

# Operator Commands

This chapter covers the following topics:

- Command Syntax
  - General Broker Commands
  - Participant-specific Commands
  - Security-specific Commands
  - Transport-specific Commands
  - XCOM-specific Commands
- 

## Command Syntax

The following command format is required to communicate with EntireX Broker, using the operator console. Parameters in UPPERCASE must be typed "as is". Parameters in lowercase must be substituted with a valid value. Operator commands have the following format:

```
/INTR tsn,command[parameter]
```

where *tsn* is the BS2000/OSD task sequence number of the EntireX Broker main task  
*command* is the operator command  
*parameter* is an optional parameter allowed by the operator command you are issuing

## General Broker Commands

The following broker commands are available:

- BROKER TRACE
- DPOOL
- DRES
- DSTAT
- ETBEND
- ETBSTOP
- FLUSH
- PSTORE TRACE

- SHUTDOWN class,server,service
- TRACE
- TRAP-ERROR

## BROKER TRACE

Alias of broker command TRACE. Modifies the setting of the broker-specific attribute TRACE-LEVEL.

### Example

➤ To set a trace level 2 for broker

- Enter command:

```
/INTR tsn,BROKER TRACE=2
```

If the console prompt is suppressed, enter an MSG command before the console command:

```
MSG partition_id
```

See TRACE-LEVEL under *Broker Attributes*.

## DPOOL

Lists all memory pools currently allocated by EntireX Broker. Start address, pool size in bytes and name of pool are provided. There can be multiple entries for a specific type of pool.

### Sample Output

```
ETBM0720 Operator typed in: DPOOL
ETBM0657 Broker pool usage:
ETBM0657 0x2338FFB8 16781380 bytes COMMUNICATION POOL
ETBM0657 0x243A9EB8 368964 bytes CONVERSATION POOL
ETBM0657 0x24404F38 233668 bytes CONNECTION POOL
ETBM0657 0x2443EF38 4395204 bytes LONG MESSAGES POOL
ETBM0657 0x24870BB8 3703876 bytes SHORT MESSAGES POOL
ETBM0657 0x24BF9398 134244 bytes PARTICIPANT POOL
ETBM0657 0x24C1AF78 36996 bytes PARTICIPANT EXTENSION POOL
ETBM0657 0x24C24798 26724 bytes PROXY QUEUE POOL
ETBM0657 0x24C2BDA8 131668 bytes SERVICE ATTRIBUTES POOL
ETBM0657 0x24C4CB98 54372 bytes SERVICE POOL
ETBM0657 0x24C5AF78 32900 bytes SERVICE EXTENSION POOL
ETBM0657 0x24D31FA8 344148 bytes SUBSCRIPTION POOL
ETBM0657 0x24D865A8 129620 bytes TOPIC ATTRIBUTES POOL
ETBM0657 0x2338F420 2952 bytes TOPIC POOL
ETBM0657 0x24DA6778 30852 bytes TOPIC EXTENSION POOL
ETBM0657 0x24C63B18 87268 bytes TIMEOUT QUEUE POOL
ETBM0657 0x24C79398 179300 bytes TRANSLATION POOL
ETBM0657 0x24CA5F38 176324 bytes UNIT OF WORK POOL
ETBM0657 0x24CD1798 391268 bytes WORK QUEUE POOL
ETBM0657 0x24DAEB98 33892 bytes PSTORE SUBSCRIBER POOL
ETBM0657 0x24DB73A8 19540 bytes PSTORE TOPIC POOL
ETBM0582 Function completed
```

## DRES

Displays EntireX Broker's resource usage for conversations, message buffers, participants, services, topics, the timeout queue, units of work, and the work queue. Resource usage provides the total number, the number of free elements, and the number of used elements.

### Sample Output

```
ETBM0720 Operator typed in: DRES
ETBM0581 Broker resource usage:
ETBM0581 Resource ----- Total # --- Free # --- Used #
ETBM0581 Conversations          4096      852    3244
ETBM0581 Long message buffers      0         0         0
ETBM0581 Short message buffers   8192    7384     808
ETBM0581 Participants            256     235         21
ETBM0581 Services                256     240         16
ETBM0581 Topics                  0         0         0
ETBM0581 Timeout Queue          1280     845     435
ETBM0581 Units Of Work           0         0         0
ETBM0581 Work Queue              256     239         17
ETBM0582 Function completed
```

## DSTAT

Displays the total number of active elements, and an optional high watermark for services, clients, servers, conversations, message buffers, topics, publishers, subscribers, and publications.

