English

FUJITSU Software BS2000

interNet Services V3.4B

Administrator Guide

Edition June 2017

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Contents

1 Preface

The interNet Services product supplements the TCP/IP functionality of openNet Server with the following standards:

- DNS Resolver and Server
- NTP Client and Server
- FTP Client and Server
- TELNET Client and Server
- OpenSSH
- Mail Sender in BS2000
- Mail Reader in BS2000
- Mail Server in POSIX

1.1 Target group of the manual

This Administrator Guide is intended for BS2000 system administrators who want to install and operate interNet Services on BS2000 OSD/BC. Knowledge of the BS2000 OSD/BC operating system and the basic concepts of TCP/IP is therefore assumed. Apart from this administrator's manual, there is also a User Guide for interNet Services, which should be available to system administrators in addition to this manual.

1.2 Summary of contents

This manual is arranged as follows:

- Chapter 2: Installing Internet Services without FTP, TELNET and Mail.

  This chapter describes the installation of the DNS, NAMED, openSSH and NTP services as POSIX program packages using the POSIX installation routine.

- Chapter 3: Installing FTP and TELNET

  This chapter describes the installation of the FTP and TELNET services via SDF command.

- Chapter 4: Configuration and operation of FTP

  This chapter describes the TLS/SSL support in the FTP server and configuration using via installation command or an option file. It also describes the startup, termination, operation and server exits of the FTP server.

- Chapter 5: Configuration and operation of TELNET

  This chapter describes the TLS/SSL support in the TELNET server and configuration using an installation command or an option file. It also describes the startup, termination, operation and server exits of the TELNET server.

- Chapter 6: Generating random numbers

  This chapter describes how random numbers are generated in BS2000 and POSIX.

- Chapters 7 - 11

  These chapters present the individual components of the interNet Services in detail. The main topics discussed here are the functionality of the components, configuration, operational notes and diagnostic options.

- Chapter 12: Specification of a cipher suite preference list

  This chapter centrally describes the format of cipher mnemonics for the preference list specification.

1.3 Licensing regulations

The licensing regulations for the OpenSSL package and the TLS-FTP patch of Peter 'Luna' Runestig are printed below.

```
LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSDstyle
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License
---------------

/* ====================================================================
 * Copyright (c) 1998-2003 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
```

```
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given
attribution
 * as the author of the parts of the library used.
```

```
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the
library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an
acknowledgement:
 *    "This product includes software written by Tim Hudson
(tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply
be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

```
/*
 * Copyright (c) 1999 - 2002 Peter 'Luna' Runestig <peter@runestig.com>
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modifi-
 * cation, are permitted provided that the following conditions are met:
 *
 *   o  Redistributions of source code must retain the above copyright
notice,
 *      this list of conditions and the following disclaimer.
 *
 *   o  Redistributions in binary form must reproduce the above copyright no-
 *      tice, this list of conditions and the following disclaimer in the do-
 *      cumentation and/or other materials provided with the distribution.
 *
 *   o  The names of the contributors may not be used to endorse or promote
 *      products derived from this software without specific prior written
 *      permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED
 * TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE
LI-
 * ABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUEN-
 * TIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEV-
 * ER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABI-
 * LITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF
 * THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
 */


NTP Licence
===========

*********************************************************************
*                                                                   *
* Copyright (c) University of Delaware 1992-2015                    *
*                                                                   *
* Permission to use, copy, modify, and distribute this software and   *
```

```
* its documentation for any purpose with or without fee is hereby     *
* granted, provided that the above copyright notice appears in all    *
* copies and that both the copyright notice and this permission       *
* notice appear in supporting documentation, and that the name        *
* University of Delaware not be used in advertising or publicity      *
* pertaining to distribution of the software without specific,        *
* written prior permission. The University of Delaware makes no       *
* representations about the suitability this software for any         *
* purpose. It is provided "as is" without express or implied          *
* warranty.                                                           *
*                                                                     *
***********************************************************************
Content starting in 2011 from Harlan Stenn, Danny Mayer, and Martin Burnicki
is:

