<tr>
<td><strong>Max buffer size</strong></td>
<td>The maximum outbound buffer size defined for the Msg-Field or Buffer-Field conversion rule.</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Sending Buffer max</strong></td>
<td>The maximum size of the outbound buffer for this trigger process to be sent.</td>
</tr>
<tr>
<td><strong>id</strong></td>
<td>The user-defined name of the buffer definition associated with the transformation used for the sending buffer.</td>
</tr>
<tr>
<td><strong>Output Message id</strong></td>
<td>The name of the message definition associated with the transformation that was entered on the Message Definitions Panel.</td>
</tr>
<tr>
<td><strong>From offset</strong></td>
<td>The number of bytes by which the response message is offset in the buffer.</td>
</tr>
<tr>
<td><strong>length</strong></td>
<td>The length of the data type field in the buffer definition associated with the transformation.</td>
</tr>
<tr>
<td><strong>name</strong></td>
<td>The user-defined name for the field in the buffer definition associated with the transformation.</td>
</tr>
<tr>
<td><strong>Buffer decimal</strong></td>
<td>The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td><strong>Message decimal</strong></td>
<td>The implied number of digits after the decimal point in the message field. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers.</td>
</tr>
<tr>
<td><strong>Buf-type</strong></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 the definitions of the buffer types, see <strong>Type on page 32.</strong></td>
</tr>
<tr>
<td><strong>To Message id</strong></td>
<td>A message field identifier. Zero (0) is a special value that signifies no field identifier.</td>
</tr>
<tr>
<td><strong>Type</strong></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, and IPA For the definitions of the message types, see <strong>Type on page 26.</strong></td>
</tr>
<tr>
<td><strong>Reqd</strong></td>
<td>Whether the field is required for a Conversion. Default value is Y. If you set this field to N, if the Data Type is numeric, and if the content of the field has not been initialized by the host-side application, the Transformer initializes the field and no editing errors occurs.</td>
</tr>
</tbody>
</table>
name  The field name you specify in the message definition associated with the transformation from which data is to be extracted.

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

The Summary of Definition Panel displays a conversion rule’s definition in detail.

### Panel Layout

```
| Conversion Identification: SXC-RR-I-CR01 |
| Max buffer size: 0 |
| Related Buffer max: 0, id: SXG-I0-BF01 |
| Related Message id: SXC-I0-RR-MS01 |

Message Field id: 0, type: OPA, name: tibss-envelope
    Truncate: S, Required: Y

Message Field id: 0, type: STR, name: IVP-ID
    Buffer offset: 0000, length: 008, name: IVP-ID
    Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT
    Truncate: Y, Required: Y
    Sep-sign: N, Right-just: N, Pad: 40
    Init len: 0, Init value:

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

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

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

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

Message Field id: 0, type: STR, name: IVP-BES-PGM-NAME
```

TIBCO Substation ES Configuration and Resources
| Buffer offset: 0160, length: 008, name: IVP-BES-PGM-NAME |
| Message decimal: 0, Buffer decimal: 0, Buf-type: TEXT |
| Truncate: Y, Required: N |
| Sep-sign: N, Right-just: N, Pad: 40 |
| Init len: 0, Init value: |

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

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

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

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

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

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

Field Descriptions

**Conversion Identification**

The name you specify for 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.
| **id** | The name of the buffer definition associated with the transformation that was entered on the Buffer Definitions Panel. |
| **Related Message id** | The name of the message definition associated with the transformation that was entered on the Message Definitions Panel. |
| **Message Field 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, and IPA. For the definitions of the message types, refer to Type on page 26. |
| **name** | The field name you specify in the message definition associated with the transformation from which data is to be extracted. |
| **Buffer offset** | The number of bytes for the field that is the starting offset in the buffer. The buffer offset positions the data at a given byte position. |
| **length** | The length of the data type field in the buffer definition associated with the transformation. |
| **name** | The field name you specify in the message definition associated with the transformation from which data is to be extracted. |
| **Message decimal** | The implied number of digits after the decimal point in the message field. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers. |
| **Buffer decimal** | The implied number of digits after the decimal point in the buffer. For example, for the number 345.67, the implied decimal is 2. This field only applies to integers. |
| **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 the definitions of the buffer types, refer to 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 Yes. The S field serves as a system field, which means no mapping to the buffer field is required. |
**Required**
Specifies whether the field is required for a conversion. The valid selections are Y(es) and N(o). The default is Y.

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

For an outbound process, if you set this field to N, if the Data Type is numeric, and if the content of the field has not been initialized by the host-side application, the field is initialized by the Transformer and no editing errors 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. The default is N. The valid selections are: N(one), L(eading) and T(rail).n

**Right-just**
The positioning of the string. You can use padding characters to fill in the data on either side, as applicable. The default is NO and strings are left justified.

**Pad**
The character for padding. You can use keyboard-accessible characters as a pad, which are placed at the beginning or at the end, depending on the configuration. The default pad character is a space.

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

**Init value**
The value or characters for filling in the field if the data is not available in the buffer.
Summary of Definition Panel (Group)

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

Panel Layout

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

|Group name : IVP Group
|    Recipe id: Substation-Admin-IMS
|    Trigger id: CICS-RR-REQUEST

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

In the Copybook Converter Panel, you select and convert a COBOL copybook member from a PDS into a buffer definition within the configuration file. Optionally, you can create a message and a conversion rule during the copybook parsing and conversion process.

Panel Layout

SXTP10----------------------- Copybook Converter ------------------------------

Command

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

Library containing Copy Member:

Additional libraries for COPY:

(Parser Debug level) ===> 0

Field Descriptions

Copy Member  The name of the copybook member to be selected from a PDS. This value is required and cannot be blank.

Buffer Identifier  Optional. The 16-character name for the buffer identifier. If you enter *, the Copy Member name is used.

Message Identifier  Optional. The 16-character name for the message identifier. If you enter *, the Copy Member name is used.

Convert Rule Id  Optional. The 16-character name for the convert rule identifier. If you enter *, the Copy Member name is used.
You can specify a convert rule identifier only if both a buffer identifier and a Rendezvous message identifier exist.

<table>
<thead>
<tr>
<th><strong>Library containing Copy Member</strong></th>
<th>The primary PDS that contains the selected copybook member. This value is required and cannot be blank.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional libraries for Copy</strong></td>
<td>The DSNs of the PDS that can contain other copybooks that are included within the copybook member.</td>
</tr>
<tr>
<td><strong>Parser Debug level</strong></td>
<td>The parser debug level. Retain the default value zero (0) unless TIBCO support personnel requests otherwise.</td>
</tr>
</tbody>
</table>
Log Viewer Panel

In the Log Viewer Panel, you select the recorded log and trace information written to a preallocated 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 specifying the values for Project, Group, and Type; or a fully qualified cluster name in the field Other VSAM Cluster Name. The value for the field Other VSAM Cluster Name is used if nonblank.

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

**Direction**
Specify 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 you specify Y, the log is treated as active (currently being updated by Substation ES). When active, the index record is reread 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 you do not specify a value, a default value of 200 is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VSAM Debug</strong></td>
<td>The debugging level for the VSAM access routines. Refrain from changing this value unless requested by TIBCO authorized personnel. The default is zero.</td>
</tr>
<tr>
<td><strong>Pgm Debug</strong></td>
<td>The debugging level for the ISPF interface program. Refrain from changing this value unless requested by TIBCO authorized personnel. The default is zero.</td>
</tr>
</tbody>
</table>
Log Entries Panel

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

Panel Layout

SXLP002---------------------- Log Entries ------------------------ Row 1 of 29
Command ===>                                                   Scroll ===> PAGE
Sel  Time       Type  Msg Id    RC  Rsn Message
- ------------- ---- -------- ---- ---- --------------------------------------
  11:09:54.0919  154 SXG1600I    0    0 Log Agent Starting - Logging to TIBLOG
  11:09:54.1126  154 SXG1800I    0    0 Parameter Log Agent Starting - Logging
  11:09:43.4368  154 SXS1000I    0    0 Starting ~ TIBCO Substation (ES) for 0
  11:09:43.4596  154 SXS1009I    0    0 Substation (ES) - Version 2.0.0
  11:09:54.7894  154 SXG2801I    0    0 Initialized ~ Substation ES - Admin &
  11:09:55.0454  154 SXG2800I    0    0 Starting ~ Substation ES - Transformer
  11:09:55.0479  0 SXT5717I    0    0 Trace level 1, 10 work threads; Cfg:T
  11:09:55.2871  154 SXG2801I    0    0 Initialized ~ Substation ES - Transfo
  11:09:55.3874  154 SXG2800I    0    0 Starting ~ Substation ES - CICS Interf
  11:09:55.4673  154 SXC3000I    0    0 CICSTS4G - Connect IIICICS to CICS Regi
  11:09:55.5539  154 SXC3001I    0    0 CICSTS4G - Connect IIICICS to CICS Regi
  11:09:55.5590  154 SXG2801I    0    0 Initialized ~ Substation ES - CICS Int
  11:09:55.5950  156 SXC3400I    0    0 CICSTS4G - Started CICS EXCI Session (1
  11:09:55.6130  156 SXC3400I    0    0 CICSTS4G - Started CICS EXCI Session (2
  11:09:55.6352  154 SXG2800I    0    0 Starting ~ Substation ES - IMS Interfa
  11:09:55.6612  154 SXI4000I    0    0 A01IMSG - Connect IIIMS to IMS Region
  11:09:55.7283  154 SXI4001I    0    0 A01IMSG - Connect IIIMS to IMS Region
  11:09:55.7326  154 SXG2801I    0    0 Initialized ~ Substation ES - IMS Inte
  11:09:55.7336  154 SXS1001I    0    0 Initialized ~ Substation ES for IVP's
  11:09:55.8350  154 SXG1800I    0    0 Parameter Log closed
  11:09:55.8362  154 SXS2120I    0    0 (B) HiMark:0 Occ:1024 Incr:0 Curr:0
  11:09:55.8363  154 SXS2121I    0    0 (F) Stress Levels Strt:768 Slow:819 St
  11:09:55.8408  154 SXS2120I    0    0 (A) HiMark:410 Occ:1024 Incr:102 Curr:
  11:09:55.8504  154 SXG2806I    0    0 Activated ~ Substation ES - CICS Inter
  11:09:55.8521  154 SXG2806I    0    0 Activated ~ Substation ES - Transforme
  11:09:55.8525  0 SXT5036I    0    0 All transformer tasks are Good To Go
  11:09:55.8525  154 SXG2806I    0    0 Activated ~ Substation ES - Admin & Op
  11:09:55.8531  154 SXG2806I    0    0 Activated ~ Substation ES - IMS Interf
  11:09:55.8610  154 SXG2806I    0    0 Activated ~ Substation ES for IVP's

Field Descriptions

Sel  If you specify s or x in the input column, a pop-up panel with all the values that
     relate to the selected message is shown.

Valid selections: s and x.
**Time**  The time at which this message was recorded by the Substation ES LTA Agent.