### Sample Output

```
ETBM0720 Operator typed in: DSTAT
ETBM0580 Broker statistics:
ETBM0580 NUM-SERVICE ..... 0
ETBM0580 Services active ..... 7
ETBM0580 NUM-CLIENT ..... 0
ETBM0580 Clients active ..... 10
ETBM0580 Clients active HWM ..... 10
ETBM0580 NUM-SERVER ..... 0
ETBM0580 Servers active ..... 10
ETBM0580 Servers active HWM ..... 10
ETBM0580 NUM-CONVERSATION ..... 0
ETBM0580 Conversations active ..... 607
ETBM0580 Conversations active HWM .. 968
ETBM0580 NUM-LONG-BUFFER ..... 0
ETBM0580 Long buffers active ..... 0
ETBM0580 Long buffers active HWM ... 0
ETBM0580 NUM-SHORT-BUFFER ..... 0
ETBM0580 Short buffers active ..... 1219
ETBM0580 Short buffers active HWM .. 1928
ETBM0580 NUM-TOPIC ..... 0
ETBM0580 Topics active ..... 0
ETBM0580 NUM-PUBLISHER ..... 0
ETBM0580 Publishers active ..... 0
ETBM0580 Publishers active HWM ..... 0
ETBM0580 NUM-SUBSCRIBER ..... 0
ETBM0580 Subscribers active ..... 0
ETBM0580 Subscribers active HWM .... 0
ETBM0580 NUM-PUBLICATION ..... 0
ETBM0580 Publications active ..... 0
ETBM0580 Publications active HWM ... 0
ETBM0582 Function completed
```

## ETBEND

Processing stops immediately. Current calls to the EntireX Broker are not allowed to finish.

## ETBSTOP

Alias of ETBEND.

## FLUSH

Flush all trace data kept in internal trace buffers to stderr (SYSOUT). The broker-specific attribute TRMODE=WRAP is required.

## PSTORE TRACE

Modifies the trace level for the Adabas persistent store (Adabas-specific attribute TRACE-LEVEL).

### Example

➤ **To set a trace level 2 for the Adabas persistent store**

- Enter command:

```
/INTR tsn,PSTORE TRACE=2
```

See TRACE-LEVEL under *Broker Attributes*.

## SHUTDOWN *class,server,service*

Shuts down the specified service immediately and stops all servers that have registered this service.

### Example

➤ **To shutdown service CLASS=RPC, SERVER=SRV1, SERVICE=CALLNAT**

- Enter command:

```
/INTR tsn,SHUTDOWN RPC,SRV1,CALLNAT
```

## TRACE

Modifies the setting of the broker-specific attribute TRACE-LEVEL.

### Sample Commands

➤ **To modify the trace level**

- Enter command, for example:

```
/INTR tsn,TRACE=0  
/INTR tsn,TRACE=1  
/INTR tsn,TRACE=4
```

See `TRACE-LEVEL` under *Broker Attributes*.

## TRAP-ERROR

Modifies the setting of the broker-specific attribute `TRAP-ERROR`.

### Sample Command

➤ To modify the setting for `TRAP-ERROR`

- Enter command:

```
/INTR tsn,TRAP-ERROR=nnnn
```

where *nnnn* is the four-digit API error number that triggers the trace handler.

See `TRAP-ERROR` under *Broker Attributes*.

## Participant-specific Commands

Within EntireX Broker nomenclature, a participant is an application implicitly or explicitly logged on to the Broker as a specific user. A participant could act as client, server, publisher or subscriber. The following participant-specific commands are available:

- CANCEL parameter
- USERLIST
- USERS parameter

### CANCEL *parameter*

Operator command CANCEL is used to delete participants from EntireX Broker. The following parameters are supported:

Parameter	Description
[USER=] <i>user_id</i>	Cancel all participants with the specified <i>user_id</i> . Non-persistent resources will be freed by the timeout manager. Prefix "USER=" is the default value and may be omitted.
SEQNO= <i>seqno</i>	Cancel the participant with the sequence number <i>seqno</i> . Non-persistent resources will be freed by the timeout manager. Operator commands USERLIST and USERS display sequence numbers of all selected participants.

### Sample Commands

#### ➤ To cancel all participant entries of user "DOE"

- Enter command:

```
/INTR tsn,CANCEL DOE
```

Or:

```
/INTR tsn,CANCEL USER=DOE
```

#### ➤ To cancel participant with sequence number "11"

- Enter command:

```
/INTR tsn,CANCEL SEQNO=11
```

## USERLIST

Operator command USERLIST displays a list of selected participant entries. The following parameters are supported:

Parameter	Description
none   *	Display all participants.
<i>user_id</i>	Display all participants with user ID <i>user_id</i> . Wildcard characters are supported.