***********************************************************************
*                                                                     *
* Copyright (c) Network Time Foundation 2011-2015                     *
*                                                                     *
* All Rights Reserved                                                 *
*                                                                     *
* Redistribution and use in source and binary forms, with or without  *
* modification, are permitted provided that the following conditions  *
* are met:                                                            *
* 1. Redistributions of source code must retain the above copyright   *
*    notice, this list of conditions and the following disclaimer.    *
* 2. Redistributions in binary form must reproduce the above          *
*    copyright notice, this list of conditions and the following      *
*    disclaimer in the documentation and/or other materials provided  *
*    with the distribution.                                           *
*                                                                     *
* THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND ANY EXPRESS  *
* OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED   *
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  *
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE    *
* LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR *
* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT   *
* OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR  *
* BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF          *
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT           *
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE   *
* USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH    *
* DAMAGE.                                                             *
***********************************************************************
```

1.4 Overview of interNet Services

interNet Services V3.4B requires the following software:

– BS2000/OSD ≥ V9.0A
– openNet Server ≥ V3.6

The following port numbers are used by default by the components included in interNet Services:

Port		**Protocol**	**Explanation**
20	tcp	FTP	File Transfer Protocol [default data]
21	tcp	FTP	File Transfer Protocol [control]
22	tcp	OpenSSH	Secure Shell
23	tcp	TELNET	TELNET
25	tcp	SMTP	Simple Mail Transfer Protocol
53	tcp/udp	DNS	Domain Name Server
80	tcp	HTTP	World Wide Web HTTP
110	tcp	POP3	Post Office Protocol - Version 3
123	udp	NTP	Network Time Protocol
143	tcp	IMAP	Internet Message Access Protocol
443	tcp	HTTPS	HTTP over TLS (Transport Layer Security) / SSL (Secure Sockets Layer)
953	tcp/udp	DNS	MDC tool (NAMED)
1235	tcp/udp	DNS	Domain Name Resolver

1.5 Changes compared to the previous version

The following list of changes provides an overview of what is new in interNet Services V3.4B and relevant to this manual.

The changes that affect the interNet Services User Guide are listed in the corresponding section of that manual.

OpenSSH

- The SSH protocol version 1 is no longer supported.
- Elliptic Curve Cryptography (ECDSA, Ed25519) is now supported.

SSL/TLS protocol versions and cipher suites for FTP, TELNET, Mail sender, and Mail reader

- Discontinuation of SSLv2
- Extension to include TLSv1.1 and TLSv1.2.
- New cipher suites for the SSL/TLS cipher suites

FTP

- New FTP option *-TVFS* for enabling/disabling TVFS.
- New commands in subevent in *FTPCMD* as a result of the new FTP server commands MLSD and MLST.

Telnet

- New value *optional* in the START-TLS option *Z tls-required*.

Mail senders in BS2000

- New options *maxQueueLifeTime*, *smtpReadMaxWaitTime*, *smtpRetryTimeBase*, and *smtpRetryTimeMaxExp* to control the maximum life expectancy of an e-mail and the waiting times for repeated dispatch attempts.
- The *retryLimit* option is no longer supported. It is replaced by the *maxQueueLifeTime* option.

Discontinued functions

- Discontinuation of crypto hardware
  Crypto hardware is no longer supported. As a result, the *tlsUseCryptoHardware* and *-Z UseCryptoHardware* options, as well as the *USE-CRYPTO-HARDWARE* configuration parameter in the SET-FTP-TELNET-PARAMETERS command are no longer available.
- The POSIX *prngd* daemon is no longer provided and is therefore not described. You can generate random numbers instead using the BS2000 PRNGD.

Modified manual structure

- The structure of the cipher mnemonics for creating preference list specifications is now described centrally in chapter “Specification of a cipher suite preference list” on page 375.

1.6 Notational conventions

This manual uses the following notational conventions:

italics
: denote file names, program names, names of management windows, parameter names, menu titles and menu options as well as commands and variables in the main body of text.

<angle brackets>
: identify variables for which you have to enter values.

[square brackets]
: indicates optional input.

{braces} ...
: indicate a list of aternatives which are separated by "|".

`fixed-pitch font`
: denotes input for the system, system output and file names in examples.

command
: Elements (names of commands and parameters) of the syntax description for commands that must be entered unchanged are highlighted in bold.


For informative texts


CAUTION!
For warnings

References

References within the manual include the page concerned and the section or chapter as required. References to topics described in other manuals include the short title of the manual. The full title can be found in the list of related publications.

SDF syntax representation

Metasyntax

Representation	Meaning	Examples
UPPERCASE LETTERS	Uppercase letters denote keywords (command, statement or operand names, keyword values) and constant operand values. Keyword values begin with *.	**HELP-SDF** **SCREEN-STEPS** = ***NO**
UPPERCASE LETTERS in boldface	Uppercase letters printed in boldface denote guaranteed or suggested abbreviations of keywords.	**GUID**ANCE**-MODE** = ***Y**ES
=	The equals sign connects an operand name with the associated operand values.	**GUID**ANCE**-MODE** = ***NO**
< >	Angle brackets denote variables whose range of values is described by data types and suffixes (see Tables 2 and 3).	**SYNTAX-F**ILE = <filename 1..54>
Underscoring	Underscoring denotes the default value of an operand.	**GUID**ANCE**-MODE** = ***NO**
/	A slash serves to separate alternative operand values.	**NEXT-FIELD** = ***NO** / ***Y**ES
(...)	Parentheses denote operand values that initiate a structure.	,**UNGUID**ED**-DIA**LOG = ***Y**ES (...) / ***NO**
[ ]	Square brackets denote operand values which introduce a structure and are optional. The subsequent structure can be specified without the initiating operand value.	**SELECT** = [***BY-ATTR**IBUTES](...)
Indentation	Indentation indicates that the operand is dependent on a higher-ranking operand.	,**GUID**ED**-DIA**LOG = ***Y**ES (...) ***Y**ES(...) **SCREEN-STEPS** = ***NO** / ***Y**ES

Table 1: Metasyntax (part 1 of 2)

Representation	Meaning	Examples
|	A vertical bar identifies related operands within a structure. Its length marks the beginning and end of a structure. A structure may contain further structures. The number of vertical bars preceding an operand corresponds to the depth of the structure.	**SUP**PORT = ***TAPE**(...) ***TAPE(...)** | **VOL**UME = ***ANY**(...) | ***ANY**(...) | | ...
,	A comma precedes further operands at the same structure level.	**GUID**ANCE**-MODE** = ***NO** / ***Y**ES ,**SDF-COM**MANDS = ***NO** / ***Y**ES
list-poss(n):	The entry “list-poss” signifies that a list of operand values can be given at this point. If (n) is present, it means that the list must not have more than n elements. A list of more than one element must be enclosed in parentheses.	list-poss: ***SAM** / ***ISAM** list-poss(40): <structured-name 1..30> list-poss(256): ***OMF** / ***SYSLST**(...) / <filename 1..54>
Alias:	The name that follows represents a guaranteed alias (abbreviation) for the command or statement name.	**HELP-SDF** Alias: **HPSDF**

Table 1: Metasyntax (part 2 of 2)

Data types

Data type	Character set	Special rules
alphanum-name	A…Z 0…9 $, #, @	
cat-id	A…Z 0…9	Not more than 4 characters; must not begin with the string PUB
command-rest	freely selectable	
composed-name	A…Z 0…9 $, #, @ hyphen period catalog ID	Alphanumeric string that can be split into multiple substrings by means of a period or hyphen. If a file name can also be specified, the string may begin with a catalog ID in the form :cat: (see data type filename).
c-string	EBCDIC character	Must be enclosed within single quotes; the letter C may be prefixed; any single quotes occurring within the string must be entered twice.
date	0…9 Structure identifier: hyphen	Input format: yyyy-mm-dd jjjj: year; optionally 2 or 4 digits mm: month tt: day
device	A…Z 0…9 hyphen	Character string, max. 8 characters in length, corresponding to a device available in the system. In guided dialog, SDF displays the valid operand values. For notes on possible devices, see the relevant operand description.
fixed	+, - 0…9 period	Input format: [sign][digits].[digits] [sign]: + oder - [digits]: 0...9 must contain at least one digit, but may contain up to 10 characters (0...9, period) apart from the sign.

Table 2: Data types (part 1 of 6)

Data type	Character set	Special rules
filename	A…Z 0…9 $, #, @ hyphen period	Input format: [:cat:][$user.] { file | file(no) | group | group { (*abs) | (+rel) | (-rel) } } :cat: optional entry of the catalog identifier; character set limited to A...Z and 0...9; maximum of 4 characters; must be enclosed in colons; default value is the catalog identifier assigned to the user ID, as specified in the user catalog. $user. optional entry of the user ID; character set is A…Z, 0…9, $, #, @; maximum of 8 characters; first character cannot be a digit; $ and period are mandatory; default value is the user's own ID. $. (special case) system default ID file file or job variable name; may be split into a number of partial names using a period as a delimiter: $name_1$[.$name_2$[...]] $name_i$ does not contain a period and must not begin or end with a hyphen; file can have a maximum length of 41 characters; it must not begin with a $ and must include at least one character from the range A...Z.

Table 2: Data types (part 2 of 6)

Data type	Character set	Special rules
filename (contd.)		#file (special case) @file (special case) # or @ used as the first character indicates temporary files or job variables, depending on system generation. file(no) tape file name no: version number; character set is A...Z, 0...9, $, #, @. Parentheses must be specified. group name of a file generation group (character set: as for “file”) group{(*abs) (+rel) (-rel)} (*abs) absolute generation number (1-9999); * and parentheses must be specified. (+rel) (-rel) relative generation number (0-99); sign and parentheses must be specified.
integer	0…9, +, -	+ or -, if specified, must be the first character.
name	A…Z 0…9 $, #, @	Must not begin with 0...9.

Table 2: Data types (part 3 of 6)

Data type	Character set	Special rules
partial-filename	A…Z 0…9 $, #, @ hyphen period	Input format: [:cat:][$user.][partname.] :cat: see filename $user. see filename partname optional entry of the initial part of a name common to a number of files or file generation groups in the form: $name_1$.[$name_2$.[...]] $name_i$ (see filename). The final character of “partname” must be a period. At least one of the parts :cat:, $user. or partname must be specified.
posix-filename	A...Z 0...9 special characters	String with a length of up to 255 characters; consists of either one or two periods or of alpha-numeric characters and special characters. The special characters must be escaped with a preceding \ (backslash); the / is not allowed. Must be enclosed within single quotes if alter-native data types are permitted, separators are used, or the first character is a ?, ! or ^. A distinction is made between uppercase and lowercase.
posix-pathname	A...Z 0...9 special characters structure identifier: slash	Input format: [/]$part_1$/.../$part_n$ where $part_i$ is a posix-filename; max. 1023 characters; must be enclosed within single quotes if alter-native data types are permitted, separators are used, or the first character is a ?, ! or ^.

Table 2: Data types (part 4 of 6)

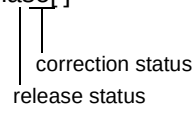
Data type	Character set	Special rules
product-version	A…Z 0…9 period single quote	Input format: [[C]'][V][m]m.naso[']  where m, n, s and o are all digits and a is a letter. Whether the release and/or correction status may/must be specified depends on the suffixes to the data type (see suffixes without-corr, without-man, mandatory-man and mandatory-corr in Table 3). product-version may be enclosed within single quotes (possibly with a preceding C). The specification of the version may begin with the letter V.
structured-name	A…Z 0…9 $, #, @ hyphen	Alphanumeric string which may comprise a number of substrings separated by a hyphen. First character: A...Z or $, #, @
text	freely selectable	For the input format, see the relevant operand descriptions.
time	0…9 structure identifier: colon	Time-of-day entry: Input format: { hh:mm:ss / hh:mm / hh } hh: hours mm: minutes ss: seconds } Leading zeros may be omitted
vsn	a) A…Z 0…9 b) A…Z 0…9 $, #, @	a) Input format: pvsid.sequence-no max. 6 characters pvsid: 2-4 characters; PUB must not be entered sequence-no: 1-3 characters b) Max. 6 characters; PUB may be prefixed, but must not be followed by $, #, @.

Table 2: Data types (part 5 of 6)

Data type	Character set	Special rules
x-string	Hexadecimal: 00…FF	Must be enclosed in single quotes; must be prefixed by the letter X. There may be an odd number of characters.
x-text	Hexadecimal: 00…FF	Must not be enclosed in single quotes; the letter X must not be prefixed. There may be an odd number of characters.

Table 2: Data types (part 6 of 6)

Suffixes for data types

Suffix	Meaning
x..y *unit*	With data type “integer”: interval specification x minimum value permitted for “integer”. x is an (optionally signed) integer. y maximum value permitted for “integer”. y is an (optionally signed) integer. *unit* with “integer” only: additional units. The following units may be specified: *days* *byte* *hours* *2Kbyte* *minutes* *4Kbyte* *seconds* *Mbyte* milliseconds
x..y *special*	With the other data types: length specification For data types catid, date, device, product-version, time and vsn the length specification is not displayed. x minimum length for the operand value; x is an integer. y maximum length for the operand value; y is an integer. x=y the length of the operand value must be precisely x. *special* Specification of a suffix for describing a special data type that is checked by the implementation. “special” can be preceded by other suffixes. The following specifications are used: *arithm-expr* arithmetic expression (SDF-P) *bool-expr* logical expression (SDF-P) *string-expr* string expression (SDF-P) *expr* freely selectable expression (SDF-P) *cond-expr* conditional expression (JV) *symbol* CSECT or entry name (BLS)

Table 3: Data type suffixes

1.7 README files

The functional changes to the current product version and revisions to this manual are described in the product-specific Readme file.

Readme files are available to you online in addition to the product manuals under the various products at *http://manuals.ts.fujitsu.com*. You will also find the Readme files on the Softbook DVD.

Information under BS2000

When a Readme file exists for a product version, you will find the following file on the BS2000 system:

`SYSRME.<product>.<version>.<lang>`

This file contains brief information on the Readme file in English or German (<lang>=E/D). You can view this information on screen using the `SHOW-FILE` command or an editor. The `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` command shows the user ID under which the product's files are stored.

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available online at *http://manuals.ts.fujitsu.com*.

2 Installing/uninstalling the services without FTP, TELNET and Mail

Please read the Release Notice supplied with the product in addition to this chapter.

You will find further information on the installation/uninstallation of the various interNet Services in the corresponding description.

2.1 Installation

The individual components of the interNet Services software package are installed by the POSIX package installation program like any POSIX program package (see the manual "POSIX Basics for Users and System Administrators").

The interNet Services V3.4B software package must be installed component by component. The DNS, NAMED, NTP and OpenSSH components are available for selection.

If you are installing interNet Services for the first time, you will need to customize the configuration files of the individual components to your specific requirements as described in the relevant chapters.

If interNet Services V3.4B is to replace a previously installed version, please uninstall the old version first.

If you uninstall an existing installation and then reinstall it ("new installation"), the modified configuration files will be saved in the */etc/tcpipsv* directory. During installation these backup copies are as a rule automatically placed in the active */etc* directory again. For details of this, see the detailed descriptions of the various components.

After installing the interNet Services and before calling the individual services, the configuration files of the individual services must be customized to individual requirements. Normally, this is done by editing the files.

If the POSIX subsystem is stopped and then restarted, the interNet Services daemons that have not been deactivated are started automatically.

Example

```
/START-POSIX-INSTALLATION
```

Function: Installs POSIX program packages (IMON support: Y)
Product name: TCP-IP-SV
Package name: DNS or NAMED or NTP or PRNGD or OPENSSH

The screen mask of the installation program for DNS installation is shown below. User entries are highlighted with **bold print**.

```
                        BS2000 POSIX package installation


  IMON support ?     :  Y   (y)  mandatory for official package
                            (n)  private package (SINLIB...)

  name of product    :  TCP-IP-SV
  package of product :  DNS                    (optional for certain products)

  version of product :              (format Vmm.n or mmn)

  correction state     :            (format aso, optional for IMON support)

  installation userid   :           (mandatory for no IMON support)


