☁ software AG

webMethods EntireX

Installing EntireX under BS2000

Version 10.5

October 2019

WEBMETHODS

This document applies to webMethods EntireX Version 10.5 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

Installing EntireX under BS2000

This document describes how to install and operate the BS2000 components of EntireX.

General Information	An overview of resources delivered.
Installing EntireX Broker	How to install and start the EntireX Broker under BS2000.
Installing the RPC Server	How to install and start the EntireX RPC Server for BS2000.
Installing EntireX Security	Provides information required for installing EntireX Security under BS2000.

Installation prerequisites are listed for all platforms centrally; see *BS2000 Prerequisites* in the EntireX Release Notes. See also *EntireX Installation Overview* for topics that apply to multiple operating systems.

> **Note:** If you want to use EntireX on BS2000 together with the Eclipse-based Designer components, you need to install the respective EntireX components under UNIX or Windows, using the Software AG Installer. See the separate Software AG Installer documentation under *http://documentation.softwareag.com* > *Cross-Product Guides* > *Software AG Installer and Update Manager*.

Related Literature

- *EntireX Administration under BS2000*
- *RPC Server for BS2000*

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documenta-tion.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.software-ag.cloud**. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.soft-wareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.

- Ask questions and find answers in our discussion forums.

- Get the latest Software AG news and announcements.

- Explore our communities.

- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.

- Search the Knowledge Center for technical information and tips.

- Subscribe to early warnings and critical alerts.

- Open and update support incidents.

- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Prerequisites

Component/Feature	Prerequisites
COBOL Wrapper	■ To compile the sources generated by the Designer component COBOL Wrapper: the COBOL2000 compiler on BS2000.
C Wrapper	■ To compile the applications generated by the Designer component C Wrapper: any ILCS-enabled C/C++ compiler on BS2000.

3   General Information

This chapter covers the following topics:

Distribution Media

All BS2000 components of EntireX are distributed on the supplied mainframe installation medium including the EntireX license certificate. A license certificate can also be sent by e-mail.

Instalation Jobs

The installation of Software AG products on mainframe platforms is performed by installation jobs. These jobs are contained in the delivered files. The System Maintenance Aid (SMA) generates the following jobs:

■ Copy the contents of the installation medium to disk.

Contents of Installation Medium

The installation medium contains the files listed in the table below. The sequence of the files, the file types, the number of library blocks needed and the space each file requires on disk are shown in the Software AG Product Delivery Report, which accompanies the installation medium. During installation, the files are loaded from the installation medium.

File Name	Type
EXX103.JOBS	EntireX Broker jobs and configuration.
EXX103.LIB	EntireX Broker components, stubs and examples.
EXX103.SYSF	Adabas persistent store FDT.
WAL842.MOD	Adabas components required to run EntireX Broker.
WAL842.SRC	Adabas components source library.
EXP103.JOBS	RPC Server for BS2000 jobs and configuration.
EXP103.LIB	RPC Server for BS2000 components.
EXP103.CSRV	C server examples library.
EXP103.COBS	COBOL server examples library.
EXP103.COBC	COBOL client examples library.

Copying the Contents of the Installation Medium to Disk

If you are not using SMA, use the procedure described below and supply the values specified below.

To copy the data sets from installation medium to disk, perform the following steps:

1. Copy the Library SRVnnn.LIB from Installation Medium to Disk

This step is not necessary if you have already copied the library SRV*nnn*.LIB from another Software AG installation medium. For more information, refer to the element #READ-ME in this library.

The library SRV*nnn*.LIB is stored on the installation medium as the sequential file SRV*nnn*.LIBS containing LMS commands. The current version *nnn* can be obtained from the **Software AG Product Delivery Report**. To convert this sequential file into an LMS library, execute the following commands:

```
/IMPORT-FILE SUPPORT=*TAPE(FILE-NAME=SRVnnn.LIBS, -
/ VOLUME=<volser>, DEV-TYPE=<tape-device>)
/ADD-FILE-LINK LINK-NAME=EDTSAM, FILE-NAME=SRVnnn.LIBS, -
/ SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT, -
/ BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ '/'
@SYSTEM 'REMOVE-FILE-LINK EDTSAM'
@SYSTEM 'EXPORT-FILE FILE-NAME=SRVnnn.LIBS'
@WRITE 'SRVnnn.LIBS'
@HALT
/ASS-SYSDTA SRVnnn.LIBS
/MOD-JOB-SW ON=1
/START-PROG $LMS
/MOD-JOB-SW OFF=1
/ASS-SYSDTA *PRIMARY
```

where `tape-device` is the device type of the installation medium, e.g. TAPE-C4

       `volser` is the VOLSER of the installation medium (see Software AG Product Delivery Report)

2. Copy the Procedure COPY.PROC from Installation Medium to Disk

To copy the procedure `COPY.PROC` to disk, call the procedure `P.COPYTAPE` in the library SRV103.LIB:

```
/CALL-PROCEDURE  (SRV103.LIB,P.COPYTAPE), -
  /   (VSNT=<volser>, DEVT=<tape-device>)
```

If you use a TAPE-C4 device, you may omit the parameter `DEVT`.

