



# **TIBCO Substation ES<sup>TM</sup>** **Operations and Administration**

*Software Release 2.14*  
*April 2020*

## 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.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

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

TIBCO, the TIBCO logo, the TIBCO O logo, TIB, Information Bus, TIBCO BusinessEvents, TIBCO Business Process Management, TIBCO BusinessWorks, TIBCO Enterprise Message Service, TIBCO Rendezvous, TIBCO Mainframe WebUI, TIBCO Mainframe RED, and TIBCO Substation ES are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

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

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

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

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2001-2020. TIBCO Software Inc. All Rights Reserved.

# Contents

|   |           |
|---|-----------|
| <b>Figures .....</b>  | <b>8</b>  |
| <b>TIBCO Documentation and Support Services .....</b>         | <b>9</b>  |
| <b>Operational Commands .....</b>                             | <b>11</b> |
| Substation ES Console Interface .....                         | 11        |
| z/OS Command Interface .....                                  | 11        |
| Overview of Operational Commands .....                        | 11        |
| Command Summary .....   | 13        |
| Display Commands .....  | 17        |
| Update Commands .....   | 25        |
| Initialization Commands .....                                 | 29        |
| Control Commands .....  | 31        |
| <b>Substation ES Operations .....</b>                         | <b>35</b> |
| Auto Recovery of Messages When EMS is the ESB .....           | 35        |
| System Property Fields .....                                  | 35        |
| Processing of Property and System Fields .....                | 36        |
| Using Correlation ID .....                                    | 36        |
| Substation ES HeartBeat Events .....                          | 39        |
| Writing Substation ES Messages to SYSLOG .....                | 40        |
| Sites with Multiple Stacks .....                              | 40        |
| Detecting Network Connection Failures with EMS ESB .....      | 40        |
| How does RVDQ ESB Work .....                                  | 41        |
| Substation RV-FT (Rendezvous Fault Tolerant) .....            | 42        |
| Sending Variable Length Messages .....                        | 42        |
| Sending Variable Length Messages Using TIBCO EMS .....        | 42        |
| Sending Variable Length Messages Using TIBCO Rendezvous ..... | 43        |
| Exceptional Cases .....                                       | 45        |
| <b>CICS Operations .....</b>                                  | <b>46</b> |
| Substation ES CICS Region Resources .....                     | 46        |
| Substation ES in a CICSplex Environment .....                 | 50        |
| Methods of Invocation .....                                   | 51        |
| Services without Conversion Rules .....                       | 53        |
| Recipe Service .....  | 53        |
| Trigger Service .....   | 55        |
| Containers and Processes .....                                | 57        |
| Multiple Containers .....                                     | 58        |
| Single Containers .....                                       | 58        |

|   |           |
|---|-----------|
| Invoking Multiple Resources .....                       | 59        |
| Using FastPath Trigger Processes .....                  | 61        |
| Defining a FastPath Trigger .....                       | 61        |
| Endpoint Configuration for Rendezvous ESB .....         | 61        |
| Endpoint Configuration for EMS ESB .....                | 62        |
| Invocation of CICS Programs to Access DB2 .....         | 63        |
| DB2 Access Considerations .....                         | 63        |
| DB2Entry and DB2Tran for Programs That Access DB2 ..... | 64        |
| Security Considerations .....                           | 64        |
| Scenarios of CICS DB2 Transaction Definition .....      | 64        |
| Trigger and Recipe Performance .....                    | 66        |
| CICS Request/Reply .....                                | 67        |
| CICS HVT Triggers .....                                 | 68        |
| Usage Samples .....                                     | 69        |
| Substation ES Trigger Flow .....                        | 69        |
| Ordered Message Delivery .....                          | 70        |
| Configuration .....                                     | 71        |
| Code Samples: Link to SXCQWRIT or SXCQWTSQ .....        | 71        |
| SXCQWRIT Parameter Fields .....                         | 72        |
| HVT Trigger Transactions .....                          | 73        |
| CICS Trigger Queue Verify Task .....                    | 74        |
| Sample Output for SXTV Commands .....                   | 74        |
| CICS 3270 Assistance Task .....                         | 75        |
| SXTH Command-line Parameters .....                      | 75        |
| Sample Output for SXTH Commands .....                   | 76        |
| CICS 3270 Dead Message Queue Task .....                 | 77        |
| Sample Output for SXTD Commands .....                   | 77        |
| CICS Security Settings .....                            | 78        |
| CICS Transaction Security for Substation ES .....       | 78        |
| Definitions of Resource Access Control Facility .....   | 79        |
| Maximum EXCI Sessions .....                             | 79        |
| Operations Cookbook .....                               | 80        |
| <b>RED for CICS Operations .....</b>                    | <b>83</b> |
| Overview of the Interface for RED .....                 | 83        |
| RED Initialization for CICS .....                       | 83        |
| CICS Transactions Using RED .....                       | 84        |
| Setting Up a Recipe for CICS RED .....                  | 86        |
| Detailed Flow of CICS Transaction .....                 | 86        |
| RED CICS Triggers .....                                 | 86        |

|   |            |
|---|------------|
| Trigger Message Flows in CICS .....                       | 87         |
| RED CICS Request/Reply .....                              | 88         |
| Request/Reply Message Flows in CICS .....                 | 89         |
| RED CICS Control Information Viewer .....                 | 90         |
| XTUV Command-line Parameters .....                        | 90         |
| Sample Output for XTUV,TRG Command .....                  | 91         |
| Sample Output for XTUV,REAL Command .....                 | 93         |
| Sample Output for XTUV,STATS Command .....                | 94         |
| <b>IMS Operations .....</b>                               | <b>96</b>  |
| IMS Security Settings .....                               | 96         |
| Setup of OTMA Security Level .....                        | 96         |
| Defining RACF for IMS Security .....                      | 98         |
| IMS Security Exits .....                                  | 98         |
| IMS Transactions .....                                    | 99         |
| IMS Events or Triggers .....                              | 99         |
| General Trigger Processing .....                          | 99         |
| Synchronous Callout (ICAL) or Request/Reply Support ..... | 99         |
| Processing of IMS OTMA Trigger Errors .....               | 100        |
| Exceptional Cases .....                                   | 100        |
| <b>Logging, Tracing and Auditing .....</b>                | <b>101</b> |
| Introduction .....  | 101        |
| Features .....  | 101        |
| Disk Logging .....  | 102        |
| LTA Disk Files .....                                      | 102        |
| Setup of LTA Disk Files .....                             | 106        |
| LTA Services Utility .....                                | 107        |
| Features .....  | 107        |
| JCL Sample (SXGLINIT and SXGLUTIL) .....                  | 108        |
| Keyword Phrases .....                                     | 109        |
| Control Parameters .....                                  | 109        |
| System Logger Stream .....                                | 110        |
| Log and Trace Data Viewer .....                           | 110        |
| TIBCO z/OS Logging API .....                              | 112        |
| Parameters .....  | 113        |
| <b>Substation ES Utilities .....</b>                      | <b>115</b> |
| Rendezvous Request-Reply Timer Utility .....              | 115        |
| Operational Commands .....                                | 115        |
| Control Parameters .....                                  | 116        |
| Examples of Control Statements .....                      | 117        |

|   |     |
|---|-----|
| RED State Monitor Utility .....   | 120 |
| Examples of Control Statement .....                                       | 121 |
| Operational Commands .....  | 121 |
| Generic Sender or Listener Utility .....                                  | 121 |
| Rendezvous Generic Sender or Listener .....                               | 122 |
| Examples .....  | 125 |
| TIBCO Enterprise Message Service Generic Sender or Listener Utility ..... | 126 |
| Generic Error Listener Utility .....                                      | 129 |

# Figures

---

Architecture in a CICSplex Environment ..... 51

Trigger Flow ..... 70

LTA Multiple Entry Panel ..... 104

LTA Message Pop-up Panel ..... 105



# TIBCO Documentation and Support Services

---

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

## Product-Specific Documentation

Documentation for TIBCO Substation ES is available on the [TIBCO Substation ES Product Documentation](#) page.

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

`TIB_substation_version_docinfo.html`

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

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

- *TIBCO Substation ES Concepts*
- *TIBCO Substation ES Mainframe Logger User's Guide*
- *TIBCO Substation ES Installation*
- *TIBCO Substation ES Operations and Administration*
- *TIBCO Substation ES Configuration and Resources*
- *TIBCO Substation ES Messages and Codes*
- *TIBCO Substation ES Release Notes*

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

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

## How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.

- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking Register on the website.

### **How to Join TIBCO Community**

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to:

<https://community.tibco.com>

# Operational Commands

---

Substation ES operational commands are also known as console commands. You enter Substation ES operational commands through Substation ES Console Interface or remotely from applications.

The operational commands section includes the following topics:

- [Substation ES Console Interface](#)
- [Overview of Operational Commands](#)
- [Command Summary](#)
- [Display Commands](#)
- [Update Commands](#)
- [Initialization Commands](#)
- [Control Commands](#)

## Substation ES Console Interface

You can enter Substation ES operational commands through Substation ES Console Interface or remotely from applications. Commands that are received and the resultant output are logged to the Substation ES log medium.

If commands must be echoed upon the master console, you can set an appropriate value for the `CONSOLE-MSG` keyword in the Substation ES `SXSSIP$1` member.

Remote applications can issue all commands, except for the following ones:

- Commands to terminate Substation ES
- Commands to start or stop any of the Substation ES interfaces

From a remote location, you can send commands from a user application or Generic Listener or Sender Utility in Substation ES.

## z/OS Command Interface

The z/OS console can check on the status of components and interfaces that are operating in the Substation ES region. For example, you can identify pending requests and manage them from this console.

Substation ES supports z/OS Command Interface, which is the default command interface for Substation ES. You can enter commands on the console by specifying a modify command. You can shut down an active Substation ES instance normally with a stop command. See the *z/OS System Commands* manual for the syntax and usage of those commands.

Two examples are shown as follows:

- `F [jobname.]identifier,HELP`: lists the available commands and keywords.
- `P [jobname.]identifier`: stops an active Substation ES instance.

## Overview of Operational Commands

Substation ES operational commands are also known as console commands. You can enter Substation ES operational commands through the Substation ES console interface or remotely from applications.

By using Substation ES operational commands, you can perform the following tasks:

- Show resource usage, statistics, counters, and so forth.

- Connect or disconnect Substation ES Interfaces to back-end systems.
- Terminate Substation ES or its interfaces.
- Initialize data configuration items, such as recipes and triggers, during run time.
- Update a subset of System Initialization Parameters (SIP) during run time.

Commands received from Console Interface are logged and the resultant output is written to the log output medium and the console terminal. The commands are divided into the following four categories:

- [Display Commands](#)
- [Update Commands](#)
- [Initialization Commands](#)
- [Control Commands](#)

Except for control commands, you can submit all commands through the batch job SXJGENSL or SXRGENSL or from a remote user application. You must enter control commands on the z/OS console.

### Command Format

The syntax and terminology are as follows:

*command,keyword1,[keyword2,keyword3, ...] [# comments]*



You can add a comment, preceded by a space and a number sign (#), at the end of a command.

The first word is the command, followed by one or more keywords, separated by commas. The **HELP**, **SHUT**, and **TERM** commands are exceptions.

Some keywords specify a value with the keyword, as in the following examples:

`RECIPE=recipe_name`

`INTF=interface_ID`

### Command and Keyword Abbreviations

Some commands and keywords have abbreviated forms. In command descriptions, abbreviations are noted with a vertical bar. The following example shows a command and its abbreviation:

**DISCONNECT|DISCONN**

You can type either **DISCONNECT** or **DISCONN** to specify this command. Similarly, the following example shows a keyword and its abbreviation:

**LISTENERS|LSNS**

You can type either **LISTENERS** or **LSNS** to specify this keyword.

### Sample Output

In addition to showing typical usage, some command descriptions provide an example of the output. The following example shows a sample output:

```
SXA2601I Console Command Entered 'SHOW,ACT,LISTENERS'
SXT5586I The following listeners are active:
SXT5585I Intf:IIXFR, Dest:tibss.ivp.20.request
SXT5585I Intf:IIXFR, Dest:tibss.ivp.20.reply
SXT5585I Intf:IIXFR, Dest:tibss.admin.sxs
```

## Command Summary

The following table summarizes all Substation ES console commands by category. If multiple keywords are enclosed within square brackets, you must specify one of those keywords with the command.

### Display

| Command      | Keyword   |
|--------------|---|
| <b>HELP</b>  | [ , sub-command]  |
| <b>SHOW,</b> | ACTIVE   ACT, LISTENERS   LSNS [ , INTF   IID=intf-id]    |
|              | ACTIVE   ACT, RECIPE   RID [ , INTF   IID=intf-id]        |
|              | ACTIVE   ACT, TRIGGER   TID [ , INTF   IID=intf-id]       |
| <b>SHOW,</b> | CICS, "CEMT Sub-command" [ , INTF   IID=intf-id]          |
|              | CICS, SXTH [ , Sub-command] [ , INTF   IID=intf-id]       |
| <b>SHOW,</b> | DISABLED   DISA [ , INTF   IID=intf-id]                   |
|              | DISABLED   DISA, LISTENERS   LSNS [ , INTF   IID=intf-id] |
|              | DISABLED   DISA, RECIPE   RID [ , INTF   IID=intf-id]     |
|              | DISABLED   DISA, TRIGGER   TID [ , INTF   IID=intf-id]    |
| <b>SHOW,</b> | EMS, STATS, RECIPE   RID=name [ , INTF   IID=intf-id]     |
| <b>SHOW,</b> | ESB   TPORT [ , INTF   IID=intf-id]                       |
| <b>SHOW,</b> | IMS, BUFFER   |
|              | IMS, STATS  |
|              | IMS, CONNECT  |
|              | IMS, XCF  |
| <b>SHOW,</b> | INTF   IID  |
|              | INTF   IID=name   |
|              | INTF   IID [ , CONNECT   CONN]                            |
|              | INTF   IID=name [ , CONNECT   CONN]                       |
| <b>SHOW,</b> | LOGSTATS  |
|              | LOGSTATS, ALL   |

| Command | Keyword   |
|---------|---|
|         | LOGSTATS ,DETAIL                                      |
| SHOW,   | LTAVERS   |
| SHOW,   | QUE   |
|         | QUE ,LOCK   |
| SHOW,   | RED ,BUFFER   |
|         | RED ,LIMIT  |
|         | RED ,SENDER=*  XCF sender name                        |
|         | RED , "CEMT Sub-commands" [,INTF   IID=intf-id]       |
|         | RED , "XTUV [, Sub-commands" ] [,INTF   IID=intf-id]  |
| SHOW,   | RECIPE   RID   TRIGGER   TID=name, INTF   IID=intf-id |
| SHOW,   | RV ,DISPQ   |
|         | RV ,DQ-DISPQ  |
| SHOW,   | SSHOT [, num]   |
|         | SSHOT ,DETAIL   DET [, num]                           |
|         | SSHOT ,ENTRY   ENT [, num]                            |
| SHOW,   | SSID  |
| SHOW,   | SUSPENDED   |
| SHOW,   | SYS [, DEBUG]   |
|         | SYS ,LOCK   |
| SHOW,   | TRCSTATS  |
|         | TRCSTATS ,ALL   |
|         | TRCSTATS ,DETAIL                                      |
| SHOW,   | UOW   |
| SHOW,   | XCF ,GRP [=Group name] [, Member name]                |

## Update

| Command         | Keyword  |
|-----------------|--|
| DISABLE,        | RECIPE   RID=name [, FOR=BES-Name, INTF   IID=intf-id]   |
|                 | RECIPE   RID=*, FOR=BES-Name [, INTF   IID=intf-id]  |
|                 | TRIGGER   TID=name [, INTF   IID=intf-id]  |
| ENABLE,         | RECIPE   RID=name [, FOR=BES-Name, INTF   IID=intf-id]   |
|                 | RECIPE   RID=*, FOR=BES-Name [, INTF   IID=intf-id]  |
|                 | SUSPENDED [, FOR=BES-Name, INTF   IID=intf-id]   |
|                 | TRIGGER   TID=name [, INTF   IID=intf-id]  |
| RECOVER,        | RECIPE   RID=name [, INTF   IID=intf-id] /* [, INTF   IID=intf-id] /* [, FOR=BES-Name, INTF   IID=intf-id] |
| REFRESH   REFR, | RECIPE   RID=name [, INTF   IID=intf-id] [, TRCLVL=value]  |
|                 | TRIGGER   TID=name [, INTF   IID=intf-id] [, TRCLVL=value]   |
|                 | COUNTERS   |
|                 | RECIPE   RID   TRIGGER   TID, ALL [, INTF   IID=intf-id]   |
|                 | TPORT=tpport-id [, INTF   IID=intf-id]   |
| RELOAD,         | RECIPE   RID=name [, INTF   IID=intf-id]   |
|                 | TRIGGER   TID=name [, INTF   IID=intf-id]  |
|                 | RECIPE   RID   TRIGGER   TID, ALL [, INTF   IID=intf-id]   |
| SUSPEND,        | *, FOR=BES-Name [, INTF   IID=intf-id]   |

## Initialization

| Command               | Keyword                                   |
|-----------------------|---|
| CONNECT   CONN,       | INTF   IID=intf-id                        |
|                       | APPLID=applid                             |
|                       | ESB   TPORT=tpport-id, INTF   IID=intf-id |
| DISCONNECT   DISCONN, | INTF   IID=intf-id                        |
|                       | APPLID=applid                             |

| Command       | Keyword                                   |
|---------------|---|
|               | ESB   TPORT=tport-id, INTF   IID=intf-id  |
| UNLOAD   UNL, | RECIPE   RID=name [, INTF   IID=intf-id]  |
|               | TRIGGER   TID=name [, INTF   IID=intf-id] |

## Control

| Command | Keyword  |
|---------|--|
| SET,    | CICS, STATS={Y N} [, INTF   IID=intf-id]   |
| SET,    | CICS, TRCLVL-ONL=value [, INTF   IID=intf-id]  |
| SET,    | CICS, TRCLVL-HVS=value [, INTF   IID=intf-id]  |
| SET,    | CICS   IMS, TRCLVL-TRIG=value [, INTF   IID=intf-id]   |
| SET,    | IMS, BUFFER-PURGE [, INTF   IID=intf-id]   |
| SET,    | IMS, FREE-TRIG [, INTF   IID=intf-id]  |
| SET,    | IMS, HOLD-TRIG [, INTF   IID=intf-id]  |
| SET,    | IMS, REL-TRNAK [, SXITRNAK   DFS\$\$TOQ]   |
| SET,    | LOGDEST=value  |
| SET,    | LOGLVL=value   |
| SET,    | LOGFWRITE=value  |
| SET,    | RED, ABEND-RESET [, INTF   IID=intf-id]  |
| SET,    | RED, TRCLVL-AMSG   TRCLVL-SSRED   TRCLVL-CICS   TRCLVL-#RED   TRCLVL-TRG   TRCLVL-VIEW=value |
| SET,    | RED, STATS   STATISTICS={Y N}, INTF   IID=intf-id  |
| SET,    | RVFT[=value], INTF   IID=intf-id   |
| SET,    | STATS={Y N}  |
| SET,    | STATS={Y N} [, INTF   IID=intf-id]   |
| SET,    | TRACE-ENTRY=CONFIG, DETAIL   |
| SET,    | TRCDEST=value  |
| SET,    | TRCLVL=value [, OTMA   OPS   RR]   |



| Command | Keyword         |
|---------|-----------------|
| SET,    | TRCFWRITE=value |
| SHUT    |                 |
| SWITCH  |                 |
| TERM    |                 |

## Display Commands

Display commands include **HELP** and **SHOW** commands.

### HELP Command

The **HELP** command lists all the available commands. If an appropriate value in Substation ES SIP is set for CONSOLE-MSG, the command and the result are echoed on the Master Console display.

The **HELP** command is also available for an individual command, such as **SHOW**, **SET**, and so on. If you specify a command with **HELP**, a list of the keywords associated with the command is displayed as follows:

```
HELP
```

```
HELP, SET
```

### SHOW Command

The **SHOW** command inquires into Substation ES statistics, usage, and other useful information.

The typical usage of the **SHOW** command is listed as follows:

```
SHOW,ACTIVE|ACT,LISTENERS|LSNS[,INTF|IID=intf-id]
```

This command shows all active listeners in Substation ES. The optional keyword **INTF|IID=intf-id** specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,ACT,LISTENERS'
SXT5586I The following listeners are active:
SXT5585I Intf:IIXFR, Dest:tibss.ivp.20.request
SXT5585I Intf:IIXFR, Dest:tibss.ivp.20.reply
SXT5585I Intf:IIXFR, Dest:tibss.admin.sxs
```

```
SHOW,ACTIVE|ACT,RECIPE|RID[,INTF|IID=intf-id]
```

This command shows all the loaded recipes in Substation ES. The optional keyword **INTF|IID=intf-id** specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,ACT,RECIPE'
SXT5588I The following recipes are active:
SXT5587I Intf:IIXFR, Since Date:2004/11/09 Time:10:35:42.3321
RID:SXC-RR-REPLY
SXT5587I Intf:IIXFR, Since Date:2004/11/09 Time:10:35:42.3335
RID:SXC-RR-ANSWER
SXT5587I Intf:IIXFR, Since Date:2004/11/09 Time:10:35:42.3340
RID:Substation-Admin
```



In the preceding example, the Since Date and Time values show when a recipe was last loaded.

```
SHOW,ACTIVE|ACT,TRIGGER|TID[,INTF|IID=intf-id]
```

This command shows all the active triggers in Substation ES. The optional keyword `INTF|IID=intf-id` specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,ACT,TRIGGER'
SXT5590I The following triggers are active:
SXT5589I Intf:IIXFR, Since Date:2004/11/09 Time:10:35:42.3909
          TID:CICS-RR-REQUEST
SXT5589I Intf:IIXFR, Since Date:2004/11/09 Time:10:35:42.3912
          TID:SXC-IVP-TRIGGER
SXT5589I Intf:IIXFR, Since Date:2004/11/09 Time:10:35:42.4173
          TID:SXI-IVP-TRIGGER
```



In the preceding example, the Since Date and Time values show when a trigger was last loaded.

**SHOW,CICS, "CEMT Sub-command" [, INTF|IID=intf-id]**

This command issues the equivalent of CEMT on the CICS backend systems. The command submits a CICS 3270 task to be executed in the interface CICS regions. Only **INQUIRE** sub-commands are supported. The keyword `INTF|IID=intf-id` specifies the interface that must be the identifier used in the Substation ES CICS SIP member.

For example:

```
SHOW,CICS, "CEMT I TAS", IID=CICS01
SHOW,CICS, "CEMT I PROG(SXCCI001)"
SHOW,CICS, "CEMT INQUIRE PROG(SXCCI001)", INTF=CICS02
```

The response output for the **CEMT** command displays only the first screen's results.

**SHOW,CICS, SXTH[, Sub-command] [, INTF|IID=intf-id]**

This command issues the equivalent of SXTH on the CICS backend systems which communicate with Substation ES. The command submits a CICS 3270 task to be executed in the interface CICS regions. The keyword `INTF|IID=intf-id` specifies the interface that must be the identifier used in the Substation ES CICS SIP member.

For example:

```
SHOW,CICS, SXTH, HVT
SHOW,CICS, SXTH, TSQ, INTF=CICS01
SHOW,CICS, "SXTH, HVT", INTF=CICS01
```

For the supported list of SXTH optional sub-commands, see [SXTH Command-Line Parameters](#).

**SHOW,COUNTERS**

In Substation ES, counters are synonymous with running totals. Substation ES gathers information from all Substation ES interfaces that support counters. Counters differ by interface types because of their disparate processing requirements.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,COUNTERS'
SXS0950I Substation (ES) Interface Information
SXS0960I # Intf Id:IIXFR Type:XFR Vers:2.3.0
SXS0965I - RID:Substation-Admin - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-C-DPL - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-COB-DPL - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-COB-DPL-START-PGM - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-COB-DPL-START-TRAN - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-DPL-WRITE-TDQ - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-DPL-WRITE-TSQ - Used:1 Errors:0
SXS0965I - RID:SXC-IVP-COB-DPL-COMPLEX - Used:1 Errors:0
SXS0965I - TID:SXC-IVP-TRIGGER - Used:3 Errors:0
SXS0962I No counter information recorded for Intf:SSMASTER
SXS0960I # Intf Id:IADMIN Type:ADMIN Vers:2.3.0
SXS0961I - Operational Commands :1
SXS0960I # Intf Id:IICICS Type:CICS Vers:2.3.0
SXS0961I - Trigger Entries Received :3
```

```
SXS0961I - Requests processed, DPL :3
SXS0961I - Requests processed, DPL/XCTL :1
SXS0961I - Requests processed, DPL/Tran :1
SXS0961I - Requests processed, DPL/TDQ :1
SXS0961I - Requests processed, DPL/TSQ :1
```

**SHOW,DISABLED|DISA[, INTF|IID=*intf-id*]**

This command shows all disabled recipes and triggers. The optional keyword INTF|IID=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,DISABLED'
SXT5593I The following recipes/triggers are disabled
SXT5594I Intf:IIXFR,Disabled Date:2004/11/09 Time:10:42:00.8670
RID:SXC-RR-REPLY, Rsn: by Request
```

**SHOW,DISABLED|DISA,LISTENERS|LSNS[, INTF|IID=*intf-id*]**

This command shows all disabled listeners. The optional keyword INTF|IID=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2602I Console Command 'SHOW,DISABLED,LSNS'
SXT5586I The following listeners are disabled:
SXT5585I Dest:tibss.CICS.C.Request
```

**SHOW,DISABLED|DISA,RECIPE|RID[, INTF|IID=*intf-id*]**

This command shows all disabled recipes. The optional keyword INTF|IID=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2602I Console Command 'SHOW,DISABLED,RID'
SXT5593I The following recipes/triggers are disabled
SXT5594I Disabled Date:2012/07/02 Time:17:22:35.559709
RID:SXC-DPL-C, Rsn: by Request
```

**SHOW,DISABLED|DISA,TRIGGER|TID[, INTF|IID=*intf-id*]**

This command shows all disabled triggers. The optional keyword INTF|IID=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2602I Console Command 'SHOW,DISA,TID'
SXT5593I The following recipes/triggers are disabled
SXT5594I Disabled Date:2012/07/02 Time:17:27:19.240171
TID:TRIGGER-TEXT, Rsn: by Request
```

**SHOW,EMS,STATS,RECIPE|RID=*name*[, INTF|IID=*intf-id*]**

This command shows the statistics of this recipe. The optional keyword INTF|IID=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601IConsoleCommand 'SHOW,EMS,STATS,RID=SXC-DPL-C,INTF=ESB01'
IID:ESB01 Recipe SXC-DPL-C Queue Statistics
Receiver Count :1
UnAcknowledged Msgs :0
Pending Msgs :0
```

**SHOW,ESB|TPORT[, INTF|IID=*intf-id*]**

This command shows the status information of all transports (ESB) within the current Substation ES instance. The optional keyword INTF|IID=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,TPORT'
SXT5669I Transport EMS-Server-FT used 0, state=Unused,
process-err=
```

```
SXT5669I Transport EMS-Server used 22, state=Active, process-err=
SXT5670I Transports active 1, in-error 0, unused 1, disabled 0
```

#### **SHOW,IMS , BUFFER**

The Information Management System (IMS) commands show the information on one or more IMS interfaces that are active within the current Substation ES instance. This command shows the information on the Open Transaction Manager Access (OTMA) buffers currently in use.

#### **SHOW,IMS , STATS**

The IMS commands show the information on one or more IMS interfaces that are active within the current Substation ES instance. This command shows the information on the OTMA interface.

The command output is as follows:

```
SXA2601I Console Command 'SHOW,IMS,STATS'
SXG2077I Operational Request Feedback for IId:IMSE
SXI4967I Count      0 - Transactions received
SXI4967I Count      0 - Transactions to IMS
SXI4967I Count      0 - Nack transactions
SXI4967I Count      0 - Transactions timed out
SXI4967I Count      0 - Ack requests for sync=conf
SXI4967I Count      0 - Msg notify exit calls
SXI4967I Count      3 - SRBs to Message Exit
SXI4967I Count      3 - SRB Buffers used
SXI4967I Count      0 - SRB Response handled locally
SXI4967I Count      0 - Trigger SRBs received
SXI4967I Count      0 - Trigger SRBs routed normally
SXI4967I Count      0 - Lost SRB messages
SXI4967I Count      1 - SRBs to Group Exit
SXI4967I Count      0 - Group Buffers used
SXI4967I Count      0 - Lost Group messages
SXI4967I Count      0 - Synchronous callouts received
SXI4967I Count      0 - Synchronous callouts replied
SXI4967I Count      0 - Trigger NAKs received
SXI4967I Count      0 - Trigger NAKs reprocessed
```

#### **SHOW,IMS , CONNECT**

The **IMS** commands show the information on one or more IMS interfaces that are active within the current Substation ES instance. This command shows the IMS connection status.

#### **SHOW,IMS , XCF**

The **IMS** commands show the information on one or more IMS interfaces that are active within the current Substation ES instance. This command shows the information on the IMS XCF connection.

#### **SHOW,INTF | IID**

This command shows the information on all interfaces and their states.

#### **SHOW,INTF | IID=*name***

This command shows the information on the specified interface and its state.

#### **SHOW,INTF | IID[ , CONNECT | CONN]**

This command shows the information on all the connection states of all interfaces.

#### **SHOW,INTF | IID=*name* [ , CONNECT | CONN]**

This command shows the information on the connection state of the specified interface.

#### **SHOW,LOGSTATS**

This command shows the information on the current active log disk file.

#### **SHOW,LOGSTATS , ALL**

This command shows the information on all currently active log disk files.

#### **SHOW,LOGSTATS , DETAIL**

This command shows the statistical information on each active log disk file.

**SHOW,LTAVERS**

This command shows the version number of the Logging, Tracing, and Auditing (LTA) agents.

**SHOW,QUE**

This command shows the internal queue values of each interface.

The command output is as follows:

```
SXA2601I Console Command Entered 'SHOW,QUE'
SXS0850I - Intf: TIBSS01 UoW Que statistics request
SXS0851I - Currently in use Entries - 0
SXS0851I - Max Entries in Wait HiWM - 0
SXS0851I - Internal Next Stored Idx - 1
SXS0851I - Internal Next Strip Idx - 1
SXS0850I - Intf: IADMIN UoW Que statistics request
SXS0851I - Currently in use Entries - 0
SXS0851I - Max Entries in Wait HiWM - 0
SXS0851I - Internal Next Stored Idx - 0
SXS0851I - Internal Next Strip Idx - 0
SXS0850I - Intf: IIXFR UoW Que statistics request
SXS0851I - Currently in use Entries - 0
SXS0851I - Max Entries in Wait HiWM - 0
SXS0851I - Internal Next Stored Idx - 0
SXS0851I - Internal Next Strip Idx - 0
SXS0850I - Intf: IICICS UoW Que statistics request
SXS0851I - Currently in use Entries - 0
SXS0851I - Max Entries in Wait HiWM - 0
SXS0851I - Internal Next Stored Idx - 0
SXS0851I - Internal Next Strip Idx - 0
```

**SHOW,QUE, LOCK**

This command freezes the system and performs the **SHOW,QUE** command. See [SHOW,SYS, LOCK](#) for an explanation of the LOCK keyword.

**SHOW,RED, BUFFER**

This command shows information on the RED interface messages pool buffers currently in use within Substation ES.

**SHOW,RED, LIMIT**

This command shows RED parameters set up on the RED interface.

**SHOW,RED, SENDER=\* | XCF Sender name**

This command shows the statistics information on the RED interface Sender threads currently in use within Substation ES. If the \* is present will show all Sender threads. Or with a specific XCF Sender name present will only show the statistics information for that one Sender thread.

You may use the console command **/D XCF,GRP, [CICS APPLID]** to get the Sender names in this CICS system. In the example below the CICS system C512WW3G was the source of the RED data.

The sample command **SHOW,RED, SENDER=\*** output is as follows:

```
SXA2601I Console Command 'SHOW,RED, SENDER=*'
SXG2690I Operations Request executed successfully
SXG2077I Operational Request Feedback for IId:REDEA1
Statistics for Member:C512WW3G.REDEA1$47#00.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#01.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#02.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#03.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#04.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#05.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#06.Active
```

```

Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#07.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#08.Active
Sent-C:200 Recv:200 R/R:200
Statistics for Member:C512WW3G.REDEA1$47#09.Active
Sent-C:200 Recv:200 R/R:200
* End-of-Data *
SXA2666I End-of-Command

```

The sample command **SHOW,RED,SENDER=REDEA1\$47#01** output is as follows:

```

SXA2601I Console Command 'SHOW,RED,SENDER=REDEA1$47#01'
SXG2690I Operations Request executed successfully
SXG2077I Operational Request Feedback for IID:REDEA1
Statistics for Member:C512WW3G.REDEA1$47#01
System:LION Job:QASS11RE RED:NONE(0) XCF:Active(3)
Sent-C:200 Recv:200 R/R:200
* End-of-Data *
SXA2666I End-of-Command

```

**SHOW,RED, "CEMT Sub-command" [,INTF|IID=*intf-id*]**

This command issues the equivalent of CEMT on the CICS backend systems. The command submits a CICS 3270 task to be executed in the RED interface CICS regions. Only INQUIRE sub-commands are supported. The keyword **INTF|IID=*intf-id*** specifies the interface that must be the identifier used in the Substation ES CICS SIP member. If the **INTF|IID=*intf-id*** is not specified, the command will send to all RED interfaces BES. For example:

```

SHOW,RED, "CEMT I TAS", IID=RED01
SHOW,RED, "CEMT I PROG(SXCCI001)"
SHOW,RED, "CEMT INQUIRE PROG(SXCCI001)", INTF=RED02

```

The response output for the CEMT command displays only the first screen's results.

**SHOW,RED, "XTUV[, Sub-command"] [,INTF|IID=*intf-id*]**

This command issues the equivalent of XTUV on the CICS backend systems which communicate with Substation ES. The command submits a CICS 3270 task to be executed in the RED interface CICS regions. The keyword **INTF|IID=*intf-id*** specifies the interface that must be the identifier used in the Substation ES CICS SIP member. If the **INTF|IID=*intf-id*** is not specified, the command will send to all RED interfaces BES. For example:

```

SHOW,RED, "XTUV, STATS"
SHOW,RED, "XTUV, TRG", INTF=RED01
SHOW,RED, "XTUV, REC", INTF=RED02

```

For the supported list of XTUV optional sub-commands, see [XTUV Command-Line Parameters](#).

**SHOW,RECIPE|RID|TRIGGER|TID=*name*,INTF|IID=*intf-id***

This command shows the active recipe or trigger configuration information that is currently loaded in Substation ES. The keyword **INTF|IID=*intf-id*** specifies an ESB interface identifier.