**Type**  The internal message type code.

**Msg Id**  The message identifier.

**RC**  The return code associated with the message.

**Rsn**  The reason code associated with the message.

**Message**  The initial part of the message description. The entire message is available if you select the pop-up panel.
Single Message Values Panel

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

Panel Layout

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

<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>
</tbody>
</table>

Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Msg Id</strong></td>
<td>An eight-character message identifier.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The internal message type.</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>The date on which this message was recorded by the Substation ES LTA Agent.</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>The time at which this message was recorded by the Substation ES LTA Agent.</td>
</tr>
<tr>
<td><strong>Union id</strong></td>
<td>A representation of what message fields were recorded for this entry. Refer to TIBCO Substation ES Messages and Codes for a description.</td>
</tr>
<tr>
<td><strong>GRIN</strong></td>
<td>The internal Global Resource Identification Number.</td>
</tr>
<tr>
<td><strong>Proc Id</strong></td>
<td>The internal process identifier for identifying a Substation ES task or subtask.</td>
</tr>
<tr>
<td><strong>Thread</strong></td>
<td>The Substation ES thread that issued the message.</td>
</tr>
<tr>
<td><strong>Stck</strong></td>
<td>The internal 64-bit Store Clock value that uniquely identifies this message.</td>
</tr>
<tr>
<td><strong>RC</strong></td>
<td>The return code associated with the message.</td>
</tr>
<tr>
<td><strong>Reason</strong></td>
<td>The reason code associated with the message.</td>
</tr>
<tr>
<td><strong>Buffer length</strong></td>
<td>A complete message description. If necessary, the message is displayed in multiple lines.</td>
</tr>
</tbody>
</table>
Chapter 2  Additional Configuration

You can use Substation ES to specify different interface configurations to meet various business requirements. For example, you can configure a single Substation ES instance to simultaneously communicate with multiple CICS regions. Alternatively, you can configure multiple Substation ES instances to communicate with the same CICS region. The configuration choice depends on your system requirements.

Topics

- Adding an Additional Substation ES CICS Interface, page 116
- Adding an Additional ESB Interface, page 117
- Adding Rendezvous ESB Interface With SSL, page 118
- Adding Substation ES Instances to a Single CICS Region, page 119
- Using Connection Factory, page 120
- Using RVDQ with Multiple Substation ES Instances, page 121
- Using RVDQ with One Substation ES Instance, page 123
- Using Opaque Field and Message Size, page 125
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 \texttt{USERHLQ.CNTL(SXCINTF)} to a new name.
   This new member must have a different \texttt{INTF-ID} keyword value. If connecting to a different CICS region, the \texttt{APPLID} keyword value must be changed.

3. Edit the Substation ES SIP member, the default being \texttt{USERHLQ.CNTL(SXSSIP$1)}.

4. Add a new \texttt{INTF-MEMBER} keyword, specifying the newly created CICS member as the keyword value. For example \texttt{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 \texttt{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 116.

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 specify identical `INTF-ID` values in the ESB SIP members.
Adding Rendezvous ESB Interface With SSL

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

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

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 ESB interface member
   `USERHLQ.CNTL(SXSIEMS1 or SXSIrv1)`. Make parameter changes that are necessary.

3. Optional. This step is required only if Substation ES Heartbeat communications will be used.
   Create a copy of the Substation ES SIP for Admin interface member
   `USERHLQ.CNTL(SXSIADM)`. In the new member, specify the corresponding ESB INTF-ID from step 2 to the new ESB-INTF-ID keyword value.

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

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

6. Change the Substation ID in the Substation ES SIP.

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

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

9. In the Substation ES startup JCL or Procedure, change the PARM card value to point to the newly created Substation ES SSP member, for example:
   `'-SSPMEM NEWSSP$1'`

To share one ESB configuration file with multiple instances of Substation ES, distinguish between recipes or triggers definitions for the Substation ES instances with groups. For details, see Defining Groups on page 21.

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

Each active Substation ES must have a unique Substation ID.

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

Substation ES uses the TIBCO Enterprise Message Service connection factory for all EMS connections. Substation ES creates a default Connection Factory Object when the ESB interface starts, then populates the Connection and Re-Connection Parameters using the definition from the Substation ES Configuration File.

Implementing Automatic Reconnect

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

```
ems_server_url, ems_server_url
```

Note that there are no spaces between the URL names. For example:

```
192.168.1.100:7888, 192.168.1.100:7888
```

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

- Attempts
- Delay
- Timeout

Use caution when setting these values or changing them from the product default settings because mainframe values need to be a little higher than a local connection.
Using RVDQ with Multiple Substation ES Instances

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

The steps described here configure three Substation ES instances:

1. Create three CICS SIP members, specifying the same INTF-ID for each. This INTF-ID must match the Interface Id defined in the ESB configuration file (on the Define Processing System Resources panel, option 4).

   In this example, the new CICS SIP members are named SXCINTF1, SXCINTF2 and SXCINTF3.

2. Create three RV ESB SIP members, setting RVDQ-YN set to Y for all three. Each must also have a unique INTF-ID.

   In this example, the new RV ESB SIP members are named SXSIRV1, SXSIRV2 and SXSIRV3.

3. Optional. This step is required only if Substation ES Heartbeat communications will be used.

   Create three copies of the Substation ES SIP for Admin interface members. Name the copies SXSIADM1, SXSIADM2 and SXSIADM3. In each new member, specify the corresponding ESB INTF-ID from step 2 to each new ESB-INTF-ID keyword value.

4. Create three Substation ES SIP members, specifying a unique SUBSTATION-ID for each, and entering INTF-MEMBER settings for the corresponding CICS SIP and RV ESB SIP members that were created in step 1 and step 2.

   In this example, the new Substation ES SIP members are named SXSSIP$1, SXSSIP$2 and SXSSIP$3:

   — SXSSIP$1 has the INTF-MEMBER for SXCINTF1, SXSIRV1, and SXSIADM1
   — SXSSIP$2 has the INTF-MEMBER for SXCINTF2, SXSIRV2, and SXSIADM2
   — SXSSIP$3 has the INTF-MEMBER for SXCINTF3, SXSIRV3, and SXSIADM3

5. Create three SSP members, specifying the SIPMEM setting for the corresponding Substation ES SIP members that were created in step 4.
In this example, the SSP members are named SXSSSP$1, SXSSSP$2 and SXSSSP$3:

- SXSSSP$1 has the SIPMEM for SXSSIP$1
- SXSSSP$2 has the SIPMEM for SXSSIP$2
- SXSSSP$2 has the SIPMEM for SXSSIP$3

6. Create three Substation ES startup JCL, using the same ESB configuration file for all three.

In this example, the Substation ES JCL members are named TIBSSES1, TIBSSES2 and TIBSSES3:

- in TIBSSES1 specify PARM='-SSPMEM SXSSSP$1'
- in TIBSSES2 specify PARM='-SSPMEM SXSSSP$2'
- in TIBSSES3 specify PARM='-SSPMEM SXSSSP$3'

7. Start all Substation ES instances.

All instances use RV distributed queue processing. The first started Substation ES instances acts as the scheduler.
This section describes the process taken to configure one Substation ES instance to use RV distributed queue processing to communicate with a single CICS region or multiple CICS regions.

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

1. Create three ESB configuration files. Each file must have a CICS BES Interface ID unique but otherwise contain identical recipe settings. The CICS BES Interface ID is set on the Transformer Configuration panel option 4.
   
   In this example, the CICS Interface IDs for the new ESB configuration files are named CICSA, CICSB and CICSC.

2. Create three CICS SIP members. Each member must have a unique INTF-ID. This INTF-ID must match the Interface Id defined in the corresponding ESB configuration file. If this is for multiple CICS regions then the APPLID must have the corresponding generic CICS application ID.
   
   In this example, the new CICS SIP members are named SXCINTF1, SXCINTF2 and SXCINTF3.

3. Create three RV ESB SIP members, setting RVDQ-YN to Y for all three. Each must also have a unique INTF-ID.
   
   In this example, the new RV ESB SIP members are named SXSIRV1, SXSIRV2 and SXSIRV3.

4. Create one Substation ES SIP member, entering all INTF-MEMBER settings for the corresponding CICS SIP and RV ESB SIP members.
   
   In this example, the new Substation ES SIP member is named SXSSIP$1.

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

6. Create one Substation ES startup JCL, using three ESB configuration files for the corresponding CICS SIP and RV ESB SIP members.
   
   In this example, the Substation ES JCL member is named TIBSES$1, and uses three configure files named TIBCFG1, TIBCFG2 and TIBCFG3:
   
   — TIBCFG1 corresponds to CICS SIP SXCINTF1 and RV ESB SIP SXSIRV1.
   
   — TIBCFG2 corresponds to CICS SIP SXCINTF2 and RV ESB SIP SXSIRV2.
   
   — TIBCFG3 corresponds to CICS SIP SXCINTF3 and RV ESB SIP SXSIRV3.
7. Start the Substation ES instance.
All interfaces use RV distributed queue processing. The last startup interface acts as the scheduler.
Using Opaque Field and Message Size

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

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

The conditions for this are:

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

The conversion then uses either the defined buffer size or the returned data size, based on the size field in the conversion rule, to create the outbound message.
This chapter describes the three utilities for the TIBCO Substation ES configuration file.

Topics

- Overview, page 128
- Conversion Utility, page 129
- Configuration File Utility, page 132
- Selective Export Utility, page 135
Before using Substation ES, you must supply the information in the configuration file to the Substation ES ESB Interface. The Substation ES ESB Interface requires this configuration file information to initialize successfully. Information includes data about connecting to messaging daemons and servers used for message communication.

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

A Configuration File must be allocated with IDCAMS. See Allocation of Configuration File on page 5 for details.

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

The Configuration File’s Conversion Utility extracts the configuration entities from a Substation ES configuration file and creates a sequential EXPORT file. You can then import the EXPORT file to a Substation ES configuration file with the Configuration File Utility. For details on how to create and allocate configuration files, see Allocation of Configuration File on page 5.

Function

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

These utility JCLs are included with Substation ES:

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

JCL Sample (SXSCFC26)

Execution is controlled by parm values and JCL statements.

```
//EXPORT EXEC PGM=SXT3CV26,REGION=0M,PARM='parm values'
// STEPLIB DD DISP=SHR,DSN=<USERHLQ>.LOAD
// TIBLOG DD SYSOUT=* 
// SYSPRINT DD SYSOUT=* 
// SYSOUT DD SYSOUT=* 
// CEEDUMP DD SYSOUT=* 
// SYSUDUMP DD SYSOUT=* 
// CONFIG DD DISP=SHR,DSN=old.config.file 
// EXPORT DD DISP=,(,CATLG),DSN=new.export.file, 
// UNIT=SYSDA,SPACE=(TRK,(20,20)), 
// DCB=(BLKSIZE=14000,LRECL=350,RECFM=VB) 
```
Table 3  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>

Parameter Values

Table 4 describes the values for the parm field of the EXEC statement. All the values are optional; the conversion process can proceed with no parm data other than SXSCFC26..

Table 4  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>Specifies the level of message display. The value for n can be between 0 and 5, with 0 providing very few messages and 5 showing the hex displays of all the records processed.</td>
</tr>
<tr>
<td>-Ixxx</td>
<td>Input DD name</td>
<td>CONFIG</td>
<td>Enables 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>Writes out the entire contents of CONFIG in export format to the EXPORT file. No selection is provided; the entire file is processed. Specify xxx as the DD name to replace EXPORT.</td>
</tr>
<tr>
<td>-P</td>
<td>Print</td>
<td>None</td>
<td>Prints each recipe and trigger with the associated Backend, ESB Endpoint, Group membership, and Conversion rules.</td>
</tr>
<tr>
<td>-R</td>
<td>Read-only</td>
<td>Update</td>
<td>Opens the CONFIG file as read-only. If combined with Import, all the updates from the import operation are lost.</td>
</tr>
<tr>
<td>-Tn</td>
<td>Table debug</td>
<td>2</td>
<td>Defines the diagnostic level for the internal TABLE processor, which ties together information from the various VSAM-keyed records.</td>
</tr>
</tbody>
</table>
Table 4  Transformer Conversion Utility Parameters

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Default</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-X</td>
<td>Hex dump</td>
<td>None</td>
<td>Sequentially reads and hex-displays the entire configuration file as a diagnostic aid.</td>
</tr>
</tbody>
</table>
Configuration File Utility

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

Functions

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

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

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

JCL Sample (SXSCFIMP and SXSCFEXP)

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

//STEP0001 EXEC PGMM=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)
//
//
Table 5 describes the DD names of the Configuration File Utility.

