

TIBCO Substation ES™

Operations and Administration

Software Release 2.10
May 2016

Important Information

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

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

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

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

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

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

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

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

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

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

TIBCO Software Inc. Confidential Information

Contents

Figures	6
TIBCO Documentation and Support Services	7
Operational Commands	9
Substation ES Console Interface	9
MVS Command Interface	9
Overview of Operational Commands	9
Command Summary	11
Display Commands	14
Update Commands	21
Initialization Commands	24
Control Commands	26
Substation ES Operations	29
Auto Recovery of Messages When EMS is the ESB	29
System Property Fields	29
Processing of Property and System Fields	30
Substation ES HeartBeat Events	30
Writing Substation ES Messages to SYSLOG	31
Sites with Multiple Stacks	32
Detecting Network Connection Failures with EMS ESB	32
Sending Variable Length Messages	32
Sending Variable Length Messages Using TIBCO EMS	32
Sending Variable Length Messages Using TIBCO Rendezvous	33
Exceptional Cases	35
CICS Operations	36
Substation ES CICS Region Resources	36
Substation ES in a CICSplex Environment	40
Methods of Invocation	41
Services without Conversion Rules	43
Recipe Service	43
Trigger Service	45
Containers and Processes	47
Multiple Containers	48
Single Containers	48
Invoking Multiple Resources	49
Using FastPath Trigger Processes	51
Defining a FastPath Trigger	51

Endpoint Configuration for Rendezvous ESB	51
Endpoint Configuration for EMS ESB	52
Invocation of CICS Programs to Access DB2	53
DB2 Access Considerations	53
DB2Entry and DB2Tran for Programs That Access DB2	54
Security Considerations	54
Scenarios of CICS DB2 Transaction Definition	55
Trigger and Recipe Performance	56
CICS Request/Reply	58
CICS HVT Triggers	59
Usage Samples	59
Substation ES Trigger Flow	59
Ordered Message Delivery	60
Configuration	61
Code Samples: Link to SXCQWRIT or SXCQWTSQ	61
SXCQWRIT Parameter Fields	62
HVT Trigger Transactions	63
CICS Trigger Queue Verify Task	64
Sample Output for SXTV Commands	64
CICS 3270 Assistance Task	65
SXTH Command-line Parameters	65
Sample Output for SXTH Commands	66
CICS 3270 Dead Message Queue Task	67
Sample Output for SXTD Commands	67
CICS Security Settings	68
CICS Transaction Security for Substation ES	68
Definitions of Resource Access Control Facility	69
Maximum EXCI Sessions	69
Operations Cookbook	70
RED for CICS Operations	73
Overview of the Interface for RED	73
RED Initialization for CICS	73
CICS Transactions Using RED	74
Setting Up a Recipe for CICS RED	75
Detailed Flow of CICS Transaction	75
RED CICS Triggers	76
Trigger Message Flows in CICS	77
RED CICS Request/Reply	78
Request/Reply Message Flows in CICS	79

RED CICS Control Information Viewer	80
XTUV Command-line Parameters	80
Sample Output for XTUV, TRG, INFO Command	81
Sample Output for XTUV, REAL Command	82
IMS Operations	84
IMS Security Settings	84
Setup of OTMA Security Level	84
Defining RACF for IMS Security	86
IMS Security Exits	86
IMS Transactions	87
IMS Events or Triggers	87
General Trigger Processing	87
Synchronous Callout (ICAL) or Request/Reply Support	87
Processing of IMS OTMA Trigger Errors	88
Exceptional Cases	88
Logging, Tracing and Auditing	89
Introduction	89
Features	89
Disk Logging	90
LTA Disk Files	90
Setup of LTA Disk Files	94
LTA Services Utility	95
Features	95
JCL Sample (SXGLINIT and SXGLUTIL)	96
Control Parameters	96
Keyword Phrases	97
Substation ES Utilities	99
Rendezvous Request-Reply Timer Utility	99
Operational Commands	99
Control Parameters	100
Examples of Control Statements	101
Generic Sender or Listener Utility	104
Rendezvous Generic Sender or Listener	105
Examples	107
TIBCO Enterprise Message Service Generic Sender or Listener Utility	108
Generic Error Listener Utility	111

Figures

Architecture in a CICSplex Environment 41

Trigger Flow 60

File Selection Panel 91

LTA Multiple Entry Panel 92

LTA Message Pop-up Panel 93

TIBCO Documentation and Support Services

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

<https://docs.tibco.com>

Product-Specific Documentation

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

`TIB_substation_version_docinfo.html`

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

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

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

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

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

How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

How to Join TIBCO Community

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

<https://community.tibco.com>

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.

MVS 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 MVS 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 *MVS 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 MVS 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
HELP	[, sub-command]
SHOW,	ACTIVE ACT, LISTENERS LSNS [, INTF IID=intf-id]
	ACTIVE ACT, RECIPE RID [, INTF IID=intf-id]
	ACTIVE ACT, TRIGGER TID [, INTF IID=intf-id]
SHOW,	CICS, "CEMT Sub-command" [, INTF IID=intf-id]
	CICS, SXTH [, Sub-command] [, INTF IID=intf-id]
SHOW,	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]
SHOW,	EMS, STATS, RECIPE RID=name [, INTF IID=intf-id]
SHOW,	ESB TPORT [, INTF IID=intf-id]
SHOW,	IMS, BUFFER
	IMS, STATS
	IMS, CONNECT
	IMS, CONNAME
SHOW,	INTF IID
	INTF IID=name
	INTF IID [, CONNECT CONN]
	INTF IID=name [, CONNECT CONN]
SHOW,	LOGSTATS
	LOGSTATS, ALL

Command	Keyword
	LOGSTATS , DETAIL
SHOW,	LTAVERS
SHOW,	QUE
	QUE , LOCK
SHOW,	RED , BUFFER
	RED , LIMIT
SHOW,	RECIPE RID TRIGGER TID=name , INTF IID=intf-id
SHOW,	RV , DISPQ
	RV , DQ-DISPQ
SHOW,	SSHOT
	SSHOT , DETAIL
	SSHOT , ENTRY
SHOW,	SSID
SHOW,	SUSPENDED
SHOW,	SYS [, DEBUG]
	SYS , LOCK
SHOW,	TRCSTATS
	TRCSTATS , ALL
	TRCSTATS , DETAIL
SHOW,	UOW