3. Copy all Product Files from Installation Medium to Disk

To copy all Software AG product files from installation medium to disk, enter the procedure `COPY.PROC`:

```
/ENTER-PROCEDURE  COPY.PROC, DEVT=<tape-device>
```

If you use a TAPE-C4 device, you may omit the parameter `DEVT`. The result of this procedure is written to the file `L.REPORT.SRV`.

When the files have been copied to disk, continue with the steps described under *Installing EntireX Broker under BS2000* and *Installing the RPC Server for BS2000*.

4 Installing EntireX Broker under BS2000

This section explains how to install and start the EntireX Broker on BS2000. It covers the following topics:

Introduction

When installing EntireX Broker, all modifications are done to the J and S elements - job control (J-elements) and parameter files (S-elements) - located in EXX103.JOBS. All job control and parameter elements contain a preconfiguration which enables you to install EntireX Broker with much less effort. Using this preconfiguration requires that all libraries be located under the same BS2000 user ID.

> **Note:** Installation prerequisites are described centrally. See *BS2000 Prerequisites*. Make sure these are met before you start installation. It is important to upgrade your libraries first.

Overview of Broker Installation Steps

This section describes the following installation steps

- Install the License Certificate
- Customize the EntireX Broker Attribute File
- Customize the ADALNK Parameters File
- Customize the EntireX Broker Startup JCL and Start EntireX Broker
- Start EntireX Broker
- Stop EntireX Broker
- Creating a Broker Persistent Store (optional)
- Set up the EntireX Broker Security Server for BS2000 (optional)

Install the License Certificate

There are two types of license file:

- One is delivered on installation medium (EXX103.LICS)

- The other, in ASCII format, you may have received by e-mail (EXX103.XML) or on a CD. To make this file available for EntireX, transfer it in binary format to BS2000, using FTP. Make sure that the target file on BS2000 is allocated with `FILE-STRUCTURE = SAM` and `BUF-LEN = STD(2)`.

  When uploading the license file to BS2000, you can use the following FTP commands to create the required file structure:

```
LITERAL FILE EXX103.LICS, RECSIZE=0, RECFORM=V, BLKSIZE=(STD,2), OPEN=UPDATE, ↵
FCBTYPE=SAM
BIN
PUT EXX103.XML EXX103.LICS
```

Customize the EntireX Broker Attribute File

ETB-ATTR is a sample broker attribute file. Customize the attribute settings to suit your needs.

To run a minimal configuration of EntireX Broker that is suitable to execute the verification programs BCOC and BCOS, set up the following parameters:

Parameter	Description
BROKER-ID=ETB<nnnnn>	Identifies the Broker to which the attribute file applies. The Broker ID must be unique per machine.
NODE=<node-id>	A DBID under which EntireX Broker is visible in the system.
IDTNAME=ADA<xxxxx>	Specifies the ID table name under which EntireX Broker will be accessible.
PORT=<port>	A free port number EntireX Broker listens on for TCP/IP communication. Depending on the system settings, free port numbers under BS2000 start with port numbers greater than 4096.

For a full description of all parameters, see *Broker Attributes*.

Customize the ADALNK Parameters File

The verification programs BCOS and BCOC, the command and information services utilities ETBCMD and ETBINFO as well as the Adabas persistent store require the ADALNK parameter IDTNAME to be set. See also *Verifying the Installation of the Broker* and step *Creating a Broker Persistent Store (optional)* below.

Parameter	Description
ADALNK IDTNAME= ADA<xxxxx>	Specifies the ID table name under which EntireX Broker is accessible.

Customize the EntireX Broker Startup JCL and Start EntireX Broker

The job control delivered with EntireX Broker uses BS2000 S-procedures. We strongly recommend you do not modify START-BROKER. This procedure is recursively called to establish the broker environment. Modifications should only be done to the PARAMETER-DECLARATION section as described below. If all EntireX Broker components are installed under the same BS2000 user ID, and all previous installation steps have been done using the default settings, no modifications to START-BROKER are necessary.