  The definition of a installation path is optional for this product.
  Please enter the full pathname of the wanted installation directory:
  /opt/TCP-IP-SV/dns



  install: DUE       help: F1      terminate: F2
--------------------------------------------------------------------------------
```

2.2 Uninstallation

As when installing the interNet Services components, uninstallation is also performed via the POSIX installation program under the TSOS ID. During uninstallation, a search is made for active interNet Services daemons, and these processes are then terminated. The termination of active daemons is logged in the syslog file */var/adm/syslog*. All files, links and procedures of the interNet Services components are then deleted.

During uninstallation, a number of configuration files are saved in the */etc/tcpipsv* directory. You will find more detailed information in the uninstallation descriptions of the different interNet Services.

2.3 Initial operation

The user can start the daemons of the components of interNet Services individually. The corresponding procedures are installed in the */usr/sbin* or */etc/init.d/* directory.

The start calls are:

- For DNS: */etc/init.d/TCP-IP-SV.dns start* or */etc/init.d/TCP-IP-SV.named start*
- For NTP: */etc/init.d/TCP-IP-SV.ntp start*
- For OpenSSH: */etc/init.d/TCP-IP-SV.openssh start*

Restart calls are also provided for all daemons and are required if a modified configuration file is to be read in during a session.

The restart calls are:

- For DNS: */etc/init.d/TCP-IP-SV.dns restart* or */etc/init.d/TCP-IP-SV.named restart*
- For NTP: */etc/init.d/TCP-IP-SV.ntp restart*
- For OpenSSH: */etc/init.d/TCP-IP-SV.openssh restart*

During the restart procedure, a check is carried out to establish whether the corresponding daemon has been started. If no active daemon is found, a normal restart is carried out.

2.4 Shutting down

The shutdown calls are:

- For DNS: */etc/init.d/TCP-IP-SV.dns stop* or */etc/init.d/TCP-IP-SV.named stop*
- For NTP: */etc/init.d/TCP-IP-SV.ntp stop*
- For OpenSSH: */etc/init.d/TCP-IP-SV.openssh stop*

The shutdown only applies until the POSIX subsystem is terminated. If an automatic restart is to be prevented when the POSIX subsystem is restarted, the daemons' programs must be made non-executable as described under the installation of the individual components. However, if required, a manual start can be performed with *mstart* instead of *start*.

All services installed on a component-by-component basis (DNS, NAMED, OPENSSH, NTP) are permanently deactivated by the POSIX uninstallation.

3 Installing FTP and TELNET

Please read the Release Notice supplied with the product in addition to this chapter.

Following the installation of the product files, another installation step is required for FTP and TELNET using the SDF command SET-FTP-TELNET-PARAMETERS. Configuration is also carried out when this step is executed.

When the SET-FTP-TELNET-PARAMETERS command is executed (see page 40), one option file each is generated for the FTP server and the TELNET server in which the FTP server parameters or TELNET server parameters are stored as options.

3.1 Installing FTP and TELNET via SDF command

The SDF command SET-FTP-TELNET-PARAMETERS offers the following functionality:

– Definition of parameters for FTP and TELNET servers and placing them in separate option files for the FTP and TELNET servers (see page 67 or page 148)

– Creation of ENTER files for the FTP and TELNET daemons

SET-FTP-TELNET-**PAR**AMETERS

FTP-SERVER-PROC=*NO / *CREATE(...)

***CRE**ATE(...)

JOB-NAME= *STD / <name 1..5>

, JOB-CLASS= *STD / <name 1..8>

, CPU-TIME= *STD / <integer 1..32767>

, PRIORITY**= *STD** / <integer 0..255>

, DEBUG= *STD / <integer 0..9>

, TRACE= *STD / <integer 0..9>

, MAXIMUM-**CONN**ECTIONS**= *STD** / <integer 1..900>

, STATION-**ID= *STD** / <integer 0..6>

, TRANSFER-**JOB-CLASS= *STD** / <name 1..8>

, TRANSFER-**CPU-TIME= *STD** / <integer 1..32767>

, TIMEOUT-VALUE**= *STD** / <integer 1..32767>

, SYSTEM-**EXIT-LEV**EL**= *STD**/<integer 0..3>

SET-FTP-TELNET-**PAR**AMETERS

, FTAC-SUPPORT**= *STD** / ***NO** / ***YES**(...)
***YES**(...)
LEVEL**= *STD** / <integer 1..2>
, JOB-CLASS= *STD / <name 1..8>
, ENTER-**F**ILE**= *STD** / <filename 1..54_without-generation-version>
, SERVER-INFORMATION-**F**ILE**= *STD** / <filename 1..54_without-generation-version>
,FTAC-USERID=*STD /<name 1..8>
, TLS-SUPPORT**= *STD** / ***NO** / ***YES**(...)
***YES**(...)
PROTOCOL= *STD / <text 1..80>
, CIPHER-SUITE= *STD / <text 1..80>
, RSA-CERTIFICATE-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, RSA-KEY-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, DSA-CERTIFICATE-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, DSA-KEY-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CA-CERTIFICATE-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CLIENT-CA-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CERT-CHAIN-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CA-REVOCATION-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, RANDOM-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, SSL-LIBRARY= *STD / ***NONE** / <filename 1..54_without-generation-version>
, VERIFY-CLIENT= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**
, VERIFY-DEPTH= *STD / <1..32767>
, SEC-CONTROL-CONN= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**
, SEC-DATA-CONN= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**
, ACCOUNTING= *STD / ***NO** / ***YES**(...)
***YES**(...)
FILE= *STD / <filename 1..54_without-generation-version>
, OPTION-FILE= *STD / <filename 1..54_without-generation-version>
, SELECTOR= *STD / <text 1..511>
, INITIAL-COMMANDS= *STD / <c-string 2..256>
, PORT-NUMBER= *STD / <integer 1..32767>
, SERVER-ENTER-FILE= *STD / <filename 1..54_without-generation-version>
, LOGGING-FILE= *STD / <filename 1..54_without-generation-version>
, ALLOW-TSOS-LOGIN= *STD / ***NO** / ***YES** / ***TLS**

SET-FTP-TELNET-**PAR**AMETERS

, TELNET-**SERVER-PROC= *NO / *CRE**ATE(...)

***CRE**ATE(...)

JOB-NAME= *STD / <name 1..5>

, JOB-CLASS= *STD / <name 1..8>

, CPU-TIME= *STD / <integer 1..32767>

, PRIORITY**= *STD** / <integer 0..255>

, DEBUG= *STD / <integer 0..9>

, TRACE= *STD / <integer 0..9>

, MAXIMUM-**CONN**ECTIONS**= *STD** / <integer 1..900>

, STATION-**ID= *STD** / <integer 0..6>

, ASCII-TABLE**= *STD** / <text 1..8>

, EBCDIC-TABLE**= *STD** / <name 1..8>

, TLS-SUPPORT= *STD / ***NO** / ***PARAMETERS(...)**

***PARAMETERS (...)**

OPTION= *STD / ***START-TLS** / ***AUTHENTICATION(...)**

***AUTHENTICATION(...)**

DEBUG= *STD / ***NO** / ***YES**

, PROTOCOL= *STD / <text 1..80>

, CIPHER-SUITE = *STD / <text 1..80>

, RSA-CERTIFICATE-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, RSA-KEY-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, DSA-CERTIFICATE-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, DSA-KEY-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, CA-CERTIFICATE-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, CLIENT-CA-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, CERT-CHAIN--FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, CA-REVOCATION-FILE = *STD / ***NONE** / <filename 1..54_without-generation-version>

, RANDOM-FILE = *STD / <filename 1..54_without-generation-version>

, SSL-LIBRARY = *STD / ***NONE** / <filename 1..54_without-generation-version>

, VERIFY-CLIENT= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**

, VERIFY-DEPTH= *STD / <1..32767>

, ENCRYPTION= *STD **/ *NO / *YES(...)**

***YES(...)**

DEBUG= = *STD / ***NO** / ***YES**

, KEY= <x-text 1..16>

, SSL-LIBRARY = *STD / ***NONE** / <filename 1..54_without-generation-version>

SET-FTP-TELNET-**PAR**AMETERS
, OPTION-FILE= *STD / <filename 1..54_without-generation-version> **, SELECTOR= *STD /** <text 1..511> **, PORT-NUMBER= *STD** / <integer 0..32767> **, SERVER-ENTER-FILE= *STD** / <filename 1..54_without-generation-version> **, LOGGING-FILE= *STD** / <filename 1..54_without-generation-version>
, START-PROCEDURE**= *NO** / ***CRE**ATE

Shared operands of FTP and TELNET

For a description of the FTP operands, see page 55.
For a description of the TELNET operands, see page 138.

START-PROCEDURE=
Specifies whether the SYSENT.TCP-IP-AP.*nnn*.START file will be created, which contains the start enter file for FTP and TELNET.

START-PROCEDURE=*NO
The file will not be created.

START-PROCEDURE=*CREATE
The file will be created.

Return codes of SET-FTP-TELNET-PARAMETERS commands

(SC2)	SC1	Maincode	Meaning/Guaranteed messages
	0	TCP9000	INSTALLATION WAS TERMINATED SUCCESSFULLY
	1	CMD0202	SYNTAX ERROR IN COMMAND (this return code occurs with syntax errors, which are identified at the level of the SDF command definitions.)
	1	TCP9002	INVALID FILE NAME: (&00)
	1	TCP9003	THE PASSWORD (&00) IS NOT VALID.
	1	TCP9004	THE INSTALLATION KEY (&00) IS NOT VALID.
	1	TCP9005	THE ID (&00) IS NOT VALID.
	1	TCP9006	THE BS2000 VERSION (&00) IS NOT VALID.
	1	TCP9007	THE LOGON NAME (&00) IS NOT VALID.
	1	TCP9008	(&00) MAXIMUM CONNECTIONS ARE IN AN INVALID AREA.
	1	TCP9009	THE (&01) PARAMETER (&00) IS NOT VALID.
	1	TCP9010	THE JOB CLASS (&00) IS NOT VALID.
	1	TCP9011	THE CPU LIMIT (&00) IS NOT VALID.
	1	TCP9012	SET (&00) PARAMETERS.
	1	TCP9014	THE PRIORITY (&00) IS NOT VALID.
	1	TCP9015	THE DCAM APPLICATION NAME (&00) IS NOT VALID.
	1	TCP9016	THE PORT NUMBER (&00) IS NOT VALID.
	1	TCP9017	THE VALUE (&00) FOR PORT MONITORING IS NOT CORRECT.
	1	TCP9018	THE VALUE (&00) FOR AUTORIZATION SERVICE IS NOT CORRECT.
	1	TCP9020	THE NUMBER OF POSITIONS FOR GENERATING THE DCAM APPLICATION NAME(&00) IS NOT VALID.
	1	TCP9021	THE STANDARD ASCII CODE TABLE (&00) IS NOT VALID.
	1	TCP9022	THE STANDARD EBCDIC CODE TABLE (&00) IS NOT VALID.
	1	TCP9023	SDF ERROR READING THE STATEMENT.
	64	TCP9200	OPENING OF (&00) NOT POSSIBLE: DVS: (&01)
	64	TCP9201	WRITE ERROR IN INSTALLATION FILE: DVS: (&00)
	64	TCP9202	INSTALLATION PROGRAM AND PARAMETER FILE ARE NOT THE SAME VERSION.
	64	TCP9203	FILE COMMANDO COULD NOT BE ISSUED.
	64	TCP9205	INSTALLATION (&00) WAS NOT FOUND. DVS:(&01)
	64	TCP9206	(&00) ENTER DATEI COULD NOT BE CREATED. DVS:(&01)
	64	TCP9207	INST. FILE (&00) CANNOT BE CLOSED. DVS:(&01)
	64	TCP9208	MSG GROUP TCP COULD NOT BE INSTALLED. CODE (&00)
	64	TCP9209	READ ERROR IN INSTALLATION FILE (&00)

3.2 Installing and uninstalling FTP and TELNET clients in POSIX

FTP and TELNET client are installed together with the SNMP subagent for FTP (see the manual “SNMP Management for openNet Server and interNet Services”) like a POSIX program package via package installation with the POSIX installation program. Any older versions of the TCP-IP-AP program package which may exist must be uninstalled beforehand.

Installation

```
/START-POSIX-INSTALLATION
```

Function: Install POSIX program packages (IMON support: Y)
Product name: TCP-IP-AP

The figure below shows the installation mask with the information for installation of the FTP client.