**SHOW,RV,DISPQ**

This command shows the current number of messages in a Rendezvous worker dispatcher queue.

**SHOW,RV,DQ-DISPQ**

This command shows the current number of messages that are in a Rendezvous scheduler dispatcher queue waiting to be assigned to a Rendezvous dispatcher worker.

**SHOW,SSHOT[, *num*]**

This command initiates a snapshot and ends only when the *num* is reached. If the *num* is not present or the *num* is greater than the system default value, the system default value will be used. The output is directed to SYSPRINT of the Substation ES task. This command summarizes the statistics of the key interactions within the Substation ES instance for the specified number of Units of Work (UOWs).

The command output is as follows:

|          |                            |                            |
|----------|----------------------------|----------------------------|
| Elapsed  | Start Time                 | End Time                   |
| 4.997445 | 2008/03/26-11:54:17.742439 | 2008/03/26-11:54:22.739884 |

|         |          |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
| #:1000  | Minimum  | Maximum  | Total    | Average  | Recorded |
| UoW     | 0.002990 | 0.036007 | 4.637820 | 0.004637 |          |
| BES     | 0.001441 | 0.027182 | 2.278445 | 0.002278 |          |
| WRK-In  | 0.000023 | 0.001935 | 0.101735 | 0.000101 |          |
| WRK-Out | 0.000808 | 0.022377 | 1.290829 | 0.001290 |          |
| ESB     | 0.000000 | 0.000000 | 0.000000 | 0.000000 |          |

**SHOW,SSHOT,DETAIL|DET[,num]**

This command has the following functions:

- Initiates a snapshot and ends only when the *num* is reached. If the *num* is not present or the *num* is greater than the system default value, the system default value will be used. The output is directed to SYSPRINT of the Substation ES task.
- Provides a summary of statistics of the key interactions within the Substation ES instance for the specified number of UOWs and the statistics for each UOW.
- Measures the duration of a UOW, the duration of a transaction call in CICS within the UOW, the duration of an EMS call for that UOW, and the time in which to allocate a UOW when a request is received.

The command output is as follows:

|           |                            |                            |
|-----------|----------------------------|----------------------------|
| Elapsed   | Start Time                 | End Time                   |
| 16.166103 | 2008/03/29-15:44:28.191172 | 2008/03/29-15:44:44.357275 |

|         |          |          |            |          |          |
|---------|----------|----------|------------|----------|----------|
| #:1000  | Minimum  | Maximum  | Total      | Average  | Recorded |
| UoW     | 0.023369 | 0.503595 | 194.623214 | 0.194623 |          |
| BES     | 0.007174 | 0.450786 | 164.750438 | 0.164750 |          |
| WRK-In  | 0.000013 | 0.034431 | 0.497021   | 0.000497 |          |
| WRK-Out | 0.003321 | 0.077685 | 9.077637   | 0.009077 |          |
| ESB     | 0.002446 | 0.075679 | 5.707911   | 0.005707 |          |

|        |              |                            |                            |
|--------|--------------|----------------------------|----------------------------|
| Entry# | UoW          | Start                      | End                        |
| #00000 | 1998987041   | 2008/03/29-15:44:28.191172 | 2008/03/29-15:44:28.402554 |
| 00000  | UoW:0.211382 | BES:0.205756               | ReqB:0.005378 ESB:0.004002 |
| #00001 | 418627837    | 2008/03/29-15:44:28.207054 | 2008/03/29-15:44:28.413057 |
| 00001  | UoW:0.206003 | BES:0.192627               | ReqB:0.012935 ESB:0.009628 |

**SHOW,SSHOT,ENTRY|ENT[,num]**

This command initiates a snapshot and ends only when the *num* is reached. If number *num* is not present or *num* is greater than the system default value, the system default value will be used. The output is directed to SYSPRINT of the Substation ES task. This command summarizes the statistics of the key interactions within the Substation ES instance for the specified number of UOWs and the detailed statistics for each UOW.

The command output is as follows:

|          |                            |                            |
|----------|----------------------------|----------------------------|
| Elapsed  | Start Time                 | End Time                   |
| 0.034772 | 2017/12/21-14:03:06.369853 | 2017/12/21-14:03:06.404625 |

|         |          |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
| #:002   | Minimum  | Maximum  | Total    | Average  | Recorded |
| UoW     | 0.034592 | 0.038062 | 0.072654 | 0.036327 |          |
| BES     | 0.009572 | 0.019561 | 0.029133 | 0.014566 |          |
| WRK-In  | 0.000901 | 0.000000 | 0.001883 | 0.000941 |          |
| WRK-Out | 0.004641 | 0.000000 | 0.011451 | 0.005725 |          |

|        |                |                            |  |
|--------|----------------|----------------------------|--|
| Entry# | UoW:1058210416 | Start                      | End  |
| #00000 | 470581944      | 2017/12/21-14:03:06.369853 | 2017/12/21-14:03:06.404445                   |
| Ix:434 | Complete UoW   | Dur:0.034592               |  |
|        | Service Name   | SXC-DPL-BYTEM              |  |
|        | DPL-In         | WRK                        | Dur:0.000982 14:03:06.372370 14:03:06.373352 |
|        | BES            | Dur:0.009572               | 14:03:06.382463 14:03:06.392035              |

```

DPL-Out WRK Dur:0.006810 14:03:06.392903 14:03:06.399713
#00001 783307686 2017/12/21-14:03:06.366563 2017/12/21-14:03:06.404625
Ix:432 Complete UoW Dur:0.038062
Service Name SXC-DPL-BYTEM
DPL-In WRK Dur:0.000901 14:03:06.366796 14:03:06.367697
BES Dur:0.019561 14:03:06.370276 14:03:06.389837
DPL-Out WRK Dur:0.004641 14:03:06.390145 14:03:06.394786

```

### SHOW,SSID

This command shows the Substation ID.

The command output is as follows:

```

SXA2601I Console Command 'SHOW,SSID'
Substation ES ID:SSES4IVP

```

### SHOW,SUSPENDED

This command shows all suspended recipe services.

The command output is as follows:

```

SXA2601I Console Command 'SHOW,SUSPENDED'
SXT5594I SUSPENDED Date:2013/11/21 Time:19:29:12.928219
RID:Substation-Admin-IMS

```

### SHOW,SYS[,DEBUG]

This command is a combination of the **SHOW,QUE** and **SHOW,UOW** commands. The **DEBUG** option shows values specific to internal queues, current usage, and views of these queues.

### SHOW,SYS,LOCK

This command freezes the system and performs the combination of the **SHOW,QUE** and **SHOW,UOW** commands.



Specifying the **LOCK** keyword provides an absolute and accurate view of the request.

Use this keyword with discretion, however, because all internal queues and statistical tables are locked. This keyword causes the Substation ES region to suspend operations until the request is complete.

### SHOW,TRCSTATS

This command shows the information on the current active trace disk file.

### SHOW,TRCSTATS,ALL

This command shows the information on all active trace disk files.

### SHOW,TRCSTATS,DETAIL

This command shows the statistical information on each active trace disk file.

### SHOW,UOW

This command shows the information on Substation ES Unit of Work (UOW) processing statistics.

The command output is as follows:

```

SXA2601I Console Command Entered 'SHOW,UOW'
SXS0800I - UoW List Statistics request
SXS0800I - UoW Total Allocations - 0
SXS0800I - UoW List Hi Water Mark - 0
SXS0800I - UoW Currently Active - 0
SXS0800I - UoW List Size (entries) - 200
SXS0800I - UoW List Size (bytes) - 3256
SXS0800I - Max Allocated Entries - 80
SXS0800I - Uow Incremental Amount - 20
SXS0800I - UoW Lock Status - 0
SXS0800I - UoW Stress Status - 0
SXS0800I - UoW Stress Restart Level - 150
SXS0800I - UoW Stress Slow Level - 160
SXS0800I - UoW Stress Stop Level - 179

```



**SHOW,XCF,GRP**[=*Group name*][,*Member name*]

The XCF commands show all XCF groups in this SYSPLEX, or show all connected members for one specified group, or show a member statistics information for Substation groups. For example:

SHOW,XCF,GRP

SHOW,XCF,GRP=CICSTS1G

SHOW,XCF,GRP=CICSTS1G,MSG#YODA

## Update Commands

Update commands include **DISABLE**, **ENABLE**, **RECOVER**, **REFRESH|REFR**, and **SUSPEND** commands.

### DISABLE Command

The **DISABLE** command instructs the Substation ES transformer to stop an individual recipe or trigger. Subsequently, no processing can be performed against that recipe or trigger.

The **DISABLE** commands are used in the following ways:

**DISABLE, RECIPE** | **RID**=*name* [ , **FOR**=*BES-Name* , **INTF** | **IID**=*intf-id* ]

This command disables an active Substation ES recipe. The **RECIPE** keyword specifies the name of the recipe to be disabled. The optional keyword **FOR**=*BES-Name* is used to filter the recipes selected for a BES or Alt-BES that is equal to *BES-Name*. The optional keyword **INTF** | **IID**=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'DISABLE,RID='SXC-IVP-C-DPL''
SXT5838I The following recipe/trigger is disabled Intf:IIXFR,
      RID:SXC-IVP-C-DPL
```

**DISABLE, RECIPE** | **RID**=\* , **FOR**=*BES-Name* [ , **INTF** | **IID**=*intf-id* ]

This command disables the active Substation ES recipes. The keyword **FOR**=*BES-Name* is used to filter the recipes selected for BES or Alt-BES that is equal to *BES-Name*. The optional keyword **INTF** | **IID**=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command 'DISABLE,RID=*,FOR=CICS1'
SXT5838I IId:ESB01 Service is Disable, RID:DPL-BYTES
```

**DISABLE, TRIGGER** | **TID**=*name* [ , **INTF** | **IID**=*intf-id* ]

This command disables an active Substation ES trigger. The **TRIGGER** keyword specifies the name of the trigger to be disabled. The optional keyword **INTF** | **IID**=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'DISABLE,TID='SXC-IVP-TRIGGER''
SXT5838I The following recipe/trigger is disabled Intf:IIXFR,
      TID:SXC-IVP-TRIGGER
```

### ENABLE Command

The **ENABLE** command instructs the Substation ES transformer to start a recipe or trigger.

The **ENABLE** commands are used in the following ways:

**ENABLE, RECIPE** | **RID**=*name* [ , **FOR**=*BES-Name* , **INTF** | **IID**=*intf-id* ]

This command enables a Substation ES recipe that is disabled. The **RECIPE** keyword specifies the name of the recipe to be enabled. The optional keyword **FOR**=*BES-Name* is used to filter the recipes selected for BES or Alt-BES that is equal to *BES-Name*. The optional keyword **INTF** | **IID**=*intf-id* specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command 'ENABLE,RID=SXC-DPL-C,FOR=CICS2'
SXT5686W IId:ESB01 No Service Recipes found
SXT5005I IId:ESB02 Recipe SXC-DPL-C has trace=5
SXT5838I IId:ESB02 Service is Enabled, RID:SXC-DPL-C
```

**ENABLE, RECIPE | RID=\*, FOR=BES-Name[, INTF | IID=intf-id]**

This command enables all Substation ES recipes that are disabled. The keyword FOR=BES-Name is used to filter the recipes selected for BES or Alt-BES that is equal to BES-Name. The optional keyword INTF | IID=intf-id specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command 'ENABLE,RID=*,FOR=CICS1'
SXT5005I IId:ESB01 Recipe DPL-BYTES has trace=5
SXT5838I IId:ESB01 Service is Enable, RID:DPL-BYTES
SXT5005I IId:ESB01 Recipe SXC-DPL-C has trace=5
SXT5838I IId:ESB01 Service is Enabled, RID:SXC-DPL-C
```

**ENABLE, SUSPENDED [, FOR=BES-Name, INTF | IID=intf-id]**

This command enables a Substation ES recipe service that is suspended. The FOR keyword is optional, and the BES-Name keyword is used to filter the recipes selected only for BES or Alt-BES that is equal to BES-Name.

The INTF | IID=intf-id keyword denotes an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command 'ENABLE,SUSPENDED'
SXT5838I Service is unsuspended, RID:SXC-DPL-C
```

**ENABLE, TRIGGER | TID=name[, INTF | IID=intf-id]**

This command enables a Substation ES trigger that is disabled. The TRIGGER keyword specifies the name of the trigger to be enabled. The optional keyword INTF | IID=intf-id specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'ENABLE,TID='CICS-RR-REQUEST''
SXT5838I The following recipe/trigger is enabled Intf:IIXFR,
TID:CICS-RR-REQUEST
```

## RECOVER Command

The **RECOVER** command instructs the TIBCO Enterprise Message Service (EMS) server to redeliver all unacknowledged messages that are stored in the EMS queue. The Substation ES recipe definition contains the queue name from which the redelivered messages originate.

Undelivered messages are the result of previous, unsuccessful processing if the Substation ES service mode is Guaranteed Always (GA). The **RECOVER** command is the only nondisruptive method for recovering undelivered messages.

**RECOVER, RECIPE | RID=name[, INTF | IID=intf-id]**

**RECOVER, \*[, INTF | IID=intf-id]**

**RECOVER, \*[, FOR=BES-Name, INTF | IID=intf-id]**

The **RECIPE** keyword identifies the name of the EMS queue that is recovered. The optional keyword **INTF | IID=intf-id** specifies an ESB interface identifier.



This command instantiates the tibems\_Session Recover API call.

For example:

```
RECOVER, *
```

|                                    |   |
|------------------------------------|---|
| <b>RECOVER</b> ,*,FOR=CICS         | Recover is for all recipes with BES referencing CICS. |
| <b>RECOVER</b> ,*,XYZ              | XYZ means nothing.                                    |
| <b>RECOVER</b> ,*,IID=ESB1         | Recover is sent only to interface ESB1.               |
| <b>RECOVER</b> ,RID=ABC            |   |
| <b>RECOVER</b> ,RECIPE=ABC         |   |
| <b>RECOVER</b> ,RID=ABC , IID=ESB1 | Recover is sent only to interface ESB1.               |

## REFRESH | REFR Command

The **REFRESH|REFR** command instructs the Substation ES transformer to load a new copy of a service (recipe or trigger) in-flight. The old recipe or trigger is deactivated and unloaded when the new copy is loaded and activated.

If the recipe or trigger was not loaded during Substation ES initialization, you can use this command to load and activate the recipe or trigger. When the command is successfully completed, the message number SXT5811I is written to the log and trace files.

The **REFRESH** command also refreshes the current counters for all the active interfaces, whether or not there has been a change in the service details.

If the recipe is in suspended state when a **REFRESH** command is issued, the command loads the recipe but the state remains in suspended mode. The newly loaded recipe becomes active again only when it has been unsuspended.

The **REFRESH|REFR** commands are used in the following ways:

**REFRESH|REFR,RECIPE|RID=name[ , INTF|IID=intf-id] [ , TRCLVL=value]**

This command loads a recipe as described previously. The **RECIPE** keyword specifies the name of the recipe to be refreshed. The optional keyword **INTF|IID=intf-id** specifies that the recipe is to be refreshed with the specified ESB interface identifier. The optional keyword **TRCLVL=value** specifies that the recipe service sets the trace level in the memory to the value indicated. However, the trace level in the configuration file won't be changed.

The command output is as follows:

```
SXA2601I Console Command 'REFR,RID=SXC-DPL-C,TRCLVL=3'
SXT5811I IId:XFREA1 Recipe SXC-DPL-C refreshed - still for
          tibss.CICS.C.Request
SXT5680I IId:XFREA1 Recipe SXC-DPL-C reloaded
          Trace Level set to 3
SXA2666I End-of-Command
```

**REFRESH|REFR,TRIGGER|TID=name[ , INTF|IID=intf-id] [ , TRCLVL=value]**

This command loads a trigger as described previously. The **TRIGGER** keyword specifies the name of the trigger to refresh. The optional keyword **INTF|IID=intf-id** specifies that the trigger is to be refreshed with the specified ESB interface identifier. The optional keyword **TRCLVL=value** specifies that the trigger service sets the trace level in the memory to the value indicated. However, the trace level in the configuration file won't be changed.

The command output is as follows:

```
SXA2601I Console Command 'REFR,TID=TRIGGER-TEXT,TRCLVL=2'
SXT5487I IId:XFREA1 Trigger TRIGGER-TEXT refreshed
          Trace Level set to 2
SXA2666I End-of-Command
```

**REFRESH|REFR,COUNTERS**

This command refreshes the Substation ES counters.

**REFRESH | REFR, RECIPE | RID | TRIGGER | TID, ALL [ , INTF | IID=*intf-id*]**

This command refreshes all recipes or triggers currently loaded in the Substation ES; including any disabled recipes or triggers. This operation will not load any new or unloaded recipe or trigger from the configuration file. The optional keyword **INTF | IID=*intf-id*** specifies that the recipes or triggers are to be refreshed with the specified ESB interface identifier.

Depending on the number of recipes or triggers to be refreshed, this command might take a long time. For example, 30 - 45 seconds.

If a recipe or trigger is in use, then the process waits half a second for work to be cleared before trying the definition refresh again. Substation ES attempts the definition refresh 10 times before abandoning the definition and proceeding to the next definition.

**REFRESH | REFR, TPORT=*tport-id* [ , INTF | IID=*intf-id*]**

This command refreshes the ESB endpoint *tport-id* which is currently loaded in the Substation ES. This operation loads the ESB endpoint from the configuration file. The optional keyword **INTF | IID=*intf-id*** specifies that the ESB endpoint is to be refreshed with the specified ESB interface identifier.

## RELOAD Command

The **RELOAD** command instructs the Substation ES transformer to reload a new copy of a service (recipe or trigger) in-flight. The old recipe or trigger is deactivated and unloaded when the new copy is loaded and activated.

This command will not load and activate the recipe or trigger, if the recipe or trigger was not loaded during Substation ES initialization.

The **RELOAD** command also refreshes the current counters for all the active interfaces, irrespective of whether there has been a change in the service details.

If the recipe is in suspended state when a **RELOAD** command is issued, the command loads the recipe in suspended mode. The suspended recipe becomes active once it has been unsuspended.

The **RELOAD** commands are used in the following ways:

**RELOAD, RECIPE | RID=*name* [ , INTF | IID=*intf-id*]**

This command loads a recipe. The **RECIPE** keyword specifies the name of the recipe to be reloaded. The optional keyword **INTF | IID=*intf-id*** specifies that the recipe is refreshed with the specified ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command 'RELOAD,RID=SXI-TRAN'
SXT5816I IId:ESBRA1 Recipe SXI-TRAN refreshed, destination
          tibss.IMS.Tran.Request
SXT5680I IId:ESBRA1 Recipe SXI-TRAN reloaded
SXA2666I End-of-Command
```

**RELOAD, TRIGGER | TID=*name* [ , INTF | IID=*intf-id*]**

This command loads a trigger as described previously. The **TRIGGER** keyword specifies the name of the trigger to reload. The optional keyword **INTF | IID=*intf-id*** specifies that the trigger is to be reloaded with the specified ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command 'RELOAD,TID=TRIGGER-
BYTES,IID=ESBRA1'
SXT5487I IId:ESBRA1 Trigger TRIGGER-BYTES reloaded
SXA2666I End-of-Command
```

**RELOAD, RECIPE | RID | TRIGGER | TID, ALL [ , INTF | IID=*intf-id*]**

This command reloads all recipes or triggers currently loaded in the Substation ES; including any disabled recipes or triggers. This operation will not load any new or unloaded recipe or trigger from the configuration file. The optional keyword `INTF | IID=intf-id` specifies that the recipes or triggers are to be reloaded with the specified ESB interface identifier.

Depending on the number of recipes or triggers to be refreshed, this command might take a long time. For example, 30 - 45 seconds.

If a recipe or trigger is in use, then the process waits half a second for work to be cleared before trying the definition reload again. Substation ES attempts the definition refresh 10 times before abandoning the definition and proceeding to the next definition.

## SUSPEND Command

The **SUSPEND** command instructs the Substation ES ESB interface to stop processing for an individual recipe, or all recipes related to a particular BES.

The suspended state is different from a disabled state where a service's counter is not cleared and the service's internals remain intact. It's recommended that a **SUSPEND** command for a particular BES should not be explicitly issued because when suspended, Substation will not unsuspend the BES when it is connected. You must manage the BES explicitly to unsuspend it.

An individual recipe service can be enabled by using the standard **ENABLE,RID=xxx** command.



If a recipe service has both BES and Alt-BES specified, the service will only be suspended when both BES and Alt-BES are down (not available).

The **SUSPEND** commands are used in the following ways:

**SUSPEND,\* ,FOR=BES-Name[ ,INTF | IID=*intf-id*]**

This command when issued internally suspends all Substation ES recipe services for BES or Alt-BES. The *BES-Name* keyword is used to filter the recipes selected for BES or Alt-BES that is equal to *BES-Name*. The optional keyword `INTF | IID=intf-id` specifies that the recipes with the specified ESB interface identifier are suspended.

The command output is as follows:

```
SXA2601I Console Command 'SUSPEND,* ,FOR=IMS '
SXA2685I Suspending all services for BES:IMS
SXT5838I Service is suspended, RID:SXI-TRAN
SXT5838I Service is suspended, RID:SXI-TRAN-AND-TRG
SXT5838I Service is suspended, RID:SXI-TRAN-BYTE
```

**SUSPEND,RECIPE | RID=*name* [ ,INTF | IID=*intf-id*]**

This command suspends a recipe. The **RECIPE** keyword specifies the name of the recipe to suspend. The optional keyword `INTF | IID=intf-id` specifies that the recipe with the specified ESB interface identifier is suspended.

The command output is as follows:

```
SXA2601I Console Command 'SUSPENDED,RID=SXI-TRAN '
SXA2686I Suspending Service RID:SXI-TRAN
SXT5838I Service is suspended, RID:SXI-TRAN
```

## Initialization Commands

Initialization commands include **CONNECT | CONN**, **DISCONNECT | DISCONN**, and **UNLOAD | UNL** commands.

### CONNECT | CONN Command

The **CONNECT | CONN** commands instruct Substation ES to connect to a specified transport, the CICS region or the IMS region.

The **CONNECT | CONN** commands are used in the following ways:

**CONNECT | CONN,INTF | IID=*intf-id***

This command reconnects the specified Substation ES CICS or IMS interface to its CICS or IMS region. The keyword **INTF | IID=*intf-id*** specifies the interface that must be the identifier used in the Substation ES CICS or IMS SIP member.

**CONNECT | CONN,APPLID=*applid***

This command reconnects all Substation ES CICS or IMS interfaces identified at Substation ES startup to the CICS or IMS region with the corresponding APPLID. The keyword **APPLID=*applid*** specifies the CICS or IMS region.

**CONNECT | CONN,ESB | TPORT=*tport-id*, INTF | IID=*intf-id***

This command, which reconnects a Substation ES transport to an EMS server or a Rendezvous daemon, is usually used after a manual disconnection for a messaging transport. The **ESB** keyword identifies the transport identifier specified in the Substation ES startup configuration. The **INTF | IID=*intf-id*** keyword specifies that the transport be connected only using the specified ESB interface.

## DISCONNECT | DISCONN Command

The **DISCONNECT** command instructs Substation ES to disconnect from a specified transport, the CICS region or the IMS region.

The **DISCONNECT** commands are used in the following ways:

**DISCONNECT | DISCONN,INTF | IID=*intf-id***

This command disconnects the specified Substation ES CICS interface from a CICS region. The keyword **INTF | IID=*intf-id*** specifies the interface.

**DISCONNECT | DISCONN,APPLID=*applid***

This command disconnects all Substation ES CICS or IMS interfaces that use the same APPLID for a CICS or IMS region within a Substation ES region. The keyword **APPLID=*applid*** identifies the CICS or IMS region.



A CICS interface is defined in a Substation ES SIP member. The **INTF** or **APPLID** value specified in the **Control** command must match that defined for a CICS interface. The Substation ES CICS SIP members are located in the *USERHLQ*. *INTF* data set.



When you use the **CONN** or **DISCONN** command with the **APPLID** keyword, the command affects all currently active Substation ES instances that have been configured to communicate with this CICS region.

**DISCONNECT | DISCONN,ESB | TPORT=*tport-id*, INTF | IID=*intf-id***

This command disconnects a Substation ES transport from an EMS server or a Rendezvous daemon. The transport does not automatically reconnect after this command is issued; a manual connect transport command must be issued.

The **ESB** keyword identifies the transport identifier specified in the Substation ES startup configuration. The **INTF | IID=*intf-id*** keyword specifies that the transport be disconnected only from the specified ESB interface.

## UNLOAD | UNL Command

The **UNLOAD** command instructs the Substation ES transformer to unload a recipe or a trigger.

The **UNLOAD** commands are used in the following ways:

**UNLOAD | UNL,RECIPE | RID=*name* [ , INTF | IID=*intf-id*]**

This command unloads a recipe from the Substation ES runtime configuration. The **RECIPE** keyword specifies the name of the recipe to unload. The optional keyword **INTF | IID=*intf-id*** specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'UNLOAD,RID='SXC-IVP-C-DPL''
SXT5825I Unload Recipe SXC-IVP-C-DPL successful
```

**UNLOAD | UNL,TRIGGER | TID=*name* [ , INTF | IID=*intf-id*]**

This command unloads a trigger from the Substation ES runtime configuration. The **TRIGGER** keyword specifies the name of the trigger to unload. The optional keyword **INTF | IID=*intf-id*** specifies an ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'UNLOAD,TID='CICS-RR-REQUEST''
SXT5825I Unload Trigger CICS-RR-REQUEST successful
```

## Control Commands

You can enter the **Control** commands on the z/OS console. The **Control** commands include **SET**, **SHUT**, **SWITCH**, and **TERM** commands.

### SET Command

The **SET** commands change various entities. For information about the command values, see "Substation ES Parameters" in *TIBCO Substation ES Installation*.

The **SET** commands are used in the following ways:

**SET,CICS,STATS={Y|N} [ , INTF | IID=*intf-id*]**

This command starts and stops the CICS interface statistics recording. The optional keyword **INTF | IID=*intf-id*** specifies a CICS interface identifier.

For the CICS interface statistics recording function to work correctly, you must set the parameter **STATISTICS** to **YES** in the **TIBSLOG** member in the **PARM** dataset.

**SET,CICS,TRCLVL-HVS=*value* [ , INTF | IID=*intf-id*]**

This command changes the trace level of CICS when tracing reliable TSQ trigger entries that pass through the interface. The range of the valid values is from 0 to 3. The optional keyword **INTF | IID=*intf-id*** specifies a CICS interface identifier.

**SET,CICS,TRCLVL-ONL=*value* [ , INTF | IID=*intf-id*]**

This command changes the level of information messages produced by CICS resident programs that act on or on behalf of Substation ES. Messages are written to the **CEEOUT** and **CEEMSG** CICS output data sets. The range of the valid values is from 0 to 5. The optional keyword **INTF | IID=*intf-id*** specifies a CICS or IMS interface identifier.

Use the transaction **SXTH,VER** to view the current CICS interface online tracing level.

**SET,CICS | IMS,TRCLVL-TRIG=*value* [ , INTF | IID=*intf-id*]**

This command changes the level of CICS or IMS trigger information recorded to the trace file. The range of the valid values is from 0 to 3. The optional keyword **INTF | IID=*intf-id*** specifies a CICS or IMS interface identifier.

**SET,IMS,BUFFER-PURGE [ , INTF | IID=*intf-id*]**

This command cleans the OTMA buffers currently in use for all IMS interfaces that are active within the current Substation ES instance. The optional keyword and IMS identifier **INTF | IID=*intf-id*** can be used to specify a particular IMS interface for cleanup.

**SET,IMS,FREE-TRIG [ , INTF | IID=*intf-id*]**

This command starts the tpipes associated with the IMS trigger processes. The optional keyword **INTF | IID=*intf-id*** specifies a particular IMS interface identifier.

**SET,IMS,HOLD-TRIG [ , INTF | IID=*intf-id*]**

This command stops the tpipes for the IMS trigger processes. The optional keyword `INTF|IID=intf-id` specifies a particular IMS interface identifier.

**SET,IMS,REL-TRNAK[,SXITRNAK|DFS\$\$TOQ]**

This command starts processing of triggers that were NAKd and requeued to the NAKd trigger queue (SXITRNAK), or were timed out by IMS (DFS\$\$TOQ). If a queue name is not specified, the default is SXITRNAK. Any other queue name is rejected.

This command might only be issued when triggers are enabled. If triggers are held, the command is rejected.

**SET,LOGDEST=value**

This command changes the destination for the Substation ES log file. The valid values are D(isk) or P(rint).

**SET,LOGLVL=value**

This command changes the level of information recorded to the log file. The range of the valid values is from 1 to 5.

**SET,LOGFWRITE=value**

This command changes the number of centiseconds (1/100 of a second) of the Force Write for the LTA Log agent. The range of the valid values is from 0 to 30.

**SET,RED,ABEND-RESET[,INTF|IID=intf-id]**

This command resets the abend counter to 0. The counter is the number of times that #RED or the application abend. The optional keyword `INTF|IID=intf-id` specifies a particular RED interface identifier. This command does not change the maximum abend value in the RED interface. The maximum abend is determined by parameter "CICS-ABEND-VALUE" in the Substation ES RED interface definitions. The default is 5. Once this is set, it cannot be changed without re-cycling the CICS region.

**SET,RED,STATS|STATISTICS={Y|N},INTF|IID=intf-id]**

This command starts and stops the CICS RED interface statistics recording. The optional keyword `INTF|IID=intf-id` specifies a RED CICS interface identifier.

**SET,RED,TRCLVL-AMSG|TRCLVL-SSRED|TRCLVL-CICS|TRCLVL-#RED|TRCLVL-TRG|TRCLVL-VIEW=value**

This command changes the level of information recorded from the following processes to the trace file:

- TRCLVL-AMSG: the RED interface for AMSG related messages. The range of the valid values is from 0 to 3.
- TRCLVL-SSRED: the RED interface for general messages. The range of the valid values is from 0 to 5.
- TRCLVL-CICS: the RED trace level for online relate messages. The range of the valid values is from 0 to 5. If it is set to zero (0) then all the following trace levels will also be set to zero (0). TRCLVL-#RED, TRCLVL-TRG and TRCLVL-VIEW.
- TRCLVL-#RED: the RED trace level only for #RED transaction relate messages. The range of the valid values is from 0 to 5.
- TRCLVL-TRG: the RED trace level only for OTRED the trigger processes relate messages. The range of the valid values is from 0 to 5.
- TRCLVL-VIEW: the RED trace level only for VIEW transaction relate messages. The range of the valid values is from 0 to 5.

**SET,RVFT[=value],INTF|IID=intf-id**



This command allows dynamic switch between the RVFT daemons. The value can be 1, 2 or 3 indicating the number of the daemon to be switched to. If no value is specified, it will be switched to the number 1 daemon in the sequence.



The command is executed for all transports in the interface that are active and which use Substation ES RV Fault Tolerant URL.

The command output of 'SET,RVFT=3,IID=ESB01' is as follows:

```
SXA2601I Console Command 'SET,RVFT=3,IID=ESB01'
SXT5470I IId:ESB01 INFO Advisory RVD.CONNECTED for RV-ESB-3URL
SXT5490I IId:ESB01 Starting RV transport: RV-ESB-3URL, svc:
          8233, network: NULL, daemon: daemon3.tibco.com:8200,
          DistQ: N
SXT5995I IId:ESB01 RV-ESB-3URL was #1 daemon1.tibco.com:8200
SXT5995I IId:ESB01 RV-ESB-3URL now #3 daemon3.tibco.com:8200
```

The command output of 'SET,RVFT,IID=ESB01' is as follows:

```
SXA2601I Console Command 'SET,RVFT,IID=ESB01'
SXT5470I IId:ESB01 INFO Advisory RVD.CONNECTED for RV-ESB-3URL
SXT5490I IId:ESB01 Starting RV transport: RV-ESB-3URL, svc:
          8233, network: NULL, daemon: daemon1.tibco.com:8200,
          DistQ: N
SXT5995I IId:ESB01 RV-ESB-3URL was #3 daemon3.tibco.com:8200
SXT5995I IId:ESB01 RV-ESB-3URL now #1 daemon1.tibco.com:8200
SXA2666I End-of-Command
```

**SET,STATS={Y|N}**

This command sets the Substation ES Statistics recording to On or Off.

**SET,STATS={Y|N},INTF|IID=*intf-id***

This command starts and stops the Substation ES and the CICS interfaces (RED and EXCI) statistics recording. The optional keyword INTF|IID=*intf-id* specifies a CICS interface identifier.

**SET,TRACE-ENTRY=CONFIG,DETAIL**

This command sets TRACE-ENTRY for the Substation ES transformer.

**SET,TRCDEST=*value***

This command changes the destination for Substation ES trace information. The valid values are D(isk) or P(rint).

**SET,TRCLVL=*value***

This command changes the level of information recorded to the trace file. The range of the valid values is from 1 to 5.

**SET,TRCLVL=*value*[,OTMA|OPS|RR]**

This command changes the level of information recorded from the following processes to the trace file:

- OTMA: the OTMA modules of the IMS interface.
- OPS: the operational command processor and the internal message process of Admin Interface.
- RR: the CICS outbound Request/Reply Trigger process. For levels greater than or equal to 3, the message SXT5328I writes to the log file.

The range of the valid values is from 1 to 5.

**SET,TRCFWRITE=*value***

This command changes the number of centiseconds (1/100 of a second) of the Force Write for LTA Trace Agent. The range of the valid values is from 0 to 30.

## SHUT Command

The **SHUT** command causes an active Substation ES instance to terminate in an orderly manner. All input is stopped and all active requests within the Substation ES instance can be completed before the final termination request is sent to all active Substation ES interfaces.

## SWITCH Command

The **SWITCH** instructs the appropriate LTA Log agent or LTA Trace agent to continue recording events on the next available disk file. If no files are available, an error is recorded to the log file and processing continues. You can use this command only if the recording medium is set to disk on the Substation ES SSP member.

**SWITCH,LOGFILE**

**SWITCH,TRCFILE**

## TERM Command

The **TERM** command causes an active Substation ES instance to terminate in an orderly manner. Termination proceeds as if a **SHUT** command were issued. However, if the Substation ES tasks do not appear to be terminating, the Substation ES master task attempts to cancel all the active tasks within each Substation ES interface.



For better results do not use this command because in-flight messages can be lost due to cancellation of active tasks.

# Substation ES Operations

You can implement different operational functions on Substation ES Interface.

The operational functions include the following:

- [Auto Recovery of Messages When EMS is the ESB](#)
- [System Property Fields](#)
- [Substation ES HeartBeat Events](#)
- [Writing Substation ES Messages to SYSLOG](#)
- [Sites with Multiple Stacks](#)
- [Detecting Network Connection Failures with EMS ESB](#)
- [Sending Variable Length Messages](#)
- [Exceptional Cases](#)

## Auto Recovery of Messages When EMS is the ESB

Substation ES attempts to recover any Guaranteed messages waiting in a queue on the TIBCO EMS server, as well as messages that are used in Guaranteed Services where delivery failed to a z/OS transaction. These are active Recipes in the configuration file, with an SMSL of GA or GS.



This section applies to CICS and IMS services.

When reconnecting to the EMS server, Substation ES automatically attempts to recover any unprocessed messages that remain in the EMS queue. This includes messages that were read but not processed because the back-end system might have been unavailable. This capability is available only for queues that are used in Guaranteed Services.

When a back-end system reconnects during a Substation session, Substation ES sends an internal recover command to the EMS server and displays information about this in the Substation log. This message is displayed, followed by messages showing which services might have been recovered:

```

SXA2650I Internal Command 'RECOVER,*,interface'
SXG2075I Requesting Auto Recovery for IID:interface
SXT5868I Recovery successful for RID:recipe-service
  
```



Only recipes with SMSL of GA or GS are recovered.

## System Property Fields

Substation ES system fields, or properties as they are known in JMS (EMS), are mostly optional, but might either be present in the message or as a property in message.

For a full list of system fields, see "System Fields" in the *TIBCO Substation ES Configuration and Resources* guide.

To ensure that the contents of these system fields are correctly processed by Substation ES and relayed to a particular interface, application or applications payload system, fields must be defined correctly in the configuration.



Only TIBCO EMS handles system fields, referred to in this case as properties, outside of the application payload. However, within EMS these system fields might occasionally appear in both a property and in the payload. When this is the case, Substation ES uses the ESB parameter **EMS-USE-PROP** to determine where to look for the fields.

For best results, configure EMS messages to use properties alone, and do not place system fields within the application payload when using EMS MAP type messages.

## Processing of Property and System Fields

Consider the following points when using system fields on input according to different message types.

### Standard Messages

Substation ES processes properties or system fields according to the message conversion rule definition.

### Multi-Container Messages

If values are specified in the properties panel, all other configuration options are ignored and only values from the properties configuration panel are selected for input.

### EMS Map and Rendezvous Messages

Substation ES processes system fields in the following order:

1. If the special indicator S in the conversion rule panel in the Trunc column is set and a property is defined for the recipe, Substation ES uses that property definition to process the system fields.
2. If the special indicator S is not set, or if there is no property defined for the recipe, Substation ES processes the fields according to the definition rules.
3. Any user properties that might be defined in the properties panel.
4. Any `correlation_id` options.
5. Any other system properties defined in the properties panel.

### EMS Bytes Messages

Substation ES first copies the message directly to the buffer area, ignoring any message and buffer options. Then Substation ES processes system fields in the following order:

1. Fields in the conversion rule panel
2. Any user properties that are defined in the properties panel
3. Any `correlation_id` options
4. Any other system properties defined in the properties panel

### EMS Text Messages

Substation ES processes system fields in the following order:

1. Only the first field in the buffer definition is used to hold the user payload
2. All fields in the conversion rule panel
3. Any user properties that are defined in the properties panel
4. Any `correlation_id` options
5. Any other system properties defined in the properties panel

## Using Correlation ID

This section describes the different usages of the Correlation ID in Substation ES for recipes and trigger services.

### Recipes

- Go to selected recipe service Recipe Details definition panel

- Use the command **EMS** go to the EMS Recipe Details Extension panel
  - Set the input field **Copy JMSCorrelation** to **Y**

```

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

Recipe Identifier      ==> SXC-RR-EXTRN-PROCESS
Input Dest Type       ==> Q   Name: tibss.CICS.RR.trigger.Request
Output Dest Type      ==> Q   Name: tibss.CICS.RR.Reply
Error Dest Type       ==> Q   Name: tibss.error.out

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

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

```

## Triggers

There are two ways to set up a Correlation Id for a trigger service

- Add **tibss-CorrelationId** property to the Trigger service definition with properties type **V** and value **\$\$SYSTEM**

```

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

Trigger Identifier    => CICS-RR-REQUEST
Publish To           => tibss.CICS.RR.trigger.Request
Publish To, Sec      =>
Reply To             =>
Error Subject        => tibss.error.out
Error Threshold      => 0   (0 - 99999) Error Percent => 0   (0 - 99)
Publish To Len (Dyn) => 0   Start => 0
Output Conversion(?) => CVR-O-TSRR-01   Msg Field Name =>
ESB Endpoint (?)     => EMS-SERVER
Trace Level          => 3   0-5
Output Method        => 2   1 Status 2 Message
                        3 Both Status and Message
Compare Text Contain in Buffer
  Start => 0
  Length => 6
  Value => SXCIRR
Undelivered Requeue  => N   (Y Requeue, N Discard - Reliable Only)

```

```

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

Output Buffer Id: BUF-IO-BF01

Property Name      B/V Value
-----
tibss-CorrelationId V  $$SYSTEM

```

- Add **tibss-CorrelationId** to output message definition and change in all output conversion rules to reflect one of the following

- For CICS EXCI and IMS BES systems that generate the trigger
  - The data area must provide the value to set tibss-CorrelationId
  - Add **tibss-CorrelationId** to message definition.