Table 5  DD Names of Configuration File Utility

<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, which is updated during a copy operation.</td>
</tr>
<tr>
<td>IMPORT</td>
<td>Points to an existing IMPORT file, which is read during an import operation.</td>
</tr>
<tr>
<td>EXPORT</td>
<td>Points to a new EXPORT file, which is written during an export operation. This file can be a preallocated disk file.</td>
</tr>
</tbody>
</table>

For a summary of the usages of the DD names, see Appendix B, Summary of Transformer Configuration File Utilities, page 197.

Parameters

Table 6 describes the values you can enter in the PARM field of the EXEC statement. In the absence of function action requests, the default action is print.

Table 6  Function Parameters of Configuration File Utility

<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>Copies all the definitions from one configuration file to another file defined by CONFIG2. The file referenced by CONFIG2 is initialized before copying starts.</td>
</tr>
<tr>
<td>-Mddn</td>
<td>Import</td>
<td>None, IMPORT</td>
<td>Reads the contents of the IMPORT file and apply to the CONFIG file. If you specify ddn, that DD name is used instead of IMPORT.</td>
</tr>
<tr>
<td>-Nddns</td>
<td>Export</td>
<td>None, EXPORT</td>
<td>Writes out the entire contents of CONFIG in export format to the EXPORT file. If you specify ddn, that DD name is used instead of EXPORT.</td>
</tr>
</tbody>
</table>
Table 7 describes the parameter values of the Configuration File Utility.

<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>Specifies the level of message display. The value for $n$ can be between 0 and 5, with 0 providing very few messages and 5 showing the hex displays of all the 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>Prints out each recipe and trigger with the associated Backend, ESB Endpoint, Group membership, and Conversion rules.</td>
</tr>
<tr>
<td>-R</td>
<td>Read-only</td>
<td>Update</td>
<td>Opens the CONFIG file as read-only. If combined with Import, all the 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 does not apply if you also specify read-only.</td>
</tr>
<tr>
<td>-V</td>
<td>Validate</td>
<td>None</td>
<td>Validates all the data structures for consistency and block identifiers (eyeball characters). If you specify -V -V, validation does not terminate after the first error occurs.</td>
</tr>
<tr>
<td>-X</td>
<td>Hex dump</td>
<td>None</td>
<td>Displays all the active definitions in the file in hex and does not show the relationships among the definitions.</td>
</tr>
<tr>
<td>-Z</td>
<td>Zero file</td>
<td>None</td>
<td>Resets the CONFIG file to empty before starting the requested processes. Note the warning below.</td>
</tr>
</tbody>
</table>

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

For a summary of the usage of the parameter options, see Appendix B, Summary of Transformer Configuration File Utilities, on page 197.
Selective Export Utility

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

Functions

The SXT3CSIE utility performs the following tasks:

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

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

This utility can also export “logical pieces” to a sequential file while deleting those entities from the configuration file. In that case, although the deletion request is entered during the export, it does not apply to the current file. The request is written to the export file and processed during the IMPORT process against the updated file by means of the Configuration File Utility.

JCL Sample (SXSCFEXS)

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

```
//STEP0002 EXEC PGM=SXT3CSIE,REGION=0M,PARM='values'
//STEPLIB DD DISP=SHR,DSN=USERHLQ.LOAD
//TIBLOG DD SYSOUT=*  
//SYSPRINT DD SYSOUT=* 
//SYSSOUT DD SYSOUT=* 
//CEEDUMP DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//CONFIG DD DISP=SHR,DSN=config.file 
//EXPORT DD DISP=(,CATLG),DSN=export.file, 
// UNIT=SYSDA,SPACE=(TRK,(20,20)), 
// DCB=(BLKSIZE=14000,LRECL=350,RECFM=VB) 
//CONTROL DD * 
(export control parameters) 
//
```
Table 8 describes the DD names of the Selective Export Utility.

**Table 8  DD Names of Selective Export Utility**

<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 preallocated disk file.</td>
</tr>
</tbody>
</table>

For a summary of the usages of DD names, refer to Appendix B, Summary of Transformer Configuration File Utilities, page 197.

**Parameter Values**

Table 9 describes the values you can enter possible in the PARM field of the EXEC statement.

**Table 9  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>Specifies the level of message display. The value for n can be between 0 and 5, with 0 providing very few messages and 5 showing the hex displays of all the 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>-P</td>
<td>Print</td>
<td>None</td>
<td>Prints out each export recipe and/or trigger with the associated Backend, ESB Endpoint, Group membership, and Conversion rules.</td>
</tr>
<tr>
<td>-R</td>
<td>Read-only</td>
<td>Update</td>
<td>Opens the CONFIG file as read-only mode.</td>
</tr>
</tbody>
</table>

For the parameter options, see Appendix B on page 197.
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*
- **REPLACE** *field* *Replacement-Value*
- **DELETE** *type* *identifier*

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

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

Entity Type Values

You must specify an entity *type* value in a export control statement. See Table 10 for the values.

**Table 10  Entity Type Values**

<table>
<thead>
<tr>
<th>Value</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>ESB</td>
<td>For ESB definitions only.</td>
</tr>
<tr>
<td>BUFFER</td>
<td>For buffer definitions only.</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>For 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

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

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

Qualifier Values

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

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

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

Table 11  Definition Qualifier Values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Exports all the related definitions of the selected type.</td>
</tr>
<tr>
<td>ONLY</td>
<td>Exports only the selected type definition and does not export the related structure definitions.</td>
</tr>
<tr>
<td>BUF</td>
<td>Exports the related buffer definitions for the selected type.</td>
</tr>
<tr>
<td>MSG</td>
<td>Exports the related message definitions for the selected type.</td>
</tr>
<tr>
<td>CVR</td>
<td>Exports the related Conversion Rule definitions for the selected type.</td>
</tr>
<tr>
<td>BES</td>
<td>Exports the related BES definitions for the selected type.</td>
</tr>
<tr>
<td>TRN</td>
<td>Exports the related ESB definitions for the selected type.</td>
</tr>
<tr>
<td>GRP</td>
<td>Exports the related Group definitions for the selected type.</td>
</tr>
</tbody>
</table>
REPLACE Control Parameter

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

<table>
<thead>
<tr>
<th>ENTITY Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKEND</td>
<td>Intf-Id</td>
<td>Interface Id field</td>
</tr>
<tr>
<td>ESB</td>
<td>URL</td>
<td>Primary URL definition</td>
</tr>
<tr>
<td></td>
<td>URL-ECF</td>
<td>EMS Connection Factory URL definition</td>
</tr>
<tr>
<td>RECIPE</td>
<td>BES-Name</td>
<td>Back-end system name specified for endpoints</td>
</tr>
<tr>
<td></td>
<td>BES-ALT-Name</td>
<td>Back-end system name specified for endpoints</td>
</tr>
<tr>
<td></td>
<td>CICS-System-Name</td>
<td>CICS AOR routing system identifier</td>
</tr>
<tr>
<td>ESB</td>
<td></td>
<td>ESB name specified for endpoints</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>ESB</td>
<td>ESB name specified for endpoints</td>
</tr>
</tbody>
</table>

Examples

Here are a few examples:

- To export a recipe definition named rcp1 and all its related structures:
  ```
  EXPORT RECIPE rcp1
  ```

- To export a recipe definition named rcp1 and all its related but completely qualified structures instead of accepting the default qualifier ALL:
  ```
  EXPORT RECIPE rcp1 BUF MSG CVR BES TRN GRP
  ```

- To export an entire file:
  ```
  EXPORT ALL * ONLY
  ```
  RVY3CFUT with the parameter -N has the same capability.

- To export all ESB definitions only:
  ```
  EXPORT ESB * ONLY
  ```
• To delete a recipe definition named rcp1 and all its related structures:
  
  DELETE RECIPE rcp1

• To delete a buffer definition named buf2 and all its related structures:
  
  DELETE BUFFER buf2
Chapter 4  

**Examples**

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

**Topics**

- Overview, page 142
- Substation ES Installation IVPs, page 143
- Substation ES Communication Programs, page 144
- Example of ESB Request – Reply DPL, page 155
- Example of CICS Trigger, page 162
- Example of CICS High Volume Trigger (HVT), page 163
- Example of ESB Request DPL XCTL, page 164
- Example ESB Invoking a CICS Transaction, page 165
- Example of ESB to TDQ, page 166
- Example of CICS TSQ Entry, page 167
- Example of ESB Request – Reply Complex DPL, page 168
- Example of CICS-Initiated Request or Reply, page 169
- Example of ESB Request – Reply IMS, page 171
- Example of ESB Request – Reply and Trigger, page 173
- Example of IMS BMP Trigger, page 175
- Example of IMS Conversational Functions, page 176
- Examples of Transform-Only Data, page 178
Overview

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

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

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

Also included is an Error listener source SXR3IELS, which listens for errors published by instances of TIBCO Substation ES. For summary usage of sample IVPs resources, refer to Appendix A, Sample IVP Resources, on page 193.

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

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

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

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

Request or Reply

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

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

Triggers

Triggers are BES (CICS or IMS) applications that initiate published data from their respected environments. Substation ES can publish data on behalf of these BES (CICS or IMS) applications. In this case, the data is translated from the z/OS data format and packaged as a TIBCO Rendezvous message for delivery to an external application that is subscribing to this data.

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

This section describes the generic IVP components in Substation ES.

Rendezvous IVP Sender Listener

The program SXR3ISL1 works as a TIBCO messaging requestor application. It publishes Substation ES IVP requesting messages based on a set of parameters passed in through SYSIN DD statements in the run JCL and then waits for a reply from Substation ES. If an error occurs, the process stops; otherwise, the process loops on the number specified in the REPEAT parameter.