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]

Command	Keyword
	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]
	TRIGGER TID=name [, INTF IID=intf-id]
	COUNTERS
	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=tport-id , INTF IID=intf-id
DISCONNECT DISCONN,	INTF IID=intf-id
	APPLID=applid
	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 , 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]

Command	Keyword
SET,	IMS , HOLD-TRIG[, INTF IID= <i>intf-id</i>]
SET,	IMS , REL-TRNAK[, SXITRNAK DFS\$\$TOQ]
SET,	LOGDEST= <i>value</i>
SET,	LOGLVL= <i>value</i>
SET,	LOGFWRITE= <i>value</i>
SET,	RED[, TRCLVL-AMSG TRCLVL]
SET,	TRACE-ENTRY=CONFIG , DETAIL
SET,	TRCDEST= <i>value</i>
SET,	TRCLVL= <i>value</i>
SET,	TRCLVL= <i>value</i> [, OTMA OPS RR]
SET,	TRCFWRITE= <i>value</i>
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 active 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 3270 results on screen.

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:

```
SXA2601I Console Command '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, CONNAME

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 TIBCO Enterprise Message Service™ 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, 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

This command initiates a snapshot and ends only when the preset value is reached. 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

This command has the following functions:

- Initiates a snapshot and ends only when the preset value is reached. 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

This command initiates a snapshot and ends only when the preset value is reached. 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
16.924783    2008/03/29-15:59:07.846085  2008/03/29-15:59:24.770868

#:1000   Minimum   Maximum   Total      Average Recorded
UoW      0.020000   0.531226   203.060447   0.203060
BES      0.005930   0.521352   175.132541   0.175132
WRK-In   0.000016   0.008004   0.383973     0.000383
WRK-Out  0.003364   0.085766   8.129159     0.008129
ESB      0.002431   0.076503   5.214009     0.005214

Entry# UoW      Start      End
#00000 1218811218 2008/03/29-15:59:07.846085 2008/03/29-15:59:08.030790
          TRG  WRK 15:59:08.019095 15:59:08.019355 Dur:0.000260
          BES 15:59:07.846086 15:59:08.019095 Dur:0.173009
          TRG-Out WRK 15:59:08.019419 15:59:08.027033 Dur:0.007614
          ESB 15:59:08.021915 15:59:08.026729 Dur:0.004814
          00000 Complete UoW:0.184705
#00001 663466177 2008/03/29-15:59:07.847841 2008/03/29-15:59:08.051230
          TRG  WRK 15:59:08.043346 15:59:08.043412 Dur:0.000066
          BES 15:59:07.847841 15:59:08.043346 Dur:0.195505
          TRG-Out WRK 15:59:08.043542 15:59:08.051173 Dur:0.007631
          ESB 15:59:08.047732 15:59:08.050923 Dur:0.003191
01      Complete UoW:0.203389
```

SHOW, SSID

This command shows the Substation ID.

The command output is as follows:

```
SXA2601I Console Command 'SHOW,SSID'
          Substation ES ID:SSES#4#IVP
```

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
```

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 a 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 Enable, 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 Enable, 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,*	
RECOVER*,FOR=CICS	Recover is for all recipes with BES referencing CICS.
RECOVER*,XYZ	XYZ means nothing.
RECOVER*,IID=ESB1	Recover is sent only to interface ESB1.
RECOVER,RID=ABC	
RECOVER,RECIPE=ABC	
RECOVER,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]

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 command output is as follows:

```
SXA2601I Console Command Entered 'REFRESH,RID='SXC-RR-REPLY''
SXT5811I Recipe SXC-RR-REPLY refreshed - still for
tibss.ivp.20.request
```

REFRESH|REFR, 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 refresh. The optional keyword **INTF|IID=intf-id** specifies that the trigger is to be refreshed with the specified ESB interface identifier.

The command output is as follows:

```
SXA2601I Console Command Entered 'REFRESH,TRIGGER=SXC-IVP-TRIGGER'
SXT5487I Trigger SXC-IVP-TRIGGER refreshed
```

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 already loaded in the Substation ES; include any disabled recipe or trigger will become enabled. 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.

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 with 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 MVS 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, TRCLVL-HVS=*value* [, **INTF|IID=***intf-id*]

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

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 are 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 queued 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 centi seconds (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[, TRCLVL-AMSG | TRCLVL]

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

- TRCLVL-AMSG: the RED interface for AMSG related messages. The range of the valid values is from 1 to 3.
- TRCLVL: the RED interface for general messages.

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 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 centi seconds (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 disks 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 operations functions include the following ones:

- [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

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
ESB-INTF-ID = <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.
HEARTBEAT-SERVICE = <i>T-Name</i>	Specifies the trigger service name that Substation ES uses to publish HeartBeat messages.
HEARTBEAT-INTERVAL = <i>seconds</i>	The interval, in seconds, between heartbeats. For example: HEARTBEAT-INTERVAL =5 To deactivate the HeartBeat feature, specify an interval of 0.
HEARTBEAT-COMMAND = <i>command</i>	Specifies a Substation ES operational command that is issued during each HeartBeat event. You can specify a maximum of five commands. For example: 000070 HEARTBEAT-COMMAND = SHOW ,INTF,CONN 000071 HEARTBEAT-COMMAND = SHOW ,COUNTERS 000072 HEARTBEAT-COMMAND = SHOW ,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 in the *USERHLQ.MSGS(SXT56)* 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..MSG
/**

```

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*.