Startup Parameter	Description	Default
LICENSE-FILE	License certificate file.	EXX103.LICS
EXX-LIB	EntireX Broker library.	EXX103.LIB
EXX-JOBS	EntireX Broker jobs library.	EXX103.JOBS
WAL-MOD	WAL module library.	WAL842.MOD
MLC-MOD	Licensing module library.	MLC127.MOD
BROKER-ATTRIBUTES	EntireX Broker attribute file.	ETB-ATTR
BROKER-VARIABLES	EntireX Broker attributes variables file.	ETB-VARS
ADABAS-PARAMETERS	Adabas ADALNK parameter file.	ETB-ADAPARM
FILE-PREFIX	File name prefix used for all files written to disk.	ETB
CONFIG-REPORT-FILE	EntireX Broker configuration report file name. *file-prefix.tsn*.CONFIG.REPORT	CONFIG.REPORT
PSTORE-REPORT-FILE	EntireX Broker Adabas persistent store report file name.*file-prefix.tsn*.PSTORE.REPORT	PSTORE.REPORT
STORAGE-REPORT-FILE	EntireX Broker storage report file name. *file-prefix.tsn*.STORAGE.REPORT	STORAGE.REPORT
LICENSE-REPORT-FILE	EntireX Broker license report file name. *file-prefix.tsn*.LICENSE.REPORT	LICENSE.REPORT
ACCOUNTING-FILE	EntireX Broker accounting file name. *file-prefix.tsn*.ACCOUNTING	ACCOUNTING
CLOGR1-FILE	EntireX Broker TRACE-LEVEL=1 command log file 1. *file-prefix.tsn*.CLOGR1	CLOGR1
CLOGR2-FILE	EntireX Broker TRACE-LEVEL=1 command log file 2. *file-prefix.tsn*.CLOGR2	CLOGR2
TASK-TYPE	Is filled during runtime with the EntireX Broker task type. Do *not* modify it !	MAIN
MAIN-TASK-TSN	Is filled during runtime with the main task TSN. The value is used by subsequently entered Broker tasks. Do *not* modify it!	*ETB

If EntireX Broker requires SERVICE-UPDATES to be set to YES, the attributes file ETB-ATTR cannot be held in LMS. It needs to be located on disk instead. This requires a change the START-BROKER procedure. See SERVICE-UPDATES.

Copy the attribute file to disk.

```
/ START-LMS
//  EXTRACT-ELEMENT ELEMENT=*LIB(LIB=EXX103.JOBS,ELEMENT=ETB-ATTR,TYPE=S)
//END
```

Change the ADD-FILE-LINK in the section below in START-BROKER in EXX103.JOBS to ↵
read the file ETB-ATTR from disk.

```
...
/   "------------------------------------------------------------------"
/   "  SET UP FILE LINKS                                               "
/   "------------------------------------------------------------------"
/   ADD-FILE-LINK LINK-NAME = ETBLIC,   FILE-NAME = &(LICENSE-FILE)
/   ADD-FILE-LINK LINK-NAME = ETBFILE,  FILE-NAME = &(BROKER-ATTRIBUTES)
/   ADD-FILE-LINK LINK-NAME = ETBVARS,  FILE-NAME = #BROKER-VARIABLES
/   ADD-FILE-LINK LINK-NAME = DDLNKPAR, FILE-NAME = #ADABAS-PARAMETERS
...
```

Start EntireX Broker

≫ **To start the broker**

■   Enter the following SDF command:

```
/ENTER-PROCEDURE *LIB(LIB=EXX103.JOBS,ELE=START-BROKER), -
/JOB-NAME=ETB,LOGGING=*NO,RESOURCES=*PAR(CPU-LIMIT=*NO)
```

We recommend using a three-character job name. The job name is taken as prefix for all sub-sequently started tasks. Because the job name is limited to eight characters, a longer job name will overwrite the suffix added by EntireX Broker. For example: EntireX Broker running with three worker tasks and NET-TCP communication, JOB-NAME=ETB, CPU-LIMIT=*NO:

```
NAME        TSN TYPE       PRI      CPU-USED CPU-MAX ACCOUNT#
ETB        5397 2 BATCH    9 255     2.2379  NTL    1
ETBCOM     5398 2 BATCH    9 255     1.3577  NTL    1
ETBWRK00   5399 2 BATCH    9 255     0.8970  NTL    1
ETBWRK01   5400 2 BATCH    9 255     0.7571  NTL    1
ETBWRK02   5401 2 BATCH    9 255     0.7445  NTL    1
ETBTCP00   5402 2 BATCH    9 255     0.6124  NTL    1
ETBTCPPX   5403 2 BATCH    9 255     0.5417  NTL    1
ETBNET00   5404 2 BATCH    9 255     0.6555  NTL    1
ETBTOM     5407 2 BATCH    9 255     6.4044  NTL    1
```

The properties assigned to the main task (ETB), e.g. JOB-CLASS, CPU-LIMIT, will be inherited by all subsequently started tasks. For CPU-LIMIT, if specified, only *NO (no time limit) and *STD are inherited.