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

- **Data Set:** USERHLQ.JCL
- **Member:** SXCRIRSL for CICS Interface IVPs
  
  SXIRIRSL for IMS Interface IVPs

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

SYSIN Parameters

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

- **Data Set:** USERHLQ.CNTL
- **Member:** SXCRIP01 for CICS Interface IVPs
  
  SXIRIP01 and SXIRIP02 for IMS Interface IVPs

Following is an example of the input control statements 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 in the **SYSIN DD Name file**. See *Table 12* for the control parameters.
Table 12  Rendezvous IVP Sender Listener Control Parameters

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BES <code>name</code></td>
<td>The BES interface ID for the tibss-BES system field. The default is a blank.</td>
</tr>
<tr>
<td>BES-DELAY <code>timevalue</code></td>
<td>The length of time (in seconds) to wait between message replies on the CICS side. The value must fall between 0 and 60. The default is 0.</td>
</tr>
<tr>
<td>CLIENT-ID <code>name</code></td>
<td>The identifier to use with the Rendezvous daemon connection.</td>
</tr>
<tr>
<td>DAEMON <code>URL:port</code></td>
<td>The Rendezvous daemon URL and TCP port.</td>
</tr>
<tr>
<td>DEBUG `ON</td>
<td>OFF`</td>
</tr>
<tr>
<td>DESCRIPTOR <code>name</code></td>
<td>The 56 characters for the field IVP-DESCR.</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>The 8 characters for the field IVP-ID.</td>
</tr>
<tr>
<td>NETWORK <code>address</code></td>
<td>The Rendezvous network parameter.</td>
</tr>
<tr>
<td>PATTERN-LEN <code>number</code></td>
<td>The creation of the variable field IVP-STRING of length $number$, populated with 1--------10--------. The default is 0.</td>
</tr>
<tr>
<td>PASSWORD <code>password</code></td>
<td>The password for the user ID previously specified and authorized to execute this process. This password depends on the Security Interface and is for the tibss-password system field. The default is a blank.</td>
</tr>
<tr>
<td>PUBLISH-ONLY</td>
<td>When included, the application sends messages but does not listen for replies.</td>
</tr>
<tr>
<td>REPEAT <code>number</code></td>
<td>The number of times to repeat the send process. The default is 1.</td>
</tr>
<tr>
<td>RESOURCE <code>name</code></td>
<td>The name of the tibss-resource system field, up to a maximum of eight characters. The default is a blank.</td>
</tr>
<tr>
<td></td>
<td>If present, this value overrides the recipe Resource Name field, which depends on the Method on Invocation. For details, see Resource Name in Recipe Details Panel on page 59.</td>
</tr>
<tr>
<td>SERVICE <code>port</code></td>
<td>The port number of the Rendezvous UDP service.</td>
</tr>
<tr>
<td>SUBJ-REP <code>name</code></td>
<td>The name of the Rendezvous reply subject to which the reply message is sent. If not PUBLISH-ONLY and SUBJ-REP is absent, then a private inbox subject name is used.</td>
</tr>
<tr>
<td>SUBJ-REQ <code>name</code></td>
<td>The Rendezvous request subject name to which the message is sent.</td>
</tr>
<tr>
<td>Keyword – Operand</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>TIMEOUT timevalue</td>
<td>The length of time (in seconds) for the response timeout. The default is 5.</td>
</tr>
<tr>
<td>USER userid</td>
<td>The RACF or back-end identifier for the user who is authorized to execute this process. This Id depends on the Security Interface. For the tibss-userid system field; the default is a blank.</td>
</tr>
<tr>
<td>WAIT timevalue</td>
<td>The length of time (in seconds with decimals) to wait between message sends. The default is 0.</td>
</tr>
<tr>
<td></td>
<td>For example: 1.5</td>
</tr>
<tr>
<td>* text</td>
<td>A comment.</td>
</tr>
</tbody>
</table>
**TIBCO Enterprise Message Service IVP Sender Listener**

The program **SXJ3ISL1** works as a TIBCO messaging requestor application. It publishes Substation ES IVP requesting messages according to a set of parameters passed in through **SYSIN DD** statements in the run JCL, then waits for a reply from Substation ES. If an error occurs, the process stops; otherwise, the process loops on the number specified in the **REPEAT** parameter.

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

Data Set: `USERHLQ.JCL`
Member: `SXCRIJSL` for CICS Interface IVPs
        `SXIRIJSL` for IMS Interface IVPs

**SYSIN Parameters**

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

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

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

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

The requesting messages that the IVP Consumer Producer publishes are determined by the keywords and operands in the **SYSIN DD Name** file. The control parameters are described in **Table 13**.

**Table 13  Control Parameters for EMS IVP Sender Listener**

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BES name</td>
<td>The BES interface ID for the tibss-BES system field. That ID can be sent as a user property field or MAP message. The default is a blank.</td>
</tr>
</tbody>
</table>
### Control Parameters for EMS IVP Sender Listener (Cont’d)

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BES-DELAY <code>timevalue</code></td>
<td>The length of time (in seconds) to wait between message replies on the CICS side. The value must fall between 0 and 60. The default is 0.</td>
</tr>
<tr>
<td>CLIENT-ID <code>name</code></td>
<td>The identifier to use with the EMS connection.</td>
</tr>
<tr>
<td>DEBUG `ON</td>
<td>OFF`</td>
</tr>
<tr>
<td>DEST-REP <code>name</code></td>
<td>The name of the EMS destination to which the reply is to be sent. If <code>PUBLISH-ONLY</code> and <code>DEST-REP</code> is absent, a temporary topic or queue will be used.</td>
</tr>
<tr>
<td>DEST-REQ <code>name</code></td>
<td>The name of the EMS subject destination to which the message is sent.</td>
</tr>
<tr>
<td>HOSTPAGE <code>code page</code></td>
<td>The value for setting the host code page. The default is <code>IBM-1047</code>.</td>
</tr>
<tr>
<td>DESCR <code>name</code></td>
<td>The 56 characters for the field <code>IVP-DESCR</code>.</td>
</tr>
<tr>
<td>EMS-PSWD <code>password</code></td>
<td>The password to connect to the EMS server.</td>
</tr>
<tr>
<td>EMS-USER <code>userid</code></td>
<td>The user ID to connect to the EMS server.</td>
</tr>
<tr>
<td>IVP-ID</td>
<td>The maximum eight characters for the field <code>IVP-ID</code>.</td>
</tr>
<tr>
<td>MSGTYPE `MAP</td>
<td>BYTES`</td>
</tr>
<tr>
<td>NETPAGE <code>code page</code></td>
<td>The value for setting the network code page. The default is <code>ISO8859-1</code>.</td>
</tr>
<tr>
<td>PATTERN-LEN <code>number</code></td>
<td>The creation of the MAP message field <code>IVP-STRING</code> of length <code>number</code>, populated with <code>1--------10--------....</code> The default is 0. For <code>MSGTYPE=BYTES</code>, the pattern message replaces the standard IVP structure. The length of the bytes message is <code>PATTERN-LEN</code>.</td>
</tr>
<tr>
<td>PROPERTIES `YES</td>
<td>NO`</td>
</tr>
<tr>
<td>PROPERTY <code>Name</code> <code>Value</code></td>
<td>The property field <code>Name</code> with <code>Value</code> that is to be added to a message as user property field, or a mapped message field. The field type depends on the <code>PROPERTIES</code> setting.</td>
</tr>
</tbody>
</table>
### Table 13  Control Parameters for EMS IVP Sender Listener (Cont’d)

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSWD  <code>password</code></td>
<td>The password for the tibss-password user property field or MAP message field. The default is a blank.</td>
</tr>
<tr>
<td>PUBLISH-ONLY</td>
<td>A directive for the program to only send a message and not listen for a reply.</td>
</tr>
<tr>
<td>REPEAT <code>number</code></td>
<td>The number of times to repeat the send process. The default is 1.</td>
</tr>
<tr>
<td>RESOURCE <code>name</code></td>
<td>The name of the tibss-resource user property or MAP message field, up to a maximum of eight characters. The default is a blank. If 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 59.</td>
</tr>
<tr>
<td>SERVER <code>URL:port</code></td>
<td>The URL and TCP port number of the EMS server.</td>
</tr>
<tr>
<td>TIMEOUT <code>timevalue</code></td>
<td>The length of time (in seconds) for the response timeout. The default is 5.</td>
</tr>
<tr>
<td>TYPE `TOPIC</td>
<td>QUEUE`</td>
</tr>
<tr>
<td>USER <code>userid</code></td>
<td>The user ID for the tibss-userid system field. The ID can be sent as a user property or MAP message field. The default is a blank.</td>
</tr>
<tr>
<td>WAIT <code>timevalue</code></td>
<td>The length of time (in seconds) to wait between message sends. The default is 0. The length of time (in seconds with decimals) to wait between message sends. The default is 0. For example, to wait one and a half seconds between sends, specify: 1.5</td>
</tr>
<tr>
<td><code>* text</code></td>
<td>A comment.</td>
</tr>
</tbody>
</table>
IVP Request or Reply TIBCO Enterprise Message Service Consumer Producer

The program SXJ3RRCP works as a TIBCO messaging transactional server application. The Consumer Producer receives request messages published by a Substation ES CICS or IMS initiated request trigger process. The Consumer Producer then sends back a response message to the reply destination which that either came with the request message, or the destination that was provided in the SYSIN parameter RESPONSE-DEST. The response message type is determined by the inbound request message type.

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

Data Set: USERHLQ.JCL
Member: SXJ3RRCP

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

SYSIN Parameters

The response messages sent by the utility are determined by the keywords and operands specified in the SYSIN DD Name file. See Table 14.

Table 14  IVP Request or Reply EMS Consumer Producer Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIENT-ID</td>
<td>name</td>
<td>The client identifier to use with the EMS connection.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>DBG</td>
<td>OFF</td>
</tr>
<tr>
<td>DELAY</td>
<td>timevalue</td>
<td>The length of time (in seconds) to wait between receive and reply processes. The value must fall between 0 and 60. The default is 0, indicating no wait.</td>
</tr>
<tr>
<td>HOSTPAGE</td>
<td>code page</td>
<td>The value for setting the host code page.</td>
</tr>
<tr>
<td>NETPAGE</td>
<td>code page</td>
<td>The value for setting the network code page.</td>
</tr>
<tr>
<td>PSWD</td>
<td>password</td>
<td>The password associated with the user ID that is used to connect to the EMS server.</td>
</tr>
<tr>
<td>REQUEST-DEST</td>
<td>destination</td>
<td>Specifies the EMS destination name on which messages are received.</td>
</tr>
</tbody>
</table>
Table 14  IVP Request or Reply EMS Consumer Producer Parameters

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

Data Set: USERHLQ.JCL
Member: SXRGILS

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

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

SYSIN Parameters

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

```
DEBUG OFF
subject tibss.ivp.>
subject tibss.ibm.>
UFLD-PFX tibss-
```
IVP IMS Conversational Sender or Listener

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

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

Data Set: USERHLQ.JCL
Member: SXIRICVS

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

The programs SXR3IELS and SXJ3IELS initiate a TIBCO messaging subscriber that receives error messages published by Substation about data transformation errors, warnings, and error conditions encountered during Substation ES IPV’s runtime processing. The sample JCL stream for error listeners is in the following:

Data Set: USERHLQ.JCL
Member: SXRRIELS for Rendezvous
       SXJRIELS for TIBCO Enterprise Message Service

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

SYSIN Parameters

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

Data Set: USERHLQ.CNTL
Member: SXRPERLS for Rendezvous
       SXJPERLS for TIBCO Enterprise Message Service

Table 15 describes the control parameters.