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**.

```
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 Buffer Definition, verify that the buffer offset 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
```

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 ==> 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
```

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 Buffer Definition, verify that the buffer offset 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	

3. In Buffer Field Details, set the decimal place to 1.

----- Buffer Field Details ---
 Command ==>

Field Name ==> IVP-OPAQUE

Field Type ==> OPA (STR, ...)

Buffer Start ==> 0

Buffer Length ==> 16384

Decimal Places ==> 1 (0-6)

Pad Character ==> or hex()

Separate Sign ==> N (N/Y)

Leading Sign ==> N (N/Y)

Justify Right ==> N (Y/N)

Initial Value ==>

Initial Value Length ==> 0

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	Reqd	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)

Sel	Flds	Bufsz	Max	EMS	S	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 IBM 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.

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.

The following table lists and describes CICS connections:

CICS Connections

Connection	Description
SXCA	The Administrative server connection.
SXCD	The Dynamic Program Link (DPL) server and application program connection.
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 of less than 16 MB. This task is typically used for older CICS programs that are nonreentrant. You can set the task on a recipe definition.
SSEX	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
Administrative	
SXCOSRVA	<p>The Substation ES Administration server, which is called under the following conditions:</p> <ul style="list-style-type: none"> • At Substation ES startup for a sanity check • During trigger processing • At CICS shutdown • At Substation ES termination
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 Substation ES Operations for detailed information.
SXCOERR	The internal Substation ES CICS program that writes Substations CICS region errors to an error queue file.
Server	
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.
Request or Reply	
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.

Program	Description
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.
IO Routine	
SXCQWRIT	The Substation ES HVT routine that is called if you must write a trigger record to the HVT queues.
CICS	
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.
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 and SXQ1 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 queue that holds 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

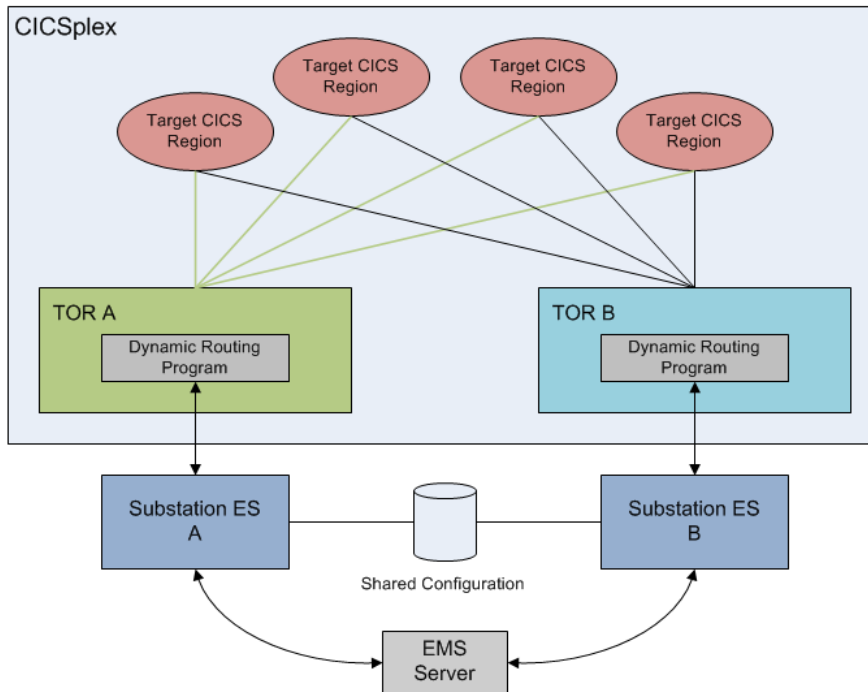
Workload Manager (WLM) of CICSplex SM uses the dynamic routing program EYU9XLOP to route eligible work requests from a requesting region to a suitable target region selected at the time the transaction is initiated.

Together with the EYU9XLOP 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> <p>Issues a start transaction with DFHCOMMAREA.</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, and uses a 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-Ownning 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 Alt BES(?):
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 a special handle. 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 else 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              OCCURS 10 TIMES
    05 MCTF-NAME           PIC X(16)      VALUE SPACES.
    05 MCTF-TYPE           PIC X(8)       VALUE SPACES
    88 MCTF-STRING         VALUE 'STRING'.
    05 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 SXCCDQW.

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(?) =>
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 lists:
 - 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      Alt BES(?):
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 Trace Level for your desired level of debugging.
8. Specify Output Method of either 2 or 3.
9. Modify Compare Text as needed.
10. Modify Undelivered Requeue as needed.
11. 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 based 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 based 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 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 based 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, Sec**
- **Reply To**
- **Publish To Len (Dyn)**
- **Compression**
- **Preserve**
- **No Body Trace**

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 to use a DB2ENTRY object, do one of the following list:

- 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 that 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

You can have a general understanding for 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:

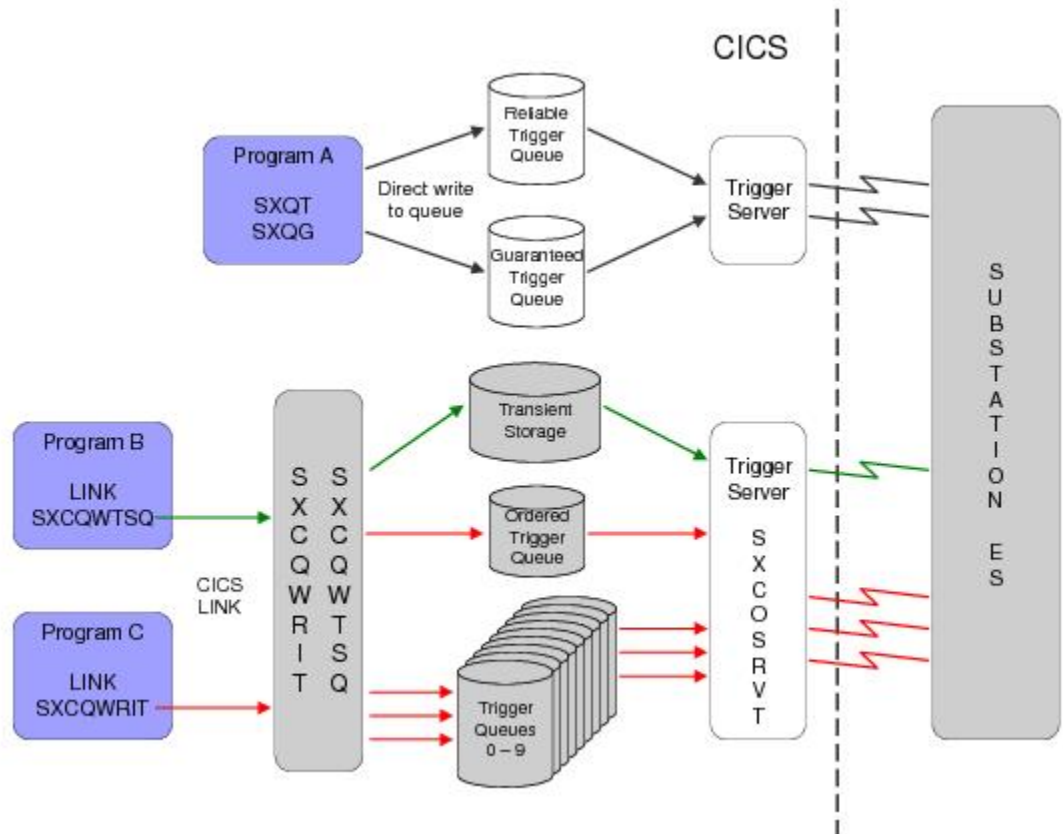
- The COBOL program SXCCI032: the associated copybook is SXCCTDQW. The transaction SXTF invokes this program.
- The C program SXC3I032: the associated header file is SXCHTDQW. The transaction SXT6 invokes this program.

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 (CICS TS 3.1 and above) 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 since 3.1 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 (CICS TS 3.1 and above) 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
TDQW-SERVICE	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.
TDQW-TSQ-Name	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.
TDQW-DATALEN	PIC S9(8) COMP	This field is the length of the data in the case of a data pointer. Otherwise, the value must be zero.
TDQW-PDATA	POINTER	This field is the pointer to the data or NULL (binary zeros). Data Pointers cannot be used for the MRO processing because the address is local to Address Space.
TDQW-CONTAINER	PIC S9(8) COMP	For TSQ data, this field is set to zero or FALSE. For a Container name, this field is set to one or TRUE.
TDQW-TDQ-TYPE	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.
TDQW-RETURN	PIC 9(8) 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.
TDQW-REASON	PIC 9(8) 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.
TDQW-RESP1	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.
TDQW-RESP2	PIC 9(8) 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
TDQW-ROLLBACK-SW	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.
TDQW-TDQ-NAME	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
SXTF/SXT6	Sends one message through HVT in IVP format.
SXTF/SXT6,HELP/?	Lists all the keywords for the available commands.
SXTF/SXT6,type(G R O S),number_of_records	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.
SXTF/SXT6,type(G R O S),number_of_records,data_length(1 - 4096000)	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. The maximum data length for the TSQ Trigger is 32000.
SXTF/SXT6,type(G R O S),number_of_records,data_length(1 - 4096000)service_name	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. The maximum data length for the TSQ Trigger is 32000.
SXTF/SXT6,Type(G R O S),number_of_records,data_lengths(1 - 4096000),service_name,#-CTN	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. The valid <i>type</i> values are G for Guaranteed, R for Reliable, O for Ordered, and S for Temporary Storage Trigger. The maximum data length for the TSQ Trigger is 32000. The valid range for #-CTN is 1 to 20.

Command	Description
SXTF/SXT6 , <i>Type</i> (G R O S), <i>number_of_records</i> , <i>data_lengths</i> (1 - 4096000), <i>service_name</i> , #-CTN, <i>delay</i>	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, 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. The maximum data length for the TSQ Trigger is 32000. The valid range for #-CTN 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
SXTV, HELP/?	Lists all the keywords for the available commands.
SXTV, queue-name, number	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.
SXTV, TSQ, queue-name	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.
SXTV, SHOW, SSSES	Displays the first 336 characters of the Substation ES anchor record.
SXTV, SHOW, HVT	Displays the first 336 characters of the Substation ES High Volume Trigger record.
SXTV, TESTHVT	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
SXTH	Displays a statistics summary for the Substation ES CICS interface queue resources.
SXTH, ALL	Displays a statistics summary for the Substation ES CICS interface queue resources.
SXTH, DEF	Displays the default information of the Substation ES CICS interface queue resources.
SXTH, ERR	Displays the details on the first error record from the error queue.
SXTH, ERR, DELETE	Displays the details on the first error record from the error queue and then deletes the record from the queue.
SXTH, ERR, PURGE	Deletes all the records from the error queue.
SXTH, HELP/?	Lists all the keywords for the available commands.
SXTH, HVT	Displays the concurrent activity of the HVT queues.
SXTH, TSQ	Displays the concurrent activity of the HV TSQ triggers.
SXTH, PTDQ, TDQ <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.
SXTH, PTSQ, TSQ <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.
SXTH, QUE	Displays the usage information on the default Trigger Queue.
SXTH, QUE, TDQ <i>Name</i>	Displays the queue usage information on the given TD queue name.

Command	Description
SXTH, TRG	Displays the concurrent active statistics for the triggers and CICS initiated request or reply processes.
SXTH, VER	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**