```

----- Define Fields in a Message -----
Row 1 to 35 of 38
Command ==>
                                     Scroll ==> CSR
(Types: DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64,
  U64, F32, F64, POR, IPA)
Message Id   ==> MSG-IO-RR-01

```

| Sel | Type       | Dec | Id  | Seq | Name                       |
|-----|------------|-----|-----|-----|----------------------------|
| --- | ---        | --- | --- | --- | -----                      |
|     | STR        | 0   | 0   | 17  | IVP-REPLY-DATE-CHAR        |
|     | STR        | 0   | 0   | 18  | IVP-REPLY-TIME-CHAR        |
|     | OPA        | 0   | 0   | 99  | tibss-envelope             |
|     | STR        | 0   | 0   | 99  | tibss-password             |
|     | STR        | 0   | 0   | 99  | tibss-userid               |
|     | <b>STR</b> | 0   | 0   | 99  | <b>tibss-CorrelationId</b> |
|     |            | 0   | 0   | 0   |                            |

- In the conversion rule, tell Substation ES how to populate the field

```

----- Define Msg-Field / Buffer-Field Conversion -----
Row 15 to 19 of 19
Command ==>
                                     Scroll ==> CSR

  Conversion Id   ==> CVR-O-TSRR-01
  Using Message   ...: MSG-IO-RR-01
  Using Buffer     ...: BUF-IO-BF01
  Max Buffer Size ==> 0

```

| Trunc | Reqd | BfFld | Msg Field                  | Num | Buffer Field        |
|-------|------|-------|----------------------------|-----|---------------------|
| ---   | ---  | ---   | -----                      | --- | -----               |
| Y     | Y    | 0     | IVP-REPLY-TIME-CHAR        | 15  | IVP-REPLY-DATE-CHAR |
| Y     | Y    | 0     | tibss-envelope             | 16  | IVP-REPLY-TIME-CHAR |
| Y     | Y    | 0     | tibss-password             | 0   |                     |
| Y     | Y    | 0     | tibss-userid               | 0   |                     |
| Y     | Y    | 16    | <b>tibss-CorrelationId</b> | 0   |                     |

- For RED CICS, there is a field to specify your own value or if not filled in, Substaton ES will generate one. In the Trigger definition
  - Add tibss-CorrelationId to message definition

```

----- Define Fields in a Message -----
Row 1 to 35 of 38
Command ==>
                                     Scroll ==> CSR
(Types: DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64,
  F32, F64, POR, IPA)
Message Id   ==> MSG-IO-RR-01

```

| Sel | Type       | Dec | Id  | Seq | Name                       |
|-----|------------|-----|-----|-----|----------------------------|
| --- | ---        | --- | --- | --- | -----                      |
|     | STR        | 0   | 0   | 17  | IVP-REPLY-DATE-CHAR        |
|     | STR        | 0   | 0   | 18  | IVP-REPLY-TIME-CHAR        |
|     | OPA        | 0   | 0   | 99  | tibss-envelope             |
|     | STR        | 0   | 0   | 99  | tibss-password             |
|     | STR        | 0   | 0   | 99  | tibss-userid               |
|     | <b>STR</b> | 0   | 0   | 99  | <b>tibss-CorrelationId</b> |
|     |            | 0   | 0   | 0   |                            |

- In the output conversion rule, use the S indicator in the Trunc column to tell Substation ES to use the internal correlation id

```

----- Define Msg-Field / Buffer-Field Conversion -----
Row 15 to 19 of 19
Command ==>
                                Scroll ==> CSR

Conversion Id    ==> CVR-O-TSRR-01
Using Message    ...: MSG-IO-RR-01
Using Buffer      ...: BUF-IO-BF01
Max Buffer Size  ==> 0

Trunc  Reqd  BfFld  Msg Field          | Num Buffer Field
-----|-----
Y      Y      0  IVP-REPLY-TIME-CHAR | 15 IVP-REPLY-DATE-CHAR
Y      Y      0  tibss-envelope       | 16 IVP-REPLY-TIME-CHAR
Y      Y      0  tibss-password       | 0
Y      Y      0  tibss-userid         | 0
S      Y      0  tibss-CorrelationId  | 0

```

## Substation ES HeartBeat Events

HeartBeat is a message or an event that is sent to a destination, informing the recipients that Substation ES is running. Additionally, the Substation ES instance issues the specified internal operational commands. The contents of the commands are then sent to a destination defined in a trigger service.

The default Substation ES installation includes the basic configuration for HeartBeat events. To implement HeartBeat, activate the service in the administration interface member SXSIADM by defining the parameters and keywords shown in the following table.

### HeartBeat Parameters

| Parameter                                  | Description  |
|--|--|
| <b>ESB-INTF-ID</b> = <i>intf-id</i>        | Determines the interface that sends the HeartBeat message. <i>intf-id</i> is the ID name of the ESB interface that sends the message.  |
| <b>HEARTBEAT-SERVICE</b> = <i>T-Name</i>   | Specifies the trigger service name that Substation ES uses to publish HeartBeat messages.  |
| <b>HEARTBEAT-INTERVAL</b> = <i>seconds</i> | The interval, in seconds, between heartbeats. For example:<br><br><b>HEARTBEAT-INTERVAL</b> =5<br><br>To deactivate the HeartBeat feature, specify an interval of 0.   |
| <b>HEARTBEAT-COMMAND</b> = <i>command</i>  | Specifies a Substation ES operational command that is issued during each HeartBeat event. You can specify maximum of five commands. For example:<br><br><b>HEARTBEAT-COMMAND</b> = <b>SHOW</b> ,INTF , CONN<br><br><b>HEARTBEAT-COMMAND</b> = <b>SHOW</b> ,COUNTERS<br><br><b>HEARTBEAT-COMMAND</b> = <b>SHOW</b> ,ACT , RID |



By default, these parameters are commented out.

When the HeartBeat feature is enabled, Substation ES issues internal commands, and then Substation ES collects the output and passes it to the HEARTBEAT trigger service for delivery to the destination specified in the trigger service definition. The trigger service is determined by the **HEARTBEAT-SERVICE** parameter.

## Writing Substation ES Messages to SYSLOG

Substation ES provides the option to write messages to SYSLOG so that you can trap and automate the setup on selected Substation ES messages.

For example, to have Substation ES send messages SXT5636, SXT5639, and SXT5673 to SYSLOG, do the following:

1. Create a member SXT56 in the *USERHLQ.MSGS* data set.
2. Add the following statements to the member SXT56:

- SXT5636 OPT=SYSLOG
- SXT5639 OPT=SYSLOG
- SXT5673 OPT=SYSLOG



The member name must be the first five characters of the message code. For example, if the messages are SXT5847 and SXT5846, the member name is SXT58.

3. Add the following DD statement to the Substation ES startup JCL or PROC:

```
/** MESSAGE LIBRARY
/**
//TIBMSG DD DISP=SHR,DSN=&USERHLQ. .MSGS
/**
```

An alternative way to write all Substation ES messages of a particular severity or higher to SYSLOG is by setting the **LOG2-SYSLOG** parameter in the System Startup Parameters (SSP) member. The **LOG2-SYSLOG** parameter determines the severity of Substation ES log messages routed to the z/OS SYSLOG and provides an automation tool that picks up and reacts to error messages produced by Substation ES.

The valid values for the **LOG2-SYSLOG** parameter are as follows:

- 0: prints all log entries.
- 4: prints all the messages that are warnings or of a greater severity.
- 8: prints all the messages of that severity and higher or those that are suffixed with E, C, or F.
- 20: prints nothing in addition to z/OS SYSLOG.

## Sites with Multiple Stacks

If the system uses multiple TCP/IP stacks, insert the following JCL statement before the Substation ES application execution steps:

```
//TCAFF EXEC PROC=TIBTCAFF,STACK=TCPIP-STACK-NAME
```

This statement identifies the stack that the application uses.

You can specify a TCP/IP stack at the ESB level. To use this facility, you must first set the ESB parameter **STACK-NAME** to the address space default. The specified **STACK-NAME** value is the TCP/IP stack that is used for all ESB connections with a blank **TCPIP Stack Name** value. If the **TCPIP Stack Name** value of the ESB connection is not blank, the value given there is the name used. This allows a single Substation ES to use multiple TCP/IP stacks from the same address space.

## Detecting Network Connection Failures with EMS ESB

If network issues disrupt the connection between EMS and Substation ES, unexpected errors might result if Substation ES does not detect the connection failure.

If a network issue occurs during the Substation ES run time, you can configure EMS to assist the Substation ES ESB in detecting the network connection failure. Substation ES then reports the correct ESB state and handles the ESB connections properly.



For best results, specify the following parameters and values in the EMS server configuration:

```
server_heartbeat_client = 5
client_timeout_server_connection = 20
```

For more details about these parameters, see the *TIBCO Enterprise Message Service User's Guide*.

## How does RVDQ ESB Work

When Substation ES is using RVDQ, each interface can start two threads. One thread acts as a Scheduler with its own separate event queue and queue dispatcher. The other thread is a Worker. The worker thread uses the Substation ES callback routine to process incoming messages for Recipe subjects. Each Substation ES RVDQ interface sets up listeners for the subjects with at least one Active Scheduler Thread. If the workloads are high it's best to have the scheduler be defined to its own Substation instance.

Though you may define a Scheduler Thread in several RV ESB interfaces, only one (1) Scheduler thread is active. The others are dormant in backup mode until the active scheduler ends or becomes invalid.

The Scheduler thread is the one that listens to RV topics. As messages are delivered from RV daemons, the active scheduler distributes them to worker threads in the RVDQ group based on **RVDQ-IDENTITY** value using a unique "inbox" topic. The actual RVDQ group identity is

*Transport-ID* + "-" + *RVDQ-IDENTITY* + "-DistQ"

For example: RV-SERVER-RVDQ-DistQ

where RV-SERVER represents **Transport-ID**, RVDQ represents **RVDQ-IDENTITY** and DistQ represents -DistQ.

So the flow of messages in RVDQ is:

1. The Scheduler in its message callback gets a RV message from the RV Daemon.
2. The Schedule re-packages the message into a DQ message.
3. The Schedule sends the DQ message to a worker thread(s).
4. The Worker thread gets the DQ message and re-creates the original message.
5. The Worker thread uses the Substation ES callback to hand the message to a Substation ES. From there it goes to the Back-end system for processing.



The Scheduler is only one (1) thread for the **RVDQ-Identity** value and so - can only use one cpu (it might bounce around from cpu to cpu - but only uses 1 cpu at a time) - so when a scheduler consumes 100% of a cpu - it is maxed out similar to CICS QR processing).

The **RVDQ-WORKERS** parameter is the number of DQ messages that can be put on the worker "INBOX". They stay on the "INBOX" until the worker callback calls the user callback and the user callback returns which in Substation ES case is the time to get a UOW and dispatch the UOW to other Substation ES internal queues.

If **RVDQ-WORKERS** is set to zero, then RVDQ will not send messages to this interface. It can be used as a Scheduler, but not as a Worker.

By setting **RVDQ-SCHEDULER** to high (value of 5) and **RVDQ-WORKERS** to zero (0), the RV ESB interface will only be used for Scheduling. This set of options allow you to have a Substation ES that is only a Scheduler controlling its priority and location (LPAR).

By setting **RVDQ-SCHEDULER** to low (value of 0) and **RVDQ-WORKERS** to a non-zero, the RV ESB interface will be used as a Worker. (note RVDQ will commandeer any Worker thread as a Scheduler if required. There must always be one valid Scheduler for a RVDQ identity).

To increase concurrent processing RVDQ messages you need to increase the number of RV ESB interfaces. Each RVDQ interface will create a worker thread to handle messages from the Scheduler thread. The RVDQ interface can be in the same or different Substation ES address spaces on the same or different LPARs.

## Substation RV-FT (Rendezvous Fault Tolerant)

Substation ES RV-FT allows the transport definition to specify from 1 to 3 RV daemon URLs. This is done by using a semi-colon (;) delimited string for the transport URL.



The value for network and service is same.

### Example:

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

ESB Id      ==> RV-SERVER
Type       ==> R   (E/R EMS or RV)
Startup    ==> Y   (Y/N Start during Substation Initialization)
Port       ==> 0
URL        ==> URL1:7500;URL2:7500;URL3:7500

Rendezvous Values:
Service     ==> 8888
Network     ==> 0   0   0   0   or
String ==>
Multicast   ==> N   (N/Y)
Multicast Address ==>

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

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

When RV-FT type URL is used, Substation ES uses the first daemon to which it can connect successfully followed by the other daemons in the order. Using the above example, *URL1:7500* is the first daemon connected, followed by *URL2:7500* and *URL3:7500* in order.

If the connectivity with the current daemon is lost, Substation ES does the following functions:

- Initiates the connection with the first daemon in the list.
- Attempts to connect with the second daemon, followed by the next daemon in the list in case of connection failure.
- Returns to the current daemon and waits for re-connection if no active daemon is found.

When a connection is made, all Subjects for the transport are set up and the messages start to process.

## Sending Variable Length Messages

Some business applications always send the same size output; other applications send out different length messages based on business criteria.

For example, when you send an image (.jpg file), the message should be a variable length. Images can vary in size from 5K, 10K, 100K, 1MB, and larger. It is not good practice to send a 1MB message with only 5K of data, so the variable length message is very important to performance. Variable length messages can be sent by using TIBCO EMS or TIBCO Rendezvous.

## Sending Variable Length Messages Using TIBCO EMS

You can use TIBCO EMS to send variable length messages.

The following steps explain how to set up Substation ES to send variable length messages using TIBCO EMS.

Examples of these definitions are in the IVP Configuration file and are used by DPL-BYTES recipes. The following recipe example shows an example of the Bytes message or the Map message with the **DATA** field as input and a variable bytes message as output with a max length of 16384.

### Procedure

1. Define a message with only one field, which is **OPAQUE**.

```
-----Define Fields in a Message ----- Row 1 to 20 of 20
Command ==>
(Types: DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64,
      F32, F64, POR, IPA)
Message Id ==> MSG-IO-OPAQUE
Sel Type Dec Id Seq Name
-----
      OPA    0    0 0  DATA
      0      0 0 0
```

2. In Buffer Definition, verify that the buffer offset (Start) is set to zero (0).

```
----- Define Fields in a Buffer ----- Row 1 to 18 of 21
Command ==>
(Types are Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, Opa)
Buffer Id ==> BUF-BYTES
Maximum Size ==> 0
Sel Type Start Len Dec Name
-----
      OPA    0    16384    0  IVP-OPAQUE
      0      0      0      0
```

3. Configure the conversion rule so that only one output field exists and the **SZ** field is set to M:

- a) Add only one field in the output conversion rule.

```
----- Define Msg-Field / Buffer-Field Conversion - Row 1 to 1 of 1
Command ==>
Conversion Id ==> CVR-IO-BYTE
Using Message ...: MSG-JMS-BYTE
Using Buffer ...: BUF-BYTES
Max Buffer Size ==> 0
Trunc Req'd BfFld Msg Field | Num Buffer Field
-----|-----
      Y    Y    1  DATA    | 1  IVP-OPAQUE
```

- b) Set the conversion rule to the **SZ** field to M.

```
----- Conversion Rules ----- Row 5 to 13 of 13
Command ==>
(Use ADD command to create a new conversion rule)
(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      B  M CVR-IO-BYTE MSG-JMS-BYTE BUF-BYTES
```

## Sending Variable Length Messages Using TIBCO Rendezvous

You can use TIBCO Rendezvous to send variable length messages.

The following steps explain how to set up Substation ES to send variable length messages using TIBCO Rendezvous.

Examples of these definitions are in the IVP Configuration file and are used by the DPL-BYTES recipes. The following recipe example shows an example of the Bytes message or the Map message with the **DATA** field as input and a variable bytes message as output with a max length of 16384.

## Procedure

1. Define a message with only one field, which is **OPAQUE**.

```
Command ==> Scroll =
(Types: DAT, OPA, MSG, STR, BOL, I8, U8, I16, U16, I32, U32, I64, U64,
      F32, F64, POR, IPA)
Message Id ==> MSG-IO-OPAQUE

Sel Type Dec Id Seq Name
-----
      OPA  0  0  0  DATA
      0  0  0
```

2. In Define Fields in a Message, set the decimal place to 1.

```
Message Id ==> MSG-IO-OPAQUE

Sel Type Dec Id Seq Name
-----
      OPA  1  0  0  DATA
      0  0  0
```

3. In Buffer Definition, verify that the buffer offset (Start) is set to zero (0).

```
----- Define Fields in a Buffer ----- Row 1 to 18 of 21
Command ==> Scroll ==> CSR
(Types are Str, Text, Pdec, Updec, Zdec, Uzdec, Ubin, Bin, Float, Opa)
Buffer Id ==> BUF-BYTES
Maximum Size ==> 0

Sel Type Start Len Dec Name
-----
      OPA  0 16384 0 IVP-OPAQUE
      0 0 0
```

4. Configure the conversion rule so that only one output field exists and the **SZ** field is set to M:
  - a) Add only one field in the output conversion rule.

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

Conversion Id ==> CVR-IO-BYTE
Using Message ...: MSG-JMS-BYTE
Using Buffer ...: BUF-BYTES
Max Buffer Size ==> 0

Trunc Req'd BfFld Msg Field | Num Buffer Field
-----|-----
      Y      Y      1  DATA | 1  IVP-OPAQUE
```

- b) Set the conversion rule to the **SZ** field to M.

```
----- Conversion Rules ----- Row 5 to 13 of 13
Command ==> Scroll ==> CSR
(Use ADD command to create a new conversion rule)
(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      B M CVR-IO-BYTE MSG-JMS-BYTE BUF-BYTES
```

## Exceptional Cases

When you use Substation ES, you might encounter and should be aware of some exceptional cases. The exceptional cases are described as follows.

### Rounding

In Substation ES Transformer, if you are doing an outbound conversion from Float to Integer with less than six implied decimals and the value is large enough, the outbound conversion always loses 1 at the last digit. This is a known rounding problem.

For example, if the input value is -327.67, it is -.32766992E 03 in the buffer. After conversion to I32 with two implied decimals, the value becomes -32766.

### Nonmatching Subjects

For an inbound process, if Substation ES Transformer cannot find a matched invoked subject, the requesting messages are ignored.

You (outside z/OS) can check through the daemon HTTP interface on the subjects to which Substation ES is currently listening. Alternatively, if you use z/OS, you can check the current Substation ES listening subjects in the Substation ES parameters log file TIBLPARM.

### Buffer Size

Substation ES Transformer provides one buffer definition for multiple conversion rules. Therefore, the automatic calculation of the maximum buffer size ignores the fields at the end of the buffer that did not map to the message fields.

### STRING Data in Buffer Fields

When using the STRING data type for a buffer field, you must add one extra byte at the end for the STRING null terminator in the buffer-field definition. Otherwise, truncation occurs and the last character of the STRING data is lost.

### Substation ES Trigger Queue Name

After selecting a queue name for triggers and using it for Substation ES CICS Interfaces, you cannot change that name dynamically or by recycling Substation ES. Instead, you must define the new queue name to CICS, change the name in the member that contains the Substation ES CICS SIP, and recycle Substation ES and the CICS regions.

### Dynamic Change EMS Message Selector

To dynamically change EMS Message Selector of an active recipe service, you must first use the Substation ES console command **UNLOAD** on the recipe service, and then follow with the **REFRESH** command to load the new copy of the service.

### Code Page Issue

Both EMS and RV allow you to change the Code Page for both the host and network side of the interface.

Sometimes processes need more control. To have total control in a ActiveMatrix BusinessWorks process, define the data fields as OPA instead of String or Text. In the ActiveMatrix BusinessWorks process, convert the data as you need and send it as an OPA. No additional conversion will be done on the data. It will be delivered to the Back End System as ActiveMatrix BusinessWorks specified.

# CICS Operations

---

This section describes how to use CICS operational functions.

The CICS operational functions include the following ones:

- [Substation ES CICS Region Resources](#)
- [Substation ES in a CICSplex Environment](#)
- [Methods of Invocation](#)
- [Services Without Conversion Rules](#)
- [Containers and Processes](#)
- [Invoking Multiple Resources](#)
- [Using FastPath Trigger Processes](#)
- [Invocation of CICS Programs to Access DB2](#)
- [Trigger and Recipe Performance](#)
- [CICS Request/Reply](#)
- [CICS HVT Triggers](#)
- [CICS Trigger Queue Verify Task](#)
- [CICS 3270 Assistance Task](#)
- [CICS 3270 Dead Message Queue Task](#)
- [CICS Security Settings](#)
- [Operations Cookbook](#)

## Substation ES CICS Region Resources

Substation ES CICS resources are defined at installation and used by Substation ES during execution.

For a description of the Substation ES CICS IVP resources, see "Sample IVP Resources" in the *TIBCO Substation ES Configuration and Resources* manual.

### Connections

Connections, which are defined as External CICS Interface (EXCI), enable communication between the Substation ES region and Substation ES CICS servers and resources.

A CICS resource can connect to 1 to 8 Substation CICS interfaces (either in the same Substation ES or different ones).

The following table lists and describes CICS connections:

#### *CICS Connections*

| Connection | Description   |
|------------|---|
| SXCA       | The Administrative sever connection.                                      |
| SXCD       | The Dynamic Program Link (DPL) server and application program connection. |

| Connection | Description  |
|------------|--|
| SXCT       | The trigger server connection for requests initiating from within the CICS region. |

### Tasks

A Substation ES region invokes tasks through an EXCI connection. Those tasks communicate with internal Substation ES CICS servers or task definitions in the CICS region to invoke Substation ES CICS servers or programs.

The following table lists and describes CICS tasks:

#### *CICS Tasks*

| Task | Description  |
|------|--|
| SXBX | The CICS mirror transaction used by a Substation ES region to invoke programs that utilize DSA storage below the 16 MB line. This task is typically used for older CICS programs that are nonreentrant. You can set the task on a recipe definition. |
| SXEX | The CICS mirror transaction used by a Substation ES region to invoke programs that use CICS extended storage. You can set the task on a recipe definition.   |
| SXPA | The internal posting task by a High Volume Trigger (HVT) queue threshold event in a Substation ES CICS region.   |
| SXPG | The internal posting task by a Guaranteed Trigger event in a Substation ES CICS region.  |
| SXPT | The internal posting task by a Reliable Trigger event in a Substation ES CICS region.  |
| SXTD | The Substation ES CICS sample program that displays the records on Dead Message Queue (DMQ) and that can delete and resend records.  |
| SXTE | An internal task with which the Substation ES CICS region writes error messages to the queue task.   |
| SXTH | The Substation ES CICS online help task, which shows Substation ES queue usages, error messages, and trigger and request or reply information.   |
| SXTO | The internal Substation ES CICS-initiated request or reply timeout task.   |
| SXTS | The task that invokes the Substation ES CICS startup program when the program list table (PLT) entries have not yet been installed.  |
| SXTU | The task that invokes Substation ES CICS Token Storage Manager.  |
| SXTV | The task that verifies and displays High Volume Trigger (HVT) queues.  |

## Programs

Substation ES region programs are invoked by tasks or linked to other programs in the CICS region. CICS programs are normally used internally, with the exception of the SXCREQR program, which is called by user applications during the CICS-initiated request or reply process.

The following table lists and describes CICS programs:

### *CICS programs*

| Program                 | Description  |
|-------------------------|--|
| <b>Administrative</b>   |  |
| SXCOSRVA                | <p>The Substation ES Administration server, which is called under the following conditions:</p> <ul style="list-style-type: none"> <li>• At Substation ES startup for a sanity check</li> <li>• During trigger processing</li> <li>• At CICS shutdown</li> <li>• At Substation ES termination</li> </ul> |
| SXCOSTRT                | The Substation ES startup program that is initiated by CICS at startup. This program must be defined to the CICS PLT startup routines.   |
| SXCOSHUT                | The Substation ES shutdown program that is initiated by CICS at shutdown. This program must be defined to the CICS PLT shutdown routines.  |
| SXCOHELP                | The Substation ES CICS program that provides help and displays information on Substation ES error messages and status. See <a href="#">Substation ES Operations</a> for detailed information.  |
| SXCOERR                 | The internal Substation ES CICS program that writes Substation's CICS region errors to an error queue file.  |
| <b>Server</b>           |  |
| SXCOSRVD                | The Substation ES DPL server that is called when you must write to a queue, initiate a CICS task with data, or invoke a program to which to pass your data in DFHCOMMAREA.   |
| SXCOSRVG                | The Substation ES DPL server is called when you must invoke a program and pass it to a data area that is greater than 32 KB with Containers, Storage Pointer, or Temporary Storage Queue (TSQ).  |
| SXCOSRVT                | The Substation ES trigger server that is activated when a trigger record or a CICS-initiated request is to be processed by Substation ES.  |
| <b>Request or Reply</b> |  |



| Program           | Description   |
|-------------------|---|
| SXCREQR           | The Substation ES CICS-initiated request or reply requester program to or from which is linked a user application for initiating a request and for receiving a user response. If an error occurs, this program can receive a Substation ES-defined nonzero return and reason code.  |
| SXCREPLY          | The Substation ES CICS-initiated request or reply program that must be invoked by Substation ES when a reply is received from a request initiated from a CICS application. The reply program checks the context information for validity, ensures that the CICS task is still running, and places the reply information of the user in the awaiting user program DFHCOMMAREA. |
| SXCRRT0           | The Substation ES CICS initiated request or reply timeout program that monitors the timeout period of user programs that initiate a request from within CICS and that issues a terminate request if the timeout value is exceeded.  |
| <b>IO Routine</b> |   |
| SXCQWRIT          | The Substation ES HVT routine that is called if you must write a trigger record to the HVT queues.  |
| <b>CICS</b>       |   |
| SXCCABND          | The internal Substation ES CICS program that performs program abend detection within the CICS region. This program recovers from User Program Abend with SXCOABND for large messages.   |
| SXCOABND          | The internal Substation ES CICS program that performs program abend handler within the CICS region. This program produces error messages for User Program Abend and cleans storage for large messages.  |
| SXCOEVPA          | The internal Substation ES CICS program that performs ECB posting of HVT queue threshold events within the CICS region.   |
| SXCOUPD           | The Substation ES CICS program that invokes Substation ES CICS Token Storage Manager.   |
| SXCOEVPE          | The internal Substation ES CICS program that performs ECB posting of Substation ES error events within the CICS region.   |
| SXCOEVPO          | The internal Substation ES CICS program that performs ECB posting of Substation ES-ordered events within the CICS region.   |
| SXCOEVPT          | The internal Substation ES CICS program that performs ECB posting of Substation ES events within the CICS region.   |
| SXCOEVPG          | The internal Substation ES CICS program that performs ECB posting of Substation ES-guaranteed events within the CICS region.  |
| SXCCODMQ          | The internal Substation ES CICS sample program that displays the records on DMQ and that can delete and resend records.   |

| Program  | Description   |
|----------|---|
| SXC3VQUE | The Substation ES CICS program that verifies and displays HVT Queues. |

### Queues

Substation ES uses TDQs for certain processing requirements.

The following table lists and describes CICS Queues:

#### *CICS Queues*

| Queue                                  | Description   |
|--|---|
| SXQG                                   | The queue that holds guaranteed outbound trigger message records.   |
| SXQ0 through SXQ9                      | The queues that hold guaranteed outbound High Volume Triggers message records.  |
| SXQ\$                                  | The queue that holds guaranteed retransmit trigger message records.   |
| SXQT                                   | The queue that holds reliable outbound trigger message records.   |
| SXQU, SXQV, SXQW, SXQX, SXQY, and SXQZ | The queues that hold reliable outbound High Volume Triggers message records.  |
| SXQ@                                   | The queue that holds reliable retransmit trigger message records.   |
| SXQE                                   | The queue that contains Substation ES CICS online error messages.   |
| SXQD                                   | The queue that holds outbound trigger records if Transformer cannot find a conversion definition or if the trigger message was not published during processing. |
| SXQM                                   | The queue that holds a mirror copy of your trigger records after processing by Substation ES.   |
| SXQO                                   | The queue that holds ordered delivery, outbound trigger message records.  |

## Substation ES in a CICSplex Environment

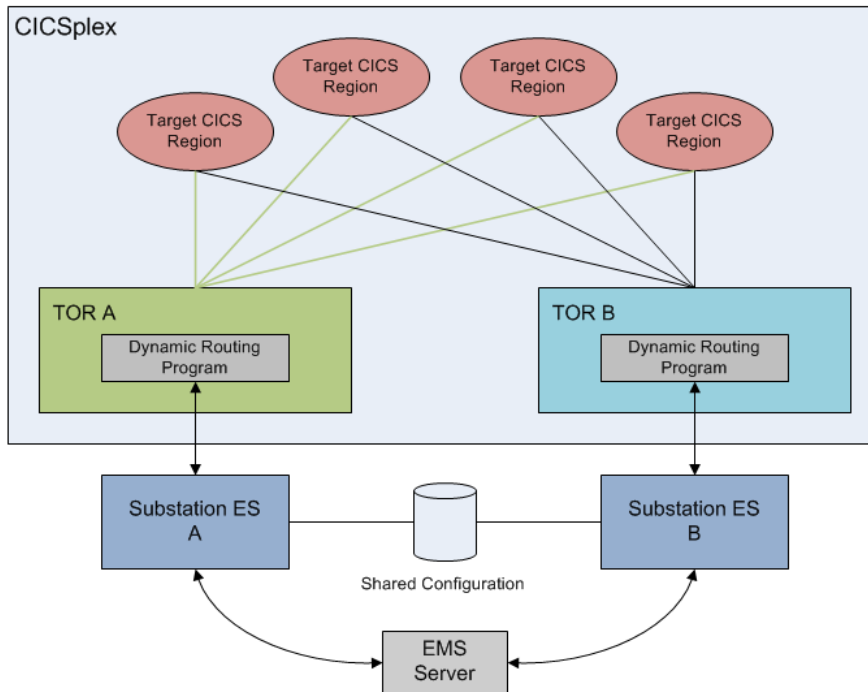
Workload Manager (WLM) of CICSplex SM uses the dynamic routing program EYU9XL0P to route eligible work requests from a requesting region to a suitable target region selected at the time the transaction is initiated.