Table 15: Control Parameters for IPV Generic Error Listener

<table>
<thead>
<tr>
<th>Keyword – Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJECT {tibss.error.}&gt;</td>
<td>The IPV error subject names to be listened to.</td>
</tr>
<tr>
<td>QUEUE or TOPIC {tibss.error.out}</td>
<td>TIBCO Enterprise Message Service only. The subject destination on which an error message is published if Transformer conversion or BES encounters errors.</td>
</tr>
<tr>
<td>PSWD</td>
<td>TIBCO Enterprise Message Service only. The password associated with the user ID if used in the session.</td>
</tr>
<tr>
<td>UFLD-PFX {tibss-}</td>
<td>The prefix of the Substation ES system field name for the processing status. We recommend that you do not change this value for IPV runs.</td>
</tr>
<tr>
<td>USER</td>
<td>TIBCO Enterprise Message Service only. The user-specified identifier if used in the session.</td>
</tr>
</tbody>
</table>

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

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

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

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

Component Summary

For the C language:

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

For the COBOL language:

- **SXCRIJSL or SXCRIRSL** – The publisher that sends the request message
- **SXC-DPL-COBOL** – The Substation ES recipe identification
- **SXC-DPL-MRO** – The Substation ES recipe identification for the DPL MRO Method of Invocation
- **SXCCI001** – The CICS COBOL application program
Execution Parameters of IVP Sender Listener

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

Rendezvous – DPL Request/Reply (LANGUAGE C)

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

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

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

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

```
DEST-REQ   tibss.CICS.Cobol.Request
DEST-REP   tibss.CICS.Cobol.Reply
TYPE       QUEUE
RESOURCE   SXCC1001
```

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

```
DEST-REQ   tibss.CICS.MRO.Request
DEST-REP   tibss.CICS.MRO.Reply
TYPE       QUEUE
RESOURCE   SXCC1001
```
Example of ESB Request – Reply DPL Container

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

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

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

Component Summary

For COBOL:
- SXCRJSL or SXCRIRSL – The publisher that sends the request message
- SXC-DPL-COBOL-Container – The Substation ES recipe identification
- SXCCI014 – The CICS COBOL application program

Execution Parameters of IVP Sender Listener

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

Rendezvous – DPL Container Request/Reply (LANGUAGE COBOL)

```plaintext
SUBJ-REQ    tibss.CICS.Cobol.Container.Request
RESOURCE    SXCCI014
REPEAT      1
TIMEOUT     300000
WAIT        0
BES-DELAY   5
```

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

```plaintext
DEST-REQ    tibss.CICS.Cobol.Container.Request
DEST-REP    tibss.CICS.Cobol.Container.Reply
TYPE        QUEUE
RESOURCE    SXCCI014
```
Example of ESB Request – Reply DPL Storage

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

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

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

Component Summary

For the C language:
- SXCRJSL / SXCRIRSL – The publisher that sends the request message
- SXC-DPL-C-Storage – The Substation ES recipe identification
- SXC3I013 – The CICS C application program

For the COBOL language:
- SXCRJSL / SXCRIRSL – The publisher that sends the request message
- SXC-DPL-COBOL-Storage – The Substation ES recipe identification
- SXCC1013 – The CICS COBOL application program

Execution Parameters of IVP Sender Listener

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

Rendezvous – DPL Storage Request/Reply (LANGUAGE C)

```plaintext
SUBJ-REQ       tibss.CICS.C.STG.Request
RESOURCE       SXC3I013
REPEAT         1
TIMEOUT        300000
WAIT           0
BES-DELAY      5
```
Rendezvous – DPL Storage Request/Reply (LANGUAGE COBOL)

SUBJ-REQ   tibss.CICS.Cobol.STG.Request
RESOURCE   SXCCI013
REPEAT     1
TIMEOUT    300000
WAIT       0

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

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

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

DEST-REQ   tibss.CICS.Cobol.STG.Request
DEST-REP   tibss.CICS.Cobol.STG.Reply
TYPE       QUEUE
RESOURCE   SXCCI013
Example of ESB Request – Reply DPL TSQ

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

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

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

Component Summary

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

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

Execution Parameters of IVP Sender Listener

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

Rendezvous – DPL TSQ Request/Reply (LANGUAGE C)

```
SUBJ-REQ   tibss.CICS.C.TSQ.Request
RESOURCE   SXC3I012
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
```
Rendezvous – DPL TSQ Request/Reply (LANGUAGE COBOL)

SUBJ-REQ  tibss.CICS.Cobol.TSQ.Request
RESOURCE  SXCCI012
REPEAT    1
TIMEOUT   300000
WAIT      0

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

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

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

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

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

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

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

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

Component Summary

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

CICS Execution Command

The command for executing transactions entered from a CICS 3270 screen is **SXTT.**
Example of CICS High Volume Trigger (HVT)

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. The Substation ES Transformer then processes the data conversion according to the selected conversion rule in the trigger recipe.

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

Component Summary

For the C language:

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

For the COBOL language:

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

CICS Execution Command

The command for executing transactions entered from a CICS 3270 screen is SXTF or SXT6. For details, see High Volume Trigger in the TIBCO Substation ES Operations and Administration manual.
Example of ESB Request DPL XCTL

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

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

Execution Parameters of IVP Sender Listener

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

Rendezvous

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJ-REQ</td>
<td>tibss.CICS.Cobol.Program.Request</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXCCI003</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>SXCIV03</td>
</tr>
</tbody>
</table>

TIBCO Enterprise Message Service

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REQ</td>
<td>tibss.CICS.Cobol.Program.Request</td>
</tr>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.Cobol.Program.Status</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXCCI003</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>SXCIV03</td>
</tr>
</tbody>
</table>
Example ESB Invoking a CICS Transaction

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

The CICS program replies by writing the reply information into the Substation ES trigger TDQ. Substation ES gets the trigger based on the defined search criteria for this trigger process and then delivers the reply and an acknowledgement 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

- SXCRJSL or SXCRIRSL — The publisher that sends the request message
- SXC-DPL-COBOL-START-TRAN — The Substation ES recipe identification
- SXCCI004 — The CICS COBOL application program
- TRIGGER-FLDS — The Substation ES trigger identification

Execution Parameters of IVP Sender Listener

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

Rendezvous

```
SUBJ-REQ   tibss.CICS.Task.Request
RESOURCE   SXTI
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
IVP-ID     SXCIV04
```

TIBCO Enterprise Message Service

```
DEST-REQ   tibss.CICS.Task.Request
DEST-REP   tibss.CICS.Task.Status
TYPE       QUEUE
RESOURCE   SXTI
REPEAT     1
TIMEOUT    300000
WAIT       0
BES-DELAY  5
IVP-ID     SXCIV04
```
Example of ESB to TDQ

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

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

Component Summary

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

Execution Parameters of IVP Sender Listener

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

Rendezvous

<table>
<thead>
<tr>
<th>SUBJ-REQ</th>
<th>tibss.CICS.TDQ.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE</td>
<td>SXQT</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>SXCIV05</td>
</tr>
</tbody>
</table>

TIBCO Enterprise Message Service

<table>
<thead>
<tr>
<th>DEST-REQ</th>
<th>tibss.CICS.TDQ.Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REP</td>
<td>tibss.CICS.TDQ.Status</td>
</tr>
<tr>
<td>TYPE</td>
<td>QUEUE</td>
</tr>
<tr>
<td>RESOURCE</td>
<td>SXQT</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>SXCIV05</td>
</tr>
</tbody>
</table>
Example of CICS TSQ Entry

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

Substation ES delivers an acknowledgement 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, see the Messages Flow section of the *TIBCO Substation ES Concepts* manual.

Component Summary

- **SXCRIJSL or SXCRIRSL** — The publisher that sends the request message
- **SXC-DPL-WRITE-TSQ** — The Substation ES recipe identification
- **SXCTSQI1** — The name of a CICS TSQ Substation ES temporary storage queue

Execution Parameters of IVP Sender Listener

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

**Rendezvous**

```
SUBJ-REQ  tibss.CICS.TSQ.Request
RESOURCE  SXCTSQI1
REPEAT    1
TIMEOUT   300000
WAIT      0
BES-DELAY 5
IVP-ID    SXCIV06
```

**TIBCO Enterprise Message Service**

```
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
```
Example of ESB Request – Reply Complex DPL

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

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

Component Summary

- SXCRJSL or SXCRIRSL — The publisher that sends the request message
- SXC-DPL-COBOL-COMPLEX — The Substation ES recipe identification
- SXCCI010 — The CICS COBOL application program

Execution Parameters of IVP Sender Listener

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

**Rendezvous**

<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>

**TIBCO Enterprise Message Service**

<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>
Example of CICS-Initiated Request or Reply

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

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

The Substation ES reply program is invoked by the reply message. The reply data is delivered to the originating CICS program. In both directions, the Substation ES Transformer converts the data according to conversion rules in the recipes and trigger.

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

Component Summary

For the C language:

- SXR3 — CICS transaction ID of the request initiator
- SXC31RR1 — The CICS C application program
- CICS-RR-REQUEST — The Substation ES trigger identification for initiating requests
- SXC-RR-EXTRN-PROCESS — The Substation ES recipe identification for generating replies
- SXCCIRR3 — The CICS COBOL application program, which receives a request from an application through DPL and replies through the COMMAREA
- SXC-RR-ANSWER — The Substation ES recipe identification for receiving replies
- SXCREPLY — The Substation ES Request or Reply module.

For the COBOL language:

- SXRC — The CICS transaction ID of the request initiator
- SXCCIRR1 — The CICS COBOL application program
- CICS-RR-REQUEST — Substation ES trigger identification for initiating requests
- **SXC-RR-EXTRN-PROCESS** — The Substation ES recipe identification for generating replies
- **SXCCIRR3** — The CICS COBOL application program, which receives a request from an application through DPL and replies through the COMMAREA
- **SXC-RR-ANSWER** — The Substation ES recipe identification for receiving replies
- **SXCREPLY** — The Substation ES Request or Reply module.

**CICS Execution Command**

COBOL entered from a CICS 3270 screen: SXRC
C entered from a CICS 3270 screen: SXR3
Example of ESB Request – Reply IMS

This IVP example illustrates how a TIBCO messaging client application initiates a request message to invoke an IMS Message Processing Program (MPP). The client uses 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. Substation ES picks up the data from there and delivers the reply to the IVP Sender Listener. In both directions, the Substation ES Transformer processes the data conversions based on the selected conversion rules in the recipe.

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

Component Summary

- SXIRIJSL or SXIRIRSL — The JCL member that submits the request initiator
- SXIR3ISL1 or SXJ3ISL1 — The IVP Sender Listener that sends the request message
- SXI-TRAN — The Substation ES recipe identification
- SXICTT01 — The IMS transaction identification
- SXICIP01 — The IMS COBOL message processing program

Execution Parameters of IVP Sender Listener

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

Rendezvous

SUBJ-REQ    tibss.IMS.Tran.Request
REPEAT      1
TIMEOUT     300000
WAIT        0
BES-DELAY   5

TIBCO Enterprise Message Service

DEST-REQ    tibss.IMS.Tran.Request
DEST-REP    tibss.IMS.Tran.Reply
TYPE        QUEUE
REPEAT      1
TIMEOUT        300000
WAIT           0
Example of ESB Request – Reply and Trigger

This example illustrates how a TIBCO messaging client application initiates a request message to invoke an IMS Message Processing Program (MPP) by using the IMS message queue through Substation ES.

The IMS MPP replies by putting information into the IO-AREA and inserting it to the IMS message queue. In addition, the MPP uses ALT-PCB to insert the output data to the destined Substation ES trigger TPipe. Substation ES picks up that data and delivers the reply and trigger messages to the IVP Sender Listener. In both directions, the Substation ES Transformer processes the data conversion according to the selected conversion rules in the recipe.

For more information on the process flow, see the TIBCO Substation ES Concepts manual.

Component Summary

- SXRIJSL or SXRIRSL — The JCL member for that submits the request initiator
- SXR3ISL1 or SXR3ISL1 — The IVP Sender Listener sending of that sends the request message
- SXI-TRAN-AND-TRG — The Substation ES recipe identification
- SXICITO2 — The IMS transaction identification
- SXICIP02 — The IMS COBOL message processing program

A version of this program, SXICIP03, demonstrates the use of variable-length IMS transaction codes (not using the first 8 bytes of the message).

Execution Parameters of IVP Sender Listener

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

Rendezvous

```plaintext
SUBJ-REQ tibss.IMS.Tran-Trig.Request
REPEAT 1
TIMEOUT 300000
WAIT 0
BES-DELAY 5
```
### TIBCO Enterprise Message Service

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEST-REQ</td>
<td>tibss.IMS.Tran-Trig.Request</td>
</tr>
<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>TIMEOUT</td>
<td>300000</td>
</tr>
<tr>
<td>WAIT</td>
<td>0</td>
</tr>
</tbody>
</table>
Example of IMS BMP Trigger

This example illustrates how a trigger process delivers data from an IMS Batch Message Program (BMP) or a 3270 device to a TIBCO messaging application through Substation ES.

The JCL is supplied with the installation of Substation ES. The IMS BMP Trigger example uses this JCL to invoke the IMS transaction.