```
Interface - Status:Active, Id:SS-KR-2-6-IVPs-IIICICS
Trigger(SXQ/G) - Status:Active, Busy:No - (R) Status:Active, Busy:No
Trigger(HVT/G) - Status:Active, Ques:3 - (R) Status:Active, Ques:2
CICS Req/Reply - Status:Active, Busy:No - (O) Status:Inact , Busy:No
R/R usage - In Use:0, HWM (Curr,Max):0,1
```

- **SXTH, HVT command**

```
HVT Activity - Inact/Posted/Act Q-Name Offload
- Status:Active , Id: SXQ0 1-0-0
- Status:Active , Id: SXQ1 1-0-0
- Status:Active , Id: SXQ2 1-0-0
- Status:Active , Id: SXQ3 1-0-0
- Status:Active , Id: SXQ4 1-0-0
- Status:Active , Id: SXQ5 1-0-0
- Status:Active , Id: SXQ6 1-0-0
- Status:Active , Id: SXQ7 1-0-0
- Status:Active , Id: SXQ8 1-0-0
- Status:Active , Id: SXQ9 1-0-0
- Status:Active , Id: SXQW 1-0-0
- Status:Active , Id: SXQX 1-0-0
- Status:Active , Id: SXQY 1-0-0
- Status:Active , Id: SXQZ 1-0-0
TDQ Server: SXCOSRVT - Res:1, Use:7692064, Status:ENABLED
```

- **SXTH, TSQ command**