You can now configure and run the verification jobs BCOS and BCOC. See *Verifying the Installation of the Broker*.

Stop EntireX Broker

⟫ **To stop the broker from a privileged user ID**

■ Enter the following command:

```
/INFORM-PROGRAM MSG='ETBSTOP',JOB-IDENTIFICATION=*TSN(TSN=tsn)
```

where *tsn* is the task number associated with the broker main task (in the example above the TSN of job name ETB)

All other tasks that were created as a result of starting the broker will be stopped automatically.

⟫ **To stop the broker from an operator console**

■ Enter the following command:

```
/INTR tsn,ETBSTOP
```

where *tsn* is the task number associated with the broker main task (in the example above the TSN of job name ETB)

All other tasks that were created as a result of starting the broker will be stopped automatically.

⟫ **To stop the broker from a non-privileged user ID**

■ Use the S-procedure `STOP-BROKER` in EXX103.JOBS

Startup Parameter	Description	Default
BROKER-ID	Depending on the communication method, the Broker ID can be specified in two different formats: ■ **TCP Transport Method** `ip:port:TCP` where *ip* is the address or DNS host name, *port* is the port number that EntireX Broker is listening on, and TCP is the protocol name	none

Startup Parameter	Description	Default
	■ **NET Transport Method** `ETBnnn:SVCmmm:NET` where *nnn* is the ID under which EntireX Broker is connected to the Adabas ID table, *mmm* is the SVC number under which the Adabas ID table can be accessed, and `NET` is the protocol name	
`ADABAS-PARAMETERS`	Adabas parameters used for NET communication method.	ETB-ADAPARM
`USERID`	If EntireX Broker is running with EntireX Security, a user ID needs to be supplied.	none
PASSWORD	If EntireX Broker is running with EntireX Security, a password needs to be supplied.	none
EXX-LIB	EntireX Broker module library.	EXX103.LIB
EXX-JOBS	EntireX Broker jobs library.	EXX103.JOBS
WAL-MOD	WAL module library.	WAL842.MOD

Set the broker ID in the `PARAMETER-DECLARATION` section and enter following command:

```
/CALL-PROCEDURE (EXX103.JOBS, STOP-BROKER)
```

Creating a Broker Persistent Store (optional)

This step may be skipped if no persistent store is required. See *Managing the Broker Persistent Store* for more information. The persistent store resides in an Adabas database, which means that Adabas must be installed. See *BS2000 Prerequisites*.

Customize job `CREATE-PSTORE-ADABAS`. Add the appropriate values in the `PARAMETER-DECLARATION` section and run the job.

```
/CALL-PROCEDURE (LIB=EXX103.JOBS,ELE=CREATE-PSTORE-ADABAS)
```

This Adabas `ADALOD` job loads the Adabas persistent store file FDT from EXX103.SYSF into the database where the Adabas persistent store is to reside.

Set the relevant attributes in the broker attribute file ETB-ATTR. See *Adabas-specific Attributes* and *Managing the Broker Persistent Store*.

> **Note:** The Adabas persistent store requires the ADALNK parameter `IDTNAME` in `ETB-ADAPARM` to be set properly. This must be the same ID table as used by the broker. See *Customize the EntireX Broker Attribute File* and *Customize the ADALNK Parameters File*.

Set up the EntireX Broker Security Server for BS2000 (optional)

≫ To activate authentication

■ Set the following two parameters in the broker attribute file to switch on security:

■ **In the DEFAULTS=BROKER section**

```
SECURITY=YES
```

■ **In the DEFAULTS=SECURITY section**

```
ACCESS-SECURITY-SERVER=YES
```

The Broker Security Server requires administrator rights and must be run under a privileged user ID.

> **Note:** If the parameter `ACCESS-SECURITY-SERVER` is set to "NO", the broker itself must be run under a privileged user ID to allow authentication. In this case, the Broker Security Server is not needed.

≫ To start the Broker Security Server

1 Set up the correct broker library within `START-SECURITY-SERVER`, because the server task does not usually run under same user ID where the module library resides.

2 Issue the following command from a privileged user ID (TSOS) to run the server:

```
/ENTER-PROCEDURE *LIB(LIB=$kkk.EXXnnn.JOBS,     -
/                    ELE=START-SECURITY-SERVER),  -
/                    JOB-NAME=SECUSERV,LOG=*NO
```

where `$kkk` is the user ID under which the broker library resides.

≫ To stop the Broker Security Server from a privileged user ID

■ Enter:

```
/INFORM-PROGRAM MSG='EOJ',JOB-IDENTIFICATION=*TSN(TSN=tsn)
```

where <*tsn*> is the BS2000 task number associated with the server.