The IMS BMP initiates the request by inserting the output data to the IMS message queue (TPipe). The TPipe delivers the IMS outbound events to Substation ES, which gets the trigger based on the defined search criteria for this trigger process. Substation ES Transformer then processes the data conversions according to the selected conversion rule in the trigger recipe.

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

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

Component Summary

- SXIVPBT — The JCL member that submits the batch request initiator
- SXICITB1 — The IMS transaction identification
- SXICIPB1 — The IMS COBOL batch message program
- SXI-TRIGGER — The Substation ES IMS interface trigger identification

Execution Parameters of JCL

```bash
//STEP1BT EXEC IMSBATCH,MBR=SXICIPB1,PSB=SXICIPB1,IMSID=&IMSID,
    // IN=SXICITB1,APARM='000001SXITRG$RYNSXIIVB

Alternatively:

//STEP1BT EXEC IMSBATCH,MBR=SXICIPB1,PSB=SXICIPB1,IMSID=&IMSID,
    // IN=SXICITB1,APARM='000001SXITRG$GYNSXIIVB
```
Example of IMS Conversational Functions

This example illustrates how a TIBCO messaging transactional client application initiates request messages to perform the IMS conversational functions through Substation ES. This example uses the IBM IMS INSTALL/IVP sample of a simple IBM phone-book application. All functions performed by that application can be executed with Substation ES IMS Conversational IVP.

You must install the phone-book application before running this IMS Conversational example. Refer to IMS Installation Vol 1: Installation Verification for installation requirements for all programs, PSBs, DBDs, MFSs, and other supporting materials used by the application.

You invoke the IMS Conversational Sender or Listener by submitting JCL with a set of function operand parameters.

The IMS Conversational Sender or Listener initiates each function, sending the request message through Substation ES to instruct the IMS application. The IMS application sends a reply message back to confirm the process status of each function. Substation ES then delivers the reply to the IMS conversational Sender or Listener and continues the conversation with the IMS application until it receives the END confirmation message from IMS.

In both directions, the Substation ES Transformer converts the data according to the conversion rules in the recipe and then publishes the output message to the IMS Conversational Listener.

For more information on the process flow, see the Messages Flow for IMS section of TIBCO Substation ES Concepts manual.

Component Summary

- SXIRICVS or SXIJCJS — JCL member that submits the IMS Conversational Sender or Listener
- SXR3ICV1 or SXJ3ICV1 — The C TIBCO transactional client application program that initiates and receives conversational request and reply messages
- SXI-CONVERSATIONAL — The Substation ES recipe identification
- IVTCB — The IBM phone book sample IMS transaction code
- DFSIVP34 — The COBOL, IBM phone book application program

When you run these IVP components, the IVP Generic Error Listener is present and listening to the subject tibss.ibm.ims.cvrs.reply.
Execution Parameters of IVP IMS Conversational Sender or Listener

Table 16 describes control parameters.

**Table 16  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>Operand,Fields</td>
<td>One message entry for a specific IMS program function in IBM sample phone book application.</td>
</tr>
<tr>
<td>ADD</td>
<td>last name,first name,ext #,ZIP code</td>
<td>Performs the ADD function to add this new record to the IMS database.</td>
</tr>
<tr>
<td>DEL</td>
<td>last name</td>
<td>Performs the DEL function of deleting this record from the IMS database.</td>
</tr>
<tr>
<td>DIS</td>
<td>last name</td>
<td>Performs the DIS function of displaying this record from the IMS database.</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td>Stipulates the sending of the message to terminate the current transactional process.</td>
</tr>
<tr>
<td>UPD</td>
<td>last name,first name,ext #,ZIP code</td>
<td>Performs the UPD function for updating 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 publish the correspond requesting message.

```plaintext
-ENTRY ADD, PEPPER, JOHN, EXT77, Z94304
-ENTRY ADD, SALTY, SANDRA, EXT44, Z94001
-ENTRY DIS, CINAMON
-ENTRY DIS, PEPPER
-ENTRY DEL, SALTY
-ENTRY ADD, CINAMON, CINDY, EXT88, Z94001
-ENTRY UPD, CINAMON, CINDY, EXT11, Z94999
-ENTRY DEL, SALTY
-ENTRY DIS, CINAMON
-ENTRY DEL, CINAMON
-ENTRY DEL, PEPPER
-ENTRY END
```

The complete output from the IMS Conversational IVP is in the following:

Data Set:  USERHLQ.DATA
Member:  SXIIOCVS
Examples of Transform-Only Data

These examples illustrate how Substation ES can transform data from different mainframe sources (databases or files) to TIBCO messaging applications.

These processes support transformations in both directions. Additionally, Substation ES can easily transformation data from platforms that publish a TIBCO Rendezvous or TIBCO Enterprise message to a mainframe format (databases or files). When the data has been transformed, it can be stored to a permanent medium by a user procedure.

Here are two examples:

- **Data Forwarder**, which transforms data from the mainframe to a message. This example uses a simple TIBCO messaging publisher to read records from a PDS file and to 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. 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.

- **Data Receiver**, which transforms from a message to the mainframe format. In this example, the Data Receiver receives the outbound data from Substation ES and writes the input to a file. The data received would have been received by Substation ES as a message, transformed to an opaque data type, and published to the Data Receiver. In the inbound direction, the Substation ES Transformer converts the data according to the 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, see the Messages Flow section of the *TIBCO Substation ES Concepts* manual.

Component Summary

For the Data Forwarder example:

- **SXRI1GDF** — The JCL member that submits the Data Forwarder sender
- **SXRI3GDF** — The C Publisher that reads the data records from a file and sends each record as a message in Opaque data type until end of the file
- **USERHLQ.DATASXGFL2RV** — Contains the data to be forwarded.
- **SXS-TRANSFORM-ONLY-FORWARD** — The Substation ES recipe identification
For the Data Receiver example:

- Output messages from the Data Receiver example
- `SXS-TRANSFORM-ONLY-RECEIVE` — The Substation ES recipe identification
- `SXRIGDR` — The JCL member that submits the Data Receiver listener
- `SXR3GDR` — The C Subscriber that listens to each message and writes each data record into a file.
- `USERHLQ.DATA(SXGRV2M1)` — The data received after the listener has been stopped.

**Execution of Data Forwarder and Receiver**

The Data Receiver must be started before the Data Forwarder because the records that have been forwarded are written to another member when received by the Receiver. See Figure 3.

*Figure 3  Example of Substation ES Data Flow*
Table 17 describes the control parameters.

**Table 17  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>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>The name of the message field used by the Forwarder to send the record to Substation ES and the name by which the Receiver gets the record content.</td>
</tr>
</tbody>
</table>
| MEMBER     | {SXGFL2RV | SXGRV2M1} | The member name of the PDS file.  
  - SXGFL2RV is used as input to the Forwarder.  
  - SXGRV2M1 is used as output of the Receiver. |
| SUBJECT    | See the note below. | The name of the subject used by the Forwarder or Receiver.                                                                                  |
| UFLD-PFX   | tibss-  | The prefix of the Substation ES communication field.                                                                                         |

These subjects are used by default. Do not change them unless you are changing the corresponding recipe definitions in the configuration file.

- **Forwarder Subject:** tibss.data.input.user
- **Receiver Subject:** tibss.data.output.user
Chapter 5  Communication of Application Messages

This chapter describes how Substation ES communicates with applications that use communication fields and notifications.

Topics

- Communication Fields, page 182
- System Fields, page 183
- User Fields, page 188
- Application Notifications, page 190
Communication Fields

Communication fields are named fields within a user message that assist in bidirectional Substation ES communication. These fields enhance the ability of an application to effectively forward additional information and receive notifications when conversing with Substation ES.

Categories

Substation ES communication fields are divided into the following categories:

- **Substation ES system fields**, which are defined in a user’s message and are used by applications and Substation ES during communication. System fields can be bidirectional. System Fields are not stored within the Substation ES communication buffer passed to a BES application. Therefore, Substation ES processes these fields differently than normal message fields.

- **Substation ES user fields**, which are part of the Application Notification and are Substation ES output-only fields. These fields enable applications to identify information regarding the application’s communication with Substation ES.

All communication field names consist of a variable prefix and a suffix constant. The prefix length is a maximum of 51 characters.

Prefix

Before Substation ES startup, you can change the prefix of the communication fields by specifying a value for the `UFLDS-PREFIX` keyword in the SIP member `USERHLQ.CNTL(SXSSIP$1)`.

If you do not specify a keyword value, the default prefix of `tibss-` applies. Otherwise, add a dash (-) after the name. Retain the default prefix value unless it is absolutely necessary to enforce a change. When changing a prefix, perform the following tasks:

- Edit all the user programs and Substation ES utilities that use the communication fields and make the necessary changes. Recompile and restart, if necessary.
- Specify a different prefix in the Substation ES SIP member.
- Change the Substation ES data configuration messages that use communication fields.
- Recycle Substation ES to activate the fields with the changed prefix.
System Fields

System fields are not always used. When they are, they are either in the user’s message or as message properties. To ensure that the contents of the fields are correctly processed and relayed to the participating interfaces by Substation ES, depending on the direction of the conversation, specify certain system fields in the definition of the configuration’s message.

Definitions

Table 18 describes the system fields, which do not include the default prefix tibss-.

Table 18  System Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>envelope</td>
<td>The Substation ES context information that is used during a CICS and IMS request or reply application communication or during IMS application conversational interaction.</td>
</tr>
<tr>
<td>userid</td>
<td>The user identifier required by Substation ES and the BES.</td>
</tr>
<tr>
<td>password</td>
<td>The user’s password required by Substation ES and the BES.</td>
</tr>
<tr>
<td>resource</td>
<td>The dynamically supplied name of the resource to be executed in the BES in the payload or message.</td>
</tr>
<tr>
<td>BES</td>
<td>The dynamically supplied name assigned to the Substation ES interface for communicating with the transaction processing systems (such as CICS and IMS) in the payload or message.</td>
</tr>
<tr>
<td>tran-code</td>
<td>The dynamically supplied user transaction name to be executed in the BES in the payload or message (CICS usage only).</td>
</tr>
</tbody>
</table>
Usage

Table 19 defines the system fields, which do not include the default prefix tibss-.

Table 19  Definitions of System Fields

<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 application’s 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Define the input message with the data-type Opaque. In the field conversion rule panel, specify Trunc=S and Req=Y.</td>
</tr>
<tr>
<td>userid</td>
<td>Input</td>
<td>The user's application sends this field.</td>
</tr>
<tr>
<td>password</td>
<td>Input</td>
<td>The user's application sends this field.</td>
</tr>
<tr>
<td>resource</td>
<td>Input</td>
<td>The user's application sends this field.</td>
</tr>
</tbody>
</table>
### Table 19  Definitions of System Fields (Cont’d)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Substation Message Direction</th>
<th>Definition and Usage</th>
</tr>
</thead>
</table>
| BES        | Input                         | The user’s application sends this field. Define the input message with the data type String. In the field conversion rule panel, specify Trunc=S and Reqd=Y or N. This field is used for both the primary or secondary BES. This can be specified on the Service Recipe or dynamically as input. Valid input values are:  
  • BES — override the BES value and skip the Alt-BES value defined on the Service.  
  • BES; — override the BES value and keep the Alt-BES value defined on the Service.  
  • BES;Alt-BES — override both BES values defined on the Service.  
  • ;Alt-BES — keep the BES value and override the Alt-BES value defined on the Service. |
| ReplyTo-Name | Output                         | The destination name to which Substation ES could not successfully write. Substation ES adds this field to the TIBCO Enterprise Message Service message as a user property when writing a TIBCO Enterprise Message Service dead message. |
| ReplyTo-Type  | Output                         | The destination type of ReplyTo-Name. Substation ES adds this field to the EMS message as a user property when writing an EMS dead message. |
Defining System Fields on an Inbound Conversion Rule

An input user application message can contain Substation ES system fields. Do the following:

1. Define the required system fields to the message definition.
2. Define the attributes on the conversion rule for the appropriate recipe definition.