```
                        BS2000 POSIX package installation


  IMON support ?     :  Y   (y)  mandatory for official package
                            (n)  private package (SINLIB...)

  name of product    :  TCP-IP-AP
  package of product :                        (optional for certain products)

  version of product :              (format Vmm.n or mmn)

  correction state     :            (format aso, optional for IMON support)

  installation userid   :           (mandatory for no IMON support)









  install: DUE       help: F1      terminate: F2
_____________________________________________________________________________
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Deinstallation

```
/START-POSIX-INSTALLATION
```

Function: Uninstall POSIX program packages
Product name: TCP-IP-AP

3.3 Frequently asked questions (FAQ)

- **Question:**
  What is the meaning of the output of the following messages after the FTP/TELNET client or FTP/TELNET server has been loaded?

```
BLS0340 UNRESOLVED EXTERNAL REFERENCES
BLS0342 ### 'YS6GSBN ……
BLS0342 ### 'YS6SOCE …….
BLS0342 ### 'YS6CLOS …….
BLS0342 ### 'YS6SHTD …….
BLS0342 ### 'YS6ERRO …….
```

  Answer:
  The programs were started either with START-PROG ..... without PROG-MODE=*ANY or the Socket subsystem SOC6 has not been started.

- **Question:**
  Connection setup with *open* in the FTP or TELNET client takes a very long time. What is the reason?

  Answer:
  For connection setup both the client and the server use DNS functions. If the associated resolver files are not correctly set, this can result in lengthy wait times. This is the case when, for example, the DNS server specified in the resolver files cannot be reached.

  The name of the resolver file in BS2000 is:

  SYSDAT.SOCKETS.*nnn*.SOC6.RESOLV or
  SYSDAT.LWRESD.*nnn*.RESOLV.CONF

- **Question:**
  What does the following FTP client message mean?

```
"time limit for server response exceeded"
```

  Answer:
  The client is waiting for an answer from the server which has not arrived after a predefined number of seconds (30 seconds). The reason for this can, for example, be that the network load is too high.
  You can often solve the problem by increasing the predefined timeout value of 30 seconds using the FTP user command *settime* (see the manual “interNet Services User Guide”).

- **Question:**
  When I transfer a file using FTP and save it as a PAM file in the target system, the string “C-DATEIENDE” is appended to it. What is the purpose of this, and how can I stop it?

  Answer:
  The string “C-DATEIENDE” is normally used to mark the exact end of a PAM file. With the FTP user command `setfile datend off` you can stop “C-DATEIENDE” being appended.

- **Question:**
  In earlier FTP versions (< V4.0) tabulator characters in a text file were automatically converted into the corresponding number of blanks. This is now no longer the case. What is the reason?

  Answer:
  In the course of supporting restart capability in BS2000-FTP as of V4.0 the default for the transfer of text files was changed from `ftyp text` to `ftyp textbin`. Consequently tabulator characters are no longer converted by default.

  If you still want to use conversion of the tabulator characters, you can select this via the *ftyp* command or via the option file:

  – *ftyp* command (see the manual “interNet Services User Guide”). Specify `ftyp text` on the client or `quote site ftyp text` on the server.

  – Option file of the FTP client (*initialCommand* option, see the manual “interNet Services User Guide”) or option file of the FTP server (*initialChildCmds* option, see page 78)

- **Question:**
  In earlier FTP versions (< V4.0), SAM files with a fixed record length for which *type binary* was specified were entirely in binary format and stored without end-of-record characters. In the current FTP version the target file contains end-of-record characters. How can I prevent this?

  Answer:
  ftyp settings are now also taken into account for SAM files with a fixed record length. If *ftyp* is not “binary”, the record structure of the original file is retained when the file is transferred. To prevent this, you must select *ftyp binary* explicitly.

- **Question:**
  In BS2000 FTP and TELNET it is possible to generate diagnostic information with the *trace* and *debug* commands (on clients) or with the *-T* and *-D* options (on servers). What is the difference between *trace* and *-T* on the one hand and *debug* and *-D* on the other?

  Answer:
  debug and *-D* generate diagnostic information which concerns the products FTP and TELNET. The highest useful level here is 2.
  trace and *-T*, by contrast, output diagnostic information generated by the sockets. The highest useful level here is 10.

- **Question:**
  I have started a second FTP or TELNET server but cannot set up a connection to it. What can be the reason?

  Answer:
  The most frequent causes of the problem are:

  – Specification of a port number for the server which had already been allocated.
  – Use of an application name in the server (*-A* option) which had already been allocated.

- **Question:**
  My FTP login is rejected by the (BS2000) FTP server with the message `invalid login`. I cannot find any reason for this behavior, however. What should I do?

  Answer:
  If possible, enable the FTP trace in the FTP server using the following command:

  ```
  /INFORM-PROGRAM 'debug 2',*TSN <tsn ftpserver>
  ```

  Repeat the login and, using the command

  ```
  /INFORM-PROGRAM 'rdProt', <tsn ftpserver>
  ```

  store the trace in the file SYSOUT.TCP-IP-AP.*nnn*.FTPD.<MMDDHHMMSS>.

  (MMDDHHMMSS is the date and time specification in the format
  Month Day Hour Minute Second).

- **Question:**
  How do I reach a BS2000-FTP server with a Web browser?

  Answer:
  You can access the POSIX directory of the ID <userid> via the following URL:

  ```
  ftp://<userid>,<account-number>@<host-name>:<port-number>/
  ```

  This will at least allow you to output the directories. Access to the BS2000 directory is not possible via a Web browser.

- **Question:**
  In FTP it is possible to use `quote <command>` to send the command <command> to the server. However, there is also `quote site <command>` and `quote site exec <command>`. What is the difference?

  Answer:

  – With `quote <command>` you send FTP commands that conform with the standard to the server.

  – With `quote site <command>` you send BS2000-specific (“proprietary”) commands to the server which are not defined in the standard. These commands include *ftyp*, *cmod*, *modc*, *file*, *setc*, *sfil*.

  – With `quote site exec <command>` you send BS2000 commands to the server to be executed. To prevent misuse of these commands in the tagret system, this variant is disabled when FTAC is used (option -*FTAClevel* >0) or via the option -*disableSiteExecCommand* (see the manual “interNet Services Administrator Guide”).

- **Question:**
  When using FTP clients with a graphical user interface (GUI) there is often no way of specifying the account number required for the connection to BS2000. What should I do in this case?

  Answer:
  In cases like this enter the account number when you enter your ID as follows:

  ```
  <userid>,<account>
  ```

- **Question:**
  When fetching a file from a BS2000 FTP server my non-BS2000 FTP client aborts execution after a certain time without data transfer actually having begun.

  Answer:
  Some FTP clients show the progress of the transfer with a progress bar. For this purpose, the clients first of all use the FTP protocol command *SIZE* to query the size of the file from the server. Generally the server must read the relevant file in full to process this command. With very large files this can naturally take quite a time, with the result that the client's timeout monitoring clears the connection.
  Unfortunately on some clients this timeout monitoring cannot be reconfigured to permit longer wait times. In addition, in many cases a *SIZE* command is sent to the server even when the progress bar has been disabled.

  If an enhancement of the configuration options can be obtained from the vendor of the FTP client, there is an option on the BS2000 FTP server to disable the *SIZE* command using the - *disableSizeCommand* option (see the manual "interNet Services Administrator Guide").

  As a client cannot require that the server should support the *SIZE* command, transfer should always function. However, you must accept that the restart mechanism no longer functions because the *SIZE* command is needed for this.

- **Question:**

  When an LMS file is transferred from an NK2 pubset to an NK4 pubset, the destination file is no longer a valid LMS file.

  Answer:
  Transfer the file initially to a non-NK4 pubset on the destination computer and then use LMS to copy the LMS library to the NK4 pubset.

  Alternatively, you can convert the LMS library to an NK4 pubset on the source computer and then use FTP to transfer it to the NK4 pubset of the destination computer.

4 FTP configuration and operation

You can configure FTP using either the SDF command or the option file.

This chapter covers the following topics related to the configuration and operation of FTP servers:

4.1 TLS/SSL support on the FTP server

You will find a general overview of SSL in the interNet Services User Guide.

The following instruments are available for providing TLS/SSL support on the FTP server:

- Option file or files
- Installation command SET-FTP-TELNET-PARAMETERS
- FTP login commands

4.1.1 Parameterization of TLS/SSL support on the FTP server

TLS/SSL support offers a wide range of setting options. You can make these settings as follows:

- With the aid of options which are stored in one or more option files and are interpreted when the FTP server is started (see section “Configuring FTP via the option file” on page 67).
- With the aid of the installation command parameters SET-FTP-TELNET-PARAMETERS (see section “Configuration of FTP using the SET-FTP-TELNET-PARAMETERS installation command” on page 55).

The table below shows the options for TLS/SSL support on the FTP server. For the individual options there are also corresponding parameters of the SET-FTP-TELNET-PARAMETERS command.

Option	Description	Page
-tlsProtocol	Choose SSL protocol versions selectively	84
-tlsCipherSuite	Specify cipher suite preference list	85
-tlsRSAcertificateFile	Specify file which contains the RSA-based X.509 server certificate in PEM format	85
-tlsRSAkeyFile	Specify file which contains the private RSA server key in PEM format	86
-tlsDSAcertificateFile	Specify file which contains the DSA-based X.509 server certificate in PEM format	86
-tlsDSAkeyFile	Specify file which contains the private DSA server key in PEM format	87
-tlsCertificateChainFile	Specify file in which all the certificates required for verification of the server certificate can be stored	88
-tlsCAcertificateFile	Specify file which contains the certificates required for authentication of the FTP client in PEM format	89
-tlsAcceptableClientCAFile	Specify file from which the names of the CAs that the server accepts as signatories of client certificates can be obtained	90
-tlsCArevocationFile	Specify file which contains the CRLs of the CAs	91
-tlsVerifyClient	Define whether the FTP client must provide a certificate for server access	91
-tlsVerifyDepth	Define verification depth	92
-tlsSecureControlConnection	Define whether the control connection from the FTP client to the FTP server is to be secured with TLS	93
-tlsSecureDataConnection	Define whether the data connection from the FTP client to the server child is to be secured with TLS	94
-tlsRandFile	Specify file from which the data for initializing the PRNG is read when the server is started	95
-tlsOpenSSLlibName	Define the LMS file from which the OpenSSL library should be dynamically loaded	96

4.1.2 FTP login commands

The FTP server supports the FTP login commands *AUTH*, *PBSZ* and *PROT* in the format required by RFC 4217:

- The *TLS*, *TLS-C*, *TLS-P* and *SSL* parameters are permissible for the *AUTH* command. *SSL*, *TLS* and *TLS-C* have the same meaning and request TLS security only for the control connection. With TLS-P, security of the data connection is also requested.
- The *PBSZ* command is only supported pro forma with the parameter 0 to ensure a login process as per RFC 2228.
- The *PROT* command is supported with the parameters C and P where
  C (= Clear) disables encryption of the data connection and
  P (= Private) enables it.
- If TLS support is enabled, the *FEAT* command [RFC 2389] reports this with additional enumeration of *AUTH TLS*, *PBSZ* and *PROT*.