> **To stop the Broker Security Server from an operator console**

■   Enter:

```
/INTR tsn,EOJ
```

where *tsn* is the BS2000 task number associated with the server.

> **To stop the Broker Security Server from a non-privileged user ID**

■   Enter the following SDF command:

```
/CALL-PROCEDURE (EXX103.JOBS, STOP-SECURITY-SERVER)
```

> **Note:** This works from all user IDs in the system.

Verifying the Installation of the Broker

> **To test the EntireX Broker installation**

1   The client and server programs that are executed by the installation verification need the following to be defined in the Broker attribute file (if the delivered attribute file is used, no changes are needed):

```
CLASS = ACLASS, SERVER = ASERVER, SERVICE = ASERVICE
```

See *Broker Attributes* for more information.

2   Start EntireX Broker. See *Starting and Stopping the Broker* in the BS2000 Administration documentation.

3   The procedures to execute the sample programs are located in library EXX103.JOBS. They, in turn, execute the `BCOC` (client) and `BCOS` (server) test programs. Customize the verification jobs before you run them. `BCOC` and `BCOS` require a `BROKER-ID` to be set up in the job control's `PARAMETER-DECLARATION` section. Since they may access an Adabas ID table (communication method XCOM), an `IDTNAME` must be provided. This `IDTNAME` is read from the `ETB-ADAPARM` element (see *Customize the ADALNK Parameters File*).

4   Start `BCOS` server program using following command:

```
/CALL-PROCEDURE (LIB=EXX103.JOBS,ELE=BCOS)
```

Wait until `BCOS` has registered the service.

5   Then start the `BCOC` client program in a different session using following command:

```
/CALL-PROCEDURE (LIB=EXX103.JOBS,ELE=BCOC)
```

Since `BCOS` sets a `WAIT` value of 60 seconds, it will terminate with a "WAIT timeout" if `BCOC` is not started within this time frame. See `WAIT` under *Broker ACI Fields*.

6   Both should now run through 10 iterations of exchanging messages.

If the initialization messages appear but the verification program fails, check for either a TCP port conflict or a problem with the ID table name or version.

Sample Programs for Client (BCOC) and Server (BCOS)

The programs `BCOC` and `BCOS` are client and server programs provided for test purposes.

BCOC Parameters

Parameter	Description	ACI Field
`-a`*`locale`*	*`locale_string`*,*`environment`*	`LOCALE-STRING,` `ENVIRONMENT`
`-b`*`broker_id`*	Broker ID (same as the database ID). This is an EntireX Broker configuration parameter. Example: `BCOC -bETB045`	`BROKER-ID`
`-c`*`class`*	Server class (part of the server definition). Default is `ACLASS`. Value must be configured in the *Broker Attributes*. Together with `server` and `service` it forms the fully qualified server name.	`SERVER-CLASS`
`-g`*`input_file_name`*	Use a file as send buffer.	
`-h`	Display usage information.	
`-i`*`num_messages[,num_uows]`*	If *`num_uows`* is not specified, the number of messages sent. If *`num_uows`* is specified, the number of UOWs and messages therein sent. Example: `bcoc -i10`	
`-j`*`reply_error`*	Test `REPLY-ERROR`.	
`-k`*`conversation_flag`*	Using conversational mode.	
`-l`	Silent mode. Suppress output.	
`-n`*`token`*	Specify the security token, if desired.	`TOKEN`

Parameter	Description	ACI Field
`-p`*`message_size`*	The size, in bytes, of the packets that will be transmitted. Used to specify send and receive length in the broker API. Example: `BCOC -p10000`	
`-r`*`compress`*	Compression level. Values: `N|Y|0-9`.	`COMPRESSLEVEL`
`-s`*`server`*	Name of server. Default is `ASERVER`. Value must be configured in *Broker Attributes*. Together with `class` and `service` it forms the fully qualified server name.	`SERVER-NAME`
`-t`*`wait_time`*	Set the wait time.	`WAIT`
`-u`*`user_id`*	User ID. With a secure broker, this will be used to perform authentication.	`USER-ID`
`-v`*`service`*	Name of service (part of the server definition). The default is `ASERVICE`. Value must be configured in *Broker Attributes*. Together with `class` and `server` it forms the fully qualified server name.	`SERVER-NAME`
`-w`*`password`*	Password. With a secure broker, this will be used to perform authentication.	`PASSWORD`
`-y`*`sleep_time`*	Delay between messages in seconds.	
`-z`*`store`*`[`*`uwstatp`*`[,`*`uowtime`*`]]`	Test unit of work.	
`-1..12`	By default, if omitted, the highest possible API version is used. Note that if you specify the API version, certain features might depend on a minimum API version.	`API-VERSION`

BCOS Parameters

Most `BCOC` parameters apply to `BCOS`, too. The following table lists additional parameters or parameters that have a different meaning.