For details on the definitions, see the descriptions of DCUI panels in Chapter 1, Configuration of Services, page 1.

--- Define Msg-Field / Buffer-Field Conversion Row 1 to 16 of 16 ---

<table>
<thead>
<tr>
<th>Conversion Id</th>
<th>==&gt; CVR-I-TSRR-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using Message</td>
<td>==&gt; MSG-I0-RR-01</td>
</tr>
<tr>
<td>Using Buffer</td>
<td>==&gt; BUF-I0-BF01</td>
</tr>
<tr>
<td>Max Buffer Size</td>
<td>==&gt; 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trunc</th>
<th>Reqd</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>8</td>
<td>IVP-REPLY-LILSECS</td>
<td>8</td>
<td>IVP-START-LILSECS</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>9</td>
<td>IVP-REPLY-LILSECS</td>
<td>9</td>
<td>IVP-REPLY-LILSECS</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>10</td>
<td>IVP-SUBJ-SFX</td>
<td>10</td>
<td>IVP-SUBJ-SFX</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>12</td>
<td>IVP-BTCH-SEQ-NO</td>
<td>11</td>
<td>IVP-BES-DELAY</td>
</tr>
<tr>
<td>S</td>
<td>Y</td>
<td>0</td>
<td>tibss-envelope</td>
<td>12</td>
<td>IVP-BTCH-SEQ-NO</td>
</tr>
<tr>
<td>S</td>
<td>N</td>
<td>0</td>
<td>tibss-password</td>
<td>13</td>
<td>IVP-START-DATE-CHAR</td>
</tr>
<tr>
<td>S</td>
<td>N</td>
<td>0</td>
<td>tibss-userid</td>
<td>14</td>
<td>IVP-START-TIME-CHAR</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td></td>
<td>15</td>
<td>IVP-REPLY-DATE-CHAR</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>0</td>
<td></td>
<td>16</td>
<td>IVP-REPLY-TIME-CHAR</td>
</tr>
</tbody>
</table>

System fields have an S defined in the Trunc column. Substation ES does not pass the contents to the back-end application buffer because they have a 0 defined in the Bffld column. Instead, Substation ES passes the contents to the required BES process for usage or validation.
Dynamic Transaction IDs

This section applies to Substation ES 2.5 and later releases only.

You can dynamically specify a user transaction that a resource is to execute under in CICS by supplying Substation ES with an input communication field, the message property `tibss-tran-code`, in the TIBCO message.

When Substation ES identifies the message property `tibss-tran-code` in an incoming message, it internally requests CICS to run the recipe under the transaction ID in `tibss-tran-code`. The following takes place in CICS:

1. Substation ES starts the transaction CSMI to handle EXCI communication.
2. CSMI switches the Transaction Id to the value of `tibss-tran-code`.
3. Security verification is done on the new transaction ID by CICS.
4. For DB2, `DB2CONN` is set up based on the new transaction ID by DB2.

Note the following when using dynamic transaction IDs:

- If `tibss-tran-code` is used, the value known as User Tran on the run-time recipe is set to Y for that instance. The Mirror Tran field on the recipe is ignored.
- The use of `tibss-tran-code` is only available for CICS transactions (not IMS).
- If the message property field `tibss-tran-code` and the map message field `tibss-tran-code` are both in a message, the property value is used.
- For MOIs other than 2, the programs `SXCOSRVD` and `SXCOSRVG` must be allowed to run under the transaction ID specified in `tibss-tran-code`.
- The transaction ID specified by `tibss-tran-code` does not have to be defined in the CICS PCT.
- CSMI normally is defined by IBM as “below the line” (`TASKDATALoc = Below`). To prevent “short on storage” situations, you might need to redefine CSMI as “above the line” (`TASKDATALoc = Any`). Since this is CSMI and is used by other processes, be sure to test this type of change to ensure that it does not affect other critical processes.
User Fields

User fields are only provided if you select the appropriate Reply Method on the Recipe or Trigger Definition panel. When in use, the fields are in the user's message and the Substation ES error message tibss-error-msg.

If the EMS-USE-PROP flag on in the ESB SIP member is enabled, the fields are included as as property fields of the user's message. User fields are only provided by the Substation ES on output.

For more information, refer to Application Notifications on page 190.

Definitions

Following is a list of the user fields, whose names do not include the default prefix tibss-.

status

If a transformation, Substation ES, or BES error occurs, then Substation ES adds the status field to the output message.

- Only one status field is added to a message. The value of the status field is incremented for each notification written to the tibss-error-msg message.
- In case of no errors, the status value is zero (0).

error-msg

This field is created if the appropriate Reply Method indicator has been set and if a transformation, Substation ES, or BES error is detected.

user field

This is the name of the field of a message defined to the Substation ES configuration that caused the transformation error. The content is a Substation ES descriptive and documented message.

error-info

This is a Substation ES warning or error informational field that is a Substation ES descriptive and documented message or a BES message.

ss-cfg-id

This is the current recipe or trigger identifier field and is always in a tibss-error-msg message.

ss-id

This is a Substation ES informational field that is always in a tibss-error-msg message.

Attributes

Table 20 lists the attributes for user fields, which do not include the default prefix tibss-.
### Table 20  User Field Attribute Summary

<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>This field, located in the user’s message segment, specifies whether the interaction with the Substation ES has succeeded.</td>
</tr>
<tr>
<td>error-msg</td>
<td>Rendezvous nested message</td>
<td>This field, located in the user’s message segment, is the Substation ES error message field.</td>
</tr>
<tr>
<td></td>
<td>EMS map message</td>
<td></td>
</tr>
<tr>
<td>error-info</td>
<td>String256 char maximum</td>
<td>This field, located in the nested error message, contains Information on an error condition detected by Substation ES.</td>
</tr>
<tr>
<td>user-field</td>
<td>String</td>
<td>This field, located in the nested error message, contains the name on the user’s input message as defined in the configuration. If a data transformation error occurs, this field also contains information on the error.</td>
</tr>
<tr>
<td>cfg-id</td>
<td>String32 char maximum</td>
<td>This field, located in the nested error message, identifies the Substation ES transformer recipe or trigger.</td>
</tr>
<tr>
<td>ss-id</td>
<td>String17 char maximum</td>
<td>This field, located in the nested error message, is the identifier of the Substation ES instance that 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 runtime failures
- Errors or failures that occur during the processing of applications requests

An Application Notification consists of a status field and a nested or submessage, or the fields are included as as property fields of the user’s message if the EMS-USE-PROP flag on in the ESB SIP member is enabled. While communicating with Substation ES, if an abnormality occurs, the TIBCO messaging application receives an Application Notification. The application can log and trace abnormalities and possibly correct them.

Depending on the direction in which the failure of execution was detected, Substation ES sends back the information relayed by Application Notification to the TIBCO messaging application along with the application’s request or reply and error message.

An Application Notification is included in messages that are delivered to the publish and error subjects as defined in the recipe and trigger definition panel.

Information Types

By specifying the appropriate Reply Method on the recipe or trigger definition, you can choose to have your external applications receive no status input or either of the following:

- The status field in all circumstances
- The status field only when an error condition exists

The field names that identify errors in the messages sent to user applications are defined in the user field. For details, see Attributes on page 188.

Structure

If a user application must receive notifications, you must specify the appropriate Reply Method for the recipe or trigger definition. The communication field tibss-status validates the delivery of notifications.
If an application receives a `tibss-status` field and the contents are nonzero, the user’s message receives an associated nested error message with the error information. When the ESB interface parameter `EMS-USE-PROP` is set to Y, or if the message is a `BYTES` or `TEXT` message, then the nested error message fields are sent as properties fields.

The format of the notification is as follows:

- Notification Control Fields within the user’s received message.
- Notification Message Fields within a nested message.

The following figures illustrate the structures of those fields.

**Figure 4  Structure of Application Notification Message**

```
tibss-status  good
fd1          good
fd2          good
fdn          ...
```

```
tibss-error-msg
```

**Figure 5  Nested Structure of Substation ES Error Message**

```
tibss-ss-id  tibss-cfg-id  tibss-error1  tibss-errorn  ...
```

```
...  and/or tibss-error-info
```

**Figure 6  Structure of Substation ES Error Messages as Properties Fields**

```
tibss-ss-id  tibss-cfg-id  tibss-error1  tibss-errorn  ...
```

```
...  tibss-status
```
The three tables in this appendix list the Substation ES sample programs, written in COBAL or C. The source for the COBOL samples is located in the USERHLQ.COB library and, for the C samples, in the USERHLQ.C library.

Table 21  Sample IVP Programs for Invoking Resources for the CICS Interface

<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>SXTT (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 21  Sample IVP Programs for Invoking Resources for the CICS Interface

<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 be written with the message content</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 be written with the message content</td>
<td>SXCTSQI1 (TSQ name)</td>
<td>Substation ES internal function</td>
</tr>
<tr>
<td>Application displaying 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>SXCCDMQ COBOL</td>
</tr>
<tr>
<td>Application publishing a message and expecting a reply from the Substation ES that invokes a CICS program using DPL</td>
<td>SXCCI010 (program)</td>
<td>SXCCI010 COBOL</td>
</tr>
</tbody>
</table>

Most z/OS COBOL data types are used in this example.

Table 22  Sample Programs for CICS Outbound Requests or Replies

<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 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 that replies to the CICS-initiated request or reply transaction</td>
<td>SXCCIRR3 (program)</td>
<td>SXCCIRR3 COBOL</td>
</tr>
</tbody>
</table>
### Table 23  Sample IVP Programs for the IMS Interface

<table>
<thead>
<tr>
<th>Description</th>
<th>Resource Name and Type</th>
<th>Program Source and Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application publishing a message and expecting a reply from Substation ES that invokes a IMS Transaction (MPP) via OTMA</td>
<td>SXICITO1 (transaction)</td>
<td>SXICIP01 COBOL</td>
</tr>
<tr>
<td>Application publishing a message and expecting a reply and a trigger from Substation ES that invokes a IMS Transaction (MPP) via OTMA</td>
<td>SXICITO2 (transaction)</td>
<td>SXICIP02 COBOL</td>
</tr>
<tr>
<td>Version of SXICITO2 demonstrating the use of a variable-length transaction name (not the first eight characters of the message)</td>
<td>SXICITO3 (transaction)</td>
<td>SXICIP03 COBOL</td>
</tr>
<tr>
<td>IMS BMP Program sending 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>
<tr>
<td>IMS BMP program using request/reply (synchronous callout or ICAL)</td>
<td>None</td>
<td>SXICIRR1 COBOL</td>
</tr>
</tbody>
</table>
Appendix B  Summary of Transformer Configuration File Utilities

The following two tables list the usage summary of the DD names and parameter combinations for each Transformer Configuration File Utility.