STAT command output is complemented by three lines. These lines first of all document whether the control connection is secured with TLS. Any algorithms with which the control and data connections are secured are also specified.

Examples

1. Non-secured control connection:

```
Protected control channel: Off
Private data channel: Off
Cipher: clear
```

2. Control connection secured with Triple DES, data connection not secured:

```
Protected control channel: On
Private data channel: Off
Cipher: DES-CBC3-SHA (168 bits)
```

3. Control and data connections secured with Triple DES:

```
Protected control channel: On
Private data channel: On
Cipher: DES-CBC3-SHA (168 bits)
```

4. Control and data connections secured with Triple DES, the control connection having to be unencrypted again after the *ccc* command:

```
Protected control channel: Off (cleared)
Private data channel: On
Cipher: DES-CBC3-SHA (168 bits)
```

4.2 Configuration of FTP using the SET-FTP-TELNET-PARAMETERS installation command

For the full command syntax and the description of the installation operands, see page 40.

SET-FTP-TELNET-**PAR**AMETERS

(...)
, FTP-SERVER-PROC=*NO / *CREATE(...)
***CRE**ATE(...)
JOB-NAME= *STD / <name 1..5>
, JOB-CLASS= *STD / <name 1..8>
, CPU-TIME= *STD / <integer 1..32767>
, PRIORITY**= *STD** / <integer 0..255>
, DEBUG= *STD / <integer 0..9>
, TRACE= *STD / <integer 0..10>
, MAXIMUM-**CONN**ECTIONS**= *STD** / <integer 1..900>
, STATION-**ID= *STD** / <integer 0..6>
, TRANSFER-**JOB-CLASS= *STD** / <name 1..8>
, TRANSFER-**CPU-TIME= *STD** / <integer 1..32767>
, TIMEOUT-VALUE**= *STD** / <integer 1..32767>
, SYSTEM-**EXIT-LEV**EL**= *STD**/<integer 0..3>
, FTAC-SUPPORT**= *STD** / ***NO** / ***YES**(...)
***YES**(...)
LEVEL**= *STD** / <integer 1..2>
, JOB-CLASS= *STD / <name 1..8>
, ENTER-**F**ILE**= *STD** / <filename 1..54_without-generation-version>
, SERVER-INFORMATION-**F**ILE**= *STD** / <filename 1..54_without-generation-version>
,FTAC-USERID=*STD /<name 1..8>
, TLS-SUPPORT**= *STD** / ***NO** / ***YES**(...)
***YES**(...)
PROTOCOL= *STD / <text 1..80>
, CIPHER-SUITE= *STD / <text 1..80_with-lower-case>
, RSA-CERTIFICATE-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, RSA-KEY-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, DSA-CERTIFICATE-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, DSA-KEY-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CA-CERTIFICATE-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CLIENT-CA-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, CERT-CHAIN-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>

SET-FTP-TELNET-**PAR**AMETERS

, CA-REVOCATION-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, RANDOM-FILE= *STD / ***NONE** / <filename 1..54_without-generation-version>
, SSL-LIBRARY= *STD / ***NONE** / <filename 1..54_without-generation-version>
, VERIFY-CLIENT= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**
, VERIFY-DEPTH= *STD / <1..32767>
, SEC-CONTROL-CONN= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**
, SEC-DATA-CONN= *STD / ***NONE** / ***OPTIONAL** / ***REQUIRE**
,ACCOUNTING=*STD / ***NO** / ***YES**(...)
***YES**(...)
FILE=*STD / filename_1..54_without-generation-version
,OPTION-FILE=*STD / filename_1..54_without-generation-version
,SELECTOR=*STD / text_1..511
,INITIAL-COMMANDS=*STD / c-string_2..256
,PORT-NUMBER =*STD / integer_1..32767
,SERVER-ENTER-FILE=*STD/ filename_1..54_without-generation-version
,LOGGING-FILE=*STD / filename_1..54_without-generation-version
,ALLOW-TSOS-LOGIN=*STD / ***NO** / ***YES** / ***TLS**
,TVFS=*STD / ***NO** / ***YES**
(...)

FTP-SERVER-PROC =
Parameters for the FTP server

FTP-SERVER-PROC=*NO
An installation may not be performed for the FTP server.

FTP-SERVER-PROC=*CREATE(...)
The FTP server should be configured. The enter file for starting the FTP server is created from the parameters entered.

JOB-NAME=
This name is used as a prefix. To construct the terminal name of the current connection, the sequence number is appended to the JOB-NAME. See also the *-appPrefix* option on page 68.

JOB-NAME=*STD
Corresponds to the entry FTPSR or the value in the installation file.

JOB-NAME=<name 1..5>
Job name

JOB-CLASS=
The job class in which the server process should run. Make sure that Enter jobs with the following parameters may be started in this job class:
CPU-LIMIT=*NO, RUN-PRIORITY=120 and START=*IMMEDIATELY.

JOB-CLASS=*STD
Corresponds to the standard job class on the system or the value in the installation file.

JOB-CLASS=<name 1..8>
Name of job class.

CPU-TIME=
Maximum CPU time available for the server process.

CPU-TIME=*STD
Corresponds to the entry NTL or the value in the installation file.

CPU-TIME=<integer 1..32767>
CPU time in seconds.

PRIORITY=
Priority with which the server process is to run.

PRIORITY=*STD
Corresponds to the entry 120 or the value in the installation file.

PRIORITY=<integer 0..255>
Server priority.

DEBUG=
Trace at user level. See also the *-debugLevel* option on page 70.

DEBUG=*STD
Corresponds to the entry 0 or the value in the installation file.

DEBUG=<integer 0..9>
Debug level.

TRACE=
Trace for TCP/IP (socket) interface. See also the *-socketTraceLevel* option on page 75.

TRACE=*STD
Corresponds to the entry 0 or the value in the installation file.

TRACE=<integer 0..10>
Socket trace level.

MAXIMUM-CONNECTIONS=
Maximum number of connections that the server is to operate. See also the *-maxConn* option on page 73.

MAXIMUM-CONNECTIONS=*STD
Corresponds to the entry 15 or the value in the installation file.

MAXIMUM-CONNECTIONS=<integer 1..900>
Maximum number of connections that the server is to operate.

STATION-ID=
Number of places with which the name of the remote computer or the job name of the FTP server is to be taken into account in the terminal name. See also the *-DSSidLength* option on page 75.

STATION-ID=*STD
Corresponds to the entry 0 or the value in the installation file.

STATION-ID=<integer 0..6>
Number of places with which the name of the remote computer or the job name of the FTP server is to be taken into account in the terminal name.

TRANSFER-CPU-TIME=
CPU time available for the job run of the child process. See also the *-logonExtension* option on page 73.

TRANSFER-CPU-TIME=*STD
Corresponds to the entry NTL or the value in the installation file.

TRANSFER-CPU-TIME=<integer 1..32767>
CPU time available for the job run of the child process.

TRANSFER-JOB-CLASS=
The job class into which the child process jobs are to be classified. See also the *-logonExtension* option on page 73.

TRANSFER-JOB-CLASS=*STD
Corresponds to the standard job class of the system or the value in the installation file.

TRANSFER-JOB-CLASS=<name 1..8>
Default: Standard dialog job class on the system.

TIMEOUT-VALUE=
Time after which the connection between the client and server will be aborted if no activity was observed in the specified period between the client and server. See also the *-timeout* option on page 74.

TIMEOUT-VALUE=*STD
Corresponds to the default setting of 3600 seconds.

TIMEOUT-VALUE=<integer 1..32767>
Timeout value in seconds.

SYSTEM-EXIT-LEVEL=
The FTP system exit is enabled with a value other than 0. If FTAC-SUPPORT=*YES is also set, the system exit is only called for the sub-events FTPBYE and FTPCMDE or otherwise for all sub-events. See also the *-systemExit* option on page 77.

SYSTEM-EXIT-LEVEL=*STD
Corresponds to the entry 0 or the value in the installation file.

SYSTEM-EXIT-LEVEL=<integer 0..3>
System exit level.

FTAC-SUPPORT=
Specifies whether or not the FTP server is to use FTAC functionality.

FTAC-SUPPORT=*STD
Default: *NO or value in the specified installation file INSTALLATION-FILE.

FTAC-SUPPORT=*NO(...)
The transfer admission check using FTAC functionality is not performed.

FTAC-SUPPORT=*YES(...)
The transfer admission check is performed using FTAC functionality.

LEVEL=
FTAC level: Level at which the FTAC transfer admission check is performed. A value of 1 indicates that access via a dialog id is not checked by FTAC but additional access is possible with a FTAC transfer admission.
A value of 2 means that access via a dialog id is also checked by FTAC (using the respective admission set). Access with the TSOS ID may not be possible from value 1. See also the *-allowTSOSLogin* option on page 80 and the *-FTAClevel* option on page 69.

LEVEL=*STD
This corresponds to value 1.

LEVEL=<integer 1..2>
FTAC level.

JOB-CLASS=
Job class in which the child processes are to run. Make sure that Enter jobs with the parameter SCHEDULING-TIME=*PARAMETERS(START=*IMMEDIATELY) may be started in this job class. The only jobs that can run in this job class are those that are started in the framework of a logon with FTAC transfer admission. See also the *-childJobClass* option on page 72.

JOB-CLASS=*STD
Corresponds to the standard batch job class on the current system.

JOB-CLASS=<name 1..8>
Name of the job class.

ENTER-FILE=
Name of the Enter file that starts the child process. See also the *-childEnterJob* option on page 71.

ENTER-FILE=*STD
Corresponds to the file SYSENT.TCP-IP-AP.*nnn*.FTPDC.

ENTER-FILE=<filename 1..54_without-generation-version>
Name of the Enter file that starts the child process.

SERVER-INFORMATION-FILE=
Name of the file for exchanging information between the server and child process. This file contains the port number, for example, under which the server can be accessed for the child process. See also the *-serverInfoFile* option on page 71.

SERVER-INFORMATION-FILE=*STD
Corresponds to the file SYSDAT.TCP-IP-AP.*nnn*.SI.

SERVER-INFORMATION-FILE=<filename 1..54_without-generation-version>
Name of the file for exchanging information between the server and child process.