### Sample Commands

#### > To display all participants

- Enter command:

```
/INTR tsn,USERLIST
```

Or:

```
/INTR tsn,USERLIST *
```

#### > To display all participants with user ID "DOE"

- Enter command:

```
/INTR tsn,USERLIST DOE
```

This produces the following output. See *Description of USERLIST Output Columns* below.

```
ETBM0720 Operator typed in: USERLIST DOE
ETBM0687 Participants:
ETBM0687 USER-ID ----- C S P U E CHR SEQNO
ETBM0687 DOE                N Y N N Y ASC 1
ETBM0582 Function completed
```

#### > To display all participants with user ID starting with uppercase "D"

- Enter command:

```
/INTR tsn,USERLIST D*
```

This produces the following output. See *Description of USERLIST Output Columns* below.

```
ETBM0720 Operator typed in: USERLIST D*
ETBM0687 Participants:
ETBM0687 USER-ID ----- C S P U E CHR SEQNO
ETBM0687 DOE                N Y N N Y ASC 1
ETBM0687 DOE1              N Y N N Y EBC 2
ETBM0687 DOE2              N Y N N Y EBC 3
ETBM0687 DOE3              N Y N N Y EBC 4
ETBM0582 Function completed
```

#### > To display all participants with 4-character user ID, starting with uppercase "D" and with uppercase "E" as third character

- Enter command:

```
/INTR tsn,USERLIST D?E?
```

This produces the following output. See *Description of USERLIST Output Columns* below.

```
ETBM0720 Operator typed in: USERLIST D?E?
ETBM0687 Participants:
ETBM0687 USER-ID ----- C S P U E CHR SEQNO
ETBM0687 DOE1                N Y N N Y EBC 2
ETBM0687 DOE2                N Y N N Y EBC 3
ETBM0687 DOE3                N Y N N Y EBC 4
ETBM0582 Function completed
```

**Description of USERLIST Output Columns**

Keyword	Description
USER-ID	User ID (32 bytes, case-sensitive). See USER-ID under <i>Broker ACI Fields</i> .
C	Client.  Y Participant is a client, otherwise "N".
S	Server.  Y Participant is a server, otherwise "N".
P	Publisher.  Y Participant is a publisher, otherwise "N".
U	Subscriber.  Y Participant is a subscriber, otherwise "N".
E	Big endian.  Y Participant is on a big-endian machine. N Participant is on a little-endian machine.
CHR	Character set.  ASC Participant is an ASCII user. EBC Participant is an EBCDIC user.
SEQNO	Sequence number of participant. Can be used for operator command CANCEL <i>parameter</i> .

**USERS parameter**

Operator command USERS displays selected user data of participant entries. The following parameters are supported:



Parameter	Description
none   *	Display all participants.
<i>user_id</i>	Display all participants with user ID <i>user_id</i> . Wildcard characters are supported.

## Sample Commands

### ➤ To display all participants

- Enter command:

```
/INTR tsn,USERS
```

Or:

```
/INTR tsn,USERS *
```

### ➤ To display all participants with user ID "DOE"

- Enter command:

```
/INTR tsn,USERS DOE
```

This produces the following output. See *Description of USERS Output Columns* below.

```
ETBM0720 Operator typed in: USERS DOE
ETBM0687 Participants:
ETBM0687 USER-ID: DOE
ETBM0687 CLIENT: N SERVER: Y PUBLISHER: N SUBSCRIBER: N
ETBM0687 SEQNO: 6 BIG ENDIAN: Y CHARSET: ASCII PUID:
ETBM0687 202073756E6578322D2D30303030324646462D2D30303030303031
ETBM0687 TOKEN:
ETBM0582 Function completed
```

## Description of USERS Output Columns

Keyword	Description
USER-ID	User ID (32 bytes, case-sensitive). See USER-ID under <i>Broker ACI Fields</i> .
CLIENT	<b>Y</b> Participant is a client, otherwise "N".
SERVER	<b>Y</b> Participant is a server, otherwise "N".
PUBLISHER	<b>Y</b> Participant is a publisher, otherwise "N".
SUBSCRIBER	<b>Y</b> Participant is a subscriber, otherwise "N".
BIG ENDIAN	<b>Y</b> Participant is on a big-endian machine. <b>N</b> Participant is on a little-endian machine.
CHARSET	<b>ASC</b> Participant is an ASCII user. <b>EBC</b> Participant is an EBCDIC user.
PUID	Internal unique ID of participant. Hexadecimal 28-byte value in printable format.
TOKEN	Optionally identifies the participant. See TOKEN under <i>Broker ACI Fields</i> .