```
TSQ Rel Trig: - Records:0
- Next Wr:0
- Next Re:0
TDQ Server: SXCOSRVT - Res:0, Use:0, Status:ENABLED
```

CICS 3270 Dead Message Queue Task

Substation ES provides SXTD, a CICS 3270 task that performs the certain functions.

The certain functions are listed 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
SXTD	Displays the first record on the DMQ.
SXTD,HELP/?	Lists all the keywords for the available commands.
SXTD,record number	Displays the number of the DMQ record.
SXTD,DEL	Deletes the first record in the DMQ.
SXTD,PURGE	Deletes all the records in the DMQ.
SXTD,RESEND	Sends the first record group on the DMQ to SXCQWRIT for resending to Substation ES.
SXTD,RESEND,ALL	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

The 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.

RED Initialization for CICS

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

The SXXFINIT program is issued from SXX3ADMIN 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 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.

Listener transactions in CICS, run by default under the #RED transaction and might execute concurrently. The number of concurrent #RED transactions are 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 is 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 are 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 SXCORED.A).

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.



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.

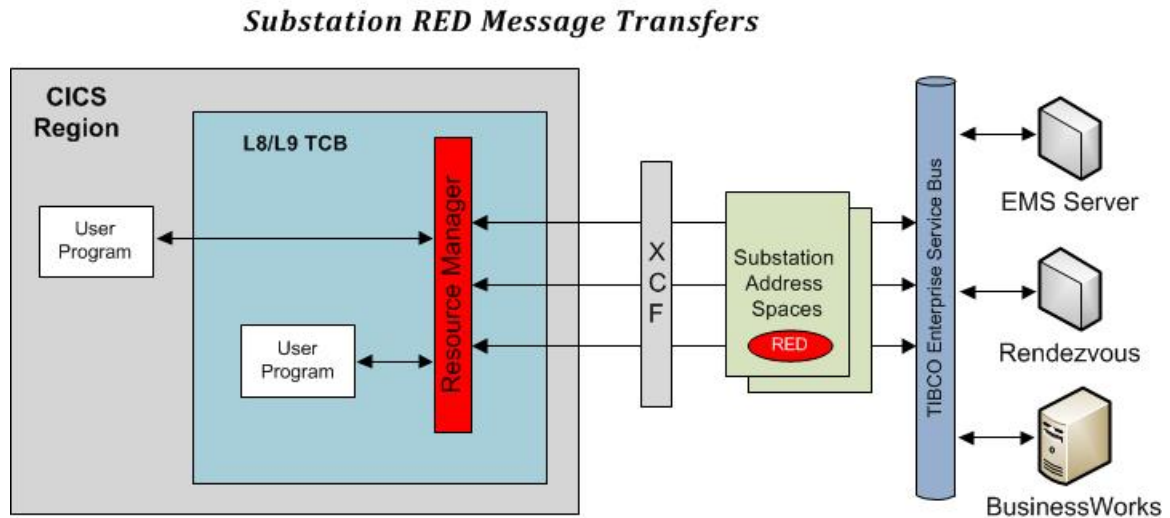
Detailed Flow of CICS Transaction

You can use RED to transact CICS.

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 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 transactions concurrent 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 listed 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. 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.
5. User application reviews a return code and a reason code for any error indication.
 6. If no errors, the response data is present as indicated by the parameters.
 7. 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 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.
 8. 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.

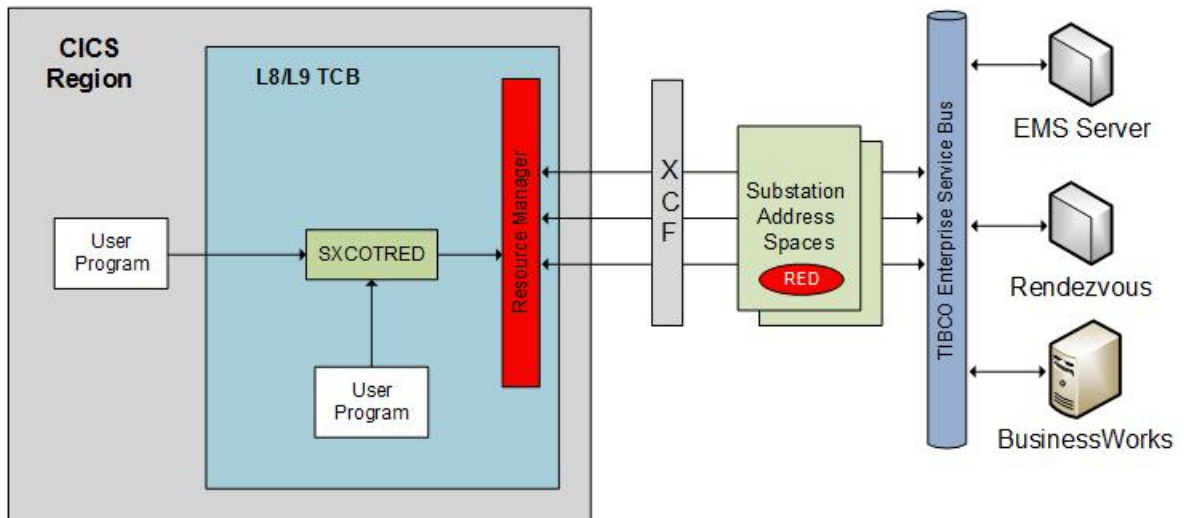
Trigger Message Flows in CICS

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. 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.
6. **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:

Substation RED Trigger Message Transfers



RED CICS Request/Reply

CICS RED provides a server module that 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

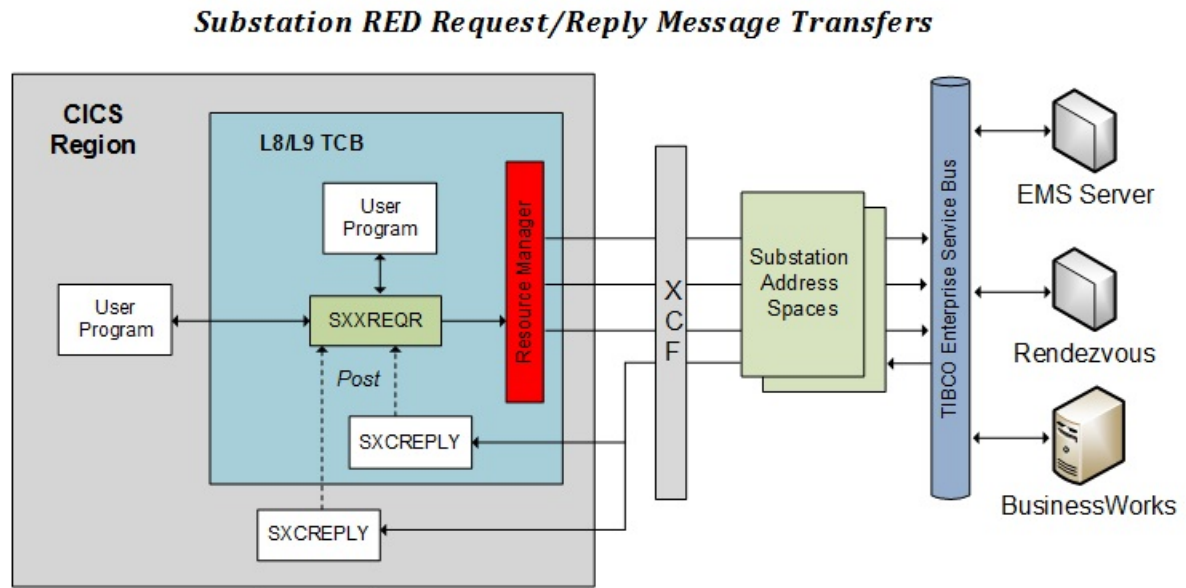
The flow is listed as follows:

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 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. 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 forward it to its destination, and then gets reply and returns it to the waiting **SXXREQR** program through a Recipe which uses program **SXCREPLY**.
6. **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
XTUV	Displays the Substation ES RED interface main storage information in CICS.
XTUV , CICS	Displays information about the Substation ES RED CICS main storage.
XTUV , command , PRINT	Displays the details on any command and print the output to CEEOUT.
XTUV , DUMP , hex-address , length	Displays the details of a specific address storage for diagnostic information.
XTUV , ERR , #	Displays the details of an error record.  The hash sign (#) is the record number of errors in the error queue.
XTUV , ERR , DELETE	Displays the details on the first error record from the error queue and then deletes the record from the queue.
XTUV , ERR , PURGE	Deletes all the records from the error queue.
XTUV , HELP ?	Lists all the keywords for the available commands.
XTUV , REAL RT	Displays real-time CICS RED counters and other values information.
XTUV , REC	Displays RED CICS receiver storage information.
XTUV , REC , INFO	Displays RED CICS receiver information.
XTUV , RED	Displays the RED main storage.
XTUV , RED , INFO	Displays the RED main information.
XTUV , STATS	Displays the RED statistics for CICS Listener.

Command	Description
XTUV , TRG	Displays all the pools and members established by the Substation ES RED interface.
XTUV , TRG , DUMP	<p>Displays the CICS RED trigger storage used by SXCOTRED.</p> <p>The RC and RSN correspond to the URL for the RED JOIN processes.</p> <ul style="list-style-type: none"> • a hyphen (-) indicates that the member has never been used. • 0 indicates that the JOIN completed successfully. • Any other value indicates an error in the member. <p>If the data continues to additional screens, press the PF7 and PF8 keys to navigate the screens.</p>
XTUV , TRG , ERR	Displays those members with non-zero values in the RC and RSN for the RED interface. If the data continues to additional screens, press the PF7 and PF8 keys to navigate the screens.

Sample Output for XTUV, TRG, INFO Command

The following one is an example of the output generated by the **XTUV , TRG , INFO** command:

XTUV, TRG, INFO command