Table 24  Summary of Configuration File Utility: DD Names

<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></td>
</tr>
</tbody>
</table>

Table 25  Summary of Configuration File Utility: Parameters

<table>
<thead>
<tr>
<th>Parameter</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>0</td>
</tr>
<tr>
<td>-Ixxx --&gt; Input DD name</td>
<td>0</td>
</tr>
<tr>
<td>-Mxxx --&gt; Import DD name</td>
<td></td>
</tr>
<tr>
<td>-Nxxx --&gt; Export DD name</td>
<td></td>
</tr>
<tr>
<td>-Oxxx --&gt; Output DD name</td>
<td>0</td>
</tr>
<tr>
<td>-P --&gt; Print</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 25  Summary of Configuration File Utility: Parameters (Cont’d)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Utility Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Batch Copy</td>
</tr>
<tr>
<td>-R --&gt; Read-only</td>
<td>0</td>
</tr>
<tr>
<td>-T --&gt; Initialize</td>
<td>0</td>
</tr>
<tr>
<td>-V --&gt; Validate</td>
<td>0</td>
</tr>
<tr>
<td>-X --&gt; Hex dump</td>
<td>0</td>
</tr>
<tr>
<td>-Z --&gt; Zero file</td>
<td>0</td>
</tr>
</tbody>
</table>

Key:
- **M** — The required parameter value for the process
- **O** — The optional parameter value for the process

For the Batch Import utility:
- Do not use the -v option to import data into an empty configuration file. If you use -v against a Rendezvous message with duplicate field names, the error SXT7075E occurs.
- Do not use the -Z option to import data into a configuration that is not empty. Otherwise, all the existing data in the configuration file is lost.
Appendix C  Conversion Formats from Rendezvous

The following two tables list the supported Rendezvous data types during data transformation (mapping) performed by Substation ES.

*Table 26  RV-to-COBOL Data Types*

<table>
<thead>
<tr>
<th>RV Field</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum 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>8</td>
</tr>
<tr>
<td>DATETIME</td>
<td>C runtime value = ctime</td>
<td>Field = 10 digits</td>
<td>9(9) COMP-5</td>
<td>4</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>Alphanumeric</td>
<td>PIC X(1)</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RV Field</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum 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>
</tbody>
</table>
Table 27  RV-to-COBOL Data Types (Numeric Fields) (Cont’d)

<table>
<thead>
<tr>
<th>RV Field</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<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>+/-999,999,999,999,999,999,999,999,999</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>+999,999,999,999,999,999,999,999,999,999</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>ZONED</td>
<td>9(18) DISPLAY</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>9(18) COMP-3</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></td>
<td>ZONED</td>
<td>S9(10)V999</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>PACKED</td>
<td>S9(10)V999 COMP-3</td>
<td></td>
<td>7</td>
</tr>
</tbody>
</table>
Table 27  RV-to-COBOL Data Types (Numeric Fields) (Cont’d)

<table>
<thead>
<tr>
<th>RV Field</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F64</td>
<td>FLOAT</td>
<td>COMP-2</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>ZONED</td>
<td>S9(15)V999 DISPLAY</td>
<td></td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>PACKED</td>
<td>S9(15)V999 COMP-3</td>
<td></td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
Appendix D  Conversion Formats from TIBCO Enterprise Message Service

The tables in this appendix list the supported TIBCO Enterprise Message Service data types during data transformation (mapping) performed by Substation ES.

Table 28  EMS-to-COBOL Data Types

<table>
<thead>
<tr>
<th>Substation Message</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum 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>

*TIBCO Enterprise Message Service representation for the field in the Substation ES Message Definition panel.

Table 29  EMS-to-COBOL Data Types (Numeric Fields)

<table>
<thead>
<tr>
<th>Substation Message</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum 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></td>
<td></td>
<td>BINARY</td>
<td>9(3) BINARY</td>
<td>+127</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ZONED</td>
<td>9(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>
</tbody>
</table>
Table 29  EMS-to-COBOL Data Types (Numeric Fields) (Cont’d)

<table>
<thead>
<tr>
<th>Substation Message</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ZONED</td>
<td>S9(5) DISPLAY</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<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>+/-999,999,999,999,999,999,999,999</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>+ 999,999,999,999,999,999,999,999</td>
<td>8</td>
<td></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>
<tr>
<td>F64</td>
<td>DOUBLE</td>
<td>FLOAT</td>
<td>COMP-2</td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>
Table 29  EMS-to-COBOL Data Types (Numeric Fields) (Cont’d)

<table>
<thead>
<tr>
<th>Substation Message</th>
<th>EMS*</th>
<th>Usage</th>
<th>COBOL Picture</th>
<th>Maximum Value</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZONED</td>
<td>S9(15)V999 DISPLAY</td>
<td></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></td>
<td>10</td>
</tr>
</tbody>
</table>

*TIBCO Enterprise Message Service representation for the field in the Substation ES Message Definition panel
Table 30 below describes the sample projects available for TIBCO ActiveMatrix BusinessWorks on IMS. In order to use the IMS Project sample programs, you must set these global variables:

<table>
<thead>
<tr>
<th>Global Variable</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>JmsProviderUrl</td>
<td>The URL of the EMS Server</td>
</tr>
<tr>
<td>JmsUserid</td>
<td>The User ID to access the EMS server</td>
</tr>
<tr>
<td>JmsPassword</td>
<td>The password associated with the EMS server user ID</td>
</tr>
</tbody>
</table>

Table 30  TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on IMS

**Substation HeartBeat event consumer**

Listens to the topic tibss.heartbeat for a TEXT message.

**Listener for IMS Callout Process Definition**

1. To start, execute the IMS BMP program.
2. TIBCO ActiveMatrix BusinessWorks listens to the queue tibss.IMS.sync.Request for a MAP message.
3. The message is sent by IMS by way of Substation.
5. Substation ES reads the reply message and forwards it to the IMS program, issuing a RESPONSE transaction. Synchronization is done using the tibss-envelope field or property.
**Table 30  TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on IMS**

<table>
<thead>
<tr>
<th>Description</th>
<th>Service Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Listener for BYTES message - Queue</strong></td>
<td>TRIGGER-BYTES</td>
<td>IMS BMP</td>
</tr>
<tr>
<td></td>
<td>(Queue)</td>
<td>SXICIPB1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COBOL</td>
</tr>
<tr>
<td><strong>Listener for MAP message - Queue</strong></td>
<td>TRIGGER-FLDS</td>
<td>IMS BMP</td>
</tr>
<tr>
<td></td>
<td>(Trigger)</td>
<td>SXICIPB1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COBOL</td>
</tr>
<tr>
<td><strong>Listener for TEXT message - Queue</strong></td>
<td>TRIGGER-TEXT</td>
<td>IMS BMP</td>
</tr>
<tr>
<td></td>
<td>(Trigger)</td>
<td>SXICIPB1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COBOL</td>
</tr>
</tbody>
</table>
Table 31 below describes the sample projects available for TIBCO ActiveMatrix BusinessWorks on CICS. In order to use the CICS Project sample programs, you must set these global variables:

<table>
<thead>
<tr>
<th>Global Variable</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>JmsProviderUrl</td>
<td>The URL of the EMS Server</td>
</tr>
<tr>
<td>JmsUsername</td>
<td>The User ID to access the EMS server</td>
</tr>
<tr>
<td>JmsPassword</td>
<td>The password associated with the EMS server user ID</td>
</tr>
</tbody>
</table>

**Table 31  TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on CICS**

<table>
<thead>
<tr>
<th>Description</th>
<th>Service Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call CICS Pgm SXCI001 - Bytes - Queue</td>
<td>DPL-BYTES (Recipe)</td>
<td>CICS SXXI001</td>
</tr>
<tr>
<td>1. Sends a message to the queue.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Receives a Bytes message response on replyProcessId.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Call CICS Pgm SXCI001 - MAP - Queue | SXC-DPL-C (Recipe) | CICS SXXI001 |
| 1. Sends a MAP message to the queue tibss.CICS.C.Request. |
| 2. Receives a MAP message response on SXCCI001ProcessId. |

| Call CICS Pgm SXCCI001 - MAP - Queue | SXC-DPL-COBOL (Recipe) | CICS SXCCI001 COBOL |
| 1. Sends a MAP message to the queue tibss.CICS.Cobol.Request. |
| 2. Receives a MAP message response on SXCCI001ProcessId. |
### Table 31  TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on CICS

<table>
<thead>
<tr>
<th>Description</th>
<th>Service Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Listener CICS Bytes Messages - Queue</strong></td>
<td>TRIGGER-BYTES (Trigger)</td>
<td>CICS Transaction SXTF or SXT6</td>
</tr>
<tr>
<td>1. Listens to the EMS queue tibss.trigger for BYTES messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Sent by Substation ES using CICS transaction SXTF or SXT6 SXTF,R,1,,TRIGGER-BYTES</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Listener CICS MAP Messages - Queue</strong></td>
<td>TRIGGER-FLDS (Trigger)</td>
<td>CICS Transaction SXTF or SXT6</td>
</tr>
<tr>
<td>1. Listens to the EMS queue tibss.trigger for MAP messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Sent by Substation ES Trigger TRIGGER-FLDS using CICS transaction SXTF or SXT6 SXTF,R,1,, TRIGGER-FLDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Listener CICS Text Messages - Queue</strong></td>
<td>TRIGGER-TEXT (Trigger)</td>
<td>CICS Transaction SXTF or SXT6</td>
</tr>
<tr>
<td>1. Listens to the EMS queue tibss.trigger for TEXT messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Sent by Substation ES Trigger TRIGGER-TEXT using CICS transaction SXTF or SXT6 SXTF,R,1,, TRIGGER-TEXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Listener for Substation Errors</strong></td>
<td>Substation Internal</td>
<td>Substation Internal</td>
</tr>
<tr>
<td>1. Listens to the EMS queue tibss.error.out for MAP Substation ES error messages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Sent by Substation ES error processing.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Listener for CICS Request-Reply Process

1. Listens to the EMS queue `tibss.CICS.RR.trigger.Request` for a MAP message.
2. Sent by Substation ES trigger `CICS-RR-REQUEST` using CICS transaction `SXRC` or `SXR3`.
3. The process sends back a reply MAP message to `tibss.CICS.RR.Reply`.
4. Substation ES recipe `SXC-RR-ANSWER` reads the reply MAP message and forwards it to the CICS program `SXC3REPLY`, which forwards the reply to the business application.

To start the test, use CICS transaction `SXR3` or `SXRC`.

<table>
<thead>
<tr>
<th>Description</th>
<th>Service Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listener for CICS Request-Reply Process</td>
<td>CICS-RR-REQUEST (Trigger)</td>
<td>CICS Transaction SXR3 or SXRC</td>
</tr>
<tr>
<td>Substation HeartBeat event consumer</td>
<td>SXC-RR-ANSWER (Recipe)</td>
<td></td>
</tr>
</tbody>
</table>

Substation HeartBeat event consumer

Listens to the EMS topic `tibss.heartbeat` for a Substation ES heartbeat TEXT message.

<table>
<thead>
<tr>
<th>Description</th>
<th>Service Name</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substation HeartBeat event consumer</td>
<td>Substation Internal</td>
<td>Substation Internal</td>
</tr>
</tbody>
</table>

Table 31  TIBCO ActiveMatrix BusinessWorks Sample Projects for Substation ES on CICS
Index

C
changes from the previous release xiv
CICS
    Recipe Details Overview  59
    Service Details Panel  65
customer support xxi

E
EMS
    Properties (Recipe) Panel  72
    Properties (Trigger) Panel  84
ESB
    Endpoint SSL Extensions Panel  52

I
IMS
    Recipe Details Panel  59
    Recipe List Panel  54
    Select Processing Panel  57
    Service Details Panel  67
    Trigger Details Panel  78

R
recipe
    EMS Properties Panel  72
Recipe Details Panel
    IMS  59
Recipe List Panel
    IMS  54

S
SSL
    ESB Endpoint SSL Extensions Panel  52
support, contacting xxi

T
technical support xxi
TIBCO_HOME xviii
trigger
    EMS Properties Panel  84
    Trigger Details Panel  78