Together with the EYU9XL0P routine, Substation ES handles the dynamic routing of the following requests:

- Eligible EXEC CICS START requests that are not associated with a terminal.
- DPL requests that are received through EXCI client programs or any function that issues an EXEC CICS LINK PROGRAM request.

The following figure illustrates the architecture in a CICSplex environment:

## Architecture in a CICSplex Environment



Substation ES is configured to interface to Terminal Owning Region (TOR) or Application-Owning Region (AOR) through EXCI. The CICSplex can route the transaction to the proper AOR in CICSplex with EYU9XLOP.

If CICSplex is set up with the proper resources in each AOR, CICSplex directs the transaction to the proper AOR for execution and returns Communications Area or Container to the originating CICS, which in turn returns the user data to Substation ES.

## Methods of Invocation

The following table describes the constraints and usage of Substation ES methods of invocation in a CICSplex environment.

### Constraints and Usage of Substation ES Methods of Invocation in CICSplex

| Recipe MOI Selection | Constraints and Usage  |
|----------------------|--|
| 2 DPL (COMMAREA)     | <p>Can route to a dynamic or static AOR region.</p> <p>Is limited to 32 KB data size for both input and output.</p> <p>Generates the least overhead.</p> <p>Uses DFHCOMMAREA for input and output.</p> |
| 3 PGM (XTCL)         | <p>Cannot dynamically route to an AOR.</p> <p>Substation ES uses a CICS XCTL, which is not routable.</p>   |

| Recipe MOI Selection | Constraints and Usage   |
|----------------------|---|
| 4 Task (Start Task)  | <p>Can route to an AOR.</p> <p>Substation ES uses a CICS START, which is routable.</p> <p>Routes TRANID to the correct AOR according to CICS TDQ definitions.</p> <p>Uses Resource Name as the Tran ID in the CICS Start command and CommArea as the FROM data.</p> <p>Is limited to 32 KB data size for input.</p>   |
| 5 TDQ Write          | <p>Can route to an AOR.</p> <p>Routes the TDQ to the correct AOR according to CICS TDQ definitions.</p> <p>Issues a write to TDQ.</p>   |
| 6 TSQ Write          | <p>Can route to an AOR.</p> <p>Routes the TSQ to the correct AOR according to CICS TSQ definitions (TSMODEL).</p> <p>Is limited to one 32 KB record in TSQ.</p> <p>Writes Temporary Queue Records; the location of the TSQ can be any AOR.</p>  |
| 9 DPL MRO            | <p>Can route to an AOR.</p> <p>Is limited to 32 KB data size for both input and output.</p> <p>Substation ES server issues a CICS LINK to Resource Name and CommArea is the FROM data.</p> <p>Uses DFHCOMMAREA for input and output.</p>  |
| 10 Container         | <p>Can route to an AOR.</p> <p>Uses Containers for Input and Output so user data can be greater than 32 KB.</p> <p>The Substation ES server issues a CICS LINK or a CICS START TRAN using the Resource Name. It can use single or multiple containers for data. When using a single container, the channel and container name can be set in the CICS panel of the recipe. The container invocation method, either Program or Transaction, can be set in the CICS panel of the recipe.</p> |
| 11 Storage           | <p>Cannot route to an AOR.</p> <p>The storage is in the Substation-Owning Region (SOR); AORs do not have visibility to that storage area.</p> <p>Uses Storage Pointers, which means this MOI cannot be routed.</p>  |

| Recipe MOI Selection | Constraints and Usage   |
|----------------------|---|
| 12 DPL TSQ           | <p>Can route to an AOR.</p> <p>Routes the TSQ to the correct AOR according to the CICS TSQ definitions (TSMODEL).</p> <p>Segments data into multiple 32 KB records.</p> <p>Passes a Temporary Queue to the user program; the location of the TSQ and user program can be any AOR.</p> |

## Services without Conversion Rules

Substation ES services can handle communications without the requirement for conversion rules. When this method is used, Substation ES services do not require the definition of conversion rules to communicate with CICS applications or create ESB messages.

Single or multiple containers in CICS can be used for automated interactions. You can follow a few easy configuration steps to implement this flawlessly and efficiently.

See [Recipe Service](#) and [Trigger Service](#) for their respective implementations.

## Recipe Service

You can prepare a recipe service without conversion rules.

Services without conversion rules allow your CICS application to receive containers as well as reply to them using a CICS channel, and supports either multiple or single containers. Unless it is specified on the service CICS options of the recipe, the name of the channel defaults to `SXC-DEF-CHANNEL`.

Using this method also provides the ability to forward destination names and other information each in their own container, allowing more flexibility in the design and usage of your application.

### Multiple Containers

When you invoke CICS resources using containers, Substation ES allows more than one container to be put into a CICS channel that is passed to a CICS application.

To use multiple containers in Substation ES, the recipe service must adhere to the following qualifications:

- Supported message types are:
  - Only EMS MAP or RV messages allow user data fields to be represented as containers with values.
  - For EMS message types, EMS properties cannot be used as data. EMS properties can only be used for Substation ES system fields, such as `tibss-*`.
- Method of Invocation is 10.
- No conversion rules are used.
- Supports invoking a program or transaction.
- Container Process is set to M.

The number of containers is determined by message fields, or by the options selected. Each field name in the message becomes a container name, and the value of the field becomes the container content. The field name must follow valid CICS name conventions, and must be no longer than 16 characters. Field names longer than 16 characters are truncated.

If the **FWD Destination Names** parameter is enabled, Substation ES creates two additional containers to send to CICS named `tibss-ListenTo` and `tibss-ReplyTo`. These containers contain the destination or subject names of this service request.

See Container (CTN) Extended Options in the **CICS Service Details** panel for settings related to the different input message types and the container feature. For more information, see the *TIBCO Substation ES Configuration and Resources* guide.

If you want to use a system field, it can be defined in the **EMS Properties** panel. For details about support fields, see "Using System Fields as Property Fields" in the *TIBCO Substation ES Configuration and Resources* guide.

## Single Containers

To use a single container, the recipe service must meet the following qualifications:

- Message type RV or EMS.
  - For EMS message types, BYTES and TEXT messages are supported.
  - For EMS message types, properties cannot be used as data and can only be used for Substation ES system fields, such as `tibss-*`.
- Method of Invocation is 10.
- No Conversion Rules are used.
- Supports invoking a program or transaction.
- Container Process is set to S.

The channel and container name to be used can be specified in the CICS options of the recipe service definition. If not specified, the default value resource name for each is used.

The following example demonstrates how to set up a recipe service with multiple containers:

```

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

Recipe Identifier      ==> SXC-DPL-NO-RULES
Listen To             ==> tibss.CICS.No.Rules.Request
Reply To              ==>
  Reply To Always      ==> Y
Error Subject          ==> tibss.error.out
Error Threshold        ==> 0 (0 - 99999) Error Percentage ==> 0
Reply To Len (Dyn)     ==> 0 Start ==> 0 Usage Limit ==> 0
Input Conversion (?) ==> (SEE CICS OPT FOR NO CONVERSION)
Output Conversion(?) ==>
Back End System (?) ==> Interface-CICS BES2(?):
  BES3 (?) ==> BES4(?):
ESB Endpoint (?) ==> EMS-SERVER SS-Id Filter: N (N or Y)
Resource Name          ==> SXCCI015
Method of Invocation ==> 10 2 DPL 3 PGM 9 DPL MRO 12 DPL TSQ
  DPL > 32K Invocations 10 Container 11 Storage
                        5 TDQ Write 6 TSQ Write 8 Command
                        4 Task/IMS Tran 7 IMS Conversational

Tran
Trace Level            ==> 0 0-5 (Debugging)
Reply Method           ==> 3 0 None 2 Message
                        1 Status 3 Both Status and Message

----- CICS Service Details -----
Command ==>

Recipe Identifier      ==> SXC-DPL-NO-RULES
Resource Name          ==> SXCCI015
Method of Invocation   ==> 10 Method of Reply ==> 3

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

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

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

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

```

## Trigger Service

You can prepare a trigger service without conversion rules. This allows your CICS application to put multiple containers using a CICS channel, and requires Substation ES Trigger Service to publish them as one message. This allows more flexibility in the design and usage of your program.

The trigger service must meet the following qualifications:

- The message type is Rendezvous or EMS:
  - For EMS message types, only MAP messages are supported.

- For EMS message types, EMS properties cannot be used as data. EMS properties can only be used for Substation ES system fields, such as `tibss-*`.
- No conversion rules are used.

Each container name becomes a message field name, and the context of the container becomes the data of the field in opaque or bytes data type. Note that string data types require special handling. Refer to the following String Data Types section.

If you must use a system field, it can be defined in the EMS Properties panel. For details of supported fields, see "Using System Fields as Property Fields" in the *TIBCO Substation ES Configuration and Resources* guide.

## String Data Types

If you want to send container data as STRING data type of field, your application has to:

- For COBOL, use the copybook member `SXCCMCTF` to nominate the containers that will be something other than an opaque field when the message is finally built.
- Move the container name to `MCTF-NAME` and set `MCTF-TYPE` to `STRING`.



If the message field to be created is opaque, then the entry does not have to be placed into this structure.

- Change the number of `MCTF-GROUP` occurrences if the supplied number is not sufficient. The default for special non-opaque fields is set to 10.
- Add a container named `tibss-TypeF` to the CICS channel containing this structure as the input to it.

Substation ES sends the fields in `MCTF-GROUP` to the message as the STRING fields. The following example shows the structure of the copybook member `SXCCMCTF`:

```
01  MCTF-HEADER-LEN-FW      PIC  S9(8)  COMP VALUE 320.
01  MCTF-CONTAINER-NAME     PIC  X(16)  VALUE  'tibss-TypeF' .

01  MCTF-GROUP .
    05  MCTF-GRP             OCCURS 10  TIMES .
        10  MCTF-NAME        PIC  X(16)  VALUE  SPACES .
        10  MCTF-TYPE        PIC  X(8)   VALUE  SPACES .
            88  MCTF-STRING   VALUE  'STRING' .
        10  FILLER           PIC  X(8)   VALUE  SPACES .
```

## Replace Config File Subjects

You can replace the `Publish To` and `Reply To` destination or subject names that are defined in the config file. The following two special containers are optional:

- `tibss-SendTo`: a value specified in the container replaces the trigger definition value for `Publish To` specified in the service definition during run time.
- `tibss-ReplyTo`: a value specified in the container replaces the trigger definition value for `Reply To` specified in the service definition during run time.



These two special containers do not become user fields.

## Business Process Example

The process to use a trigger service with multiple containers is listed as follows:

1. Create the business data containers required for a channel.
2. Create a container named `SXCQWRIT` with the link function require parameters information from the copybook `SXCCTDQW`.



Make sure that the field service name has a valid value and includes other fields properly filled in.

3. Set the **TDQW-CONTAINER** field of the copybook **SXCCTDQW** to 2.
4. Optionally, the business process can create a container whose name is **tibss-SendTo** and whose content is the destination that the **SendTo** message should be published to.
5. Optionally, the business process can create a container whose name is **tibss-ReplyTo** and whose content is the destination name that the **ReplyTo** message is sent to.
6. Create a channel link to **SXCQWRIT**.

An example of this process can be found in **USERHLQ.COB(SXCCI032)**.

The following example demonstrates how to set up a trigger service without a conversion rule:

```

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

Trigger Identifier    => TRIGGER-NO-RULES
Publish To           => tibss.trigger.no.rules
Publish To, Sec       =>
Reply To             => tibss.trigger.no.rules.reply
Error Subject        => tibss.error.out
Error Threshold       => 0      (0 - 99999)   Error Percent => 0 (0 - 99)
Publish To Len (Dyn)  => 0      Start => 0
Output Conversion(?)  =>                               Msg Field Name => q
ESB Endpoint         (?) => EMS-SERVER
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 => 0
  Value  =>
Undelivered Requeue  => N      (Y Requeue, N Discard - Reliable Only)

```

## Containers and Processes

Substation ES supports two CICS container types: multiple containers and a single container. All containers within UoW are transmitted over a single channel.

The value of the following fields relating to containers can be updated in the CICS panel of a recipe under Container Extended Options:

- **Channel Name:** you can use the field to specify the channel name that you want Substation ES to use during the CICS process. If present, this field overrides the channel default name of **SXCOSRVGCHANNEL** for all container processes.
- **Single CTN Name:** you can use the field to specify the container name that you want Substation ES to use during the CICS single container invocation process. If present, this field overrides that setting for all CICS single Container processes using the recipe.
- **Container Invocation:** you can use the field to specify the method of container invocation. You can either link to a program or start a transaction in a CICS region.
- **Container Process:** you can use the field to specify the type of container to be used. This can be a single container, or multiple containers driven by message fields. The number of containers is determined by message fields.
- **FWD Destination Names:** you can use the field to send the destination and subject name to CICS. If this parameter is enabled and the process has multiple containers without conversion rules, then Substation creates two additional containers named **tibss-ListenTo** and **tibss-ReplyTo** and sends them to CICS. These containers contain the corresponding destination or subject names of this service.

You can also use the MOI 10 recipe resource invocation to specify whether the CICS resource invoked has to be a program or a transaction. When selecting a transaction invocation, no response to the calling

program is possible and you must specify 0 or 1 for Reply Method, where 0 indicates no reply and 1 indicates a status only reply.

## Multiple Containers

With the multi-container support process, you can add multiple containers to a single resource with one call, when invoking CICS resources using containers.

The process puts multiple containers into the CICS channel, which is then passed to the program. This allows greater flexibility in program design and use. **Container Process** in the CICS panel indicates how the different input message types and container features are handled.

To use multiple containers, perform the following steps:

1. In the **Container Process** field in the CICS panel (SXTPO5I), specify M for the message derived type.
2. When configuring the input message type, consider the following points:
  - Do not specify an input conversion rule.
  - EMS properties cannot be used as data value replacements because they are only used for directing a recipe service process such as `tibss-BES` or `tibss-resource`.
  - EMS Bytes messages, Text messages and Stream messages are passed as User Data to one input container. Bytes and Stream types are handled as binary data (OPA). Text type is handled as string.
  - For EMS Map and Rendezvous Messages, each message field is a separate container with the container name being the same as the field name supplied in the message. The type of data in the container is based on the message field type.
3. When configuring the output message type, consider the following points:
  - Do not specify an output conversion rule.
  - EMS properties cannot be populated with user data. However, they can be added in the Recipe properties panel with constant or system data such as `tibss-envelope`, `tibss-ss-id`, `tibss-cfg-id`, and so on.
  - For EMS Bytes, Text and Stream messages, if one container is received from BES, a message is created using all the data from that single output container. If multiple output containers are received, a Substation ES error is generated. No data response message is created.
  - For EMS Map or Rendezvous messages, when one or more output containers exists, each container becomes an opaque field type in the message with a name that matches the container name.
4. During CICS execution, consider these points:
  - Multiple input containers can be created. One or more output containers are allowed, depending on the output message type.
  - The only Reply Methods that are supported are 0 (none) and 1 (status). For a Container Invocation of T (transaction), there is no reply message with data.

## Single Containers

Substation ES supports single container input and output from a program. The input and output containers have the same name as the program to be called, while the default channel name used is `SXCOSRVGCHANNEL`.

To use a single container, perform the following steps:

1. In the **Container Process** field in the CICS panel (SXTPO5I), specify S for single container type.
2. When configuring Input Message Type, consider these points:

- EMS Bytes messages, Text messages and Stream messages are passed as user data to input container. Bytes and Stream types are handled as binary data (OPA). Text type is handled as string.
  - EMS Map and Rendezvous messages are processed according to conversion rule and are passed as user data to input container.
3. When configuring the Output Message Type, consider these points:
- EMS Bytes messages, text messages and Stream messages are created by using all the data in the single output container.
  - EMS Map messages and Rendezvous messages are created by using the output conversion rule and the single output container as the user buffer to generate fields. The names of the fields are derived from the conversion rule.
4. During CICS execution, consider these points:
- Only one input container is created and only one output container is allowed. The name of both containers (input and output) is based on whether **Single CTN Name** is specified. If a user program creates more than one container, only the expected named container is used. All others are ignored.
  - The only Reply Methods that are supported are 0 (none) and 1 (status). For a Container Invocation of T (transaction), there is no reply message with data.

## Invoking Multiple Resources

The Invoke Multiple CICS Resources feature allows one RV or EMS message received by Substation ES to produce up to five similar invocations to different resources in a CICS address space.

Substation ES listens to an RV subject or an EMS destination as a normal recipe service does. When the recipe service has multiple resources defined, this service synchronously processes multiple CICS invocations using the same data content. Up to four additional resource fields or a total of five resources can be used. This allows the Substation ES CICS interface to run up to five transactions, all with the same characteristics and user payload.

To invoke multiple resources, the recipe service must meet these qualifications:

- Message type is RV or EMS.
- For EMS message types:
  - MAP, BYTES, and TEXT message types are supported.
  - EMS Recipe Limit is not supported.
  - The `tibss-BES` property is supported.
- Method of Reply must be specified as zero (0) - NONE.
- The same conversion rule is used for all invoke resources.
- Alt-BES is supported.
- The `tibss-resource` property can be used to change first resource value. Additional resources are executed as specified in the recipe service.
- Method of Invocation is one of the following:
  - 2 DPL
  - 3 PGM
  - 4 Task

- 5 TDQ Write
- 6 TSQ Write



This feature cannot be used in conjunction with the multi-container feature. That is, multi-execution of multiple containers resources is not supported.

The presence of Resource 2 through 5 in the recipe indicates additional CICS invocations. The resources are executed serially. If a transaction for a resource abends, Substation ES records the error and does not execute other resources that might still be eligible.

Substation ES does not stop the resources from executing until they are either completed successfully or the business process abends. If the business process simply returns an error indicator in the return message, Substation ES does not recognize that and continues processing the other resources. To stop the multi-resource execution, the business process must abend or receive an error indicator from EXCI.

The following example demonstrates how to set up a recipe service that invokes multiple CICS resources. The recipe service invokes the SXC3I001 program three times and the SXCCI001 program twice.

```

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

Recipe Identifier      ==> SXC-DPL-MULTI-EXEC
Listen To             ==> tibss.CICS.Multi.Exec.Request
Reply To              ==>
  Reply To Always      ==> Y
Error Subject         ==> tibss.error.out
Error Threshold        ==> 0      (0 - 99999)      Error Percentage ==> 0
Reply To Len (Dyn)     ==> 0      Start ==> 0        Usage Limit ==> 0
Input Conversion (?)   ==> CVR-I-BYTE      (SEE CICS OPT FOR NO CONVERSION)
Output Conversion(?)   ==> CVR-O-BYTE
Back End System (?)   ==> Interface-CICS    BES2(?):
  BES3 (?) ==>                               BES4(?):
ESB Endpoint (?)      ==> EMS-SERVER        SS-Id Filter: N (N or Y)
Resource Name         ==> SXC3I001
Method of Invocation   ==> 2      2 DPL      3 PGM      9 DPL MRO      12 DPL TSQ
  DPL > 32K Invocations      10 Container      11 Storage
                             5 TDQ Write        6 TSQ Write  8 Command
                             4 Task/IMS Tran    7 IMS Conversational Tran

Trace Level           ==> 0      0-5 (Debugging)
Reply Method          ==> 0      0 None        2 Message
                             1 Status        3 Both Status and Message

----- CICS Service Details -----
Command ==>

Recipe Identifier      ==> SXC-DPL-MULTI-EXEC
Resource Name         ==> SXC3I001
Method of Invocation   ==> 2 Method of Reply ==> 0

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

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

Multi-Execution
  Resource 2: SXC3I001   3: SXC3I001   4: SXCCI001   5: SXCCI001

```

## Using FastPath Trigger Processes

To improve the performance when processing trigger messages, Substation ES has FastPath trigger definitions. These definitions have a shorter instruction path and minimal API calls to TIBCO Rendezvous or TIBCO Enterprise Message Service. The result is faster processing of trigger messages and less CPU used to accomplish trigger publishing.

To determine whether your Trigger Service is eligible for Fast Path operations, enter an operational command of your choice and display the trigger service. You can see text next to the **Undelivered Requeue** label stating whether FastPath is supported.

### Defining a FastPath Trigger

The following steps show you how to configure a FastPath trigger.

For details on the fields and their options, see the section in the Trigger Details panel in the *TIBCO Substation ES Configuration and Resources* manual.

#### Procedure

1. From the main ISPF panel, select option 7, Define Triggers - Outbound Convert / Destinations.
2. On the command line, enter **ADD** to create a new trigger.
3. Select the ESB Endpoint to be used by the trigger and press Enter.
4. Optionally, select a conversion rule and press PF3.
5. Specify Trigger Identifier.
6. Specify a destination name for the **Publish To** field.
7. Specify a destination name for the **Publish To Sec** field if needed.
8. Specify Trace Level for your desired level of debugging.
9. Specify Output Method of either 2 or 3.
10. Modify Compare Text as needed.
11. Modify Undelivered Requeue as needed.
12. Configure the endpoint, as described in the appropriate section:
  - [Endpoint Configuration for Rendezvous ESB](#)
  - [Endpoint Configuration for EMS ESB](#)

### Endpoint Configuration for Rendezvous ESB

If the endpoint is a Rendezvous ESB:

- By default, the trigger service creates the message using data from BES buffer in one field named **DATA**. The size is the User Data Length.
- Optionally, you can specify a dynamic destination name length and start position in the **Publish To Len (Dyn)** field.



The **PROP** field is not supported with FastPath, and can not be used.

#### Definition Rules

When using the FastPath process with a trigger service, consider the following requirements if you are using Rendezvous ESB.



The CICS Request/Reply (REQR) can deploy FastPath Trigger for both EMS and Rendezvous ESBs. The REQR process automatically adds a **tibss-envelope** field to the message when FastPath is configured.

To use FastPath to process a trigger that is configured with a Rendezvous endpoint, the trigger definition must meet the following requirements:

- The trigger definition must use **Output Method** 2 or 3 only.
- The trigger definition must not include settings for these fields:
  - **Conversion Rule**
  - **Publish To, Sec**

## Endpoint Configuration for EMS ESB

When you use the FastPath process with a trigger service, consider the following requirements if the endpoint is EMS ESB.

1. Modify **Output Dest Type** as needed.
2. Modify the **Output Message** properties as needed:
  - a. Modify Priority as needed.
  - b. Modify Expiration as needed.
  - c. Modify Message Type as needed.
    - **B**(ytes): send as a bytes message with user data from BES buffer. The size is determined by the User Data Length.
    - **T**(ext): sends a text message with the buffer data. The TEXT message ends with a null-terminator.
    - **S**(tream): sends a stream message with user data from BES buffer. The size is determined by the User Data Length.
    - **M**(ap): sends a MAP message using data from BES buffer in one field named **DATA**. The size is determined by the User Data Length.



- The **PROP** field is not supported with FastPath, and should not be used.
- The EMS ESB does not support the use of the **Publish To Len (Dyn)** field with FastPath.

### Definition Rules

When using the FastPath process with a trigger service, consider the following requirements if you are using an EMS ESB.



The CICS Request/Reply (REQR) can deploy a FastPath Trigger for both EMS and Rendezvous ESBs. The REQR process automatically adds a **tibss-envelope** field to the message when FastPath is configured.

In order to use FastPath to process a trigger that is configured with an EMS endpoint, the ESB and trigger definitions must meet the following requirements:

- The ESB SIP member must have EMS-USE-PROP set to **Y**.
- The trigger definition must use **Output Method** 2 or 3 only.
- The trigger definition must NOT include settings for these fields:
  - **Conversion Rule**

- **Publish To Len (Dyn):** (must be zero (0))
- **Compression:** (must be 'N')
- **Preserve:** (must be 'N')
- **No Body Trace:**(must be 'N')

## Invocation of CICS Programs to Access DB2

You can write programs in COBOL, PLI, C, and Assembler to access DB2. For Substation ES to execute these programs, they must be able to take input and deliver output in one of following ways:

- **COMMAREA:** data is presented to the program in COMMAREA. Upon return from the program, the content of COMMAREA is considered the output of the program and is sent back to Substation ES.

The maximum input and output length of the COMMAREA is 32,000 characters.

- **Container:** data is presented to the program in CICS Container whose name is the same as the program name. The program returns its response in the same Container. In the absence of response from the user program, the program deletes the Container. Upon return from the program, the content of the Container is returned to Substation ES. If no Container is found, the response is No Response.

A Container can be any length. A container size is constrained only by the available user storage in the CICS address space.

- **TSQ name in COMMAREA:** data is presented to the program in CICSTSQ. The eight-character TSQ name is in COMMAREA. The program reads the TSQ to obtain input. Output from the program is placed in the same TSQ. Upon program return, the content of the TSQ is used by Substation ES as the response. If the TSQ is not found, then a NULL response is returned.

The size of input and output data is only limited by TSQ constraints and storage to support the size of the TSQ.

- **Storage pointer in COMMAREA:** data is presented to the program as a pointer to a storage area and length. Those two fields are in the COMMAREA. The program can access the storage directly. Upon return from the user program, the content of the storage area is used by Substation ES as the response.

## DB2 Access Considerations

The CICS region with which Substation ES interfaces is called the SOR. An application that runs the CICS region is called the AOR.



The default transaction ID under which Substation runs programs is SXEX.

You can define CICS programs to access DB2 in the following ways:

- **The program executes in the SOR that can execute under the SXEX transaction ID.**

This is the typical Substation ES method of executing a CICS program. Define the recipe as you would a normal recipe. In the Recipe Details panel, do not set a value for the **Mirror Tran** field and set User Tran to N.

- **The program executes in the SOR and must execute under a transaction ID other than SXEX.**

Two variations apply:

- The transaction ID to execute the program cannot point to the program DFHMIRS.

In the Recipe Details panel, set User Tran to Y. The transaction then starts as CICS Mirror Transaction (CSMI), and then switch to the **Mirror Tran** field.

- The transaction ID to execute the program can point to program DFHMIRS.

Define the transaction to execute DFHMIRS. In the Recipe Details panel, set Mirror Tran to the transaction ID under which the program must run. Set User Tran to N.

- **The program executes in AOR that can execute under the SXEX transaction ID.**

Define the recipe as you would a normal recipe. In the Recipe Details panel, do not set a value for the **Mirror Tran** field and set User Tran to N.

Define the SXEX transaction in AOR as you would define it in SOR.

The program definition in SOR must contain the REMOTE ATTRIBUTES for routing the program to the proper AOR for execution.

Be sure to set up the program definition in AOR correctly.

- **The program executes in AOR and must execute under a transaction ID other than SXEX.**

Define the recipe as you would a normal recipe. In the Recipe Details panel, Mirror Tran specifies the transaction ID under which the program must run. Set User Tran to N.

Define the Mirror Tran transaction in SOR and AOR with the program name of DFHMIRS.

The program definition in SOR must contain the REMOTE ATTRIBUTES for routing the program to the proper AOR for execution.

Be sure to set up the program definition in the AOR correctly.

## DB2Entry and DB2Tran for Programs That Access DB2

If your CICS program requires a DB2ENTRY object, do one of the following:

- If the DB2 program runs in SOR under SXEX, define a DB2TRAN object in SOR for SXEX.
- If the DB2 program runs in SOR under Mirror Trans, define a DB2TRAN object in SOR for the Mirror Tran transaction ID in the recipe.
- If the DB2 program runs in AOR under SXEX, define a DB2TRAN object in AOR for SXEX.
- If the DB2 program runs in AOR under Mirror Trans, define a DB2TRAN object in AOR for the Mirror Tran transaction ID in the recipe.

## Security Considerations

The user identifier under which the program executes is the same as the parameter USERID defined in the CICS interface parameters member. The transaction identifier under which the program executes depends on the Substation ES recipe, as follows:

- If the recipe does not specify Mirror Tran, the program runs under the Substation ES default transaction SXEX.
- If the recipe specifies Mirror Tran and User Tran is set to N, the program runs under Mirror Tran.
- If the recipe specifies Mirror Tran and User Tran is set to Y, the program starts under CSMI and then runs under Mirror Tran.

Your security system must allow the program and transaction to execute. You might have to add or modify security definitions for proper program execution to occur.

## Scenarios of CICS DB2 Transaction Definition

Three DB2 transaction definition scenarios are described as follows:

- **The user DB2 program does not need a CICS DB2Tran definition and can run under the transaction ID SXEX or Mirror Tran.**



Define the recipe in Substation ES as you would any other CICS program.

- **The user DB2 program needs a CICS DB2Tran definition or cannot run under transaction ID SXEX or Mirror Tran, and does not need to be routed.**

Do the following items for the CICS environment:

- Define the user DB2 program to CICS that uses COMMAREA, Container, or TSQ for input and output.
- Define the transaction ID to run the program for DB2Entry matching, if necessary.
- Define DB2Entry, if necessary.
- Define DB2Tran, if necessary.

Do the following items for security:

- Allow CSMI to link to the user DB2 program.
- Allow Userid in Substation ES to run CSMI, DFHMIRS, and the user DB2 program.

Determine the following definitions:

- CSMI: determines if the following IBM default definitions are appropriate for running your program. If not, redefine and test them:  
     TASKDATAloc: Below  
     TASKDATAKey: User  
     DYNAMIC: No  
     ROUTABLE: No
- DFHMIRS: determines if the following IBM default definitions are appropriate for running your program. If not, redefine and test them:  
     DATAlocation: Below  
     DYNAMIC: No

Do the following items for the Substation ES environment:

- Define the message as you would with other recipes.
- Define the buffer as you would with other recipes.
- Define the conversion as you would with other recipes.
- Define the recipe:

Define the fields as you would with other recipes.

Set **Method of Invocation** to 2 DPL, 10 Container, 11 Storage, or 12 DPL TSQ.

**Optional.** If the user DB2 program must run under a transaction ID other than SXEX, set Mirror Tran to the transaction ID for the CICS environment.

The user DB2 program initially starts for the transaction CSMI and the program DFHMIRS, which then changes the transaction ID to the value you defined as the mirror transaction. This value is for matching DB2TRAN entries and thus DB2Entry for the DB2 program. EIBTRANID is set to Mirror Tran so that the user DB2 program sees itself as running under the transaction ID of the Mirror Tran value.

- **The user DB2 program needs a CICS DB2Tran definition. Otherwise, the program cannot run under transaction ID SXEX and must be routed.**

Do the following items for the CICS environment:

- Define the user DB2 program to CICS that uses COMMAREA, Container, or TSQ for input and output in the AOR.
- Define the transaction ID to run the program in the AOR with program name of DFHMIRS.
- Define DB2Entry in the AOR, if necessary.
- Define DB2Tran in the AOR, if necessary.
- Define the transaction ID to run the program in the CICS that interfaces to the SOR with the program name DFHMIRS, using REMOTE ATTRIBUTES to define how to route the transaction to the AOR.

Do the following items for the Substation ES environment:

- Define the message as you would with other recipes.
- Define the buffer as you would with other recipes.
- Define the conversion as you would with other recipes.
- Define the recipe:
  - Define the fields as you would with other recipes.
  - Set Method of Invocation to 2 DPL, 10 Container, 11 Storage, or 12 DPL TSQ (if DPL TSQ is used), and then direct TSQ requests in AOR to SOR for the IDs that start with SXCT.
  - Set Mirror Tran to the transaction ID for the CICS environment.
  - Set User Tran to N.

## Trigger and Recipe Performance

The following section gives a general description of the components that affect the performance of Substation ES trigger and recipe processes.

### Trigger Processing

Substation ES triggers make heavy use of the CICS intrapartition file called DFHINTRA, whose physical record size can have a significant effect on the performance of Substation ES trigger and HVT processes.

Note the following items:

- The standard Substation ES trigger size is limited by the record size of DFHINTRA.
- Substation ES HVT segments large messages into multi-TDQ records. The larger the DFHINTRA record size (maximum 32 KB), the less segmentation is required. Less segmentation improves performance.
- Reliable trigger messages do not use the RRS facility. In case of an error in message delivery to the EMS server or Rendezvous endpoint, the message could be lost. Because of the reduced recovery, there is less overhead in processing a reliable message. HVT Reliable Triggers that must be segmented (because the message does not fit in one TDQ buffer) also use the TSQ. For those types of messages, TSQ tuning might be required.

Guaranteed trigger messages must use the RRS facility to ensure message delivery to the EMS server. Rendezvous does not support guaranteed delivery.

- The HVT TSQ facility only uses TSQ records. It does not use the RRS facility. If the message cannot be delivered to the ESB, the message is lost. If Substation ES is unavailable, HVT TSQ are stored in CICS Temporary Storage File (DFHTEMP). If CICS is re-cycled, the messages in TSQ are lost unless the system uses recoverable TSQs.

- Specify the **TS** = (*buffers, strings*) parameter on the CICS SIT to monitor the TSQ (DFHTEMP) statistics on the use of resources. A good starting point for the TS setting is as follows:

```
buffers = 5 + WORKERS in CICS definition + HVT-WORKERS
strings = WORKERS in CICS definition + HVT-WORKERS
```

- Specifying **TD** = (*buffers, strings*) parameter on the CICS SIT to monitor the DFHINTRA statistics on the use of resources. A good starting point for the TD setting is as follows:

```
buffers = 5 + WORKERS in CICS definition + HVT-WORKERS
strings = WORKERS in CICS definition + HVT-WORKERS
```

- Trigger use based on ServiceName lookup is more efficient than Buffer Value Match. This is because the Trigger list is kept sorted in alphabetic order and the match process stops after either finding a ServiceName match or going past the first possible location in the sorted order.

For faster matching, frequently used Trigger ServiceNames should be at the beginning of the list. That is, the service names should begin with "A".

Following is an example of the CICS Transient Data statistics. The highlighted numbers must always be zero for best performance.

```
Transient
Data
Transient data reads. . . . . : 1,380,551
Transient data writes . . . . . : 2,179,981
Transient data formatting writes. . . . . : 0
Control interval size . . . . . : 18,432
Control intervals in the DFHINTRA dataset . . : 94,320
Peak control intervals used . . . . . : 94,320
Times NOSPACE on DFHINTRA occurred. . . . . : 11
Transient data strings. . . . . : 3
Times Transient data string in use. . . . . : 3,560,530
Peak Transient data strings in use. . . . . : 3
Times string wait occurred. . . . . : 1,958,001
Peak users waiting on string. . . . . : 12
Transient data buffers. . . . . : 40
Times Transient data buffer in use. . . . . : 7,379,700
Peak Transient data buffers in use. . . . . : 15
Peak buffers containing valid data. . . . . : 40
Times buffer wait occurred. . . . . : 0
Peak users waiting on buffer. . . . . : 0
```

## Recipes (Request or Reply)

The **WORKERS** parameter specifies the number of concurrent CICS sessions that are started to service Substation ES requests. For CICS business applications, this is the maximum number of transactions that can be run in CICS concurrently.