```
Location 1FC1F590 Size:5516 Status:1
Group Name:TIBQA Max Len:60000 Rep-TO:5000 XMC-TO:10
Pool 1 Max Size 600 Listener:P1AMSG
Members:10 Pacing:300 Retry:0 Last:6 Lost:0 Sent:0 Error:0 Join:0
Pool 2 Max Size 8192 Listener:P2AMSG
Members:02 Pacing:500 Retry:0 Last:22 Lost:0 Sent:0 Error:0 Join:0
Member 01 St:0 RC:- RSN:- URL:TIBQA.C512WW3G#01.P1AMSG#01
Member 02 St:0 RC:- RSN:- URL:TIBQA.C512WW3G#02.P1AMSG#02
Member 03 St:0 RC:- RSN:- URL:TIBQA.C512WW3G#03.P1AMSG#03
Member 04 St:0 RC:- RSN:- URL:TIBQA.C512WW3G#04.P1AMSG#04
Member 05 St:0 RC:- RSN:- URL:TIBQA.C512WW3G#05.P1AMSG#05
Member 06 St:0 RC:- RSN:- URL:TIBQA.C512WW3G#06.P1AMSG#06
```

Field Descriptions for Output of XTUV,TRG,INFO

Field	Description
Status	<p>This field indicates the availability of the RED Trigger program SXCOTRED. Valid values are 1 and 0.</p> <p>A value of 0 indicates an error, and any user program that links to SXCOTRED receives an error.</p>
Group Name	The field value shows the group name that the RED interface uses to communicate.
Max Len	The value shows that the AMSG-MAXLEN value that the RED interface uses as the maximum message length. If the application program attempts to send data of longer length through SXCOTRED , it fails.

Field	Description
Rep-TO	The wait time in milliseconds that the guaranteed Trigger process to wait for the response from Substation ES.
XMC-TO	The maximum time in seconds that an XCF send trigger message operation is given to complete successfully.
Max Size	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 Max Size field but smaller or equal to the value in the field, RED handles the message differently, resulting in processing overhead.
Listener	The target member name that the RED interface uses to publish trigger messages.
Members	The number of concurrent message worker threads that the RED interface started to service trigger messages.
Pacing	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.
Retry	The times SXCOTRED attempts to resend a reliable and guaranteed trigger message if the first send fails.
Last	The worker thread most recently used to send a trigger message.
Lost	The total number of reliable confirm and guaranteed trigger messages that have been resent after receiving a timeout error (with RED reason code 8110).
Sent	The total number of trigger messages that have been sent. This count includes any resend messages following a failure.
Error	The total number of trigger messages that were not sent because of an error.
Join	The total number of RED API JOIN calls that have been done.

Sample Output for XTUV, REAL Command

The following one is an example of the output generated by the **XTUV,REAL** command:

XTUV, REAL command