## Security-specific Commands

### DSECSTAT

Displays the number of successful and failed Security authentications and Security authorizations.

#### Sample Output

```
ETBM0720 Operator typed in: DSECSTAT
ETBM0579 Security Authentications - successful: 20 failed: 0
ETBM0579 Security Authorizations - successful: 0 failed: 0
```

### RESET userid

Resets the Security context for the specified user ID.

#### Sample Output

```
ETBM0720 Operator typed in: RESET EXXBATCH
ETBM0578 Reset ACEE for SAF-ID EXXBATCH : 20 instances found
```

### SECURITY TRACE

Modifies the trace level for the EntireX Security (security-specific attribute TRACE-LEVEL).  
Broker-specific attribute SECURITY=YES must be set.

### Example

➤ **To set a trace level 2 for EntireX Security**

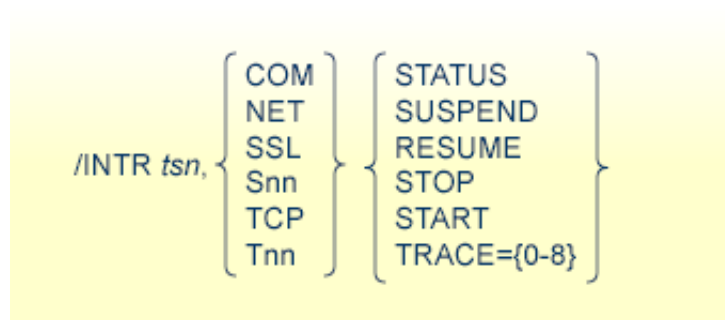
- Enter command:

```
/INTR tsn,SECURITY TRACE=2
```

See TRACE-LEVEL under *Broker Attributes*.

## Transport-specific Commands

Transport-specific commands are available for Adabas/Entire Net-Work communicators, SSL communicators and TCP communicators; the COM command can be used for all communicators. The following command syntax applies:



### COM *parameter*

This command is executed by all configured transport communicators. The following parameters are supported:

Parameter	Description
STATUS	Displays the current status of the transport communicator.
SUSPEND	Used to suspend the transport communicator. The transport communicator is halted but will not shut down. User requests receive response code 148.
RESUME	Resume a suspended transport communicator. If the communicator was not suspended before, an error message will be displayed.
STOP	Stop an active or suspended transport communicator. The transport communicator will shut down. All transport-specific resources will be freed. User requests receive response code 148.
START	Start a transport communicator that was previously stopped. If the communicator was not stopped before, an error message will be displayed.
TRACE	<p>Sets the trace level for the transport method. If the global trace level (see TRACE) is set with command  <i>/INTR tsn,TRACE=n</i>            this applies to <i>all</i> transport methods. This command will also override any existing transport-specific settings. If you subsequently enter command  <i>/INTR tsn, TCP TRACE=n</i>            only the trace level for TCP/IP transport is modified.</p> <p><b>Note:</b>            With commands <i>TCP Tnn</i>, and <i>SSL Snn</i>, the trace level is set for <i>all</i> TCP and SSL communicators respectively. Setting a trace level for a single TCP or SSL instance is not supported. For example: although it is possible to submit the command  <i>/INTR tsn,T01 TRACE=1</i>            this command sets the trace level for all TCP communicators.</p>

### Sample Output

```

ETBM0720 Operator typed in: COM STATUS
ETBW0718 TCP Communicator 0 currently active
ETBW0718 TCP Communicator 1 currently active
ETBW0718 SSL Communicator 0 currently suspended
ETBW0718 NET Communicator 0 currently suspended
XCO0039I 00113 Total number of commands = 17
XCO0057I 00113 Operator entry active
ETBM0720 Operator typed in: COM SUSPEND
ETBM0721 TCP Communicator 0 suspended
ETBM0721 TCP Communicator 1 suspended
ETBM0721 SSL Communicator 0 suspended
ETBM0721 NET Communicator 0 suspended

```

### NET parameter

This command is executed by X-COM, the Adabas/Entire Net-Work communicator. See command COM above for a list of supported parameters.

## Sample Output

```
ETBM0720 Operator typed in: NET STATUS
ETBW0718 NET Communicator 0 currently active
XCO0039I 00113 Total number of commands = 17
XCO0057I 00113 Operator entry active
```

## SSL parameter

This command is executed by all SSL communicators. See command COM above for a list of supported parameters.

## Sample Output

```
ETBM0720 Operator typed in: SSL STATUS
ETBW0718 SSL Communicator 0 currently active
```

To manipulate a specific communicator instance (max. five instances can be started), use the command S00, S01, S02, S03 or S04 for the respective SSL instance.