To raise the level of concurrency, increase that number. Bear in mind that the **WORKERS** setting affects the value you specify on the CICS **MAXTASK** parameter.

## SXEX CICS Priority

The default priority for the SXEX transaction is 1, and it must be changed to your installation normal business transactions priority plus 1. For example, if your normal business transaction priority is 20, you must set your SXEX priority to 21.

This gives Substation ES transactions better execution and faster response time.

## CICS Request/Reply

This section explains CICS Request/Reply (CRR) operations. The CICS Request/Reply facility provides an easy-to-use method for requesting information from ESB services.

A request can be sent by any business process simply by calling SXCREQR. The business process is a CICS program written in COBOL, C, PL1, or Assembler. The basic flow is as follows:

1. The business process formats a **SXCREQR** parameter list. This tells **SXCREQR** where the outgoing data is, where to put the response, and how long to wait for the response.
2. The business process then does a CICS LINK to **SXCREQR** with a parameter list in **COMMAREA**.
3. When **SXCREQR** returns to the business process, either the response data is presented or an error is indicated.



*USERHLQ.COB(SXCCIRR1)* provides an example of a COBOL program using **SXCREQR**.

The size of the input to **SXCREQR** is limited to the size of the **COMMAREA**. The outgoing message size is limited to the value of the **RR-MAXSIZE** control parameter in the CICS interface definition for Substation ES. This normally is 8192, but can be changed to a value from 2048 to 32000.

The output can be returned as data in the same **COMMAREA** as the input, or to a TSQ or storage pointer.

### Detailed Flow

The following steps gives the Request/Reply flow in more detail:

1. In the CICS user program, configure the **SXCREQR** parameter.
2. In the CICS user program, link to **SXCREQR** with parameters and data in **COMMAREA**.
3. **SXCREQR** validates parameters and returns an error code if any are invalid.
4. **SXCREQR** gets a CRR token. There are 100 of these reusable tracking control blocks available.
5. **SXCREQR** creates a TSQ entry for user data.
6. **SXCREQR** posts an ECB, which wakes up **SXCOSRVA**, a Substation ES administration program.
7. If one is not already active, **SXCREQR** starts an **SXTO** transaction for timeout processing.
8. **SXCREQR** waits for a response, either from the timeout module or the response module.
9. The **SXCOSRVA** task returns to Substation ES indicating an S **SXCREQR** event.
10. Substation ES starts a new transaction for program **SXCOSRVT**.
11. **SXCOSRVT** gets data from TSQ and returns the data to Substation ES.
12. Substation ES finds a matching trigger definition using a buffer value match.
13. Substation ES publishes message according to the trigger definition, using the **tibss-envelope** value.
14. The ESB process gets the message, processes it, and issues a response message that includes the **tibss-envelope**.
15. Substation ES consumes the response message using a recipe definition.
16. Substation ES starts a new transaction in CICS for **SXCREPLY** using the response data and envelope.
17. **SXCREPLY** finds the correct CRR token using the envelope value.
18. **SXCREPLY** moves the response data to the appropriate area, whether the **COMMAREA**, TSQ, or Storage Pointer, and posts the **SXCREQR** ECB.
19. **SXCREQR** wakes up and returns to caller.
20. The CICS user program can now process the response data or error condition.

## CICS HVT Triggers

This section explains CICS HVT triggers that you can choose to use in certain situations.

Consider using the CICS HVT capability of Substation ES if you require any of the following features:

- You expect a high volume of messages or queue records to be sent from CICS regions. The throughput consideration would be around 200 messages per second.

- You require ordered delivery of information.
- Your applications might send more than 32 KB of information. Trigger data can exceed 32 KB.
- You use TIBCO Enterprise Message Service as your transport and require guaranteed message delivery for triggered information.
- The information to be sent can reside in DFHCOMMAREA, TSQ, storage, or a container.
- There are two facilities for storing data during the HVT process:
  - Transient Data Queues (TDQ). To use TDQ, your program links to SXCQWRIT.
  - Temporary Storage Queues (TSQ). To use TSQ, your program links to SXCQWTSQ.

Both programs use the same input format.

If you are using the Container or TSQ input option for HVT, the process works in an MRO environment (with proper MRO definitions). The storage pointer option does not work in a MRO environment.

Substation ES provides the program SXCQWTSQ, which any CICS application can link to. SXCQWTSQ creates a new TSQ entry (name) for each link to SXCQWTSQ. The TSQ created is limited to one record with a maximum size of 32000. The TSQ normally uses auxiliary storage. Using TSMODEL allows you to change the normal operation to MAIN.

HVT using TSQ has no recover or backout. Once the link to SXCQWTSQ is done the TSQ entry is created and scheduled for delivery to Substation ES. The HVT TSQ request could be sent to the ESB before the business program that links to SXCQWTSQ ends. Because Substation ES processes HVT TSQ entries as fast as possible, the entities are not processed in any order.

The previous methods of directly writing to TDQ are still supported, but impose the same restrictions as the previous versions of the product. All future enhancements will be made according to the HVT concept. HVT is a major enhancement to enable Event Driven Architecture (EDA) for High-Volume Trigger processing in CICS.

## Usage Samples

Substation ES contains fully functional C and COBOL code samples on how to call and use HVT.

The code samples can help you test and understand how HVT operates. They are listed as follows:

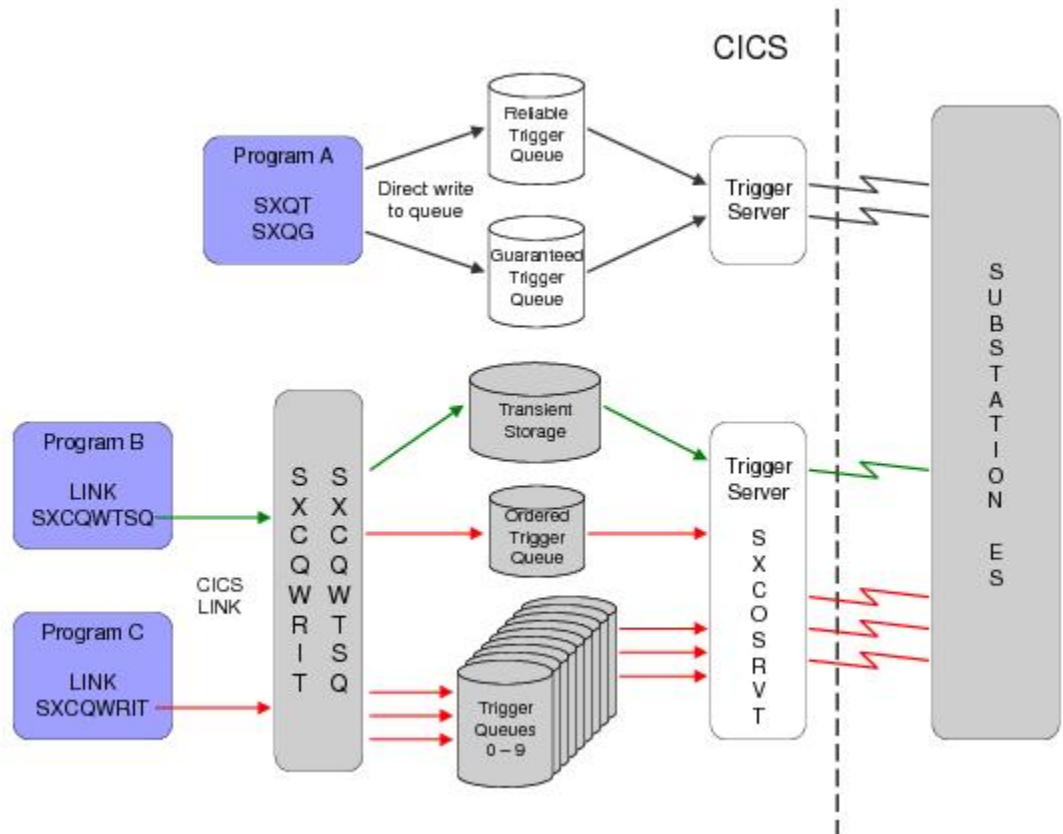
- The COBOL program SXCCI032: the associated copybook is SXCCTDQW. The transaction SXTF invokes this program.
- The C program SXC3I032: the associated header file is SXCHTDQW. The transaction SXT6 invokes this program.

## Substation ES Trigger Flow

The section describes the trigger flow of Substation ES.

The following figure illustrates the trigger flow.

## Trigger Flow



As shown in the Trigger Flow figure , you use standard CICS calls and facilities for HVT. The process proceeds as follows:

1. A user program sets fields in the copybook or structure to inform `SXCQWRIT` of what and where the information is to which it is sent. The fields to populate are normally the type of a record, the data length, and the data; or an address of the data.
  - The data can also be located in TSQ. Just specify a TSQ name, which can be made MRO capable.
  - The data can also be in a CICS container, which can be made MRO capable.
2. The user program calls `SXCQWRIT` or `SXCQWTSQ` with `EXEC CICS LINK`.
3. `SXCQWRIT` adds data to TDQ and `SXCQWTSQ` adds data to TSQ known to Substation ES.
4. Information passed back to the user program contains a return code, a reason code, and the name of the TDQ or TSQ to which the program wrote. Always examine the return code.



If the return code is nonzero, check the rollback field. If this indicator is Y, you must perform a rollback because partial records might have been written to TDQ, causing unpredictable results. Links using TSQ cannot be rolled back.

## Ordered Message Delivery

If ordered messages are sent, data is written to the TDQ `SXCO`.

There is only one of these TDQs, and synchronization is applied to preserve the order of messages. Specify the correct TDQ type in the parameters passed to the `SXCQWRIT` routine.

## Configuration

By default, you can use two guaranteed queues (TDQ type R) and TSQ (type S) and one reliable queue (TDQ type G) even if Substation ES has not been connected.

You can configure the Substation ES startup to use more than the defaults, depending on the number of CICS applications that write to those queues. If you have a high volume of Guaranteed type messages to deliver or a large volume of data to process, it is recommended that you set the **HVT-WORKERS** parameter value to the maximum of 10.

For additional information on setting up and configuring Substation ES, see the *TIBCO Substation ES Installation* manual.

When a transaction links to SXCQWRIT the first time, SXCQWRIT is allocated to one of the queues. All subsequent links to SXCQWRIT by that task use the same allocated queue. A transaction can link to SXCQWRIT multiple times, but there is no switching between queues for that transaction (task). For example, if the TDQ SXQ0 is selected on the first link to SXCQWRIT, all further links to SXCQWRIT use SXQ0. When another task first links to SXCQWRIT, SXCQWRIT is allocated a different queue. That way, the number of “CICS enqueues” due to tasks with the same TDQ is greatly reduced. Links to SXCQWTSQ for the TSQ facility do not have an enqueue problem.

SXCQWRIT can handle data lengths and container information of sizes greater than 32 KB. Remember that many large data records occupy huge storage areas and can impact performance. Large records are broken up into multiple Transient Data Records before they are sent to the Substation ES trigger server. User data located in TSQs are still limited to the CICS limitations of 32,767, which is the maximum for a signed halfword. Using containers is far more efficient.

Programs can combine both links to SXCQWRIT, SXCQWTSQ, and WRITEQ to SXQT or SXQG in them. SXCQWTSQ can handle data lengths and Container information of sizes up to 32,000.

## Code Samples: Link to SXCQWRIT or SXCQWTSQ

COBOL and C code samples are listed as follows, which demonstrate how to link to **SXCQWRIT**.

- **COBOL**

```
MOVE LOW-VALUES TO TDQW-TSQ-Name
SET TDQW-PDATA TO ADDRESS OF DATAAREA.
MOVE LENGTH OF DATAAREA TO TDQW-DATALEN.
MOVE {'R', 'G', 'O', 'S'} TO TDQW-TDQ-TYPE.
EXEC CICS LINK PROGRAM('SXCQWRIT' or 'SXCQWTSQ')
              COMMAREA(SXCCTDQW)
              LENGTH(TDQW-Header-Len)
              RESP(DFHRESP1)
              RESP2(DFHRESP2)
END-EXEC.
IF TDQW-RETURN NOT = TDQW-NORMAL
  TAKE-ACTION.
```

- **C**

```
short int sTDQW = sizeof(SXCHTDQWR);
SXCHTDQWR TDQW;

memset(&TDQW.Tsq_Name[0],0x00,16);
TDQW.pData = address of data;
TDQW.DataLen = length of data;
TDQW.Tdq_Type = {'R', 'G', 'O', 'S'};
EXEC CICS LINK PROGRAM('SXCQWRIT' or 'SXCQWTSQ')
              COMMAREA(&TDQW)
              LENGTH(sTDQW);
if (TDQW.Return != 0)
  Take_action;
```

## SXCQWRIT Parameter Fields

The following table describes the fields of the **SXCQWRIT** parameter.

### *SXCQWRIT Parameter Fields*

| Field                 | Type              | Description  |
|-----------------------|-------------------|--|
| <b>TDQW-SERVICE</b>   | PIC X(40)         | If this field is nonzero or non-blank, Substation ES uses Trigger Service Name to process this message and ignores the Compare Text Contain in the Buffer area of the Trigger Definition.  |
| <b>TDQW-TSQ-Name</b>  | PIC X(16)         | This field is the name of the TSQ in which the data is located. If this field is nonzero or non-blank, SXCQWRIT looks for the data in either the CICS temporary storage or a Container. If the data location is a TSQ, SXCQWRIT sends all the records in the TSQ to the Substation trigger server and then deletes the TSQ. SXCQWTSQ only writes the first TSQ input record. |
| <b>TDQW-DATALEN</b>   | PIC S9(8)<br>COMP | This field is the length of the data in the case of a data pointer. Otherwise, the value must be zero.   |
| <b>TDQW-PDATA</b>     | POINTER           | This field is the pointer to the data or NULL (binary zeros).<br><br>Data Pointers cannot be used for the MRO processing because the address is local to Address Space.  |
| <b>TDQW-CONTAINER</b> | PIC S9(8)<br>COMP | For TSQ data, this field is set to zero or FALSE. For a Container name, this field is set to one or TRUE.  |
| <b>TDQW-TDQ-TYPE</b>  | PIC X(1)          | This field is the type of trigger data to which the data is to be sent. The valid values are G for Guaranteed, R for Reliable, o for Ordered, and s for TSQ.   |
| <b>TDQW-RETURN</b>    | PIC 9(8)<br>COMP  | This field is the return status from SXCQWRIT or SXCQWTSQ. A zero value means all is in order. A nonzero value means a problem exists.   |
| <b>TDQW-REASON</b>    | PIC 9(8)<br>COMP  | This field is the reason code from the error if the TDQW-RETURN value is nonzero. Otherwise, the value in the field is zero.   |
| <b>TDQW-RESP1</b>     | PIC 9(8) COMP     | This field is the CICS Response code (eibresp) from an error if TDQW-REASON depicts a CICS error. Otherwise, the value in the field is zero.   |
| <b>TDQW-RESP2</b>     | PIC 9(8)<br>COMP  | This field is the CICS Response2 code (eibresp) from an error if TDQW-REASON depicts a CICS error. Otherwise, the value in the field is zero.  |



| Field                   | Type     | Description  |
|-------------------------|----------|--|
| <b>TDQW-ROLLBACK-SW</b> | PIC X(1) | This field depicts if any successful TDQ writes have been completed. If the value is Y and an error has occurred, the user transaction must issue EXEC CICS SYNCPOINT ROLLBACK to remove the partial data from the Substation Trigger Queue. The value N means no data has been written to that queue. |
| <b>TDQW-TDQ-NAME</b>    | PIC X(4) | This is a reserved field that contains the CICS TDQ to which the transaction is allocated. On the first link to SXCQWRIT, the value in the field is LOW-VALUES (0x00). For all subsequent links to SXCQWRIT, leave the value as is.  |

All other areas are reserved.

## HVT Trigger Transactions

The following table shows the command-line parameters for sample HVT tasks.

### *SXTF and SXT6 Command-Line Parameters*

| Command  | Description   |
|--|---|
| <b>SXTF/SXT6</b>   | Sends one message through HVT in IVP format.  |
| <b>SXTF/SXT6,HELP-?</b>  | Lists all the keywords for the available commands.  |
| <b>SXTF/SXT6,type(G R O S),number_of_records</b>   | Sends multiple IVP trigger messages through HVT in <i>type</i> . The valid <i>type</i> values are G for Guaranteed, R for Reliable, O for Ordered, and S for Temporary Storage Trigger.   |
| <b>SXTF/SXT6,type(G R O S),number_of_records,data_length(1 - 4096000)</b>                  | Sends multiple messages through HVT in <i>type</i> and <i>data_length</i> . The valid <i>type</i> values are G for Guaranteed, R for Reliable, O for Ordered, and S for Temporary Storage Trigger.<br><br>The maximum data length for the TSQ Trigger is 32000.   |
| <b>SXTF/SXT6,type(G R O S),number_of_records,data_length(1 - 4096000),service_name</b>     | Sends multiple messages through HVT in <i>type</i> and <i>data_length</i> to <i>service_name</i> (the trigger identifier). The valid <i>type</i> values are G for Guaranteed, R for Reliable, O for Ordered, and S for Temporary Storage Trigger.<br><br>The maximum data length for the TSQ Trigger is 32000.  |
| <b>SXTF/SXT6,Type(G R O S),number_of_records,data_length(1 - 4096000),service_name,CTN</b> | Sends multiple messages through HVT in <i>type</i> and <i>data_length</i> to <i>service_name</i> (the trigger identifier), using CTN number of containers.<br><br>The valid <i>type</i> values are G for Guaranteed, R for Reliable, O for Ordered, and S for Temporary Storage Trigger.<br><br>The maximum data length for the TSQ Trigger is 32000. The valid range for CTN is 1 to 20. |

| Command  | Description   |
|--|---|
| <b>SXTF</b> , <i>Type</i> (G R O S), <i>number_of_records</i> , <i>data_length</i> (1 - 4096000), <i>service_name</i> , <i>CTN</i> , <i>delay_in_seconds</i>             | Sends multiple messages through HVT in <i>type</i> and <i>data_length</i> to <i>service_name</i> (the trigger identifier), using <i>CTN</i> number of containers, and with a delay of <i>delay</i> in seconds between calls to SXCQWRIT or SXCQWTSQ. The valid <i>type</i> values are G for Guaranteed, R for Reliable, O for Ordered, and S for Temporary Storage Trigger. |
| <b>SXT6</b> , <i>Type</i> (G R O S), <i>number_of_records</i> , <i>data_length</i> (1 - 4096000), <i>service_name</i> , <i>CTN</i> , <i>delay_in_seconds</i> .01, #Tasks | The maximum data length for the TSQ Trigger is 32000. The valid range for <i>CTN</i> is 1 to 20.  |

The messages sent are either the default IVP messages or, if *data\_length* is specified, a record of length *data\_length* filled with 1-----10-----20-----.

## CICS Trigger Queue Verify Task

SXTV reads HVT queues and verifies that the record headers are in proper sequence.

Any errors cause printed error messages on CICS CEELLOG in either of the following forms:

- "Queue: *queue name* Record: *number* - Length Invalid - Exp: *number* Got: *number*"
- "Queue: *queue name* Record: *number* - Number Invalid - Exp: *number* Got: *number*"



No data is changed or removed from the queues.

The following table describes the command-line parameters for SXTV.

### SXTV Command-line Parameters

| Command   | Description   |
|---|---|
| <b>SXTV</b> , <b>HELP</b> -?                    | Lists all the keywords for the available commands.  |
| <b>SXTV</b> , <i>queue-name</i> , <i>number</i> | Displays record <i>number</i> in TDQ for the specified <i>queue-name</i> . If <i>number</i> is not specified, the default to display is the first record.                     |
| <b>SXTV</b> , <b>TSQ</b> , <i>queue-name</i>    | Displays the first 336 characters of the first record in the TSQ entry for the specified queue, where <i>queue-name</i> is the 16-character queue.                            |
| <b>SXTV</b> , <b>SHOW</b> , <b>SSS</b>          | Displays the first 336 characters of the Substation ES anchor record.   |
| <b>SXTV</b> , <b>SHOW</b> , <b>HVT</b>          | Displays the first 336 characters of the Substation ES High Volume Trigger record.  |
| <b>SXTV</b> , <b>TESTHVT</b>                    | Tests all the HVT transient data queues for data integrity. The output message All queues tested - 0 errors found means that all HVT queues have been tested and have passed. |

## Sample Output for SXTV Commands

The section shows an output example of an **SXTV** command.

The following output example is for the **SXTV** command:

```
SXTV, SXQE, 2
Record Number: 2, Record size: 144(0090)
00000000 0000 00000008 00000BF4 00000000 00000000 | .....4.....|
00000016 0010 0A040000 00000032 49323BA9 95E05E35 | .....zn\;.|
```

```

00000032 0020 00000000 00000000 00000000 00000000 | .....|
00000048 0030 00000000 E2E7C3D6 E2D9E5C1 E2E7C3D6 | ....SXCOSRVASXCO|
00000064 0040 E2D9E5C1 00000000 00000000 00000000 | SRVA.....|
00000080 0050 00000000 00000000 00000000 00000000 | .....|
00000096 0060 00000000 00000000 00000000 00000000 | .....|
00000112 0070 00000000 00000000 00000000 00000000 | .....|
00000128 0080 00000000 00000000 00000000 00000000 | .....|

```

## CICS 3270 Assistance Task

Substation ES provides SXTH, a CICS 3270 task that performs the following items:

- Show the details of a Substation ES CICS interface queue usage.
- Display a detailed version of the Substation ES CICS error message record.
- Display a connected or previously connected Substation ES trigger and CICS-initiated request or reply process statistics.

## SXTH Command-line Parameters

The following table describes the command-line parameters for SXTH.

| Command  | Description  |
|--|--|
| <b>SXTH</b>  | Displays a statistics summary for the Substation ES CICS interface queue resources.  |
| <b>SXTH, ALL</b>   | Displays a statistics summary for the Substation ES CICS interface queue resources.  |
| <b>SXTH, DEF</b>   | Displays the default information of the Substation ES CICS interface queue resources.  |
| <b>SXTH, ERR</b>   | Displays the details on the first error record from the error queue.   |
| <b>SXTH, ERR, DELETE</b>   | Displays the details on the first error record from the error queue and then deletes the record from the queue.  |
| <b>SXTH, ERR, PURGE</b>  | Deletes all the records from the error queue.  |
| <b>SXTH, HELP-?</b>  | Lists all the keywords for the available commands.   |
| <b>SXTH, HVT</b>   | Displays the concurrent activity of the HVT queues.  |
| <b>SXTH, TSQ</b>   | Displays the concurrent activity of the HV TSQ triggers.   |
| <b>SXTH, PTDQ, TDQ</b><br><i>Name, start record, number of records</i> | Reads the TDQ records from the specified Transient Data Queue and produces a hex format printout of the queue record. Directs the output to CEEMSG of the CICS region. |
| <b>SXTH, PTSQ, TSQ</b><br><i>Name, start record, number of records</i> | Reads the TSQ records from the specified TSQ and produces a hex format printout of the queue record. Directs the output to CEEMSG of the CICS region.                  |
| <b>SXTH, QUE</b>   | Displays the usage information on the default Trigger Queue.   |
| <b>SXTH, QUE, TDQ</b><br><i>Name</i>                                   | Displays the queue usage information on the given TD queue name.   |

| Command          | Description   |
|------------------|---|
| <b>SXTH, TRG</b> | Displays the concurrent active statistics for the triggers and CICS initiated request or reply processes. |
| <b>SXTH, VER</b> | Displays the date and timestamp of the module compiled.   |

## Sample Output for SXTH Commands

The output examples of the **SXTH** commands are listed as follows:

- **SXTH command**

```
TDQueue Name: SXQD - Records:0, Triggerlevel:0 (Dead Msg)
TDQueue Name: SXQE - Records:0, Triggerlevel:1 (Error)
TDQueue Name: SXQG - Records:0, Triggerlevel:1 (Guaranteed)
TDQueue Name: SXQO - Records:0, Triggerlevel:1 (Ordered)
TDQueue Name: SXQT - Records:0, Triggerlevel:1 (Reliable)
TDQueue Name: SXQW - Records:0, Triggerlevel:0 (HVT Rel)
TDQueue Name: SXQX - Records:0, Triggerlevel:0 (HVT Rel)
TDQueue Name: SXQ$ - Records:0, Triggerlevel:0 (ReXmit Gur)
TDQueue Name: SXQ@ - Records:0, Triggerlevel:0 (ReXmit Rel)
TDQueue Name: SXQ0 - Records:0, Triggerlevel:0 (HVT Gur)
TDQueue Name: SXQ1 - Records:0, Triggerlevel:0 (HVT Gur)
TDQueue Name: SXQ2 - Records:0, Triggerlevel:0 (HVT Gur)
TDQ Server: SXCOSRVT - Res:0, Use:16, Status:ENABLED
```

- **SXTH, ERR command**

```
Substation ES CICS Error TD Queue display
SS Error Date & Time - 2004/12/01 -- 10:57:05.031
SS Internal Cmd:11 - CICS EIBFN value :X'E02'
SS Error RC :8 - SS Error Reason :3482
CICS dfhresp1 :27 - CICS dfhresp2 :2
Program Name :SXCOSRVD
CICS Resource :SXCREPLY - SS Resource :SXCREPLY
```

- **SXTH, TRG command**

```
SXTH, TRG
2 of 8 Intf allocated - 8 Active - Trace Level 0
Interface 1 - Status:Active, Id:QA111CEA-CICSE1
Trigger Que (Gur) - Status:Active, Busy:No, Que (Rel) - Status:Active, Busy:No
Trigger HV (Gur) - Status:Active, Ques:2
Trigger HV (Rel) - Status:Active, Ques:2
Trigger HV (Ord) - Status:Active, Busy:No
CICS Req/Reply - Status:Active, Busy:No
Interface 2 - Status:Active, Id:QA111CEA-CICSEA2
Trigger Que (Gur) - Status:Active, Busy:No, Que (Rel) - Status:Active, Busy:No
Trigger HV (Gur) - Status:Active, Ques:2
Trigger HV (Rel) - Status:Active, Ques:2
Trigger HV (Ord) - Status:Active, Busy:No
CICS Req/Reply - Status:Active, Busy:No
```

- **SXTH, HVT command**

```
SXTH, HVT
HVT Activity - Inact/Posted/Act Q-Name Offload
1-2-3-4-5-6-7-8
- Status:Active , Id: SXQ0 1-0-0-0-0-0-0-0
- Status:Inactive, Id: SXQ1 1-0-0-0-0-0-0-0
- Status:Inactive, Id: SXQU 0-0-0-0-0-0-0-0
- Status:Active , Id: SXQV 1-1-1-1-1-1-1-1
Trigger Server: SXCOSRVT - Res:7, Use:6993, Status:ENABLED
```

- **SXTH, TSQ command**

```
SXTH, TSQ
HV TSQ Triggers: - Records :0
- Next Write:1447
- Next Read :1447
- TSQ Count :0
- List Count:0
```

```
- List HWM :0
Trigger Server: SXCOSRVT - Res:0, Use:21685, Status:ENABLED
```

## CICS 3270 Dead Message Queue Task

Substation ES provides SXTD, a CICS 3270 task that performs certain functions as follows:

- Display a detailed version of the Substation ES CICS dead message record.
- Delete a record from the dead message queue.
- Resubmit a record group from the DMQ to Substation ES through SXCQWRIT.

If you resend the task, records only go to the HVT queues. You must set the TRIG-HVT flag to on for the task to process the records.

The source for this task is in *USERHLQ.COB(SXCCODMQ)*.

The following table describes the command-line parameters for the task SXTD.

### SXTD Command-line Parameters

| Command                   | Description   |
|---------------------------|---|
| <b>SXTD</b>               | Displays the first record on the DMQ.   |
| <b>SXTD,HELP-?</b>        | Lists all the keywords for the available commands.                                  |
| <b>SXTD,record number</b> | Displays the number of the DMQ record.  |
| <b>SXTD,DEL</b>           | Deletes the first record in the DMQ.  |
| <b>SXTD,PURGE</b>         | Deletes all the records in the DMQ.   |
| <b>SXTD,RESEND</b>        | Sends the first record group on the DMQ to SXCQWRIT for resending to Substation ES. |
| <b>SXTD,RESEND,ALL</b>    | Sends all the record groups on the DMQ to SXCQWRIT for resending to Substation ES.  |

## Sample Output for SXTD Commands

The output examples of the **SXTD** commands are listed as follows:

- **SXTD command**

```
Date / Time :2007/01/22 -- 11:33:30.887
Record Type :R - Reliable Record Occurance: 1
Record Length : 58500 Queue Record Length:18304 EOD:No
Return Code :00004 Reason Code:44101 No Definition for Record
0000 0000 F1606060 60606060 60F1F060 60606060 *1-----10-----*
0016 0010 606060F2 F0606060 60606060 60F3F060 *---20-----30---*
0032 0020 60606060 606060F4 F0606060 60606060 *-----40-----*
0048 0030 60F5F060 60606060 606060F6 F0606060 *50-----60---*
0064 0040 60606060 60F7F060 60606060 606060F8 *-----70-----8*
0080 0050 F0606060 60606060 60F9F060 60606060 *0-----90-----*
0096 0060 606060F1 F0F06060 60606060 60F1F1F0 *---100-----110*
0112 0070 60606060 606060F1 F2F06060 60606060 *-----120-----*
0128 0080 60F1F3F0 60606060 606060F1 F4F06060 *130-----140---*
0144 0090 60606060 60F1F5F0 60606060 606060F1 *-----150-----1*
0160 00A0 F6F06060 60606060 60F1F7F0 60606060 *60-----170---*
0176 00B0 606060F1 F8F06060 60606060 60F1F9F0 *---180-----190*
0192 00C0 60606060 606060F2 F0F06060 60606060 *-----200-----*
0208 00D0 60F2F1F0 60606060 606060F2 F2F06060 *-210-----220--*
```

```
0224 00E0 60606060 60F2F3F0 60606060 606060F2 *-----230-----2*
CLEAR/PF3: end PF7: prev PF8: next Current DateTime: (01/22/07 21:05:46)
```

- **SXTD,DEL command**

SXC3803I 1 Record Deleted from the Dead Queue

- **SXTD,PURGE command**

SXC3803I All Records Deleted from the Dead Queue

- **SXTD,RESEND, ALL command**

SXC3801I 7 Record(s) Resent from the Dead Queue to HVT

## CICS Security Settings

This section describes the security settings in CICS.

It includes the following topics:

- [CICS Transaction Security for Substation ES](#)
- [Definitions of Resource Access Control Facility](#)
- [Maximum EXCI Sessions](#)

## CICS Transaction Security for Substation ES

Several settings enable you to enforce security in the CICS region for a transaction being invoked from Substation ES.

### General Settings

Consider these two general settings:

- Change the Substation ES CICS SXCD Connection definition by setting the **Attachsec** parameter to **Identify**.
- Set the **SEC** and **XTRAN** parameters in the CICS SIT as follows:

```
SEC=YES
XTRAN=YES
```

### Specific Settings

Consider performing the following procedures for the specific settings:

#### Procedure

1. Copy the Substation ES CICS Resource transaction definition SXEX to SXEU.
2. In the Substation ES Transformer configuration panels:
  - Select the Define Message Contents panel and add the **tibss-userid** field. Define it as a field of type STR on the incoming application message.
  - Select the Define Conversion Relationships panels and set the **tibss-userid** field to Trunc=S and Req=Y.  
  
This field does not need to match `buffer-field` because **tibss-userid** is passed in a system header to back-end systems.
3. Select the Define Recipes panel and change the Substation ES configuration recipe definition that requires a security check. Set the panel option of Mirror Tran to SXEU.

4. Refresh the recipe with a Substation ES administrative command.

## Definitions of Resource Access Control Facility

This section describes procedures for setting up Substation ES security in the CICS environment.

The steps assume that a minimal CICS security environment already exists in Resource Access Control Facility (RACF). For additional information on setting up an initial CICS security environment, see the *CICS Transaction Server for z/OS RACF Security Guide* from IBM.

### Procedure

1. Optional: If you do not set up RACF for CICS security, you can set a minimal Substation ES for CICS security with the following RACF commands:

```
TSO SETROPTS GENERIC(TCICSTRN)
TSO RDEFINE TCICSTRN * UACC(READ)
TSO RDEFINE TCICSTRN SXEU UACC(NONE)
```



Setting minimal Substation ES security with the preceding RACF commands is not an operational recommendation by either TIBCO or IBM.

2. After setting security (or verifying that security exists in RACF), pick a user ID that is authorized to execute the most sensitive transactions.



The following steps assume that the user ID is MAXCICS and that the CICS STARTED procedure is CICSPROC.

3. Set up the STARTED class or its equivalent to run Substation ES under the user ID MAXCICS:

```
TSO RDEFINE STARTED (CICSPROC.*) STDATA(USER(MAXCICS) GROUP(CICSGRP)
PRIVILEGED(NO) TRUSTED(NO) )
```

4. Enter the following commands:

```
TSO PERMIT SXEU CLASS(TCICSTRN) ACCESS(READ) GENERIC ID(MAXCICS)
```

5. Activate the RACF class TCICSTRN:

```
TSO SETROPTS CLASSACT(TCICSTRN)
```

6. Optional: Define additional profiles with RDEFINE to limit the authority for specific transactions, as necessary.

For guidance on setting up RACF, see the *Security Server RACF Security Administrator's Guide*, *Security Server RACF Command Language Reference*, and *Transaction Server for z/OS CICS RACF Security Guide* from IBM.

## Maximum EXCI Sessions

The IBM default is a maximum of 100 EXCI sessions for each address space.

This limit applies to all CICS and Substation ES regions. This is a runtime limit and is checked by IBM at the EXCI Session Allocation time.



If Substation ES is interfacing with more than one CICS, the combined total number of EXCI sessions in that Substation ES must be less than 100.

A message similar to the following one is displayed in Substation ES when the limit is exceeded:

```
SXC3101E CICS EXCI Call Error,TRAN:{trandid},NetName:{EXCI Name},
          CICS IRC Code:56 X'38' Subreason2:-2147483548
          RC:8 RSN:3101N:3101
ID3: 0002 16          608 {CICS Name} {Program Name}
```

The IRC command return code 56 is a NO\_FREE\_LACBE\_FOR\_LOGON code, and indicates that there is "No free LACBE for Logon".

Code 608, one of the response and reason calls returned on the EXCI calls, is an IRC\_LOGON\_FAILURE code.

Together, these two codes indicate that the 100 pipe limit for each address space has been reached. Each CICS address space can have a maximum of 100 pipes connected, whether this is to one CICS or several CICS regions.

For more information, see the IBM documentation on CICS Transaction Server for z/OS, in particular the section on EXCI pipe allocation in Installation Guide.

## Operations Cookbook

This section describes the operations that you might perform and the related procedures.