FTAC-USERID=
ID that can be entered instead of $FTAC to show at login that the transfer admission check is to take place via an FTAC transfer admission. See also the *-FTACuserId* option on page 72.

FTAC-USERID=*STD
Corresponds to the default $FTAC.

FTAC-USERID=<name 1..8>
FTAC user ID.

TLS-SUPPORT=
Defines whether TLS/SSL security is enabled for the FTP server.

TLS-SUPPORT=*STD
Default: *NO.

TLS-SUPPORT=*NO
The FTP server does not implement security for the connections by means of TLS.

TLS-SUPPORT=*YES(...)
The FTP server implements (on principle) security for the connections by means of TLS.

PROTOCOL=
See the *-tlsProtocol* option on page 84.

PROTOCOL=*STD
Default: ALL –SSLv2

PROTOCOL=<text 1..80>
Specification of the TLS/SSL protocol to be used.

CIPHER-SUITE=
See the *-tlsCipherSuite* option on page 85.

CIPHER-SUITE=*STD
Default: ALL:!EXP:!ADH

CIPHER-SUITE=<text 1..80_with-lower-case>
Specification of the encryption algorithms to be used.

RSA-CERTIFICATE-FILE=
See the *-tlsRSAcertificateFile* option on page 85.

RSA-CERTIFICATE-FILE=*STD
Default: *NONE

RSA-CERTIFICATE-FILE=*NONE
No RSA certificate file is specified.

RSA-CERTIFICATE-FILE=<filename 1..54_without-generation-version>
Name of the RSA certificate file.

RSA-KEY-FILE=
See the *-tlsRSAkeyFile* option on page 86.

RSA-KEY-FILE=*STD
Default: *NONE

RSA-KEY-FILE=*NONE
No RSA key file is specified.

RSA-KEY-FILE=<filename 1..54_without-generation-version>
Name of the RSA key file.

DSA-CERTIFICATE-FILE=
See the *-tlsDSAcertificateFile* option on page 86.

DSA-CERTIFICATE-FILE=*STD
Default: *NONE

DSA-CERTIFICATE-FILE=*NONE
No DSA certificate file is specified.

DSA-CERTIFICATE-FILE=<filename 1..54_without-generation-version>
Name of the DSA certificate file.

DSA-KEY-FILE=
See the *-tlsDSAkeyFile* option on page 87.

DSA-KEY-FILE=*STD
Default: *NONE

DSA-KEY-FILE=*NONE
No DSA key file is specified.

DSA-KEY-FILE=<filename 1..54_without-generation-version>
Name of the DSA key file.

CA-CERTIFICATE-FILE=
See the *-tlsCACertificateFile* option on page 89.

CA-CERTIFICATE-FILE=*STD
Default: *NONE

CA-CERTIFICATE-FILE=*NONE
No CA certificate file is specified.

CA-CERTIFICATE-FILE=<filename 1..54_without-generation-version>
Name of the CA certificate file.

CLIENT-CA-FILE=
See the *-tlsAcceptableClientCAFile* option on page 90.

CLIENT-CA-FILE=*STD
Default: *NONE

CLIENT-CA-FILE=*NONE
No file is specified.

CLIENT-CA-FILE=<filename 1..54_without-generation-version>
Name of the file with the certificates of the accepted CAs.

CERT-CHAIN-FILE=
See the *-tlsCertificateChainFile* option on page 88.

CERT-CHAIN-FILE=*STD
Default: *NONE

CERT-CHAIN-FILE=*NONE
No CA certificate chain file is specified.

CERT-CHAIN-FILE=<filename 1..54_without-generation-version>
Name of the CA certificate chain file.

CA-REVOCATION-FILE=
See the *-tlsCArevocationFile* option on page 91.

CA-REVOCATION-FILE=*STD
Default: *NONE

CA-REVOCATION-FILE=*NONE
No CA revocation file is specified.

CA-REVOCATION-FILE=<filename 1..54_without-generation-version>
Name of the CA revocation file.

RANDOM-FILE=
See the *-tlsRandFile* option on page 95.

RANDOM-FILE=*STD
Default: SYSDAT.TCP-IP-AP.*nnn*.FTPD.RAND

RANDOM-FILE=<filename 1..54_without-generation-version>
Name of the random numbers file.

SSL-LIBRARY=
See the *-tlsOpenSSLlibName* option on page 96.

SSL-LIBRARY=*STD
Default: LMS library to which the IMON logical ID SYSLNK refers.

SSL-LIBRARY=*NONE
No LMS library is specified.

SSL-LIBRARY=<filename 1..54_without-generation-version>
Name of the LMS library which contains the OpenSSL dynamically loadable module.

VERIFY-CLIENT=
See the *-tlsVerifyClient* option on page 91.

VERIFY-CLIENT=*STD
Default: *NONE

VERIFY-CLIENT=*NONE
No certificate is requested from the FTP client.

VERIFY-CLIENT=*OPTIONAL
A certificate is requested from the FTP client. However, if no certificate or only an invalid one is returned, the FTP client is still granted access.

VERIFY-CLIENT=*REQUIRE
A certificate is requested from the FTP client. However, if no certificate or only an invalid one is returned, the FTP client is denied access.

VERIFY-DEPTH=
See the *-tlsVerifyDepth* option on page 92.

VERIFY-DEPTH=*STD
Default: 1

VERIFY-DEPTH=<integer 1..32767>
Number of certificates between the client certificate and the certificate which is known to the FTP server (including the latter).

SEC-CONTROL-CONN=
See the *-tlsSecureControlConnection* option on page 93.

SEC-CONTROL-CONN=*STD
Default: *OPTIONAL

SEC-CONTROL-CONN=*NONE
The control connection is never secured with TLS. Corresponding requests from the client are rejected.

SEC-CONTROL-CONN=*OPTIONAL
The control connection is secured if the client requests this.

SEC-CONTROL-CONN=*REQUIRE
No login is permitted via a non-secured control connection.

SEC-DATA-CONN=
See the *-tlsSecureDataConnection* option on page 94.

SEC-DATA-CONN=*STD
Default: *OPTIONAL

SEC-DATA-CONN=*NONE
The data connection is never secured with TLS. Corresponding requests from the client are rejected.

SEC-DATA-CONN=*OPTIONAL
The data connection is secured if the client requests this.

SEC-DATA-CONN=*REQUIRE
No data transfer is permitted via a non-protected data connection.

ACCOUNTING=
Specifies whether FTP accounting records are to be collected. See also the *-acctActive* and *-acctFile* options on page 79.

ACCOUNTING=*STD
Default: *NO

ACCOUNTING=*NO
The accounting records are not collected.

ACCOUNTING=*YES(...)
Accounting records are collected.

FILE=
Name of the accounting file.

FILE=*STD
This corresponds to the file SYSDAT.TCP-IP-AP.*nnn*.FTPD.ACCOUNTING.

FILE=<filename 1..54_without-generation-version>
Name of the accounting file.

OPTION-FILE=
File in which the subsequent options are to be stored.

OPTION-FILE=*STD
Default: SYSDAT.TCP-IP-AP.*nnn*.FTPD.OPT

OPTION-FILE=<filename 1..54_without-generation-version>
Name of the option file.

SELECTOR=
Selector for FTP exit routines. See also the *-convSelector* option on page 76.

SELECTOR=*STD
Default: No exits

SELECTOR=<text 1..511>
Specifies the selector for the FTP exit routines.

INITIAL-COMMANDS=
See the *-initialChildCmds* option on page 78.

INITIAL-COMMANDS=*STD
No commands are sent to the child process.

INITIAL-COMMANDS=<c-string 2..256>
Specifies the commands to be sent to the child process.

PORT-NUMBER=
Port number under which the FTP server is started. See also the *-portNumber* option on page 74.

PORT-NUMBER=*STD
Default: 21

PORT-NUMBER=<integer 0..32767>
Specifies the port number under which the FTP server is started.

SERVER-ENTER-FILE=
Name of the start file for the FTP server.

SERVER-ENTER-FILE=*STD
Default: SYSENT.TCP-IP-AP.*nnn*.FTPD

SERVER-ENTER-FILE=<filename 1..54_without-generation-version>
Specifies the name of the start file for the FTP server.

LOGGING-FILE=
Name of the logging file for the FTP server.

LOGGING-FILE=*STD
Default: SYSOUT.TCP-IP-AP.*nnn*.FTPD

LOGGING-FILE=<filename 1..54_without-generation-version>
Specifies the name of the logging file for the FTP server.

ALLOW-TSOS-LOGIN=
See the *-allowTsosLogin* option on page 80.

ALLOW-TSOS-LOGIN=*STD
As in the older versions, the behavior is determined by the level setting for FTAC support.

ALLOW-TSOS-LOGIN=*NO
A login attempt is rejected when the TSOS ID is used.

ALLOW-TSOS-LOGIN=*YES
A login attempt is permitted when the TSOS ID is used.

ALLOW-TSOS-LOGIN=*TLS
A login attempt is permitted only when the TSOS ID is used via a connection secured with TLS/SSL.

TVFS=
Enables or disables the TVFS (Trivial Virtual File System), see option *-TVFS* on page 97.

TVFS=*STD
Default: *NO

TVFS=*NO
TVFS is not enabled.

TVFS=*YES
TVFS is enabled, i.e DVS and POSIX files are accessible via a homogeneous POSIX like file system.

4.3 Configuring FTP via the option file

When the SET-FTP-TELNET-PARAMETERS command is executed (see page 40) an option file is generated in which the various FTP server parameters are stored. The default file name for the option file is:

SYSDAT.TCP-IP-AP.*nnn*.FTPD.OPT

Under $TSOS, the execution ID.

If a file with this name exists in the FTP server's execution ID it is interpreted when the FTP server is started via the FTP server start file (default file name: SYSENT.TCP-IP-AP.*nnn*.FTPD), and the FTP server is configured accordingly.

You can thus make changes to the FTP server parameters via the option file without needing to start installation again using the SET-FTP-TELNET-PARAMETERS command.

If you want to use a file with a different name as the option file, enter the following option in the start file:

-M *option-file-name*

Option file and parameter line options

You can also specify the parameter line options supported in earlier versions. In this case you can choose between the familiar option name consisting of one character and the new, more easily recognizable name consisting of several characters.

Mixed application of the option file and options in the parameter line is also possible. Here only option names comprising one character are permitted in the parameter line. If a particular option is specified in both the parameter line and in the option file, the option specified in the parameter line has priority.

Notation of the options in the option file

The various options must be entered in the option file according to the following rules:

- Each option must be in a separate line
- If an option's arguments extend over more than one line, each line that is to be continued must be terminated with the continuation character "\".
- A line beginning with the character "#" in column 1 is ignored when the file is read in.
- No distinction is made between upper and lower case in the option names.