## TCP parameter

This command is executed by TCP communicators. See command COM above for a list of supported parameters.

## Sample Output

```
ETBM0720 Operator typed in: TCP STATUS
ETBW0718 TCP Communicator 0 currently active
ETBW0718 TCP Communicator 1 currently active
```

```
ETBM0720 Operator typed in: TCP RESUME
ETBM0721 TCP Communicator 0 resumed
ETBM0721 TCP Communicator 1 resumed
```

To manipulate a specific communicator instance (max. five instances can be started), use the command T00, T01, T02, T03 or T04 for the respective TCP instance.

## Sample Output

```
ETBM0720 Operator typed in: T00 STATUS
ETBW0718 TCP Communicator 0 currently active
```

```
ETBM0720 Operator typed in: T01 STATUS
ETBW0718 TCP Communicator 1 currently active
```

## Sample Transport Commands

### ➤ To display status of all transport communicators

- Enter command:

```
/INTR tsn,COM STATUS
```

➤ **To suspend first TCP communicator**

- Enter command:

```
/INTR tsn,T00 SUSPEND
```

➤ **To stop all SSL transport communicators**

- Enter command:

```
/INTR tsn,SSL STOP
```

## XCOM-specific Commands

### Note:

All operator commands beginning with "X" belong to X-COM, the Adabas/Entire Net-Work communicator. The following commands operate only on the Adabas transport mechanism: XCQES, XHALT, XPARM, XSTART, XSTAT and XUSER. These commands have no effect on functions not related to the Adabas transport mechanism.

XEND and XSTOP function independently of the transport mechanism. (They stop the Broker's processing immediately, whereby existing calls to the EntireX Broker are not allowed to finish.)

### XABS

Displays the current number, and the highest number, of used bytes in the Adabas attached buffer pool to the console.

### Note:

This command operates on the Adabas transport mechanism only. It has no effect on functions not related to the Adabas transport mechanism.

### Sample Output

```
ETBM0720 Operator typed in: XABS
XC00090I 00113 Attached buffer usage
XC00091I 00113 Number of bytes in use = 0
XC00092I 00113 Highest number of bytes in use = 6400
```

### XCQES

Displays the current number, and the highest number, of Adabas command queue elements to the console.

### Note:

This command operates on the Adabas transport mechanism only. It has no effect on functions not related to the Adabas transport mechanism.

### Sample Output

```
ETBM0720 Operator typed in: XCQES
XC00030I 00113 Number of active CQEs = 0
XC00031I 00113 Highest number of active CQEs = 1
```

### XEND

Alias of ETBEND.

### XHALT

New calls to the EntireX Broker are temporarily rejected. Processing is resumed by issuing the XSTART operator command. XHALT is an alias for command NET SUSPEND.

### Note:

This command operates on the Adabas transport mechanism only. It has no effect on functions not related to the Adabas transport mechanism.



## Sample Output

```
ETBM0720 Operator typed in: XHALT
ETBM0721 NET Communicator 0 suspended
```

## XPARAM

Displays the values of Adabas SVC, database ID, number of CQEs, number of attached buffers, and the application name for the Adabas transport to the console.

### Note:

This command operates on the Adabas transport mechanism only. It has no effect on functions not related to the Adabas transport mechanism.

## Sample Output

```
ETBM0720 Operator typed in: XPARAM
XCO0032I 00113 Parameters for this session:
XCO0033I 00113 SVC = 249
XCO0034I 00113 NODE = 00113
XCO0035I 00113 NCQE = 00100
XCO0036I 00113 NABS = 10000
XCO0037I 00113 User application = ETBNUC
```

## XSTART

Processing of new calls to the EntireX Broker, interrupted with the XHALT command, is resumed. XSTART is an alias of command NET RESUME.

### Note:

This command operates on the Adabas transport mechanism only. It has no effect on functions not related to the Adabas transport mechanism.

## Sample Output

```
ETBM0720 Operator typed in: XSTART
ETBM0721 NET Communicator 0 resumed
```

## XSTAT

Displays the EntireX Broker statistics as console messages.

### Note:

This command operates on the Adabas transport mechanism only. It has no effect on functions not related to the Adabas transport mechanism.

## XSTOP

Alias of ETBEND.

## XUSER

Displays the current number, as well as the highest number, of users actively issuing commands using the Adabas transport mechanism to the console.

**Note:**

The number of users displayed with this operator command will not represent all of the Broker clients and servers but only the subset of users issuing commands using the Adabas transport mechanism. Command and Information Services provides comprehensive information about all Broker clients and servers.