As a refresher, the related terms are listed as follows:

- **AOR:** Application-Owning CICS Region.
- **SOR:** the CICS region to which Substation ES connects.
- **SXEX:** the standard mirror transaction ID under which Substation ES executes programs.

### Run a CICS transaction ID with other than SXEX.

Use the method MOI 2 (DPL) and program *yyyyyyyyyy* with transaction ID *xxxx*.

1. In the recipe, set Mirror Tran to *xxxx*, Program to *yyyyyyyyyy*, MOI to 2, and User Tran to Y.
2. Define ProgramName *yyyyyyyyyy* in the SOR region.

Program *yyyyyyyyyy* then runs in the SOR with transaction ID *xxxx*.

### Run a CICS program with SXEX on a different CICS region from the one to which Substation ES connects.

Use the method MOI 2 (DPL) and program *yyyyyyyyyy* with transaction ID SXEX.

1. In the recipe, set Trans to blank, Program to *yyyyyyyyyy*, MOI to 2, and User Tran to N.
2. Define ProgramName *yyyyyyyyyy* in the AOR.
3. Define the transaction SXEX in the AOR with the program DFHMIR.
4. Define ProgramName *yyyyyyyyyy* in the SOR with RemoteSystem *zzzz* and RemoteName *yyyyyyyyyy*.  
TRANSID is ignored in this situation.

Program *yyyyyyyyyy* then runs in the AOR with transaction ID SXEX.

### Run a CICS program with a transaction ID other than SXEX on a different CICS region from the one to which Substation ES connects.

Use the method MOI 9 (DPL MRO) and Program *yyyyyyyyyy* with transaction ID *xxxx*.

1. In the recipe, set Trans to *xxxx*, Program to *yyyyyyyyyy*, MOI to 9, and User Tran to Y.
2. Define ProgramName *yyyyyyyyyy* in the AOR.
3. Define ProgramName *yyyyyyyyyy* in the SOR with RemoteSystem *zzzz* and RemoteName *yyyyyyyyyy*.  
TRANSID is ignored in this situation.

Program *yyyyyyyyyy* then runs in the AOR with transaction ID *xxxx*.



**Run a CICS program with Containers and a transaction ID other than SXEX on a different CICS region from the one to which Substation ES connects.**

Use the method MOI 10 (DPL Container) and program *yyyyyyyyyy* with transaction ID *xxxx*.

1. In the recipe, set Trans to *xxxx*, Program to *yyyyyyyyyy*, MOI to 10, and User Tran to Y.
2. Define ProgramName *yyyyyyyyyy* in the AOR.
3. Define ProgramName *yyyyyyyyyy* in the SOR with RemoteSystem *zzzz* and RemoteName *yyyyyyyyyy*.  
TRANSID is ignored in this situation.

Program *yyyyyyyyyy* then runs in the AOR with transaction ID *xxxx*.

**Run a CICS program with Containers and transaction ID SXEX on a different CICS region from the one to which Substation ES connects.**

Use the method MOI 10 (DPL Container) and program *yyyyyyyyyy* with transaction ID SXEX.

1. In the recipe, set Mirror Tran to blank, Program to *yyyyyyyyyy*, MOI to 10, and User Tran to N.
2. Define ProgramName *yyyyyyyyyy* in the AOR.
3. Define ProgramName *yyyyyyyyyy* in the SOR with RemoteSystem *zzzz* and RemoteName *yyyyyyyyyy*.  
TRANSID is ignored in this situation.

Program *yyyyyyyyyy* then runs in the AOR with transaction ID SXEX.

**Run a CICS program with a Storage Pointer and a transaction ID other than SXEX on a different CICS region from the one to which Substation ES connects.**

Substation ES does not support this configuration because a storage pointer is passed. Consider wrapping the business application program with a new application program that receives and sends data with Containers.

**Run a CICS program with a Storage Pointer and transaction ID SXEX on a different CICS region from the one to which Substation ES connects.**

Substation ES does not support this configuration because a storage pointer is passed. Consider wrapping the business application program with a new application program that receives and sends data with Containers.

**Run a CICS program with TSQ and a transaction ID other than SXEX on a different CICS region from the one to which Substation ES connects.**

Use the method MOI 12 (DPL TSQ) and program *yyyyyyyyyy* with transaction ID *xxxx*.

1. In the recipe, set Trans to *xxxx*, Program to *yyyyyyyyyy*, MOI to 12, and User Tran to Y.
2. Define ProgramName *yyyyyyyyyy* in the AOR.
3. Define ProgramName *yyyyyyyyyy* in the SOR with RemoteSystem *zzzz* and RemoteName *yyyyyyyyyy*.  
TRANSID is ignored in this situation.

4. In the AOR, define TSQMODEL for SXCW+, sending the request to the SOR.

Program *yyyyyyyyyy* then runs in the AOR with transaction ID *xxxx*.

**Run a CICS program with TSQ and transaction ID SXEX on a different CICS region from the one to which Substation ES connects.**

Use the method MOI 12 (DPL TSQ) and program *yyyyyyyyyy* with transaction ID *xxxx*.

1. In the recipe, set Trans to blank, Program to *yyyyyyyy*, MOI to 12, and User Tran to N.
2. Define ProgramName *yyyyyyyy* in the AOR.
3. Define ProgramName *yyyyyyyy* in the SOR with Remote System *zzzz* and RemoteName *yyyyyyyy*.  
TRANSID is ignored in this situation.
4. In the AOR, define TSQMODEL for SXCW+, sending the request to the SOR.

Program *yyyyyyyy* then runs in the AOR with transaction ID SXEX.

# RED for CICS Operations

---

This section describes the functions and usage of RED functions in CICS.

It includes the following topics:

- [Overview of the Interface for RED](#)
- [RED Initialization for CICS](#)
- [CICS Transactions Using RED](#)
- [RED CICS Triggers](#)
- [RED CICS Request/Reply](#)
- [RED CICS Control Information Viewer](#)

## Overview of the Interface for RED

When communicating with CICS regions and applications within the CICS regions, TIBCO Substation Interface for RED provides three major functions that user processes can use.

These functions include the following items:

- Synchronous Trigger: directly publish messages to Substation with the following options:
  - No feedback (Fire and Forget).
  - Immediate feedback that it has reached its destination (Confirmation signal).
  - Feedback that it has reached its destination, and that the destination service has either persisted or processed the message or delivered the message to an adjacent execution service.

This is typically used for guaranteed delivery.
- Synchronous Message Transfer (Requests or Requests expecting a Reply): the ability to have a conversation with a Substation ES partner or external service routine connected to ESB.
- Multi-interface support: A CICS region can interface to one or two Substation ES RED interfaces - either in the same Substation ES or in two different Substation ES in the same or different LPARs.

## RED Initialization for CICS

During CICS initialization as a part of the PLT, program SXX3STRT is executed. The program sets up the Mainframe RED environment in a CICS address space.

The SXXFINIT program is issued from SXX3ADMN once for the entire CICS address space and remains in effect until CICS termination. Therefore, the parameters and settings established for this call must be compatible with all the transactions that use RED operations and functions. All subsequent transactions execute using these initial SXXFINIT values, most of which can be overwritten by specific application usages. There is only one SXXFINIT and only one SXXFTERM call allowed for the entire CICS address space.

As a part of the Mainframe RED initialization in CICS, the following procedures occur:

1. Program SXXACRM is loaded and made resident.
2. Program SXXACNFG is loaded to determine Initialization Parameters and then released.
3. Determine that a PCM is available for the LPAR in which the CICS region executes as specified in SXXACNFG.
4. Enable and start a thin layered Resource Manager using Program SXXACRM.
5. Issue a SXXFINIT call using the values present in SXXACNFG, and list the values on CICS JESLOG.

When the initialization runs correctly, the following items are listed on CICS JESLOG:

```
SXX7700I RED for CICS - Starting Initialization
SXX7701I RED for CICS - Initialized
XX8901I Mainframe RED Started 2016/04/11-11:34:16:734
XX8911I Mainframe RED - V2.3.0.36 GA 04-10
SXX7771I RED Monitor at - URL:TIBMON.<cics applname>
SXX7705I RED Services - 2.3.0.30 04-11 Stg - Started
SXX3700I Mainframe RED CICS Initialization complete
```

6. The TCBs required for Mainframe RED internals are started in the CICS address space.
7. As a part of SXX3STRT, specific Mainframe RED joins and connects are issued. These are used to handle SXCOTRED and SXXRTXN calls. All transactions linking to SXCOTRED use these JOIN and/or Connect definitions.
8. Also part of SXX3STRT, a "shared" CICS GETMAIN is issued as an anchor control block with a z/OS token so it can be found by other transactions.

Besides issuing SXXFINIT, SXX3ADMN also sets up Resource Manager (TRUE) (the SXXACRM module ) for handling CICS RED API requests from user applications. These API requests are listed as follows:

- SXXMJOIN and SXXMLEAV
- SXXMRECM, SXXMRECP and SXXMFREE
- SXXMSEND, SXXMSNDC and SXXMSNDR



CICS transactions can never issue SXXFINIT, or SXXFTERM.

At CICS shutdown, SXX3COUT is executed as part of the shutdown PLT. This program terminates the CICS RED environment and removes the Resource Manager program. It also frees up the shared storage and z/OS token.

Functions of SXXACRM (TRUE) are to handle the following items:

- Mainframe RED requests (for example, SXXMSEND, and SXXMRECM).
- Mainframe RED SXXMJOIN and SXXLCONN and ensure that at transaction termination that the JOIN or CONNECT is terminated.
- Commit processing.
- Abend processing (Rollback).

## CICS Transactions Using RED

This section describes the way the Substation ES interfaces with a CICS using the RED environment to run programs and functions in a CICS address space.



The CICS address space and the Substation ES address space do not have to be on the same LPAR.

More than one Substation ES RED interface can communicate with a CICS address space. The Substation ES RED interfaces can be in the same or different Substation ES address spaces on the same or different LPARs.

Listener transactions in CICS, run by default under the #RED transaction and might execute concurrently. The number of concurrent #RED transactions is controlled by a Substation ES start-up parameter in the RED interface called **CICS-MAIN-MAX**. The value for this keyword dictates the maximum number of #RED transactions that are started.

As messages queue in the RED that might create a backlog, additional #RED transactions are started up to the Substation ES configuration maximum "CMMSG-WORKERS" value. Once the backlog of messages in the RED is reduced, #RED transactions terminate and only two #RED transactions remain active.



There must always be one #RED transaction for the system to function correctly.

Substation ES Interface for RED connects to CICS using EXCI and starts a RED listener for messages from Substation ES to perform appropriate recipe functions. The #RED transaction and program SXCO#RED run on a L8 TCB, so you must have sufficient L8 TCB resources to run the requests from Substation ES.

The functions performed by Substation ES recipes are listed as follows:

1. Method of Invocation 2 (DPL) and Method of Invocation 9 (DPL MRO)

SXCO#RED sets up a CICS CommArea and issues **LINK** to the Recipe Resource program. Because SXCO#RED is doing a link, the program runs under the Userid (and security) that started #RED. (Most often, this is Substation ES by way of the program SXCOREDADA).

The values for Remote Region Execution are used so that the link can be a MRO function to another CICS Region. Also, if Mirror Transaction is set, SXCO#RED changes the eibtrnid value to match. However, the original transaction ID is #RED. Because #RED is not terminating the transaction, each LINK has the same CICS task ID. SXCO#RED does a "CICS SYNCPOINT COMMIT or ROLLBACK" after each LINK to a user program. Also, SXCO#RED monitors memory being held by the task. When that memory goes over 100,000 bytes, SXCO#RED recycles itself to reduce memory issues

2. Method of Invocation 4 (Task)

SXCO#RED sets up a CICS CommArea and issues **START** to the Recipe Resource transaction ID. **START** uses the Userid and Remote Region Execution System Name if specified on the recipe. SXCO#RED does not check CICS max task or SOS, so you must be careful not to flood a CICS region. It would be advisable to set a transaction class up for transactions that would cause a very high concurrency usage.

3. Method of Invocation 10 (Container)

Both Single and Multi-Container can be used for both **LINK** and **START**. **LINK** uses the values for Remote Region Execution if they are present. **START** uses USERID and Remote Region Execution System Name if they are specified on the recipe.

All Methods of Reply are supported:

- 0 for no reply
  - 1 for status only
  - 2 for reply message
  - 3 for reply status and message
- 
- For START Method of Invocation, the response is from #RED (not the application), Business Responses from a "START" MOI must be done through the RED TRIGGER processing.
  - All RED transactions run in reliable mode. When LINK or START is completed, a SYNCHPOINT is issued.



There will be a rollback issued:



- If an error occurs on send response back to Substation ES.
- If for a guaranteed recipe, Substation ES can publish the response.

## Setting Up a Recipe for CICS RED

You can set up a recipe to use CICS RED.

### Procedure

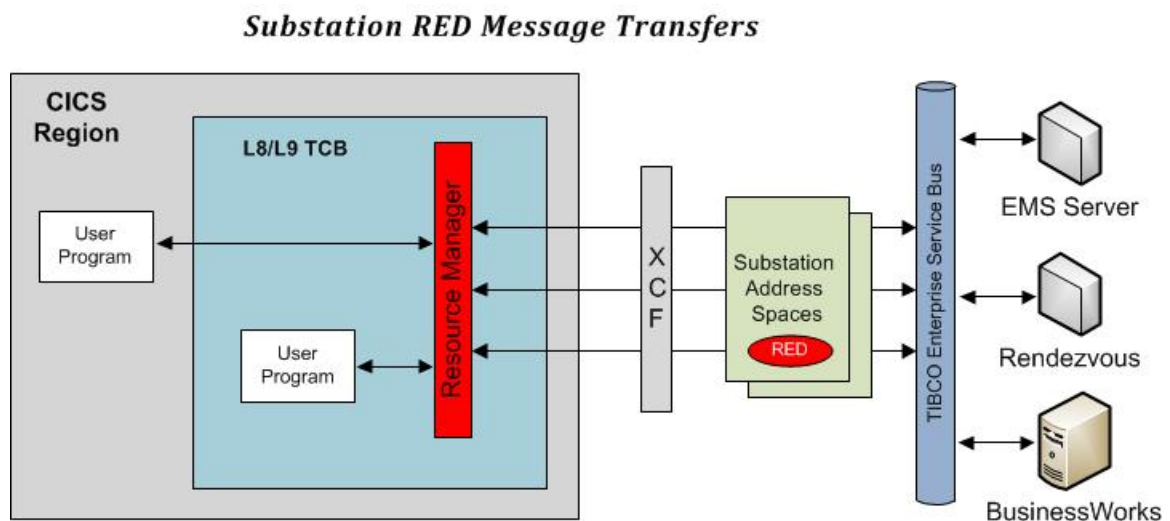
1. Set up a RED interface.
2. Set up a BES which matches your RED interface ID.
3. Create a recipe for MOI 2, 4, 9 or 10 that uses the BES ID for the RED interface.

## Detailed Flow of CICS Transaction

The following describes the flow of the CICS RED transaction.

1. Substation ES receives an ESB message.
2. Substation ES formats the CICS input area that the program or transaction is expecting.
3. Substation ES sends the input area to CICS RED.
4. CICS RED uses input area and Substation ES request to execute the function.
5. CICS RED replies using user response data to Substation ES for reply method of 1, 2, or 3.
6. Substation ES formats the ESB message from CICS RED response data.
7. Substation ES sends the ESB message.

The following figure lists the detailed flow of CICS transaction:



## RED CICS Triggers

CICS RED provides a server module that a user application can link to the called **SXCOTRED** which provides an immediate synchronous method.

**SXCOTRED** is a reentrant THREADSAFE and OPENAPI, and accessible by multiple concurrent transactions without locking issues. **SXCOTRED** runs on an L8/L9 TCB and receives parameters from the user application having to do with the following items:

- Message location
- Message size
- RED Trigger service definition in Substation ES on how to handle message feedback

The flow is as follows:

1. User application sets up message data in the following items:
  - a. A Container (Single or Multiple)
  - b. A storage location
  - c. CICS DFHCOMMAREA
2. User application sets up the **SXCOTRED** parameter information.
  - a. The Services ID to use (Substation ES RED Trigger definition)
  - b. Type of data (Container, Storage Location or DFHCOMMAREA)
  - c. Type of Response (Container, TSQ, CommArea, Data location)
  - d. Timeout value
3. User application does CICS LINK to SXCOTRED.
4. Two methods of SSES selection are available; Primary/Alternative and Round-Robin.
5. SXCOTRED sends the message immediately to Substation ES. If no Substation ES is available, an error is immediately generated and returned to caller.
  - a. The Substation ES accepts the message, and then sends it to its destination based on the Service definition it has.
    - "RELIABLE" messages are considered delivered when a Substation ES accepts the message. This type of message is the most efficient but may get dropped if there is a problem delivering the message to the end destination.
    - "GUARANTEED" messages are considered delivered when a Substation ES delivers the message to its defined destination. Anything other than a confirmed delivery to its defined destination results in an immediate error which is returned to the calling program.
    - "ORDERED" is the same as "GUARANTEED".
  - b. If Substation ES fails, **SXCOTRED** returns an error condition to the caller.
6. User application reviews a return code and a reason code for any error indication.
7. If no errors, the response data is present as indicated by the parameters.
8. The user application can link to **SXCOTRED** multiple times under the same Unit-Of-Work which forwards the message to the Substation ES. There is no special processing for a user application that abends, or issues a ROLLBACK. Once the call to **SXCOTRED** is done, the RED message process is committed and there is no back out process.
9. When the user application ends, a last link to **SXCOTRED** is made to terminate the connection with the Substation ES.

This method is useful for Processes that need immediate data publishing of many messages to the ESB environment such as Credit Card updates, EDI processes, and so on.



For more details of how to configure trigger processes with RED interface you can refer to the TIBCO Substation ES™ Configuration and Resources guide under the section Using Trigger Processing with TIBCO RED.

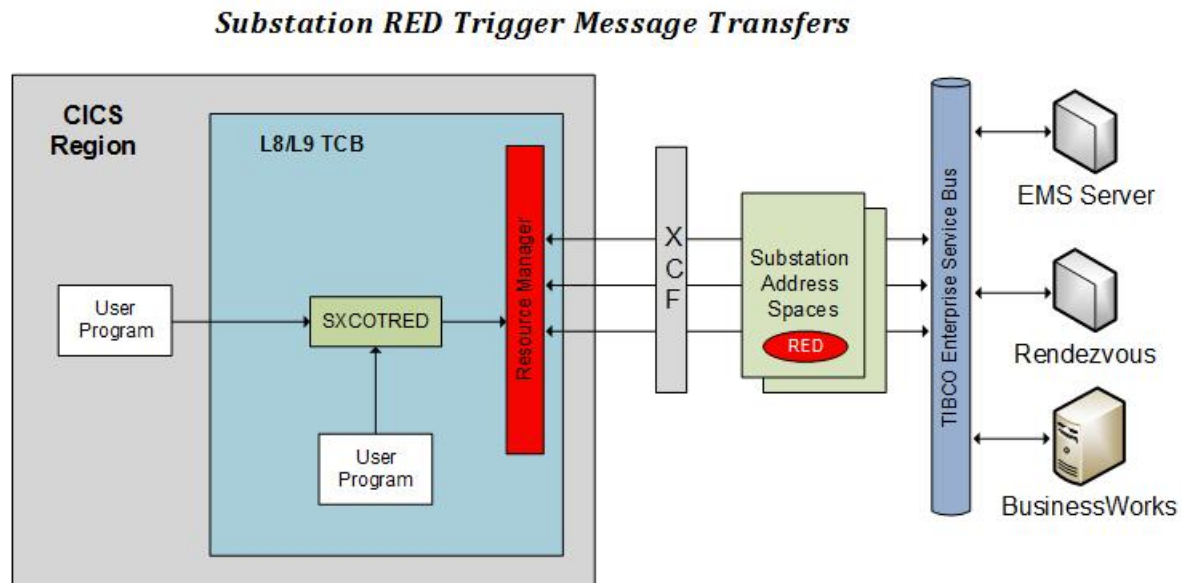
## Trigger Message Flows in CICS

When **SXCOTRED** receives control, it does a series of steps.

1. Gets parameters from either CommArea or Container.
2. Analyses parameters for validity.
3. Checks for active RED interface in CICS.

4. Gets access to the user data based on input passed.
5. Two methods of SSES selection are available; Primary/Alternative and Round-Robin.
6. Sends the RED message to Substation ES using Send, Send with Confirm, or Send with Reply indicating the RED Service to use.
  - All Substation ES (could be one or many on same or different LPARs) monitoring **SXCOTRED** requests validate Service availability.
  - **SXCOTRED** transfers the message to Substation ES.
  - **SXCOTRED** waits for response (if required) with a timeout.
  - Substation ES handles the message and forward it to its destination, and then gets reply and returns it to the waiting **SXCOTRED** program.
7. **SXCOTRED** returns the results of the call in the parameter control block and response message as indicated.

The following figure shows the trigger message transfer in CICS:



## RED CICS Request/Reply

CICS RED provides a server module that a user application can link to the called **SXXREQR** which provides an immediate synchronous method.

**SXXREQR** is a reentrant safe thread and an open API, and accessible by multiple transactions concurrent without locking issues.

**SXXREQR** runs on L8/L9 TCBs and receives parameters from the user application having to do with the following items:

- Message location
- Message size
- Trigger Service to use
- RED Trigger service definition in Substation ES on how to handle message feedback

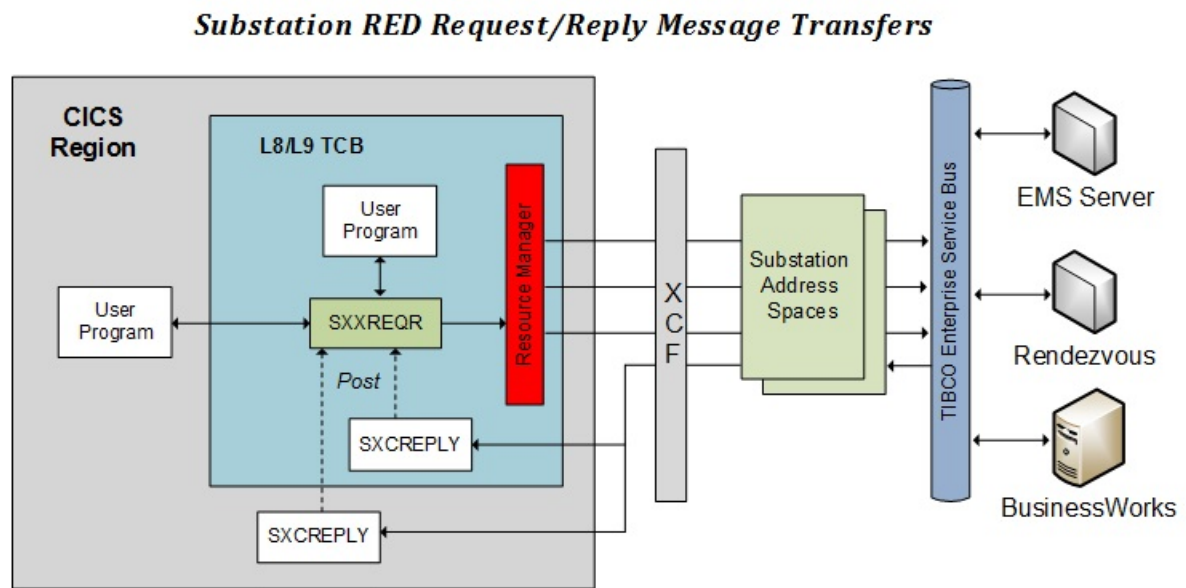
The flow is as follows:

1. User application sets up message data in the following items:



- a. A Container (Single)
  - b. A storage location (it cannot be used for MRO configuration)
  - c. CICS DFHCOMMAREA (limited to 32K)
2. User application sets up the **SXXREQR** parameter information.
  - a. The Services ID to use (Substation ES RED Trigger definition)
  - b. Type of data (Container, Storage Location or DFHCOMMAREA)
  - c. Type of Response (Container, TSQ, CommArea, Data location)
  - d. Timeout value
3. User application does CICS LINK to **SXXREQR**.
4. **SXXREQR** sends the message immediately to Substation ES. If no Substation ES is available, an error is immediately generated and returned to the caller.
  - a. The Substation ES accepts the message, and then sends it to its destination based on the Service definition it has. The message contains a TOKEN (tibss-envelope) that must be added to the response message. The response message along with the token are sent back to Substation ES on the "reply-to" destination.
  - b. If Substation ES fails, **SXXREQR** returns an error condition to the caller.
5. User application reviews a return code and a reason code for any error indication.
6. On normal return (for example, no errors and zero return code), user application uses response data as indicated from Container, TSQ, CommArea, or Data location.

The following figure shows how the Substation RED synchronous transfers:



### Request/Reply Message Flows in CICS

When **SXXREQR** receives control, it does a series of steps.

#### Procedure

1. Get parameters from either CommArea or Container.

2. Analyze parameters for validity.
3. Check for active RED interface in CICS.
4. Get access to the user data based on input passed.
5. Two methods of SSES selection are available; Primary/Alternative and Round-Robin.
6. Send the RED message to Substation ES using one of three functions (Send, Send with Confirm, or Send with Reply) indicating the RED Service to use.
  - All Substation ES (could be one or many on same or different LPARs) monitoring **SXXREQR** requests validate Service availability.
  - **SXXREQR** transfers the message to Substation ES.
  - **SXXREQR** waits for response (if required) with a timeout. The timeout value is 20 seconds by default.
  - Substation ES handles the message and forwards it to its destination, and then gets a reply and returns it to the waiting **SXXREQR** program through a Recipe which uses program **SXCREPLY**.
7. **SXXREQR** returns the results of the call in the parameter control block and response message as indicated.

## RED CICS Control Information Viewer

Substation ES provides XTUV, a CICS 3270 task that performs the following items:


- Shows the details of a Substation ES RED interface token in CICS.
- Displays the details of the Substation ES RED interface main storage in CICS.
- Displays the CICS RED trigger or recipe services information in CICS.
- Displays a specific address for storage information.

## XTUV Command-line Parameters

The following table describes the command-line parameters for XTUV.

### *XTUV Command-Line Parameters*

| Command                                | Description  |
|--|--|
| <b>XTUV</b>                            | Displays real-time CICS RED counters and other information.                    |
| <b>XTUV,command,PRINT</b>              | Displays the details on any command and prints the output to CEEOUT.           |
| <b>XTUV , DUMP ,hex-address,length</b> | Displays the details of a specific address storage for diagnostic information. |
| <b>XTUV , DUMP , CICS</b>              | Displays the TSQ main storage.   |
| <b>XTUV , DUMP , RED</b>               | Displays RED main storage  |
| <b>XTUV , DUMP , REC</b>               | Displays RED Receiver Storage.   |
| <b>XTUV , DUMP , TRG</b>               | Displays RED Trigger storage.  |

| Command                       | Description  |
|-------------------------------|--|
| <b>XTUV,ERR,#</b>             | Displays the details of an error record.<br> The hash sign (#) is the record number of an error in the error queue. |
| <b>XTUV,ERR,DELETE</b>        | Displays the details of the first error record from the error queue and then deletes the record from the queue.  |
| <b>XTUV,ERR,PURGE</b>         | Deletes all the records from the error queue.  |
| <b>XTUV,HELP ?</b>            | Lists all the keywords for the available commands.   |
| <b>XTUV,REAL RT</b>           | Displays real-time CICS RED counters and other information.  |
| <b>XTUV,REC</b>               | Displays RED CICS receiver storage information.  |
| <b>XTUV,RED</b>               | Displays the RED main storage.   |
| <b>XTUV,STATS,&lt;url&gt;</b> | Displays the RED statistics for CICS Listener.   |
| <b>XTUV,TRG[,NEXT][, #]</b>   | Displays all the pools and members established by the Substation ES RED interface. Optional operand NEXT page or NEXT,# to next # of interface. Where # is the number 1 to 6 interface.              |

### Sample Output for XTUV,TRG Command

The following is an example of the output generated by the **XTUV,TRG** command:

```
XTUV TRG,NEXT 4
```

```
CICS:C512WW3G, Location 2021B000, Size:15296
Max Len:60000, Rep-TO:5000, XMC-TO:2, Mode:R
Intf:REDEA1, Pri:1, Active, NoStress(0), NoThrottle(0), SSID:QA111RE1
Pool 1 Max Size:1000 ThdCnt:0 URL:QA111RE1.REDEA1#P1.P1AMEA1
Pacing:0 Retry:0 Lost:0 Sent:0 Error:0 Join:0
Pool 2 Max Size:8192 ThdCnt:0 URL:QA111RE1.REDEA1#P2.P2AMEA1
Pacing:0 Retry:0 Lost:0 Sent:0 Error:0 Join:0
Intf:REDEA2, Pri:1, Active, NoStress(0), NoThrottle(0), SSID:QA111RE1
Pool 1 Max Size:1000 ThdCnt:0 URL:QA111RE1.REDEA2#P1.P1AMEA2
Pacing:0 Retry:0 Lost:0 Sent:0 Error:0 Join:0
Pool 2 Max Size:8192 ThdCnt:0 URL:QA111RE1.REDEA2#P2.P2AMEA2
Pacing:0 Retry:0 Lost:0 Sent:0 Error:0 Join:0
Intf:REDEA3, Pri:1, Active, NoStress(0), NoThrottle(0), SSID:QA111RE1
Pool 1 Max Size:1000 ThdCnt:0 URL:QA111RE1.REDEA3#P1.P1AMEA3
Pacing:0 Retry:0 Lost:0 Sent:0 Error:0 Join:0
Pool 2 Max Size:8192 ThdCnt:0 URL:QA111RE1.REDEA3#P2.P2AMEA3
Pacing:0 Retry:0 Lost:0 Sent:0 Error:0 Join:0
```

More... (Press Enter)

### Field Descriptions for Output of XTUV,TRG

| Field           | Description   |
|-----------------|---|
| <b>CICS</b>     | The current CICS region applid.                       |
| <b>Location</b> | Memory address of the RED CICS Trigger control block. |

| Field                    | Description   |
|--------------------------|---|
| <b>Size</b>              | The size of the memory area for the location mentioned.   |
| <b>Max Len</b>           | The value shows the <b>AMSG-MAXLEN</b> value that the RED interface uses as the maximum message length. If the application program attempts to send data of longer length through <b>SXCOTRED</b> , it fails.   |
| <b>Rep-TO</b>            | The wait time in milliseconds that the guaranteed Trigger process waits for the response from Substation ES.  |
| <b>XMC-TO</b>            | The maximum time in seconds that an XCF send trigger message operation is given to complete successfully.   |
| <b>Mode</b>              | R for Round-Robin; F for Primary-Alternate.   |
| <b>Intf</b>              | RED interface name, display up to total of eight RED interfaces.  |
| <b>Pri</b>               | The value shows that the parameter of <b>AMSG-PRIORITY</b> been set for the RED interface connection priority to the CICS region.   |
| <b>Active / InActive</b> | Status of interface.  |
| <b>NoStress</b>          | The number of XCF interfaces that have been under stress condition.   |
| <b>NoThrottle</b>        | States whether we had a throttle and what the count of throttle has been.   |
| <b>SSID</b>              | Substation ES ID followed by Substation ES RED interface ID.  |
| <b>Pool</b>              | The related information of buffer pool #, where # can be 1, 2, or 3.  |
| <b>Max Size</b>          | The maximum message length that is allowed to be sent within a selected pool. If a message is larger than a specified value in the <b>Max Size</b> field but smaller or equal to the value in the field, RED handles the message differently, resulting in processing overhead. |
| <b>ThdCnt</b>            | The number of concurrent message worker threads that the RED interface started to service trigger messages.   |
| <b>URL</b>               | The target member name that the RED interface uses to publish trigger messages.   |
| <b>Pacing</b>            | The number of microseconds that the RED interface trigger request waits after sending a trigger message from the buffer pool, before returning to the CICS application.   |
| <b>Retry</b>             | The number of times <b>SXCOTRED</b> attempts to resend a reliable and guaranteed trigger message if the first send fails.   |
| <b>Lost</b>              | The total number of reliable confirm and guaranteed trigger messages that have been resent after receiving a timeout error (with RED reason code 8110).   |

| Field        | Description  |
|--------------|--|
| <b>Sent</b>  | The total number of trigger messages that have been sent. This count includes any resend messages following a failure. |
| <b>Error</b> | The total number of trigger messages that were not sent because of an error.   |
| <b>Join</b>  | The total number of RED API JOIN calls that have been done.  |

### Sample Output for XTUV,REAL Command

The following is an example of the output generated by the **XTUV,REAL** command:

#### XTUV,REAL

```
11:10:51.384
Receivers Active      : 1 <== #RED not running due to Abend Limit
Receivers Busy       : 1
Receivers started    : 1
Messages In Queue    : 0
High Water Mark (HWM) : 0
Message Receives     : 13
Message Replies      : 0
Message Errors       : 13
Abends (count, limit) : 7 of 5 <== Manual Intervention Required
Trigger Messages Sent : 11
```

#### Field Descriptions for output of XTUV,REAL

| Field                        | Description   |
|------------------------------|---|
| <b>Receivers Active</b>      | The current number #RED receiver transactions that are active. #RED receivers process messages destined for CICS and can run concurrently. The maximum number of #RED receivers can be controlled by the parameter <b>CICS-MAIN-MAX</b> in the RED interface. |
| <b>Receivers Busy</b>        | Total number of #RED receiver transactions that have been in busy mode.   |
| <b>Receivers started</b>     | Total number of CICS Start transactions issued to execute concurrent receivers.   |
| <b>Messages In Queue</b>     | Current number of messages waiting in the RED pool that still need to be processed.   |
| <b>High Water Mark (HWM)</b> | Highest count of messages that were waiting in the RED pool to be processed.  |
| <b>Messages Receives</b>     | The total number of messages received.  |
| <b>Messages Replies</b>      | The total number of replies sent if it is required.   |
| <b>Messages Errors</b>       | The total number of messages in error.  |

| Field                        | Description  |
|------------------------------|--|
| <b>Abends (count, limit)</b> | The total number of #RED transactions or user programs that abended. The maximum limit of abend allowable #RED restarts. The initial limit value is 5. Disconnect and connect your RED interface to have this abend count value reset. |
| <b>Trigger Messages Sent</b> | The total number of trigger messages sent.   |

## Sample Output for XTUV,STATS Command

The following is an example of the output generated by the **XTUV,STATS** command:

XTUV STATS

```

LPAR:LION Job:CICSPRO Member:TIBSS.MSGS#YODA
XCF Status      :Active    RED Status      :Active    Buffer OvrFlow  :0
Msgs Discarded  :0         Msgs Expired   :0         Msgs Held      :0

Buffer Size      :8460     Msgs > Buff Sz :0
Msgs Received    :156     Largest Size(R):1020   Max Msg Occ     :0
Msgs in Pool     :0       Buf Size Ovr   :0       HWM Buffers    :1
Msg Reply        :78      Msg Reply(SNC) :0       Msg SNC NACK   :0
Req/Reply        :0       Req/Reply Fail :0       Msg RC4        :0
SNC Sig(S)       :78      SNC Sig(R)     :0
Mux Executes     :156     Mux Abend      :0       Mux Msg Lost   :0

```

### Field Descriptions for output of XTUV,STATS

| Field           | Description   |
|-----------------|---|
| LPAR            | The name of the Mainframe HUB being used for this URL registration.                     |
| Job             | Name of the job creating the RED URL member.  |
| Member          | RED url target node.  |
| XCF Status      | How XCF sees this entry.  |
| RED Status      | How RED sees this entry.  |
| Buffer OvrFlow  | Number of times a message arrived and triggered the throttle event.                     |
| Msgs Discarded  | Number of messages that were thrown away for error situations.                          |
| Msgs Expired    | Number of messages that were removed from the buffer pool due to a timestamp expiration |
| Msgs Held       | Number of messages with locks in the buffer pool.                                       |
| Buffer Size     | Size of each buffer set on the "JOIN".  |
| Msgs > Buff Sz  | Number of messages received that were greater than buffer size but less than Max size.  |
| Msgs Received   | Number of messages received for this buffer pool url.                                   |
| Largest Size(R) | Size of largest message received.   |

| Field          | Description  |
|----------------|--|
| Max Msg Occ    | Number of times max Msg size received.   |
| Msgs in Pool   | Number of messages in the buffer pool (whether given to a receive api or not). |
| Buf Size Ovr   | Number of messages received that were over buffer size.                        |
| HWM Buffers    | High water Mark of messages in the buffer pool.                                |
| Msg Reply      | Number of messages received that were replies to a SEND-R api.                 |
| Msg Reply(SNC) | Number of messages received that were confirmation events to a SEND-C.         |
| Msg SNC NAck   | Number of messages received that were non-confirmation events to a SEND-C.     |
| Req/Reply      | Number of messages sent (Req/Reply).   |
| Req/Reply Fail | Number of R/R Msgs not getting reply.  |
| Msg RC4        | Number of times XCF returned a return code of 4.                               |
| SNC Sig(S)     | SNC confirmation signals sent.   |
| SNC Sig(R)     | SNC confirmation signals received.   |
| Mux Executes   | Message User EXIT execution count.   |
| Mux Abend      | Message User EXIT Abend count.   |
| Mux Msg Lost   | Message User EXIT Abend Lost Msgs.   |

# IMS Operations

---

You can learn about the Substation ES capabilities from this section.