```

20:00:01.149
Receivers Active      : 9
Receivers started    : 887
Current Msgs waiting : 0
High #Msgs waiting   : 3
Messages Received     : 1262
Messages Replied to  : 1262
Messages in Error     : 0
Headers invalid       : 0
Abends detected       : 0
Restart Limit         : 5
Trace Level           : 0

```

Field Descriptions for output of XTUV, REAL

Field	Description
Receivers Active	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 CICS-MAIN-MAX in the RED interface.
Receivers started	Total number of CICS Start transactions issued to execute concurrent receivers.
Current Msgs waiting	Current number of messages waiting in the RED pool that still needs to be processed.
High #Msgs waiting	Highest count of messages that are waiting in the RED pool to be processed.
Messages Received	The total number of messages received.
Messages Replied to	The total number of replies sent if it is required.
Messages in Error	The total number of messages in error.
Headers Invalid	Total number of messages received where the header information might have been invalid, incorrect or not received with a message. All messages received by #RED require a header portion.
Abends detected	The total number of #RED transactions or user programs that abended.
Restart Limit	A decrementing count of allowable #RED restarts remaining. The value is decremented after an abend is detected and is used to stop recursive abends. The initial value is 5. Disconnect and connect your RED interface to have this value reset.
Trace Level	The active RED interface on-line trace level (0-5).

IMS Operations

You can learn about the Substation ES capabilities from the 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	OTMASE=N	/SEC OTMA NONE
	OTMASE=P and the security flag value N.	/SEC OTMA PROFILE In the profile, the security flag value is N.
CHECK	OTMASE=C	/SEC OTMA CHECK
	OTMASE=P and the security flag value C.	/SEC OTMA PROFILE In the profile, the security flag value is C.
FULL	OTMASE=C	/SEC OTMA FULL
	OTMASE=P and the security flag value F	/SEC OTMA PROFILE In the profile, the security flag value is F.
PROFILE	OTMASE=P	/SEC OTMA PROFILE



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 12 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 onto 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 roll backs 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.

If you are using IMS version 10 or higher, 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 items:

- 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

You can have 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:

```
15:29:21.3234 SXA2601I Console Command
'REFR,RID=' SXC-DPL-COBOL-START-TRAN-LONG-
```

- In full with additional identifying information, if available, for example:

```
15:25:27.3478 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.

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

Log Viewer

Command --->

ISPF Log:

Project _____

Group _____

Type _____

Other VSAM Cluster Name:

Cluster Name . . 'TIBCO.SXS.LOG.DISKF1'

Display Format ---> 1 (1, 2)

Direction ---> B (F-Forward B-Backward)

Active Log? ---> Y (Y/N)

Number to Read ---> 200 (Number to read initially)

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

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

Enter END command to terminate.

The following table describes the fields in this panel.

File Selection Panel Fields

Field	Description
ISPF Log	This field specifies a log to view either by providing values for Project , Group , and Type ; or by specifying a fully qualified cluster name in the Other VSAM Cluster Name field. If this field is nonblank, the value for the Other VSAM Cluster Name field is in effect.
Display Format	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.
Direction	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.
Active Log	If the value is Y, the log is treated as being active (currently being updated by Substation ES). If the log is active, the index record is reread for each screen refresh to obtain the most current records.
Number to Read	This field specifies the number of recorded entries to display. If you do not specify a value, the default of 200 is in effect.
VSAM Debug	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).
Pgm Debug	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

Command ---->				Scroll --> CSR		
Sel	Time	Type	Msg Id	RC	Rsn	Message
-						
	17:28:41.6576	154	SXG2806I	0	0	Activated ~ Substation ES for IVP's
	17:28:41.6480	154	SXG2806I	0	0	Activated ~ Substation ES - CICS Inter
	17:28:41.6475	154	SXG2806I	0	0	Activated ~ Substation ES - Transforme
	17:28:41.6473	154	SXG2806I	0	0	Activated ~ Substation ES - Admin & Op
	17:28:41.6337	154	SXG1801I	0	0	Parameter Log closed
	17:28:41.5860	156	SXC3400I	0	0	CICSTS2G - Started CICS EXCI Session (
	17:28:41.5678	156	SXC3400I	0	0	CICSTS2G - Started CICS EXCI Session (
	17:28:41.5319	154	SXS1001I	0	0	Initialized ~ Substation ES for IVP's
	17:28:41.5311	154	SXG2801I	0	0	Initialized ~ Substation ES - CICS Int
	17:28:41.5252	154	SXC3001I	0	0	CICSTS2G - Connect IICICS to CICS Regi
	17:28:41.3869	154	SXC3000I	0	0	CICSTS2G - Connect IICICS to CICS Regi
	17:28:41.3175	154	SXG2800I	0	0	Starting ~ Substation ES - CICS Interf
	17:28:41.1610	154	SXG2801I	0	0	Initialized ~ Substation ES - Transfor
	17:28:40.9660	0	SXT5717I	0	0	Trace level 1, 10 work threads; Cfg:TI
	17:28:40.9651	154	SXG2800I	0	0	Starting ~ Substation ES - Transformer
	17:28:40.6973	154	SXG2801I	0	0	Initialized ~ Substation ES - Admin &
	17:28:40.3707	154	SXS1009I	0	0	Substation (ES) - Version 2.3.0
	17:28:39.9021	154	SXS1000I	0	0	Starting ~ TIBCO Substation (ES) for O
	17:28:39.9006	154	SXG1800I	0	0	Parameter Log Agent Starting - Logging
	17:28:39.8791	154	SXG1600I	0	0	Log Agent Starting - Logging to TIBLOG

The following table describes the fields in the panel.

LTA Multiple Entry Panel Fields

Field	Description
Sel	If you specify S in the input column, a pop-up panel with all the values that relate to the selected message is displayed.
Time	This field shows the time this message was recorded by the LTA Agent.
Type	This field shows the code of the internal message type.
Msg Id	This field shows the message identifier.
RC	This field shows the return code associated with the message.
Rsn	This field shows the reason code associated with the message.
Message	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
Msg Id	This is an eight-character message identifier.
Type	This is the internal message type.
Date	This is the date on which this message was recorded by the LTA Agent.
Time	This is the time at which this message was recorded by the LTA Agent.