Description of the options

The individual options are described below:

- With the options that correspond to the parameter line options already supported, the name, which consists of one character, is specified as an alias name.
- The options which govern TLS/SSL support in the FTP server are marked by the prefix "tls" in the option name.

-appPrefix | -A

The *-appPrefix* option is used to specify a name prefix. This name prefix is used to form the terminal name by appending a serial number. The job started by the server for each client is given this name prefix as the job name.

-appPrefix **-A**
<job-name-prefix>

<job-name-prefix>
Name prefix
Default: FTPSR

-FTAClevel | -B

The *-FTAClevel* option is used to specify the FTAC level. The FTAC level specifies the level at which the FTAC transfer admission check is performed.

-FTAClevel **-B**
0 | **1** | **2**

0

FTAC is not used.
0 is the default

1

Access via a dialog ID is not checked by FTAC. Access is also possible with an FTAC transfer admission.

2

Access via a dialog ID is also checked by FTAC on the basis of the relevant admission set.

If the FTAC level > 0, it may be that access is not possible with the TSOS ID (see the *-allowTsosLogin* option on page 80).

-childName | -C

The *-childName* option is used to specify the path for calling the child program.

-childName **-C**
<path-for-child-call>

<path-for-child-call>
Path for calling the child program.
Default: FTPDC

-debugLevel | -D

The *-debugLevel* option is used to specify the debug level. The debug level defines whether or which diagnostic information on the FTP server run is written to the logging file.

-debugLevel **-D**
<integer 0..9>

<integer 0..9>
Debug level.
“0” disables logging. The higher the value, the more information is placed in the logging file.
Default: 0

-serverInfoFile | -E

The *-serverInfoFile* is used to specify a file for exchanging information between the server and child processes. This file contains, for example, the port number under which the server can be reached by the child process. This only applies for child processes which are started in the context of a login via an FTAC transfer admission.

-serverInfoFile **-E**
<file-name 1..54>

<file-name 1..54>
Name of the file used for exchanging information between the server and child processes.
Default: SYSDAT.TCP-IP-AP.*nnn*.SI

-childEnterJob | -F

The *-childEnterJob* option is used to specify an enter file which starts the child process if login takes place via an FTAC transfer admission.

-childEnterJob **-F**
<file-name 1..54>

<file-name 1..54>
Name of the enter file.
Default: SYSENT.TCP-IP-AP.*nnn*.FTPDC

-childJobClass | -J

The *-childJobClass* option is used to specify a job class in which the child processes run which are started via an FTAC transfer admission.

Ensure that Enter jobs can be started with the parameter SCHEDULING-TIME= *PARAMETERS(START=*IMMEDIATELY) in this job class.

-childJobClass **-J**
<job-class 1..8>

<job-class 1..8>
Job class in which the child processes run.
The default is system-dependent.

-FTACuserId | -K

The *-FTACuserId* option is used to specify a pseudo ID which is used at login to show that login is not to be performed with a genuine BS2000 user ID but with an FTAC transfer admission.

-FTACuserId **-K**
<userid 1..8>

<userid 1..8>
Pseudo ID.
Default: $FTAC

-logonExtension | -L

The *-logonExtension* option can be used to specify additional information for the logon command for starting the FTP child. This includes above all the job class and the CPU limit for the interactive job.

-logonExtension **-L**
<logon-ext 1..511>

<logon-ext 1..511>
Additional information for the logon command for starting the FTP child.
The default is system-dependent.

-maxConn | -N

The *-maxConn* option is used to specify the maximum number of connections which the FTP server is to handle simultaneously.

-maxConn **-N**
<integer 1..900>

<integer 1..900>
Maximum number of connections which the FTP server is to handle simultaneously.
Default: 15

-timeout | -O

The *-timeout* option is used to select the timeout interval for the connection between the FRP server and FTP client. If no activity is detected between the server and the client in the specified time, the connection is cleared down.

-timeout **-O**
<number-of-seconds>

<number-of-seconds>
Timeout interval in seconds.
Default: 3600

-portNumber | -P

The *-portNumber* option is used to specify the port number under which the FTP server can be reached.

-portNumber **-P**
<integer 1..65535>

<integer 1..65535>
Port number under which the FTP server can be reached.
Default: 21

-DSSidLength | -S

The *-DSSidLength* is used to define the name of the remote computer or job name of the FTP server to be included in the terminal name.

-DSSidLength **-S**
<integer 0..6>

<integer 0..6>
Number of digits to be included.
Default: 0

-socketTraceLevel | -T

The *-socketTraceLevel* option is used to specify the socket trace level. The socket trace level specifies whether or which diagnostic information of the TCP/IP (socket) system is to be written to the logging file.

-socketTraceLevel **-T**
<integer 0..10>

<integer 0..10>
Socket trace level
“0” disables logging. The higher the value, the more information is placed in the logging file.
Default: 0

-convSelector | -U

The *-convSelector* option is used to enable and disable user-defined exit routines of the FTP server. Details on this are provided in the section “Exit mechanisms for the FTP server and FTP client” on page 128.

-convSelector **-U**
<selector-definition 1..511>

<selector-definition 1..511>
Selector definition
Default: <empty-string>, in other words no user-defined exit routines.

-verbose | -V

The *-verbose* option is the short form of *-debugLevel 1* and *-socketTraceLevel 1*.

-verbose **-V**

-systemExit | -X

The *-systemExit* option is used to enable and disable the system exit.

-systemExit **-X**
0 | **1**

0
System exit is disabled.
0 is the default.

1
System exit is enabled.

-initialChildCmds | -Z

The *-initialChildCmds* option is used to specify commands which are sent to the child program directly after it has started. The commands suitable for use here are above all those which make permanent settings, such as the RFC 959 commands *type*, *stru* and *mode* and the BS2000 proprietary commands *ftyp*, *setc*, *modc*, *sfil* and *cmod*. In special cases it may make sense to use the RFC 959 commands *dele* and *cwd* and the BS2000 proprietary command *file*. The BS2000 proprietary commands are described in the manual “interNet Services User Guide”. Please note: In contrast to the method of use described there, the proprietary commands may not be prefixed with a “site” in the option argument. Each command in the command string must be terminated with `\n` or `\N`.

You can use the *-initialChildCmds* option to change defaults, for example. Thus with `–initialChildCmds FTYP text\nSFIL pademptyrec on\n` you can set the FTYP default to `text` (this is the default in older FTP versions) and have empty SAM records padded with a blank.

-initialChildCmds **-Z**
<string 2..511>

<string 2..511>
String containing the commands.
Default: <empty-string>, in other words no command.

-acctActive

The *-acctActive* option is used to enable and disable collecting of FTP accounting data (FTP accounting).

-acctActive
ON | **OFF**

ON
FTP accounting is enabled.

OFF
FTP accounting is disabled.
OFF is the default.

-acctFile

The *-acctFile* option is used to specify the name of the accounting file for FTP accounting in which the accounting records are stored. The accounting file is a SAM file with variable record length.

If an accounting file with this name already exists for FTP accounting, this is updated when the FTP server starts up.

-acctFile
<accounting-file-name1..54>

<accounting-file-name 1..54>
Name of the accounting file for FTP accounting.
Default: SYSDAT.TCP-IP-AP.*nnn*.FTPD.ACCOUNTING

-allowTsosLogin

The *-allowTsosLogin* option is used to specify whether or under what conditions a login is possible under the user ID TSOS.

Until now, a login under the user ID TSOS was rejected if FTAC support was enabled (`-FTAClevel > 0`, see page 69).

With the *-allowTsosLogin* option you can now specify whether a TSOS login is possible regardless of the setting of the *-FTAClevel* option:

- If you do not use the *-allowTsosLogin* option, whether or not a TSOS login is possible depends on the *-FTACLevel* option, as in interNet Services V2.0.
- If you use the *-allowTsosLogin* option, the following argument-dependent conditions described below apply regardless of FTAC activation.

-allowTsosLogin
NO | **TLS** | **YES**

NO
A login attempt under TSOS is always rejected.

TLS
When "`TLS`" is specified, login can only take place via a connection secured with TLS/SSL.

YES
There are no restrictions for the TSOS login.

For security reasons only the value "NO" or "TLS" should be used. "NO" should be preferred.

-defaultFTACsecurityLevel

The *-defaultFTACsecurityLevel* option is used to define an FTAC security level that is to be assigned to the FTP partners. If a lower value is specified in an admission set, the associated basic function is not available to FTP partners.

If, for example, the majority of FT partners are assigned a security level of 80, and if FTP partners are to have access to the same basic functions as these FT partners, then *-defaultFTACsecurityLevel 80* should be used.

-defaultFTACsecurityLevel
<integer 1..100>

<integer 1..100>
Security level to be assigned to the FTP partners
Default: 100

-disableSiteExecCommand

The *-disableSiteExecCommand* option is used to specify whether the server supports the proprietary FTP protocol command *SITE EXEC*. This command allows the FTP client user to execute BS2000 commands on the server and thus opens up great potential for misuse. Consequently this command has to date always been disabled when FTAC support is activated (`-FTAClevel > 0`, see page 69), while it is still supported in all other cases.

With the *-disableSiteExecCommand* option you can enable or disable the *SITE EXEC* command regardless of the *-FTAClevel* option:

- If you do not use the *-disableSiteExecCommand* option, the usability of the command depends on the *-FTACLevel* option, as in interNet Services < V3.0.
- If you use the *-disableSiteExecCommand* option with the argument “YES”, the command is never available.
- If you use the *-disableSiteExecCommand* option with the argument “NO”, the command is always available.

-disableSiteExecCommand
NO | **YES**

NO
Support of the *SITE EXEC* command is not disabled.

YES
Support of the *SITE EXEC* command is disabled.

-disableSizeCommand

The *-disableSizeCommand* option is used to define whether the server supports the *SIZE* command. The *SIZE* command is described in the manual “interNet Services User Guide”.

A problem in conjunction with the *SIZE* command is that the *SIZE* command generally needs to read the file to be investigated fully before it can send a correct message about the file size. Execution of the *size* command is thus resource-intensive and time-consuming, especially with large files.

The result of this is

- a weakness with regards to “Denial of Service” attacks,
- problems with some FTP clients.

At the start of transfer these clients execute the *SIZE* command so as to display a progress bar indicating the relative progress of the transfer. Disregarding the question as to whether this progress bar justifies the relevant file being read twice by the server, the problem is encountered on at least one of the clients that it clears the connection after a fixed period, which cannot be modified by the user, without a message being issued by the server. This client is thus not suitable for transferring very large files. You must bear in mind that the client issues the *SIZE* command even if the progress bar is disabled.

If you need to support clients of this type, is makes sense to disable the *SIZE* command with the *-disableSizeCommand* option. You must bear in mind here that it may then not be possible to execute the client commands *reget* and *reput* on the server involved.