It includes the following topics:

- [IMS Security Settings](#)
- [IMS Transactions](#)
- [IMS Events or Triggers](#)
- [Exceptional Cases](#)

## IMS Security Settings

IMS can provide optional RACF security-checking capabilities in OTMA environments.

The optional RACF security-checking capabilities are listed as follows:

- **Client bids:** determining whether OTMA clients can connect to IMS for the purpose of sending end-user messages to IMS for processing.
- **IMS commands:** entered by end-users attached to OTMA clients.
- **IMS transactions:** entered by end-users attached to OTMA clients.
- **Asynchronous hold queues:** which verifies issuers of RESUME TPIPE, including Substation ES. (Messages are placed in the asynchronous queues by triggers.)

The OTMA security level for an IMS system determines whether IMS calls RACF to perform authorization checking for the above activities. However, regardless of the OTMA security level, IMS always invokes certain security exits, if they exist, including the Command Authorization Exit and the Security Reverification Exit.

IMS invokes the Transaction Authorization Exit according to two factors:

- The OTMA security level for IMS. If the Transaction Authorization Exit is in IMS, IMS always invokes it if the OTMA security level is NONE.
- Whether RACF is invoked to process transaction authorization and the RACF resulting return code. Other OTMA security levels, namely, CHECK and FULL, result in IMS invoking RACF for transaction authorization processing.

If the Transaction Authorization Exit is in IMS, IMS invokes it if and only if RACF does not deny authorization. That is, if RACF grants a user-ID authorization to a transaction or if the transaction is not secured by RACF, IMS invokes the exit routine. Otherwise, IMS does not invoke that routine.

## Setup of OTMA Security Level

The OTMA security levels include NONE, CHECK, FULL, and PROFILE. Decide which one best meets your installation requirements.

You establish the OTMA security level for IMS with the IMS startup parameter OTMASE= in the DFSPBxxx member of the IMS procedure library or with the /SECURE OTMA command. By default, the OTMA security level in IMS is FULL (or F). To override the default, specify a different value for OTMASE. The valid values for the OTMASE= parameter are as follows:

- N (None)
- C (CHECK)
- F (FULL, the default value)
- P (PROFILE)



Alternatively, after the IMS startup, you can specify or change the OTMA security level by issuing **NONE**, **CHECK**, **FULL**, or **PROFILE** on the **/SECURE OTMA** command.

The **OTMASE=** startup parameter setting and the **/SECURE OTMA** command do the same thing. Each establishes the OTMA security level for IMS. The **/SECURE OTMA** command enables you to override the OTMA security level set by the **OTMASE=** parameter during IMS initialization. With the **/SECURE OTMA** command, you can change the OTMA security level without reinitializing IMS.

Although the **/SECURE OTMA** command overrides the **OTMASE=** value, the OTMA security level specified with the **/SECURE OTMA** command is not maintained across an IMS restart. When IMS restarts, the OTMA security level is established by either the value of the **OTMASE=** keyword or its default, **OTMASE=F**, which applies under either of these conditions:

- The **OTMASE=** keyword is not coded in startup parameters.
- The **OTMASE=** keyword is specified in the startup parameters without a value.

The following table shows you how to set up each of the OTMA security levels.

*Startup Parameters and OMS Commands for OTMA Security Levels*

| Security Level | Startup Parameter                              | IMS Command   |
|----------------|--|---|
| NONE           | <b>OTMASE=N</b>                                | <b>/SEC OTMA NONE</b>   |
|                | <b>OTMASE=P</b> and the security flag value N. | <b>/SEC OTMA PROFILE</b><br>In the profile, the security flag value is N. |
| CHECK          | <b>OTMASE=C</b>                                | <b>/SEC OTMA CHECK</b>  |
|                | <b>OTMASE=P</b> and the security flag value C. | <b>/SEC OTMA PROFILE</b><br>In the profile, the security flag value is C. |
| FULL           | <b>OTMASE=C</b>                                | <b>/SEC OTMA FULL</b>   |
|                | <b>OTMASE=P</b> and the security flag value F  | <b>/SEC OTMA PROFILE</b><br>In the profile, the security flag value is F. |
| PROFILE        | <b>OTMASE=P</b>                                | <b>/SEC OTMA PROFILE</b>  |



If **PROFILE** or **P** is set, the security flag value in each message received through OTMA is checked to determine whether the level **NONE**, **CHECK**, or **FULL** must apply to that message.

### **OTMASE=N or /SECURE OTMA NONE**

If the OTMA security level is **NONE**, RACF is not invoked by IMS. **OTMASE=N** and **/SECURE OTMA NONE** establish an IMS-wide security level, that is, IMS takes the same action for each message received by OTMA.

For the OTMA security level **NONE**, that means that IMS does not invoke RACF for the following tasks:

- Client-bid security checking for client-bid messages received.
- IMS command authorization for command messages received via OTMA.
- IMS transaction authorization for initial input messages received via OTMA.

## OTMASE=C or /SECURE OTMA CHECK

If the OTMA security level is CHECK, RACF is invoked by IMS and OTMA. Like the OTMA security level NONE, CHECK (or C) is also an IMS-wide security level, that is, IMS takes the same action for each message received by OTMA.

For the OTMA security level CHECK, IMS invokes RACF for the following tasks:

- Client-bid security checking for client-bid messages.
- User ID validation and ACEE creation for OTMA client applications and inducer IDs.
- IMS command authorization for command messages received via OTMA.
- IMS transaction authorization for transaction input messages received via OTMA.
- Authorization checking for subsequent IMS resources, such as transactions, databases, segments, fields, or other resources, that are requested during source-transaction processing, when the application issues a CHNG call and an AUTH call and performs a deferred conversational program-to-program message switch.

## Defining RACF for IMS Security

If RACF is set up for IMS transactions and commands, you must establish a minimal setup.

One symptom of missing RACF definitions is the appearance of messages such as the following:

```
DFS3187W RACF NOT ACTIVE FOR RESUME TPIPE CLASS=RIMS RC=04. RACF
EXIT RC=04 REASON CODE=00. IMS SSID
```

The following steps are presented as a guideline:



The group IMSCMDS is assumed to have full authority to all IMS commands and transactions.

1. Define the following three IMS RACF classes as generic profile checking classes:

```
TSO SETROPTS GENERIC(CIMS RIMS TIMS) GENCMD(CIMS RIMS TIMS)
```

2. Define a single, generic profile in the classes:

```
TSO RDEFINE CIMS (*) OWNER(SYS1) UACC(NONE)
TSO RDEFINE RIMS (*) OWNER(SYS1) UACC(NONE)
TSO RDEFINE TIMS (*) OWNER(SYS1) UACC(NONE)
```

3. Allow access to RACF group IMSCMDS:

```
TSO PERMIT * CLASS(CIMS) ACCESS(READ) GENERIC ID(IMSCMDS)
TSO PERMIT * CLASS(RIMS) ACCESS(READ) GENERIC ID(IMSCMDS)
TSO PERMIT * CLASS(TIMES) ACCESS(READ) GENERIC ID(IMSCMDS)
```

4. Activate the classes:

```
TSO SETROPTS CLASSACT(CIMS RIMS TIMS)
```

You can define additional profiles with RDEFINE to limit the authority for specific transactions or commands, or to secure asynchronous hold queues, as necessary. For additional guidance on setting up RACF, see *SecureWay Security Server RACF Security Administrator's Guide*, *SecureWay Security Server RACF Command Language Reference*, and *IMS Version 13 System Administration* from IBM.

## IMS Security Exits

If present, signon, command (DFSCCMD0), and transaction (DFSCTRN0) authorization exits might be called after RACF authorization occurs.

These exits override the RACF authorization status, whether it is approved or denied.



Consider the effects of having these exits in place when defining security.

## IMS Transactions

This section describes the way the OTMA Interface, which is part of the IMS interface and Substation ES, handles transactions.

### Request-Reply Guaranteed

When using Substation Guaranteed services, the OTMA client (TIBCO Substation) performs these steps:

1. Specify `SyncLevel=1` (`Syncpt`) in the OTMA message prefix.
2. Specify `commit mode 1`.  
This is the only option that is used.
3. The transaction is received by Substation ES, sent to OTMA, and then to IMS.
4. After an ISRT by the IMS application on IOPCB to TPIPE, the output is returned to OTMA then to Substation.
5. OTMA waits for an ACK/NACK from Substation.
6. Substation delivers the message to the ESB and waits for a response.
7. Depending on the outcome of operations, Substation sends an ACK/NACK to OTMA.
8. IMS commits or rolls back the transaction.
9. Substation ES closes its UOW.

## IMS Events or Triggers

Substation ES supports both the receiving and publishing of IMS Asynchronous and Synchronous events or messages. Programs such as MPPs or BMPs can publish events by Substation directly onto ESB.

Substation ES does not require the OTMA exits to be installed in order to deliver outbound events. To enable outbound events, you must update the IMS member `DFSTDYxx` and define IMS descriptors. Product installation sample member `DFSYDT0` in `USERHLQ.PROC` contains the recommended default Substation ES definitions.

### General Trigger Processing

Substation ES supports both guaranteed and reliable triggers or events.

Asynchronous callouts exist in the IMS terminology. To use asynchronous callouts from MPR, BMP, IMS Fast Path, or Java processing facilities, use the `ISRT` and `PURG` calls to issue a transaction to a destination defined in `DFSYDTx` with a TPIPE that matches either the **TRIG-RDQ-NAME** or the **TRIG-GDQ-NAME** parameter in the Substation ES IMS interface definition.

Substation ES receives the transaction, and passes it on to TIBCO Enterprise Message Service or Rendezvous. No response is sent to the transaction, and only an ACK is sent to IMS. A reliable trigger is ACKd immediately upon receipt by Substation ES, while a guaranteed trigger is ACKd when Substation ES receives acknowledgment from TIBCO Enterprise Message Service or Rendezvous.

A sample program written in COBOL, `SXICIPB1`, is provided in the COB library.

### Synchronous Callout (ICAL) or Request/Reply Support

The Substation ES OTMA interface supports the use of synchronous callouts (ICAL), also known as request/replies.

To use synchronous callouts from an MPR, BMP, IMS Fast Path, or Java processing facilities, use the `ICAL` call to send a request to a destination defined in `DFSYDTx` with a TPIPE that matches the **TRIG-SYNC** parameter in the Substation ES IMS interface definition. Substation ES receives the request, and

passes it on to TIBCO Enterprise Message Service or Rendezvous. When the reply is received, Substation ES posts the response to the ICAL issuer.

A sample program written in COBOL, `SXICIRR1`, is provided in the COB library.

## Processing of IMS OTMA Trigger Errors

This section describes how to resolve two error conditions.

### Buffer Overflow

If the Substation ES OTMA client interface receives a message from OTMA, an error condition results, such as a buffer overflow (identified by the message `SXI4712E` or `SXI4703E`). That overflow occurs because the size of either the SRB buffer or the Substation ES communications buffer is too small to accommodate the message. Subsequently, the Substation ES OTMA client returns a negative acknowledgement (NAK) to OTMA and issues an error message to the log. The invalid transaction is queued to the special destination `SXITRNAS`.

To resolve that error situation, do one of the following:

- If the SRB buffer size causes the error (indicated by the `SXI4712E` message), change the **BUFFLEN-SRB** parameter to accommodate the size specified in the message.

If the communication buffer size causes the error, (as depicted by the `SXI4703E` message), change the **BUFFLEN-TRANS** parameter to accommodate the size specified in the message.

To activate the changes, recycle Substation ES so that it can handle the message with the increased size, and then issue the **REL-TRNAK** command when ready to reprocess the NAKd triggers. No loss of IMS queue buffers will occur.

- Issue an IMS Dequeue command to empty the queue; all data in queue is then lost. You need not recycle Substation ES.

### Network Failures

If an ESB connection that relates to IMS is lost or broken, IMS trigger processing stops. Inflight guaranteed triggers that have not been acknowledged might be NAKd, and there is also a very tiny window where inflight reliable triggers might be NAKd as well. All NAKd triggers are queued to the special destination `SXITRNAS`. When the ESB connection is reestablished, trigger processing restarts. NAKd triggers, however, are not automatically processed. To process them, issue the **SET, IMS, REL-TRNAK** command.

### Guaranteed Trigger NAK Tpipe

In the unlikely scenario where a guaranteed trigger message is NAKd (failed delivery), the message is queued back in IMS to a Tpipe named `SXITRNAS`. To reprocess NAKd triggers, issue the **SET, IMS, REL-TRNAK** command. If an error occurs again when processing NAKd triggers, the current message is NAKd again and the **REL-TRNAK** processing is halted.

## Exceptional Cases

This section describes some exceptional cases that you must be aware of when you use Substation ES.

### IMS System Response and Non-reply Messages

When you are using the IMS transaction method of invocation, and do not receive an expected reply message, the TIBCO messaging application might expect to receive IMS system responses.

To allow for this situation, the recipe must define the outbound conversion rule with a large enough buffer size to accommodate the IMS system responses. Alternately, you can define an extra field and map it as an optional field at the end of the inbound buffer.

# Logging, Tracing and Auditing

---

You can implement and interpret the Substation ES Log, Trace, and Audit Agents.

For details, see the following links:

- [Introduction](#)
- [Features](#)
- [Disk Logging](#)
- [LTA Services Utility](#)

## Introduction

The logging, tracing, and auditing facilities in Substation ES record events that occur during Substation ES execution, mainly to provide users and administrators with a recording facility of accurate real-time information of events, audits, definitions, and statistics produced by Substation ES.

Although the events recorded are written to different output destinations, this guide refers to the logging, tracing, and auditing facilities collectively as LTA Agents, which are executed as sub-tasks within Substation ES.

### Log Agent

Information recorded by Log Agent is targeted for Substation ES users and administrators. Log entries provide information on Substation ES-executed events, console interactions, statistics, and messages, which can be informational messages, error messages, or warnings.

All the messages in the Log files are detailed in *TIBCO Substation ES Messages and Codes*. You can route the log information to a disk file.

### Trace Agent

Information recorded by the Trace Agent is targeted for TIBCO Support and development staff. The Trace file contains the same entries as the Log file with additional event detail, storage snaps, and function level tracing. None of the information written to the Trace Agent destination is expected to be documented or meaningful to users.

You can route the Trace information to a disk file.

### Audit Agent

Targeted for Substation ES users, security staff, and administrators, information recorded by the Audit Agent is on the following:

- Substation ES System Startup Parameters and System Initialization Parameters
- Access and security validations and violations that occur during Substation ES execution

All the messages in the Audit files are detailed in *TIBCO Substation ES Messages and Codes*. Currently, the Audit Agent is not yet fully implemented and you cannot route the Audit information to a disk file.

## Features

This is an overview for the destination, formats, and debug file of the LTA Agents.

### Destination

You can route information recorded by the LTA Agents as a formatted print file or to a disk file defined in Substation ES. Make your selection by specifying the correct destination in the System Startup Parameters.

## Formatting

If you specify an Agent destination to print (P), the information can be formatted in one of the following manners:

- As a single line. Message entries that exceed 80 characters are truncated. For example:

```
SXA2601I Console Command 'REFR,RID='SXC-DPL-COBOL-START-TRAN-LONG-
```

- In full with additional identifying information, if available, for example:

```
SXA2601I Console Command 'REFR,RID='SXC-DPL-COBOL-START-TRAN-LONG-NAME' '
```

- In full with all the details available to the Agent. Normally, this format is for tracing information only. For example:

```
009C 2011/05/05 15:31:14.6141 00000003 05002889 0 0 ID5: Mod: 66 Line: 180
Function: Flush_Admin_Buffer
SXA2601I Console Command 'REFR,RID='SXC-DPL-COBOL-START-TRAN-LONG-NAME' '
```

## Debug File

The Log and Trace Agents have their own debug file in which information is recorded with respect to their execution. The amount of information is determined by the value of the LOGDBLVL and TRCDBLVL keywords.



Do not change the LOGDBLVL and TRCDBLVL value unless requested by a TIBCO representative. The content in the debug file is intended for TIBCO support and development staff only.

## Disk Logging

Substation ES Log and Trace Agents can write recorded information to disk files.

To log information to disk files, you must perform some additional tasks before the files can be used and Substation ES is started.



For JES2 systems, do not enable the disk logging feature until after successful execution of the Substation ES IVPs.



If you intend to use VSAM DISK LOGGING or TRACING on a heavily used Substation ES, you can reduce the overhead of the VSAM disk I/O by increasing the VSAM "RECSZ" parameter found in JOB <userhlq>.JCL(SXGLDEF). For Log VSAM file, change 2400 to 8096 (the size will vary based on machine and disk throughput). For the Trace VSAM file, change 9600 to 16384. To get the best results for your installation, TIBCO recommends that you try different sizes (8K, 12K, 16K, etc).

## LTA Disk Files

Each LTA Agent can have multiple active disk files with a maximum of five per agent in the Substation ES region. Disk files to be used by the LTA Agents must have a corresponding DD Name on the LOGDSK-DDN or TRCDSK-DDN keyword.

You must define and format LTA disk files before they can be used.

### WARM or COLD Options

Information recorded to disk files can be kept or overwritten across multiple restarts of Substation ES. This feature is available if you specify the appropriate value on the LOGDSK-START and TRCDSK-START keywords. The default is to retain information. Two options are available:

- The COLD option, which indicates that previously recorded information is overwritten. All disk files for the specified Agent are reset to empty and the existing information within these files cannot be accessed again. The information is lost.

- The WARM option, which indicates that when Substation ES starts, the information recorded in the previous executions is retained and new information is recorded following the most recent entry in the last used file. The information is retained.

## Reuse

The LTA Agents can reuse currently full disk files without offloading information. This feature is available if you specify the appropriate value on the LOGDSK-WRAP and TRCDSK-WRAP keywords. Two options are available:

- If you specify Y for the -WRAP keyword, that means that when all the files for the Agent are full, the oldest information is overwritten. This option is recommended for trace files with a high volume of recorded information or if old information does not need to be kept or archived. Y is the suggested value for testing and development environments.
- If you specify N for the -WRAP keyword and a file is full, the Agent switches to the next available file. The file that is deemed full must be archived or the file reset if the file is to be reused.

The default value of the keywords is N, meaning not to reuse the disk file until the data is archived (offloaded) or reset.



If all the files for an Agent are full, Substation ES suspends operations and reroutes information to the formatted print file. If that fails, the applicable LTA Agent is disabled. Avoid having LTA Agent files become full by archiving their information in a timely manner. Currently, no automatic archiving is available.

You can view the information on the status of the trace files on disk with the **SHOW,TRCSTATS<,ALL>** operator command.

## Search and Display Facility

If log and trace information is written to disk files, you can view the recorded information online in the Substation ES panels.

To access the LTA display panels:

1. Execute the following command:  
**TSO EX 'USERHLQ.CLIST(SXGSSPNL)'**
2. Select Option 2, LTA Search and Display Facility.

The following describes the related panels.

### File Selection Panel

```

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

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

Other VSAM Cluster Name:
  Cluster Name . . 'TIBQA.SXS.V2R7M0.FY.LOG.DISKF1'

Display Format ==> 2          (1, 2)
Direction      ==> F          (F-Forward B-Backward)
Number to Read ==> 9999      (Number to read initially)

VSAM Debug ==> (0 - 4)

```

Pgm    Debug ==&gt;    (0 - 4)

Enter END command to terminate.

The following table describes the fields in this panel.

*File Selection Panel Fields*

| Field                 | Description  |
|-----------------------|--|
| <b>ISPF Log</b>       | This field specifies a log to view either by providing values for <b>Project</b> , <b>Group</b> , and <b>Type</b> ; or by specifying a fully qualified cluster name in the <b>Other VSAM Cluster Name</b> field. If this field is nonblank, the value for the <b>Other VSAM Cluster Name</b> field is in effect. |
| <b>Display Format</b> | Format 1 is one recorded entry per line (truncated if necessary). Format 2 is a two-line display with the description on the second line.  |
| <b>Direction</b>      | This field has two options: Forward and Backward. Backward starts with the most recently recorded entries and reads backward for the specified number of records, thus showing the most current activity.  |
| <b>Number to Read</b> | This field specifies the number of recorded entries to display. If you do not specify a value, the default of 200 is in effect.  |
| <b>VSAM Debug</b>     | This field defines the debugging levels for the VSAM access routines. Do not change this value unless requested by authorized TIBCO personnel. The default is 0 (zero).  |
| <b>Pgm Debug</b>      | Defines the debugging levels for the ISPF interface program. Do not change this value unless requested by authorized TIBCO personnel. The default is 0 (zero).   |

*LTA Multiple Entry Panel*

SXLP002----- Log Entries ----- Row 1 of 29  
 Command ==> Scroll ==> PAGE

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



```

11:09:55.8525 0 SXT5036I 0 0 All transformer tasks are Good To Go
11:09:55.8525 154 SXG2806I 0 0 Activated ~ Substation ES - Admin & Op
11:09:55.8531 154 SXG2806I 0 0 Activated ~ Substation ES - IMS Interf
11:09:55.8610 154 SXG2806I 0 0 Activated ~ Substation ES for IVP's
***** Bottom of data *****

```

The following table describes the fields in the panel.

#### *LTA Multiple Entry Panel Fields*

| Field          | Description  |
|----------------|--|
| <b>Sel</b>     | If you specify S in the input column, a pop-up panel with all the values that relate to the selected message is displayed. |
| <b>Time</b>    | This field shows the time this message was recorded by the LTA Agent.  |
| <b>Type</b>    | This field shows the code of the internal message type.  |
| <b>Msg Id</b>  | This field shows the message identifier.   |
| <b>RC</b>      | This field shows the return code associated with the message.  |
| <b>Rsn</b>     | This field shows the reason code associated with the message.  |
| <b>Message</b> | This is the initial part of the message description. The entire message is displayed if you select the pop-up panel.       |

#### *LTA Message Pop-up Panel*

```

----- Single Message Values ----- Row 1 to 4 of 4
Command --->

Msg Id: SXS1000I      Union id: 0      Stck: B724D4951AD33081
Type: 154             GRIN: 0          RC: 0
Date: 2002/02/05      Proc Id: 83886173    Reason: 0
Time: 12:30:11.9549   Thread: 0

Message-Dependent Values:
  Buffer length - 41

-----
| No additional information
|
| Message:
| Starting TIBCO Substation (ES) for z/OS
| ***** Bottom of data *****

```

The following table describes the fields in the panel.

#### *LTA Message Pop-up Panel*

| Field         | Description                                    |
|---------------|--|
| <b>Msg Id</b> | This is an eight-character message identifier. |
| <b>Type</b>   | This is the internal message type.             |

| Field           | Description  |
|-----------------|--|
| <b>Date</b>     | This is the date on which this message was recorded by the LTA Agent.  |
| <b>Time</b>     | This is the time at which this message was recorded by the LTA Agent.  |
| <b>Union id</b> | This is the representation of what message fields were recorded for this entry. See <i>TIBCO Substation ES Messages and Codes</i> for a description. |
| <b>GRIN</b>     | This is the internal Global Resource Identification Number.  |
| <b>Proc Id</b>  | This is the internal process identifier, which identifies a Substation ES task or sub-task.  |
| <b>Thread</b>   | This is the thread of Substation ES that issued the message.   |
| <b>Stck</b>     | This is the internal 64-bit Store Clock value, which uniquely identifies this message.   |
| <b>RC</b>       | This is the return code associated with the message.   |
| <b>Reason</b>   | This is the reason code associated with the message.   |
| <b>Message</b>  | This is a complete message description and can be in multiple lines, if necessary.   |

## Setup of LTA Disk Files

This section describes how to define and initialize LTA disk files.

### Definition of LTA Disk Files

JCL is available for defining a default set of Substation ES log and trace files. The following table lists the recommended values.

#### *Default Values of LTA Disk Files*

| IDCAMS Parameter             | Log File Values | Trace File Values | Notes   |
|------------------------------|-----------------|-------------------|---|
| <b>NUMBERED</b> <sup>1</sup> | Yes             | Yes               | You must specify a value for the <b>NUMBERED</b> parameter because it designates that the file is a Relative Record data set. |
| <b>REUSE</b>                 | Yes             | Yes               |   |
| <b>SHR</b>                   | (2, 3)          | (2, 3)            |   |

| IDCAMS Parameter            | Log File Values | Trace File Values | Notes  |
|-----------------------------|-----------------|-------------------|--|
| <b>RECSZ</b> <sup>2</sup>   | (2400,2400)     | (9200,9200)       | Because the entries recorded in the LTA disk files are of fixed length, the average size and the maximum record size of the <b>RECSZ</b> parameter must be equal. The minimum value to be specified for the average record size is 1048 bytes. |
| <b>RECORDS</b> <sup>3</sup> | (4000)          | (3000)            | You can change the value of the <b>RECORDS</b> parameter to suit your site's environment. The installation default creates two files each for the Log and Trace Agents.  |

Space usage for the default Log and Trace files is based on a 3390 model 3 drive, in which the Log allocates 13 cylinders and the Trace File allocates 34 cylinders of disk storage.



If you do not set the **WRAP** parameter to on, be careful of changing the number of records to be less than the default value. If you set the number of records too low, many entries are recorded and files become full very quickly.

### Initialization of LTA Disk Files

Before Substation ES can use the newly defined LTA disk files, you must format them with the Substation ES LTA Services Utility, which formats one file per execution. This format routine of the utility creates control and summary information for internal use on the first record of each file. Additionally, the utility formats each record within the file with specific header information.

Sample JCL in the **SXGLINIT** member is available for initializing the default Log and Trace disk data sets of Substation ES.

## LTA Services Utility

The LTA Services Utility supports Substation ES LTA Agents.

### Features

The LTA Services Utility is a z/OS batch program with the capability of formatting, archiving, copying, resetting, and printing LTA disk files.

#### Printing

The LTA Services Utility can reproduce the contents of LTA disk files in a formatted print file. The utility can first archive the recorded information within the LTA disk files, and then print it as required.

#### Copying

The LTA Services Utility can copy the information within an LTA disk file to a sequential data set (with no change to the file's status indicator) and print the copied information simultaneously.

For example, the following will copy and print an LTA VSAM disk file:

```
' -CY -F2 '
```

#### Resetting

The LTA Services Utility can reset the status of an LTA disk file and mark it as empty without archiving the recorded information. Use this function only if the information is not to be retained.

For example, the following will reset an LTA VSAM disk file:

'-RY'

## Archiving

The LTA Services Utility can archive the information within an LTA disk file to a sequential data set. When archived, the file is marked as empty and can be reused by the LTA Agents. Archiving combines the copy and reset functions of the LTA Services Utility. Currently, no automatic archiving is available.

For example, the following will archive from VSAM, copy, and reset:

'-IV -CY -RY'

## JCL Sample (SXGLINIT and SXGLUTIL)

A sample procedure and JCL are provided to execute the LTA Services Utility.

```
//LTAINIT PROC FILE=USERHLQ.LOG.DISKF1
//*
//LOGINIT EXEC PGM=SXL3LOGS,REGION=OM,
// PARM='-FX -IV -XY'
//STEPLIB DD DISP=SHR,DSN=USERHLQ.LOAD
//SXLLOG1 DD DISP=SHR,DSN=&FILE
//SXLDEBUG DD SYSOUT=* DEBUG MESSAGES FROM VSAM ACCESS
//SXLPRINT DD SYSOUT=* FORMATTED OUTPUT
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD DUMMY
//ARCHIVE DD DISP=(,DELETE),DSN=&&ARCHIVE,UNIT=SYSDA,
// SPACE=(TRK,(20,40)),
// DCB=(RECFM=VB,LRECL=1048,BLKSIZE=18000)
// PEND
```

The following table describes the DD Names.

### Services Utility DDNames

| DD Name  | Description   |
|----------|---|
| SXLLOG1  | This DD Name specifies the LTA VSAM file for Substation ES and is required for the -IV operand.   |
| SXLDEBUG | This DD Name is for utility diagnostics.  |
| SXLPRINT | This DD Name is used if formatted printing is requested.  |
| ARCHIVE  | <p>You must supply this DD Name when specifying the Copy or Archive keyword. This is an output file that must have RECFM=VB.</p> <p>You must supply this DD Name when specifying the PRINT keyword phrases and -IA operand to print an archived file. This is the DD Name of an input archive file.</p> |

## Keyword Phrases

The following table describes the keyword phrases that combine operands to perform the common functions of the LTA Services Utility. These phrases are also specified on the EXEC statement's **PARM** parameter.

### *Keyword Phrases for the LTA Services Utility*

| Keyword Phrase | Operand Default | Description   |
|----------------|-----------------|---|
| TAIL           | -B -N20         | Print the last 20 records in the file.  |
| PRINT          | None            | Same as allowing all defaults. Print the entire log file with FMT1.                                     |
| FORMAT         | -FX -IV -XY     | Create a log file on a VSAM RRDS.   |
| ARCHIVE        | -FX -IV -CY -RY | Copy all the records in the log to the ARCHIVE DD and reset the log to empty. Do not print the records. |

## Control Parameters

The functions that the LTA Services Utility must perform are determined by the keywords and operands on the EXEC statement's **PARM** parameter.

The following table lists the supported values.

### *Services Utility Control Parameters*

| Keyword | Operand          | Description   |
|---------|------------------|---|
| -B      | None             | Access the log backward, starting with the most current entry.  |
| -C      | {N   Y, N}       | Copy the records to the ARCHIVE DD Name. The default is N.  |
| -F      | {1   1, 2, 3, X} | Specify the format of the report. Set the value to X if no report is desired.   |
| -G      | {2   0 - 4}      | Specify the debug level for Log Services. The value 0 means no messages; 4 means many messages.   |
| -H      | 0 - 4            | Specify the debug level for VSAM access. The values are the same as those for -G. Reset to 0 (zero) if the SXLDEBUG DD statement is missing.  |
| -I      | {V   V, A}       | Specify the input to processing: VSAM (default) or Archive (print an archive file). If the input is Archive, the -C and -R keywords cannot use the Y operand and you must not specify the -B keyword. |
| -N      | 0-99999999       | Specify the number of records to read. The default is no limit.   |
| -R      | {N   Y, N}       | Reset the VSAM cluster to empty. Reset the first two records only and do not rewrite the entire file. The default is N.   |

| Keyword | Operand    | Description   |
|---------|------------|---|
| -S      | 0-99999999 | Specify the number of records to skip. The default is none.   |
| -X      | {N   Y, N} | Create a log from a newly defined cluster, which must be done before other operations can be performed on the log. The default is N.<br><br>Must specify or default -FX -IV -RN -CN with -XY. |



If you intend to use VSAM DISK LOGGING or TRACING on a heavily used Substation ES, you can reduce the overhead of the VSAM disk I/O by increasing the VSAM "RECSZ" parameter found in JOB *userhlq.JCL(SXGLDEF)*. For Log VSAM file, try changing 2400 to 8096 (the size will vary based on machine and disk throughput). For the Trace VSAM file, try changing 9600 to 16384. To get the best results for your installation, you will have to try different sizes (8K, 12K, 16K, etc).

## System Logger Stream

Substation ES Log and Trace Agents can write recorded information to z/OS System Logger Stream.

To log information to z/OS System Logger Stream, perform some additional tasks to define the coupling facility log stream before Substation ES is started. To see the detailed steps of the JCL, see Defining System Logger Streams in the Substation ES Installation guide.

## Log and Trace Data Viewer

Log and trace data can optionally be written to the z/OS System Logger.

In order to view these logs you can go to the CLIST library, and either execute the SXGSSPNL clist, and select option 3, labeled "System Logger Search and Display Facility", or execute the SXLVIEW clist, which displays the TIBCO Log Stream Viewer panel.

```

----- TIBCO Software Inc. -----
Main Entry Panel                                     2019/07/10
                                                    15:22

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

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

Select option  ==>
1  Configuration
2  LTA Search and Display Facility
3  System Logger Search and Display Facility

ENTER = PROCEED   END PFKey = RETURN

Copyright TIBCO Software Inc. 1998 - 2019

----- TIBCO Log Stream Viewer -----
Command ==>

STREAM _____ HUBNAME _____

```

```

JOBNAME _____
JOBID   _____
UOW. .  _____
CORRID  _____

          YYYY MM DD  HH MM SS
START DATE/TIME . 2020 04 13 15 11 07
END   DATE/TIME . 2020 04 14 23 59 59

```

Press PF3 to terminate or ENTER to continue

In the Log Viewer panel, you must fill in the **STREAM** name you used in your setup for the log, trace, or stats file.