Parameter	Description	ACI Field
`-g`*`output-file-name`*	Name of output file for receive buffer (if not running in silent mode).	
`-i`*`num_messages`*`[,`*`num_uows`*`]`	If *`num_uows`* is not specified, the number of messages accepted. If *`num_uows`* is specified, the number of UOWs and messages therein that are accepted before deregistration. Example: `bcos -i10`	
`-i`*`unit_of_work_flag`*	Test unit of work.	

5 Installing the RPC Server for BS2000

The EntireX RPC Server for BS2000 allows standard RPC clients to communicate with RPC servers on the operating system BS2000. It supports the programming languages COBOL and C. This chapter covers the following topics:

For Natural RPC servers, see *Setting Up a Natural RPC Environment* in your Natural documentation.

Step 1: Define a Server-side Mapping Container

If you are using or plan to use server-side mapping files, you need to set up a server-side mapping container. A server-side mapping file is a Designer file with extension .svm. See *Server Mapping Files for COBOL*. If this step is omitted, the RPC server will start without the server-side mapping container. This means that server programs cannot make use of special COBOL syntax and features. See *When is a Server Mapping File Required?* in the Designer documentation.

The server-side mapping container stores the content of server-side mapping files, which are used at runtime to marshal and unmarshal the RPC data stream. This enables the RPC server to support special COBOL syntax. The server-side mapping container is technically an ISAM file that needs to be defined and initialized. Each RPC server requires its own server-side mapping container.

Customize S-procedure `CREATE-SVM-FILE` in `EXP103.JOBS`.

Procedure Parameter	Description	Default
EXP-LIB	RPC Server for BS2000 load library.	EXP103.LIB
SVM-FILE	Name of server-side mapping container (ISAM file) to store the contents of Designer server-side mapping files.	SVMFILE
SYSOUT-FILE	SYSOUT file name.	RPC.SYSOUT.CREASVM

The name of the server-side mapping container (ISAM file) must correspond to the `SVM-FILE` parameter specified in the RPC Server for BS2000 startup job control. See *Step 4: Customize the RPC Server for BS2000 Startup JCL*.

See also *Server-side Mapping Files* in the RPC Server for BS2000 documentation.

Step 2: Customize the RPC Server for BS2000 Configuration File RPC-CONFIG

RPC-CONFIG in EXP103.JOBS contains the RPC server parameters. If the default settings are used, only the `BROKERID` parameter needs to be set up according to your environment. The RPC Server for BS2000 will then run in a default configuration.

Depending on the communication method, the Broker ID has two formats:

■ **TCP Transport Method**

```
ip:port:TCP
```

where `ip`   is the address or DNS host name,

   `port` is the port number that EntireX Broker is listening on, and

   `TCP` is the protocol name

■ **NET Transport Method**

```
ETBnnn:SVCmmm:NET
```

where `nnn` is the ID under which EntireX Broker is connected to the Adabas ID table,

   `mmm` is the SVC number under which the Adabas ID table can be accessed, and

   `NET` is the protocol name

If the provided parameter file is used, the RPC server will run as a COBOL server. Besides the RPC service (`RPC/SRV1/CALLNAT`), it will register the DEPLOYMENT and the EXTRACTOR service. This corresponds to the delivered settings in the broker attribute file on BS2000 and other platforms. The server is configured to run with a fixed number of 3 worker task replicates. Security is turned off.

See also *Configuring the RPC Server*.

The default configuration file `RPC-CONFIG`:

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                     *
*           EntireX RPC Server v8.1 Configuration File                *
*                                                                     *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
* * * * * * * * * * EntireX Broker Parameters * * * * * * * * * * * * *
*
BROKERID=<ipaddr>:<port>:TCP                Broker ID if TCPIP is used
* BROKERID=ETB<nnnnn>::NET                  Broker ID if NET is used
*
SERVERNAME=SRV1
SERVICE=CALLNAT
CLASS=RPC
*
TIMEOUT=300                                 Seconds
LOGON=YES                                   EntireX Broker Logon
*
* CODEPAGE=EDF041
*
```