Field	Description
Union id	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.
GRIN	This is the internal Global Resource Identification Number.
Proc Id	This is the internal process identifier, which identifies a Substation ES task or sub-task.
Thread	This is the thread of Substation ES that issued the message.
Stck	This is the internal 64-bit Store Clock value, which uniquely identifies this message.
RC	This is the return code associated with the message.
Reason	This is the reason code associated with the message.
Message	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

A 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
NUMBERED ¹	Yes	Yes	You must specify a value for the NUMBERED parameter because it designates that the file is a Relative Record data set.
REUSE	Yes	Yes	
SHR	(2,3)	(2,3)	
RECSZ ²	(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 RECSZ parameter must be equal. The minimum value to be specified for the average record size is 1048 bytes.

IDCAMS Parameter	Log File Values	Trace File Values	Notes
RECORDS ³	(4000)	(3000)	You can change the value of the RECORDS 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.

A 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 an MVS 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 them 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 one is the parameter to 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 one is the parameter to 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 one is the parameter to 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.

The following list is a description of each DD Name that is specific to the 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
//TIBDEBUG DD SYSOUT=* DEBUG MESSAGES FROM VSAM ACCESS
//TIBPRINT 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	You must supply this DD Name when specifying the Copy or Archive keyword. This is an output file that must have the DCB information where RECFM=VB.

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.
-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. Must specify or default -FX -IV -RN -CN with -XY.

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.

Keyword Phrase	Operand Default	Description
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	Copie all the records in the log to the ARCHIVE DD and reset the log to empty. Do not print the records.

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
SHUT or TERM	Terminates the execution of Request-Reply Timer Utility.
STATS	Prints all current totals and statistics.

Console Command	Description
REFRESH	Prints all the current totals and statistics and resets or clears counters and storage for the next capture interval.
PRTLVL=#	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
request-subject	The name of the subject on which the Timer Utility listens for requests in order to capture the time and information about the request.
reply-subject	<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 computes the elapsed the time the conversation took place.</p> <p>You can specify up to three reply-subject names for a request subject.</p>
field-name	The name for the context value according to which the request or reply match is performed.
field-type	<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"> • STR: string • OPA: opaque • U32 or I32: 32-bit unsigned integer or 32-bit signed integer • CE: Substation ES CICS envelope <p>If you specify CE as the type, you need not specify field-name and field-length. With CE as field-type, the Timer Utility identifies the envelope that originates from CICS.</p>
field-length	<p>The length of the field associated with field-type. Normally, you specify field-length only if you define field-type as a string or opaque.</p> <p>The valid range is 1 to 36 bytes. The default is 20 bytes.</p>

Parameter	Description
print-ex-ctt	<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 print-ex-ctt and print-ex-rtt are specified, the Timer Utility prints processes that exceed either limit, according to the set prtlvl level.</p>
print-ex-rtt	<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 print-ex-rtt and print-ex-ctt are specified, the Timer Utility prints processes that exceed either limit, according to the level set in the prtlvl parameter.</p>
prtlvl	<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 field-type value is CE, to view the content of the envelope, set prtlvl 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 a thousandth 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
CICS Send Time	The time stamp of the Substation ES CICS SXCREQR routine that receives the request from the user application in the CICS region.
Task#	The task number of the user CICS application that issued the request.
Idx#	The internal storage array allocation that the Substation ES CICS Request or Reply Facility used to hold the context or envelope.
Req Time	The time at which Timer Utility received the request message.
Reply Time	The time at which the Timer Utility received the reply message.
CICS Trip Time	The time difference between when the message is sent by the CICS routine and when the reply message is received by the timer routine.
Round Trip Time	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 Summary report
1	No normal runtime messages Start and end timestamps summary message Summary report
2	Round-trip runtime messages Start and end timestamps summary message Summary report
3	Runtime messages: <ul style="list-style-type: none"> • Round trip time • Request time • Response time Start and end timestamps summary message Summary report
4	Runtime messages: <ul style="list-style-type: none"> • Round trip time • Request time • Response time • Slot data on request message • Matching field name • Matching Data • Slot number Start and end timestamps summary message Summary report

Print Level	Messages
5	<p>Runtime messages:</p> <ul style="list-style-type: none"> • Round trip time • Request time • Response time • Slot data on request message • Matching field name • Matching Data • Slot number • Slot data on request message: hex dump • Reply data: hex dump <p>Start and end timestamps summary message</p> <p>Summary report</p>

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.

Data Type	Description
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. This field is added as a property field to the EMS message, and as 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: *SXRGNSL*

Before using this utility, you must specify the Rendezvous transport parameters, for example:

```
//SET IVPNETW='-network 127.0.0.1 -service 7555'
//SXRGEN 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.
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> .

Keyword	Operand	Description
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 tibss-password 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 DEST-RECV parameter so that the published message has a reply-to 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 tibss- . 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 tibss-userid 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 WAIT plus the value of WAIT-ON-REPEAT.</p>

Keyword	Operand	Description
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 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 ones are examples showing how to send Substation ES commands to different Substation systems.

- Send a show counters command for Substation ID SS-ES-4-TEST:

```
USER TIBCO
PSWD password
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD PROP,tibss-ss-id,"SS-ES-4-TEST"
FLD STR,SXS-COMMAND,"SHOW,COUNTERS"
END
/*
```

- Send a show active listeners command for Substation ID SS-ES-4-PROD:

```
USER TIBCO
PSWD password
DEST-SEND-DEFAULT tibss.admin.sxs.request
FLD PROP,tibss-ss-id,"SS-ES-4-PROD"
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. Overrides DEST-TYPE-DEFAULT in a group.
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>	<p>Defines a field for the message.</p> <p>Where:</p> <p><i>type</i> is I32, U32, PROP, 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 tibss-password system field. The default is a blank.

Keyword	Operand	Description
PUBLISH-ONLY		Specifies that the utility only sends messages and does not wait or receive any messages. Be sure to specify a DEST-RECV parameter so that the published message has a reply-to 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		Enables TLS1.
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 tibss- . 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 tibss-userid system field. The default is a blank.

Keyword	Operand	Description
WAIT	<i>timevalue</i>	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. The time to delay between each message group is determined by the value of WAIT plus the value of WAIT-ON-REPEAT .
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-RECV tibss.admin.reply
FLD STR,SXS-COMMAND,"REFR,RID=SXC-DPL-C"
WAIT 3
END
DEST-SEND tibss.CICS.C.Request
DEST-RECV 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.
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.)
```