You can optionally fill in the following fields to reduce the amount of output displayed.

| Fields                 | Description   |
|------------------------|---|
| <b>HUBNAME</b>         | By default the Log Viewer will use the system default HUB, the default value is the SMF ID of the LPAR. If you want to use a different HUB, you can specify the HUB name in this field. |
| <b>JOBNAME</b>         | Full or partial jobname, for example, MYJOBNM or MYJOB* to select output from a single or multiple jobs.  |
| <b>JOBID</b>           | Full JOBID in the format STCnnnnnn, JOBnnnnnn, or TSUnnnnnn.  |
| <b>UOW</b>             | Full or partial Unit of work ID, for example ,1234 to show all UOWs containing the string 1234.   |
| <b>CORRID</b>          | Full or partial correlation ID, for example, 1234 to show all correlation IDs containing the string 1234.   |
| <b>START DATE/TIME</b> | Start date & time for data, defaults to 24 hours ago.   |
| <b>END DATE/TIME</b>   | End date & time for data, defaults to the end of the current day.   |

When you hit enter you are presented with the selected data in a normal browse interface, where all the usual browse commands work.

There are a few logger specific commands:

| Commands | Description  |
|----------|--|
| #C       | <p>Displays a popup window containing the jobname, Unit of Work, and Correlation Id of the current line, or the line where the cursor is located if within the data. This allows you to pick a specific job, Unit of Work, and Correlation Id.</p> <p>Click PF7 to go to the previous line, and PF8 to go to the next line.</p> <p>Click PF3 to go to the selection menu, and refine the amount of data to be displayed.</p> |
| #I       | <p>Displays a popup window containing the selection criteria from the previous menu.</p>   |
| #S       | <p>Displays a popup window containing the Correlation Id, Unit of Work and various Statistics from the record pointed to by the cursor.</p> <p>Click PF7 to go to the previous line, and PF8 to go to the next line.</p>   |

### TIBCO z/OS Logging API

TIBCO provides API call to facilitate an application to log (write data) to a z/OS System Logger Data Stream.

This section describes the concepts and details of the TIBCO z/OS Log API call. To code this API call in Assembler, C, COBOL, Fortran or PL/I on LE, follow the rules of the individual language for making calls to the API.

Use the following API writes to log or trace to a z/OS System Logger Data Stream. The API use the z/OS function IXGWRITE but to different stream names.

The APIs can be found in Substation ES library PDSE.

- For LE programs use: *USERHLQ.library(SXGAWLOG)*.
- For NON-LE program use: *USERHLQ.library(SXGFWLOG)*

When an application uses API, include above object in the binder step during compilation.

#### SXGAWLOG

SXGAWLOG library writes data to a z/OS System Logger Data Stream for logging and tracing.

It works in the following manner:

1. The program is used to create the logger record structure (ie. DATA)
2. The program calls SXGAWLOG with appropriate parameters.
3. SXGAWLOG determines which entry point was used.
  - a. It finds the matching HUB name running on the LPAR.
    - In case no match is found, it displays return code 12 and reason code 8101.
  - b. It calls the HUB process for logging.
  - c. If the HUB has logging setup, then the message is written. Otherwise, an error message is displayed.
  - d. After the z/OS stream write, the HUB returns the RC return code and REASON code to SXGAWLOG, which in turn returns the values to the caller.



4. Each call to SXGAWLOG is a standalone request. The HUB value can be the same or different between calls.
5. The SXGAWLOG is a synchronous call.

### COBOL Call

```
CALL "SXGAWLOG"
  USING BY REFERENCE HUB-NAME,
        BY REFERENCE WS-TYPE,
        BY REFERENCE WS-LOGREC,
        BY REFERENCE WS-REASON,
        RETURNING WS-RETURN
END-CALL.
```

### C Declaration

```
int SXGAWLOG(
    char *HUB,
    char *TYPE,
    void *LOGREC,
    int *REASON);
```

## Parameters

The following table describes the parameters:

| Parameters | Description   |
|------------|---|
| TYPE       | This is the address of a one character value indicating data streams to write to <ul style="list-style-type: none"> <li>• L - log data stream only</li> <li>• T - trace data stream only</li> <li>• B - both log and trace data streams</li> <li>• A - both log and trace data streams</li> <li>• Anything else - log data stream only</li> </ul> |
| HUB        | This is the address of a 4 character value indicating the HUB Name  |
| LOGREC     | This is the address of the length followed by the data to be written to the stream (since the length is a half word, the amount of data is limited to 32767). The actual max for the data is based on the z/OS System Logger Stream definition.   |
| REASON     | This is the address of a fullword (int) that will contain a code number based on the results of the call.   |
| int        | This is the return code of the call (0 means call was successful).  |

The following table describes the reason codes in Substation ES:

| Codes | Description  |
|-------|--|
| 8101  | Indicates there is no active HUB with the name (4 characters) provided.<br>Application program terminated with return code 12. |
|       | <b>User Response:</b> Check the HUB is up and running and the HUB name used is correct.  |



All other reason codes are returned from the z/OS function IXGWRITE.

**Example:** COBOL format of data structure and the call.

```
01 HUB-NAME          PIC X(4) VALUE SPACES.
01 WS-TYPE           PIC X VALUE 'L'.
01 WS-REASON         PIC S9(8) COMP.
01 WS-RETURN         PIC S9(8) COMP.
01 WS-LOGREC.
   05 WS-LOG-LEN      PIC S9(4) COMP-1.
   05 WS-LOG-DATA     PIC X(32767) VALUE SPACES

CALL "SXGAWLOG"
   USING BY REFERENCE HUB-NAME,
         BY REFERENCE WS-TYPE,
         BY REFERENCE WS-LOGREC,
         BY REFERENCE WS-REASON,
         RETURNING WS-RETURN
END-CALL.
```

**Example:** C format of data structure and the call.

```
typedef struct
{
    short length;          //length of data in next array
    char data[32767];
} logrec;

#include "SXGHWLOG.h"
int RC;
int reason;
RC = SXGAWLOG(
    "HUBA",
    "L",
    &logrec,
    &reason);
```

# Substation ES Utilities

This section covers the utilities available in Substation ES.

- [Rendezvous Request-Reply Timer Utility](#)
- [Generic Sender or Listener Utility](#)
- [Generic Error Listener Utility](#)

## Rendezvous Request-Reply Timer Utility

Request-Reply Timer Utility records the time a request is delivered and the elapsed time until a reply is received for the same message.

This utility performs the following tasks:

- Measures the elapsed time from when a request is received until a reply is delivered.
- Summarizes the multiple request or reply timings for the transmission interval.
- Optionally prints the timing information for each Request or Reply event captured.

To capture the timing of the request or reply sequence, a field named in the conversational message is identified. The value of this field must be constant during the conversation. For the timer utility to capture multiple request or reply conversations, the value of the field identified in the message must be unique for each request. A message sequence number can normally be used for this purpose.

The Timer Utility was originally used for CICS outbound request or reply conversations and can be used for bidirectional conversations. For CICS-outbound requests, the Substation ES envelope field identifies conversations. The Timer Utility understands the content of the envelope and can reveal detailed information on the request when the Timer Utility is in use for that purpose.



Rendezvous INBOX communication is not supported.

The sample JCL stream is listed as follows:

- Data Set: `USERHLQ.JCL`
- Member: `SXRTMER`

Before using this utility, you must specify the correct Rendezvous transport parameters.

Also, review and select the appropriate Request-Reply Timer Control Parameters. See [Request-Reply Timer Control Parameters](#).

## Operational Commands

The Timer Utility has a z/OS console interface for assisting in the control of the Timer execution and allowing continuous processing.

The following table describes the console commands that control the execution of the Request-Reply Timer Utility.

### *Request-Reply Timer Console Commands*

| Console Command            | Description  |
|----------------------------|--|
| <b>SHUT</b> or <b>TERM</b> | Terminates the execution of Request-Reply Timer Utility. |
| <b>STATS</b>               | Prints all current totals and statistics.                |

| Console Command | Description   |
|-----------------|---|
| <b>REFRESH</b>  | Prints all the current totals and statistics and resets or clears counters and storage for the next capture interval. |
| <b>PRTLVL=#</b> | Changes the current print level to the value specified by #. The valid values are from 0 to 5.                        |

Two examples are listed as follows:

- `F timer job name,SHUT` or `P timer job name`
- `F timer job name,PRTLVL=3`

## Control Parameters

The following table describes the Request-Reply Timer parameters.

### *Request-Reply Timer Control Parameters*

| Parameter              | Description   |
|------------------------|---|
| <b>request-subject</b> | The name of the subject on which the Timer Utility listens for requests in order to capture the time and information about the request.   |
| <b>reply-subject</b>   | <p>The name of the subject on which the Timer Utility listens for the reply. This parameter enables the Timer Utility to match the request information to the reply and compute the elapsed time the conversation took place.</p> <p>You can specify up to three reply-subject names for a request subject.</p>   |
| <b>field-name</b>      | The name for the context value according to which the request or reply match is performed.  |
| <b>field-type</b>      | <p>The field type associated with the field name used as the context field request or reply conversation.</p> <p>The valid types are as follows:</p> <ul style="list-style-type: none"> <li>• STR: string</li> <li>• OPA: opaque</li> <li>• U32 or I32: 32-bit unsigned integer or 32-bit signed integer</li> <li>• CE: Substation ES CICS envelope</li> </ul> <p>If you specify CE as the type, you need not specify <b>field-name</b> and <b>field-length</b>. With CE as <b>field-type</b>, the Timer Utility identifies the envelope that originates from CICS.</p> |
| <b>field-length</b>    | <p>The length of the field associated with <b>field-type</b>. Normally, you specify <b>field-length</b> only if you define <b>field-type</b> as a string or opaque.</p> <p>The valid range is 1 to 36 bytes. The default is 20 bytes.</p>   |

| Parameter           | Description  |
|---------------------|--|
| <b>print-ex-ctt</b> | <p>The keyword that causes a printing of the entries that exceed the CICS round-trip time specified for the keyword value. Normally, quite a few entries can be printed during the sniffing of messages. Using this parameter reduces the amount printed if you are only looking for CICS round-trip messages that exceed a specific time value. The time stamp used is obtained from the message CICS Envelope.</p> <p>The time value is in seconds and up to one-thousandth of a second, for example, 1.25, .123, or 0.035.</p> <p>If both <b>print-ex-ctt</b> and <b>print-ex-rtt</b> are specified, the Timer Utility prints processes that exceed either limit, according to the set <b>prtlvl</b> level.</p> |
| <b>print-ex-rtt</b> | <p>The keyword that causes a printing of the entries that exceed the round-trip time specified for the keyword value. Normally, quite a few entries can be printed during the sniffing of messages. Using this parameter reduces the amount printed if you are only looking for round-trip messages that exceed a specific time value.</p> <p>The time value is in seconds and up to one-thousandth of a second, for example, 1.25, .123, or 0.035.</p> <p>If both <b>print-ex-rtt</b> and <b>print-ex-ctt</b> are specified, the Timer Utility prints processes that exceed either limit, according to the level set in the <b>prtlvl</b> parameter.</p>  |
| <b>prtlvl</b>       | <p>The level of the amount of information printed during the execution of the Timer Utility. If a high throughput of requests is to be processed, it is recommended that you specify a level of 2 or less.</p> <p>The valid values are 0 to 5. The default is 0.</p> <p>If the <b>field-type</b> value is CE, to view the content of the envelope, set <b>prtlvl</b> to 3.</p>   |

## Examples of Control Statements

The section contains examples of control statements.

### Example: User Application to Substation ES Conversation

```
request-subject    my-request-subject
reply-subject      my-reply-subject
field-name         sequence-number
field-type         U32
```

### Example: CICS-Initiated Request or Reply Conversation

```
request-subject    my-request-subject
reply-subject      my-reply-subject
field-type         CE
prtlvl             3
print-ex-rtt       0.055
```

### Timer Utility

All dates are in the format CCYY/MM/DD.

All time values are represented in thousandths of a second.

### CICS-Initiated Request Or Reply

If the **field-type** value is CE, the print level is 3, and the Timer Utility times a CICS-initiated request or reply conversation, the following fields are displayed:

```
SXG8645I CICS Send Time :2003/09/09 -- 11:17:06.504 -Task#:565 Idx:0
SXG8642I Req Time, Reply Time :11:17:06.824, 11:17:06.863
SXG8652I Round Trip Time :0.039 (sec.ths)
```

### CICS Envelope Fields

| Field                  | Description   |
|------------------------|---|
| <b>CICS Send Time</b>  | The time stamp of the Substation ES CICS SXCREQR routine that receives the request from the user application in the CICS region.  |
| <b>Task#</b>           | The task number of the user CICS application that issued the request.   |
| <b>Idx#</b>            | The internal storage array allocation that the Substation ES CICS Request or Reply Facility used to hold the context or envelope.   |
| <b>Req Time</b>        | The time at which Timer Utility received the request message.   |
| <b>Reply Time</b>      | The time at which the Timer Utility received the reply message.   |
| <b>CICS Trip Time</b>  | The time difference between when the message is sent by the CICS routine and when the reply message is received by the timer routine.   |
| <b>Round Trip Time</b> | The elapsed time the external application took to produce the reply. This length of time includes the time it took the network to deliver the message to and from the external application. |

### Normal Request or Reply

An example of a normal request or reply is listed as follows. The first two lines show the context field name and the field value used by the Request-Reply Timer Utility.

```
sequence-number :2147483647 Reqst Idx:0
sequence-number :2147483647 Reply Idx:0
SXG8642I Req Time, Reply Time :14:42:50.012, 14:42:50.086
SXG8652I Round Trip Time :0.074 (sec.ths)
```

### REFRESH Command Output

An example of the output of the **REFRESH** command is listed as follows. For an explanation, see *TIBCO Substation ES Messages and Codes*.

```
SXG8110I z/OS Console Command:REFRESH
SXG8014I Current time is - 12:44:57.560
SXG8655I Interval Started :2003/09/09 -- 12:43:01.566
SXG8656I Interval Ended :2003/09/09 -- 12:44:42.076
SXG8659I Interval Elapsed Time :100.510 (secs.ths)
SXG8650I Hi Water Mark :1
SXG8651I No of Interval Events :2
SXG8653I Round Trip Time (Avg) :0.041 (sec.ths)
SXG8654I Messages per Second :0.020 (sec.ths)
SXG8658I Refresh completed. All entries cleared
```

## Print Output Levels

The following table shows the messages by printing output level.

### *Messages by Print Output Level*

| Print Level | Messages   |
|-------------|--|
| 0           | No normal runtime messages<br>Summary report   |
| 1           | No normal runtime messages<br>Start and end timestamps summary message<br>Summary report   |
| 2           | Round-trip runtime messages<br>Start and end timestamps summary message<br>Summary report  |
| 3           | Runtime messages: <ul style="list-style-type: none"> <li>• Round trip time</li> <li>• Request time</li> <li>• Response time</li> </ul> Start and end timestamps summary message<br>Summary report  |
| 4           | Runtime messages: <ul style="list-style-type: none"> <li>• Round trip time</li> <li>• Request time</li> <li>• Response time</li> <li>• Slot data on request message</li> <li>• Matching field name</li> <li>• Matching Data</li> <li>• Slot number</li> </ul> Start and end timestamps summary message<br>Summary report |

| Print Level | Messages   |
|-------------|--|
| 5           | Runtime messages: <ul style="list-style-type: none"> <li>• Round trip time</li> <li>• Request time</li> <li>• Response time</li> <li>• Slot data on request message</li> <li>• Matching field name</li> <li>• Matching Data</li> <li>• Slot number</li> <li>• Slot data on request message: hex dump</li> <li>• Reply data: hex dump</li> </ul> Start and end timestamps summary message<br>Summary report |

## RED State Monitor Utility

The RED State Monitor Utility SXX3NMON is given a list of RED URLs (up to 30) to monitor and reports state change on each of them. This allows the user to modify it and add business processes when a RED URL becomes “active” or “in-active”. Look at function “call\_user\_process” at the end of the source. You can add code here to handle either condition (“active” or “in-active”). Currently, the routine just prints out a message.



Your code runs in-line to the one thread that is monitoring all the RED URLs in the program’s input, so serialization happens.

The SXX3NMON source is provided in *USERHLQ.C* which is written in C and uses the TIBCO Mainframe RED product. The JCL sample is as follows:

- Data Set: *USERHLQ.JCL*
- Member: SXXBMON

### SYSIN Parameters

The commands or messages sent by the utility are determined by the keywords and operands in the SYSIN DD Name file. See the following table.

#### *RED State Monitor Control Parameters*

| Parameter           | Description  |
|---------------------|--|
| <b>DEBUG</b>        | This parameter sets debug level of messages. It is recommended that you retain the default (0) unless TIBCO Support requires the output.<br>Valid values: 0 - 5. Default value: 0. |
| <b>HUB</b>          | Terminates the execution of RED State Monitor Utility.   |
| <b>URL&lt;#&gt;</b> | Specifies the URL to be monitored. where # is the sequence number 1 to 30.   |



## Examples of Control Statement

```
//SYSIN DD * INPUT PARAMETERS
DEBUG 0
URL1    TIBSS.REDADM
URL2    TIBMON.C512WW3G
URL3    TIBMON.SSES#4#IVP
```

Below are the URL TIBMON.C512WW3G state change messages output from the monitor utility.

```
18:20:53.833 - TIBMON.C512WW3G DOWN
18:21:54.940 - TIBMON.C512WW3G UP
```

## Operational Commands

The Monitor Utility has a z/OS console interface for assisting in the control of the Monitor execution and allowing continuous processing. The following table describes the console commands that control the execution of the RED State Monitor Utility.

### RED State Monitor Control Command

| Command                    | Description   |
|----------------------------|---|
| <b>SHUT</b> or <b>TERM</b> | Terminates the execution of RED State Monitor Utility.        |
| <b>STATS</b> , <URL>       | Prints all current totals and statistics for the monitor URL. |

Two examples are listed as follows:

- F monitor job name,SHUT or P monitor job name
- F monitor job name,STATS,URL

Given below is an example of the output of the **STATS** command with URL TIBSS.MSGS#YODA

```
Stats command Url:'TIBSS.MSGS#YODA'
LPAR:CAFE Job:C512WW3 Member:TIBSS.MSGS#YODA
XCF Status      :Active      RED Status      :Active      Buffer OvrFlow :0
Msgs Discarded  :0           Msgs Expired    :0           Msgs Held      :0
Buffer Size     :8460        Msgs > Buff Sz :0
Msgs Received   :6           Largest Size(R):1084   Max Msg Occ    :0
Msgs in Pool    :0           Pool Overflows :0           HWM Buffers    :1
Msg Reply       :3           Msg Reply(SNC) :0           Msg SNC NACK    :0
Req/Reply       :0           Req/Reply Fail :0           Msg RC4         :0
SNC Sig(S)      :3           SNC Sig(R)     :0
Mux Exec        :7           Mux Abend      :0           Mux Msg Lost    :0
```

For the field description of the output please refer to [Field Descriptions for output of XTUV,STATS](#)

## Generic Sender or Listener Utility

The Substation ES Generic Sender or Listener Utility provides general basic messaging support to and from Substation ES.

The Substation ES Generic Sender or Listener Utility enables you to do the following tasks:

- submit Substation ES operational or console commands from a remote application.
- submit IMS operator commands from a remote location through Substation ES to the IMS back-end system and receive the returned output.
- submit basic Rendezvous messages to other Rendezvous clients.

Substation ES accepts operational commands from user applications, such as the generic Sender or Listener Utility. The Substation ES Administrative Interface accepts and processes the commands sent from external applications by means of Rendezvous messages.

IMS commands can also be sent through messages to Substation ES, which in turn sends the commands to the IMS Interface and to the IMS back-end system. User applications must pass the correct user ID or password (or both) for the IMS operational commands to be executed successfully.

For Substation ES to accept operational or administrative commands from messages, you must specify a recipe with the correct Method of Invocation (MOI) and back-end system. Two sample recipes are supplied during installation, as follows:

- **Substation-Admin** for Substation ES
- **Substation-Admin-IMS** commands for IMS command processing

The following table describes the data types supported by the Generic Sender or Listener Utility. These data types are part of the **FLD** control parameter in the SYSIN DD Name file.

#### *Data Types Supported by Generic Sender or Listener Utility*

| Data Type | Description   |
|-----------|---|
| I32       | 32-bit signed integer.  |
| U32       | 32-bit unsigned integer.  |
| PROP      | Character string. The maximum size supported is 80 bytes. You must enclose the String value within double quotation marks.<br><br>This field is added as a property field to the EMS message, and as a message field to the RV message. |
| STR       | Character string. The maximum size supported is 80 bytes. You must enclose the String value within double quotation marks.  |

### Rendezvous Generic Sender or Listener

The SXR3GEN program is in *USERHLQ.LOAD*.

The JCL sample is as follows:

- Data Set: *USERHLQ.JCL*
- Member: *SXRGENSEL*

Before using this utility, you must specify the Rendezvous transport parameters, for example:

```
//SET IVPNETW='-network 127.0.0.1 -service 7555'
//SXRGENSEL EXEC PGM=SXR3GEN,PARM='&IVPNETW'
```

### SYSIN Parameters

The commands or messages sent by the utility are determined by the keywords and operands in the SYSIN DD Name file. See the following table.

#### *Rendezvous Generic Sender or Listener Control Parameters*

| Keyword     | Operand  | Description  |
|-------------|----------|--|
| DEBUG   DBG | OFF   ON | Enables debug facility messages. It is recommended that you retain the default (OFF) unless TIBCO Support requires the output. |

| Keyword           | Operand                  | Description   |
|-------------------|--------------------------|---|
| DEST-RECV         | <i>subject name</i>      | Specifies the subject name to receive the message reply. Overrides DEST-RECV-DEFAULT in a group.  |
| DEST-RECV-DEFAULT | <i>subject name</i>      | Specifies the default subject name if DEST-RECV is not specified. The default is <code>tibss.admin.sxs.reply</code> .   |
| DEST-SEND         | <i>subject name</i>      | Specifies the subject name to which messages are sent. Overrides DEST-SEND-DEFAULT in a group.  |
| DEST-SEND-DEFAULT | <i>subject name</i>      | Specifies the default subject name if DEST-SEND is not specified. The default is <code>tibss.admin.sxs.request</code> .   |
| END               |                          | Denotes the end of the keywords for a message command definition, sometimes referred to as a group.   |
| FLD               | <i>type, name, value</i> | <p>Defines a field for the message.</p> <p>Where:</p> <p><i>type</i> is I32, U32, PST, or STR.</p> <p><i>name</i> is the name of the mapped field.</p> <p><i>value</i> is the data to be associated with the field.</p> <p>The message can contain multiple fields.</p> |
| PSWD              | <i>password</i>          | Specifies the password for the user ID previously specified and authorized to execute this command or process. This keyword is Security Interface dependent and is for the <b>tibss-password</b> system field. The default is blank.                                    |
| PUBLISH-ONLY      |                          | Specifies that the utility only sends messages and does not wait or receive any messages. Be sure to specify a <b>DEST-RECV</b> parameter so that the published message has a <b>reply-to</b> field.  |
| REPEAT            | <i>number</i>            | Specifies the number of times to send this message. The default is 1.   |
| TIMEOUT           | <i>number</i>            | Specifies the time (in seconds) to wait for the response message. The permitted range is 0 to 60 seconds. The default is 5.   |
| UFLD-PFX          | <i>prefix</i>            | Specifies the prefix for all Substation ES system field names. The default is <code>tibss-</code> . It is recommended that you do not change this value for IVP runs.   |

| Keyword        | Operand          | Description  |
|----------------|------------------|--|
| USER           | <i>userid</i>    | Specifies the RACF or back-end user ID for the user who is authorized to execute this command or process. This keyword is Security Interface dependent and is for the <b>tibss-userid</b> system field. The default is a blank.  |
| WAIT           | <i>timevalue</i> | <p>Specifies the time (in seconds) to wait between message or command sends. The default is 0. This is the interval a delay occurs after an END keyword is detected.</p> <p>The time to delay between each message group is determined by the value of <b>WAIT</b> plus the value of <b>WAIT-ON-REPEAT</b>.</p>  |
| WAIT-ON-REPEAT | <i>timevalue</i> | <p>Specifies the time (in seconds) to wait after each send of the command that is being repeated. The default is 0.</p> <p>0 is a special value that means the next message must be published as soon as the current response (or TIMEOUT) occurs.</p> <p>For non-zero values, the process uses this value minus the response time of the previous message (or TIMEOUT). If the calculation is greater than zero, then the process waits for that amount of time before publishing the next message.</p> |
| *              | <i>text</i>      | Displays the comment.  |

## Examples

The section provides the examples related to the Substation ES commands.

### Example 1

The following example shows how to send multiple Substation ES commands to a Substation ES running system.

The following example issues Substation ES commands, refreshes a recipe, waits for five seconds, disconnects the CICS Interface from the CICS region, and waits for 10 seconds. It then repeats the reconnect of the Substation ES CICS Interface to the CICS region three times, waiting for two seconds between each occurrence.

```
USER TIBCO
PSWD password
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD STR,SXS-COMMAND,"REFR,RID=Admin-Recipe"
WAIT 5
END
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD STR,SXS-COMMAND,"DISCONN,INTF=IICICS"
WAIT 10
END
REPEAT 3
WAIT-ON-REPEAT 2
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD STR,SXS-COMMAND,"CONN,INTF=IICICS"
END
/*
```

### Example 2

The following examples show how to send Substation ES commands to different Substation systems.

- Send a show counters command for Substation ID SS4TEST:

```
USER TIBCO
PSWD password
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD PROP,tibss-ss-id,"SS4TEST"
FLD STR,SXS-COMMAND,"SHOW,COUNTERS"
END
/*
```

- Send a show active listeners command for Substation ID SS4PROD:

```
USER TIBCO
PSWD password
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD PROP,tibss-ss-id,"SS4PROD"
FLD STR,SXS-COMMAND,"SHOW,ACT,LSNS"
END
/*
```

### Example 3

When issuing IMS operational commands, display the IMS OTMA status:

```
USER TIBCO
PSWD password
DBG OFF
DEST-SEND tibss.admin.sxi *Issue IMS authorised command*
FLD STR,SXS-COMMAND,"/DIS OTMA"
END
/*
```

### Example 4

Send a simple message to a listening application:

```
DEST-SEND tibss.test.msg
FLD STR,F1-String,"General custom message"
FLD U32,Field-U32-Name,1234
END
/*
```

## TIBCO Enterprise Message Service Generic Sender or Listener Utility

The SXJ3GEN program is in *USERHLQ*.LOAD.

The JCL sample is as follows:

- Data Set: *USERHLQ*.JCL
- Member: SXJGENSL

Before using this utility, you must specify the EMS transport parameters, for example:

```
//SET TIBESB='/-server tcp://your-EMS-Server-URL:7222 '
//SXJGEN EXEC PGM=SXJ3GEN,PARM='&TIBESB'
```

### SYSIN Parameters

The commands or messages sent by the utility are determined by the keywords and operands specified in the SYSIN DD Name file. See the following table.

#### EMS Generic Sender or Listener Control Parameters

| Keyword           | Operand                 | Description  |
|-------------------|-------------------------|--|
| DEBUG DBG         | OFF ON                  | Enables debug facility messages. It is recommended that you retain the default (OFF) unless TIBCO Support requires the output. |
| DEST-RECV         | <i>destination name</i> | Specifies the topic or queue to receive the message reply. Overrides DEST-RECV-DEFAULT in a group.                             |
| DEST-RECV-DEFAULT | <i>destination name</i> | Specifies the default topic or queue if DEST-RECV is not specified. The default is <code>tibss.admin.sxs.reply</code> .        |
| DEST-SEND         | <i>destination name</i> | Specifies the topic or queue to which messages are sent. Overrides DEST-SEND-DEFAULT in a group.                               |
| DEST-SEND-DEFAULT | <i>destination name</i> | Specifies the default topic or queue if DEST-SEND is not specified. The default is <code>tibss.admin.sxs.request</code> .      |
| DEST-TYPE         | TOPIC QUEUE             | Specifies the type of destination to be used.<br>Overrides DEST-TYPE-DEFAULT in a group.                                       |

| Keyword           | Operand                  | Description   |
|-------------------|--------------------------|---|
| DEST-TYPE-DEFAULT | TOPIC QUEUE              | Specifies the default type of destination if DEST-TYPE is not specified. The default is QUEUE.  |
| END               |                          | Denotes the end of the keywords for a message command definition, sometimes referred to as a group.   |
| EMS-USER          | <i>name</i>              | Specifies the user ID with which to connect to the EMS server.  |
| EMS-PSWD          | <i>password</i>          | Specifies the password with which to connect to the EMS server.   |
| FLD               | <i>type, name, value</i> | Defines a field for the message.<br>Where:<br><i>type</i> is I32, U32, PROP, or STR.<br><i>name</i> is the name of the mapped field.<br><i>value</i> is the data to be associated with the field.<br>The message can contain multiple fields. |
| PSWD              | <i>password</i>          | Specifies the password for the user ID previously specified and authorized to execute this command or process. This keyword is Security Interface dependent and is for the <b>tibss-password</b> system field. The default is a blank.        |
| PUBLISH-ONLY      |                          | Specifies that the utility only sends messages and does not wait or receive any messages. Be sure to specify a <b>DEST-RECV</b> parameter so that the published message has a <b>reply-to</b> field.  |
| REPEAT            | <i>number</i>            | Specifies the number of times to send this message. The default is 1.   |
| SSL-TRACE         |                          | Enables EMS SSL trace.  |
| SSL-DEBUG-TRACE   |                          | Enables EMS SSL debug trace.  |
| SSL-FLIPSMODE     |                          | Enables FIPS 140-2 mode.  |
| SSL-AUTH-ONLY     |                          | Specifies that this ESB connection uses SSL-only for authentication.  |
| SSL-CIPHERS       | <i>names</i>             | Specifies the IBM cipher codes to be used for encryption.   |
| SSL-SSLV3         |                          | Enables SSL V3.   |
| SSL-TLS1.1        |                          | Enables TLS1.1  |

| Keyword              | Operand          | Description   |
|----------------------|------------------|---|
| SSL-EXPECTEDHOSTNAME | <i>name</i>      | The name of the EMS server name that is being interfaced with.  |
| SSL-KEYRINGFILE      | <i>name</i>      | Specifies the Resource Access Control Facility (RACF) ring name to be used.   |
| SSL-LABEL            | <i>name</i>      | Specifies the RACF identity name.   |
| SSL-VERIFYHOSTNAME   |                  | Enables host verification.  |
| SSL-PKPASSWORD       | <i>name</i>      | Specifies the PK PASSWORD.  |
| TIMEOUT              | <i>number</i>    | Specifies the time (in seconds) to wait for the response message. The default is 5.   |
| UFLD-PFX             | <i>prefix</i>    | Specifies the prefix for all Substation ES system field names. The default is <code>tibss-</code> . It is recommended that you do not change this value for IVP runs.   |
| USER                 | <i>userid</i>    | Specifies the RACF or back-end user ID for the user who is authorized to execute this command or process. This keyword is Security Interface dependent and is for the <b>tibss-userid</b> system field. The default is a blank.   |
| WAIT                 | <i>timevalue</i> | <p>Specifies the time (in seconds) to wait between message or command sends. This is the interval a delay occurs after an END keyword is detected. The default is 0.</p> <p>The time to delay between each message group is determined by the value of <b>WAIT</b> plus the value of <b>WAIT-ON-REPEAT</b>.</p> |
| WAIT-ON-REPEAT       | <i>timevalue</i> | Specifies the time (in seconds) to wait after each send of the message or command that is being repeated. The default is 0.   |
| *                    | <i>text</i>      | Displays the comment.   |

### Example

Refresh a recipe, wait for three seconds, and send a request or reply:

```

DBG OFF
USER VERNON
PSWD password
DEST-SEND tibss.admin.sxs
DEST-RCV tibss.admin.reply
FLD STR,SXS-COMMAND,"REFR,RID=SXC-DPL-C"
WAIT 3
END
DEST-SEND tibss.CICS.C.Request
DEST-RCV tibss.CICS.RR.Reply
FLD STR,IVP-ID,"SXCIV01 "
FLD STR,IVP-DESCR,"Substation ES - EMS test CICS C"

```



```

FLD STR,IVP-SUBJ-SFX," "
FLD I32,IVP-START-LILSECS,0
REPEAT 1
END
/*

```

## Generic Error Listener Utility

The Substation ES Generic Error Listener Utility receives and prints Application Notification messages published by Substation ES. The sources SXJ3IELS and SXR3IELS are in the *USERHLQ.C* data set.

### Functions

The Substation ES Generic Error Listener Utility performs the following tasks:

- Receives and prints Substation ES data transformation notification messages.
- Receives and prints Substation ES operational and execution notification messages.

The utility listens to subjects or destination names you provide on the SYSIN DD Name. The name must correspond to the subject name on the recipe or trigger definition panel in the **Error Subject** field.

When an error or user notification is delivered during execution, a message is written to the subject defined in the **Error Subject** field. Even though these messages are processed by your application, the Substation ES Generic Error Listener Utility can serve as an alternate method for listening to the subjects and for printing the associated notifications forwarded by Substation ES.

### JCL Sample

The sample JCL is as follows:

- Data Set: *USERHLQ.JCL*
- Member:
  - SXRRIELS (Rendezvous)
  - SXJRIELS (EMS)

Before using this utility, you must specify the Rendezvous or TIBCO Enterprise Message Service transport parameters.

Also, review and select the appropriate Generic Error Listener Utility Control Parameters. See [Generic Error Listener Utility Control Parameters](#).

### SYSIN Parameters

The messages to which the Substation ES Generic Error Listener Utility listens are determined by the keywords and operands in the SYSIN DD name file. See the following table.

#### *Generic Error Listener Utility Control Parameters*

| Keyword     | Operand           | Description   |
|-------------|-------------------|---|
| SUBJECT     | {tibss.error.>}   | Rendezvous only. These are the error subject names.   |
| QUEUE/TOPIC | {tibss.error.out} | EMS only. This is the destination to which an error message is published if an ESB conversion or back-end system encounters errors. |

| Keyword  | Operand  | Description  |
|----------|----------|--|
| PSWD     |          | EMS only. This is the password associated with the user ID if a special identification is to be used in the session.                                   |
| UFLD-PFX | {tibss-} | This is the prefix of the Substation ES system field name for the processing status. It is recommended that you do not change this value for IVP runs. |
| USER     |          | EMS only. This is the user ID specified by you if a special identification is to be used in the session.   |

The examples are listed as follows:

- **Rendezvous**

```
subject    my-error-subject
subject    my-errors.>
subject    substation.test.errors.*.subject
```

- **TIBCO Enterprise Message Service**

```
queue      my-error-output
topic      substation.test.errors (Applies to topic only.)
```