```
* KERNELSECURITY=YES                         EntireX Broker Security
* USERID=<userid>
* PASSWORD=<password>
*
*
* * * * * * * * * EntireX RPC Server Parameter  * * * * * * * * * * *
*
RESTARTCYCLES=3                              default is 15
* TRACELEVEL=NONE                            NONE, STANDARD or ADVANCED
*
* EntireX RPC Cobol Server Configuration
* -------------------------------------------------------------------
SVM=PREFERRED
DEPLOYMENT=YES
EXTRACTOR=YES
MARSHALLING=(LANGUAGE=COBOL)
*
* EntireX RPC C Server Configuration
* -------------------------------------------------------------------
* EXTRACTOR=YES
* MARSHALLING=(LANGUAGE=C)
*
* Start up a fixed number of workers
* -------------------------------------------------------------------
WORKERMODEL=(FIX,3)
*
* Balance the load of available workers
* -------------------------------------------------------------------
* WORKERMODEL=(SCALE,2,5)
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Step 3: Customize ADALNK Parameter File RPC-ADAPARM

The RPC Server for BS2000 requires ADALNK parameters when a local communication with an EntireX Broker on same machine is desired (XCOM communication). Set up the IDTNAME in RPC-ADAPARM in EXP103.JOBS under which the broker has registered at the Adabas ID table.

Parameter	Description
ADALNK IDTNAME=ADA<xxxxx>	Specifies the ID table name under which EntireX Broker is accessible.

Step 4: Customize the RPC Server for BS2000 Startup JCL

The job control delivered with RPC Server for BS2000 makes use of BS2000 S-procedures. We strongly recommend you do not modify `START-RPC-SERVER`. This procedure is recursively called to establish the RPC server environment. Modifications should only be done to the `PARAMETER-DECLARATION` section as described below. If all RPC server components are installed under the same BS2000 user ID and all previous installation steps have been done using the default settings, no modifications to `START-RPC-SERVER` are necessary to run a COBOL server and execute the delivered examples.

Procedure Parameter	Description	Default
EXP-JOBS	RPC Server jobs library.	EXP103.JOBS
EXP-LIB	RPC Server load library.	EXP103.LIB
EXX-LIB	EntireX Broker load library.	EXX103.LIB
WAL-MOD	WAL library.	WAL842.MOD
PROGRAM-LIB	Server module library. Additional server module libraries can be included in the BLSLIB chain. (See the `SET UP FILE LINKS` section in the `START-RPC-SERVER` job control.) There are two sample server libraries delivered: EXP103.COBS COBOL sample server module library. See *Client and Server Examples for BS2000* in the COBOL Wrapper documentation. EXP103.CSRV C sample server module library.	EXP103.COBS
ADABAS-PARAMETERS	Adabas parameters used for XCOM communication method.	RPC-ADAPARM
RPC-CONFIG-FILE	RPC configuration parameter file.	RPC-CONFIG
SVM-FILE	Server-side mapping container (ISAM file) to store server-side mapping files.	SVMFILE
PROC-NAME	The name of the `START-RPC-SERVER` procedure.	START-RPC-SERVER
WORKER-JOB-NAME	Job name of the worker tasks.	RPCWORK
LOG-FILE-PREFIX	File name prefix used for the SYSOUT files of the main and worker tasks. The following SYSOUT files are generated: for the main task: `log-file-prefix.tsn`.RPCMAIN for each worker task: `log-file-prefix.tsn`.RPCWORK	RPC.
WORKER-JOB-CLASS	Job class of the worker tasks	*STD

Procedure Parameter	Description	Default
WORKER-CPU-LIMIT	CPU limit of the worker tasks. If this parameter is set to *NO, the user ID requires the permission to run jobs with TIME='NTL' in the job class assigned	*STD
CRTE-LIB	BS2000 Common Runtime Environment (CRTE) library	$.SYSLNK.CRTE
STUB-TRACE-LEVEL	Trace level of the EntireX Broker stub. For diagnostic purposes it can be set to 1, 2 or 3	0
LOGGING	The logging parameter is passed to the worker task job control and the SYSJ elements executed	*NO
WORKER-PARMS	Is filled during runtime with the worker task parameters. Note: Do *not* modify it !	*RPC
MAIN-TASK-TSN	Is filled during runtime with the main task TSN. Note: Do *not* modify it!	*RPC

Starting the RPC Server

≫ **To start the RPC Server for BS2000**

■ Use the following SDF command:

```
/ENTER-PROCEDURE *LIB(LIB=EXP103.JOBS,ELE=START-RPC-SERVER), -
/JOB-NAME=RPCMAIN,LOG=*NO
```

Stopping the RPC Server

≫ **To stop the RPC Server for BS2000 from a privileged user ID**

■ Enter the command:

```
/INFORM-PROGRAM MSG='STOP',JOB-IDENTIFICATION=*TSN(TSN=tsn)
```

where `tsn` is the task number associated with the RPC Server for BS2000 main task (in the example above the TSN of `RPCMAIN`)

All other tasks that were created as a result of starting the RPC Server for BS2000 will be stopped automatically.

> **To stop the RPC Server for BS2000 from an operator console**

■ Enter the command:

```
/INTR tsn,STOP
```

where `tsn` is the task number associated with the RPC Server for BS2000 main task (in the example above the TSN of `RPCMAIN`)

All other tasks that were created as a result of starting the RPC Server for BS2000 will be stopped automatically.

> **To stop the RPC Server for BS2000 from a non-privileged user ID**

■ Use S-procedure STOP-RPC-SERVER in EXP103.JOBS.

Startup Parameter	Description	Default

BROKER-ID	Depending on the communication method, the broker ID can be specified in two different formats:
	
	■ **TCP Transport Method**
	
	`ip:port:TCP`
	
	where `ip` is the address or DNS host name,
	    `port` is the port number that EntireX Broker is listening on, and
	    `TCP` is the protocol name
	■ **NET Transport Method**
	
	`ETBnnn:SVCmmm:NET`
	
	where `nnn` is the ID under which EntireX Broker is connected to the Adabas ID table,
	    `mmm` is the SVC number under which the Adabas ID table can be accessed, and
	    `NET` is the protocol name
		none

CLASS	The class name under which the RPC server is registered at the EntireX Broker.	RPC

Startup Parameter	Description	Default
SERVER	The server name under which the RPC server is registered at the EntireX Broker.	SRV1
SERVICE	The service name under which the RPC server is registered at the EntireX Broker.	CALLNAT
USERID	If EntireX Broker is running with EntireX Security, a user ID needs to be supplied	none
PASSWORD	If EntireX Broker is running with EntireX Security, a password needs to be supplied	none
EXX-JOBS	EntireX Broker jobs library	EXX103.JOBS
EXX-LIB	EntireX Broker module library	EXX103.LIB
WAL-MOD	WAL module library	WAL842.MOD

Set the broker ID in the `PARAMETER-DECLARATION` section and enter following command:

```
/CALL-PROCEDURE (EXP103.JOBS, STOP-RPC-SERVER)
```

Verifying the Installation of the RPC Server for BS2000

On platform BS2000 the basic COBOL RPC server example `CALC` is also delivered in LMS library *EXP103.COBS*. To verify the installation, use this example together with the *EntireX IDL Tester*. The example also corresponds to the RPC examples delivered on other platforms, for example UNIX and Windows. For more information, see *CALC* under *Client and Server Examples for BS2000* in the COBOL Wrapper documentation.

6 Installing EntireX Security under BS2000

Installing EntireX Security for Broker Kernel

This section describes the steps for installing EntireX Security for Broker kernel under BS2000. The installation procedure has the following steps:

- Modify the Broker Attribute File
- Start (Restart) Broker Kernel

Modify the Broker Attribute File

≫ To modify the Broker attribute file

1   Insert the following parameter in the section `DEFAULTS=BROKER` of the Broker attribute file:

```
SECURITY=YES
```

2   Modify the `DEFAULTS=SECURITY` section of the Broker attribute file according to your requirements. These parameters are used to adjust the security settings. See *Security-specific Attributes*. Authorization checks are currently not available.

> **Note:** Setting `SECURITY=YES` will load the provided load module USRSEC from the EXX load library assigned by LINK-NAME ETBLIB. This module will perform privileged operations, such as executing the SRMUINF macro for various users, and requires Broker running under TSOS.

Start (Restart) Broker Kernel

The Broker must be restarted to pick up changes to the Broker attribute file and to initialize Broker kernel under BS2000 to perform security checks.

Basic installation of EntireX Security for Broker kernel is now complete.

Installing EntireX Security for Applications Using Broker

This section describes the steps for installing EntireX Security for Broker stub under BS2000.

> **Note:** If you are running your application(s) with ACI version 7 or below, the following steps are required to install EntireX Security for the Broker stubs in all environments where applications execute either as clients or servers. These steps are not required if you are running your application(s) with ACI version 8 or above.

Link the Security Components

For applications running on BS2000 using ACI 7 or below, the Broker stub security component `NA2PETS` must be linked with the stub `BROKER`. In addition, `LLM SECUEXIT` must be made available. The following steps are required:

■ Relink all applications that contain stub `BROKER` to include module `NA2PETS`.

■ Assign the EXX load library by using an `ADD-FILE-LINK` statement with `LINK-NAME ETBUSER`.

> **Notes:**

1. These steps are needed for backward compatibility if your applications issue any commands using ACI version 7 or below. Applications using ACI version 8 or above do not require these additional components in the stub.

2. For ACI version 7 or below, these components must be added to the stub environment utilized by the application.

Installation of EntireX Security for Broker stubs is now complete. Now you can install the security components for the Broker stubs on the remaining operating systems where your application components are located.