-disableSizeCommand
NO | **YES**

NO
Support of the *SIZE* command is not disabled.
NO is the default.

YES
Support of the *SIZE* command is disabled.

-tlsProtocol

OpenSSL supports Versions 3 of the SSL protocol and also Versions 1, 1.1 and 1.2 of the TLS protocol. Some of these protocols can be activated selectively using the *-tlsProtocol* option.

-tlsProtocol
[**+** | **-**] {**SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **All** } ...

+
The protocol specified after this sign is permissible.

If neither “+” nor “-” is specified, this has the same effect as specifying “+”.

-
The protocol specified after this sign is not permissible.

SSLv3
SSL protocol Version 3

Version 3 of the SSL protocol displays some security-related deficiencies and should therefore not be used if possible.

TLSv1
TLS protocol Version 1

TLSv1.1
TLS protocol Version 1.1

TLSv1.2
TLS protocol Version 1.2

ALL
All protocols are to be enabled.

All -SSLv3 is the default.

Example

The specifications `-tlsProtocol TLSv1 TLSv1.1 TLSv1.2` and `-tlsProtocol All -SSLv3` have the same effect as long as no support of the future TLS version 1.3 is added to the FTP.

-tlsCipherSuite

The *-tlsCipherSuite* option is used to specify a cipher suite preference list. If this option is not specified, a default preference list is used.

-tlsCipherSuite
<specification>

<specification>
Specification of a a cipher suite preference list, see chapter “Specification of a cipher suite preference list” on page 375.
ALL: !EXP: !ADH:!RC4 is the default.

-tlsRSAcertificateFile

The *-tlsRSAcertificateFile* option is used to specify a file which contains the RSA-based X.509 server certificate in PEM format. This file can also contain the private RSA server key. However, generally the certificate and key are stored in different files. In this case the key file is specified using the *-tlsRSAkeyFile* option (see page 86).

-tlsRSAcertificateFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file which contains the RSA-based X.509 client certificate in PEM format.

***NONE**
No file with RSA certificates is used.
*NONE is the default.

-tlsRSAkeyFile

The *-tlsRSAkeyFile* option is used to specify a file which contains the private RSA server key in PEM format.

If both an X.509 client certificate and a private server key are contained in the same file (see the *-tlsRSAcertificateFile* option on page 85), the *-tlsRSAkeyFile* option need not be specified.

As it should be possible to start up the FTP server automatically in unattended operation, no passphrase may be entered for the private server key at server startup. You must therefore remove any existing encryption of the private key with a passphrase. In this event, ensure that unauthorized persons cannot access this key.

-tlsRSAkeyFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file which contains the private RSA server key.

***NONE**
No separate file is used for the RSA server key.
*NONE is the default.

-tlsDSAcertificateFile

The *-tlsDSAcertificateFile* option is used to specify a file which contains the DSA-based X.509 server certificate in PEM format. This file can also contain the private DSA server key. However, generally the certificate and key are stored in different files. In this case the key file is specified using the *-tlsDSAkeyFile* option (see page 87).

-tlsDSAcertificateFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file which contains the DSA-based X.509 client certificate in PEM format.

***NONE**
No file with DSA certificates is used.
*NONE is the default.

-tlsDSAkeyFile

The *-tlsDSAkeyFile* option is used to specify a file which contains the private DSA server key in PEM format.

If both an X.509 client certificate and a private server key are contained in the same file (see the *-tlsDSAcertificateFile* option on page 86), the *-tlsDSAkeyFile* option need not be specified.

As it should be possible to start up the FTP server automatically in unattended operation, no passphrase may be entered for the private server key at server startup. You must therefore remove any existing encryption of the private key with a passphrase. In this event, ensure that unauthorized persons cannot access this key.

-tlsDSAkeyFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file which contains the private DSA server key.

***NONE**
No separate file is used for the DSA server key.
*NONE is the default.

-tlsCertificateChainFile

The *-tlsCertificateChainFile* option is used to specify a file in which all certificates can be stored which are required for verification of the server certificate. The first certificate in this file is the server certificate. The remaining certificates must form an unbroken chain, starting with the certificate of the CA which issued the server certificate, through to the root certificate of a CA which can be verified directly by the FTP client. The certificates in the chain must be sorted in such a way that the root certificate is in last place.

The specified file is only required if the server certificate was issued by a CA that is not known to the FTP clients and verification can thus not be performed by the FTP clients without the certificate chain being sent. This mechanism requires that RSA and DSA certificates should not be used simultaneously for the server, as the file is used for both variants.

-tlsCertificateChainFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file in which all certificates are stored which are required for verification of the server certificate.

***NONE**
No file is specified.
*NONE is the default.

-tlsCAcertificateFile

The *-tlsCAcertificateFile* option is used to specify a file containing the CA certificates in PEM format which are required for FTP client authentication. The individual PEM certificates are arranged sequentially in the file.

You can process the file with a text editor of your choice when you wish to add or delete certificates. The individual certificates are registered in the file as follows:

```
-----BEGIN CERTIFICATE-----
< CA certificate in Base64 encoding >
-----END CERTIFICATE-----
```

Text outside these sequences is ignored by the FTP server and can therefore be used to identify the certificates which, owing to the ASN.1/Base64 encoding, are available in non-readable form.

-tlsCAcertificateFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
: Name of the file containing the certificates in PEM format which are required for FTP client authentication.

***NONE**
: No file is specified.
*NONE is the default.

-tlsAcceptableClientCAFile

When client authentication is enabled, the server notifies the clients upon TLS/SSL connection setup of the names of the CAs which it accepts as signatories of client certificates. These name are taken from the certificates in the file specified by the *-tlsAcceptableClientCAFile* option. The individual certificates in PEM format are arranged sequentially in this file.

You can process the file with a text editor of your choice when you wish to add or delete certificates. The individual certificates are registered in the file as follows:

```
-----BEGIN CERTIFICATE-----
< CA certificate in Base64 encoding >
-----END CERTIFICATE-----
```

Text outside these sequences is ignored by the FTP server and can therefore be used to identify the certificates which, owing to the ASN.1/Base64 encoding, are available in non-readable form.

-tlsAcceptableClientCAFile
<file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file.

***NONE**
No file is specified.
*NONE is the default.

-tlsCArevocationFile

The *-tlsCArevocationFile* option is used to specify a file which contains the CRLs (Certificate Revocation Lists) of the Certificate Authorities (CAs). (Certificates issued by a Certificate Authority can be declared invalid by publication of a Certificate Revocation List (CRL).)

-tlsCArevocationFile
file-name 1..54> | ***NONE**

<file-name 1..54>
Name of the file which contains the CRLs of the Certificate Authorities.

***NONE**
No file with CRLs is specified.
*NONE is the default.

-tlsVerifyClient

The *-tlsVerifyClient* option is used to define whether an FTP client requires a certificate to access a server.

-tlsVerifyClient
NONE | **OPTIONAL | REQUIRE**

NONE
The FTP server does not request a certificate from the FTP client.
NONE is the default.

OPTIONAL
The FTP server requests the FTP client to send its certificate. If the client refuses to do this or supplies an invalid certificate, access is nevertheless allowed.

REQUIRE
The FTP client must transfer a valid certificate, otherwise access is refused.

-tlsVerifyDepth

The *-tlsVerifyDepth* option is used to define the verification depth, in other words the maximum permissible number of certificates between the FTP client certificate and the certificate which is known to the FTP server.

Here you must note the following:

- If the value 1 (default) is specified as the maximum depth, the client certificate must have been signed directly by a Certificate Authority (CA) that the FTP server knows for it to be accepted.
- If the maximum depth is exceeded, the connection is cleared, unless mandatory verification of the FTP client certificate has been disabled with *-tlsVerifyClient NONE* (see page 91) or *-tlsVerifyClient OPTIONAL*.
- Specifying the depth as 0 is meaningless. In this case only self-signed certificates would be permissible.

-tlsVerifyDepth
<depth>

<depth>
Maximum permissible number of certificates between the FTP client certificate and the certificate which is known to the FTP server.
Default: 1

-tlsSecureControlConnection

The *-tlsSecureControlConnection* option is used to define whether the control connection from the FTP client to the FTP server is to be secured with TLS.

-tlsSecureControlConnection
NONE | **OPTIONAL | REQUIRE**

NONE
The control connection is never secured, in other words a corresponding *AUTH* command (see page 54) is rejected with a negative return code.
NONE is the default.

OPTIONAL
The control connection is secured when this is requested by the client.

REQUIRE
A login is only permitted if the control connection is secured beforehand.

-tlsSecureDataConnection

The *-tlsSecureDataConnection* option is used to define whether the data connection from the FTP client to the server child should be secured with TLS.

As the data connection can only be secured if the control connection is secured, it makes no sense to select a weaker setting for *-tlsSecureControlConnection* (see page 93) than for *-tlsSecureDataConnection*. Consequently *-tlsSecureControlConnection* will, if required, automatically be raised to the same value as *-tlsSecureDataConnection*.

-tlsSecureDataConnectionr
NONE | **OPTIONAL | REQUIRE**

NONE

The control connection is never secured, in other words a corresponding *PROT* command (see page 54) is rejected with a negative return code.
NONE is the default.

The setting “NONE” makes sense if, for example, you only want to offer the option of transferring the password in encrypted form but are not ready or, because of the server performance, not able to offer encryption of the files transferred. However, here you must bear in mind that a large number of Windows FTP clients do not permit the option of just securing the control connection.

OPTIONAL

The data connection is secured when this is requested by the client.

REQUIRE

Data transfer is only permitted if the data connection is secured beforehand.

-tlsRandFile

The *-tlsRandFile* option is used to specify a file from which data is read for initializing the pseudo random numbers generator (PRNG) when the server is started. When the server is shut down, the relevant data from the PRNG is read to this file so that it can be used the next time the server is started.

If several FTP servers are operated in parallel, a separate file must be defined for each server. If the BS2000 subsystem PRNGD is active, no file is required to initialize the PRNG.

CAUTION!

This file may not be accessible to unauthorized people.

-tlsRandFile
<file-name 1..54>

<file-name 1..54>
Name of the file which contains the data for initializing the PRNG.
Default: SYSDAT.TCP-IP-AP.*nnn*.FTPD.RAND

-tlsOpenSSLlibName

The *-tlsOpenSSLlibName* option is used to specify the LMS file from which the OpenSSL library is dynamically loaded. It may be necessary to specify a name other than the default name if, for example, the OpenSSL library is also used by other products.

Dynamic loading of the OpenSSL library can be expedited with the aid of DAB using caches. If the OpenSSL library is used jointly by a number of products, the size of the DAB buffer used is reduced.

-tlsOpenSSLlibName
<openssl-libname>

<openssl-libname>
Name of the LMS file from which the OpenSSL library is to be dynamically loaded.
Default: LMS file to which the IMON logical ID SYSLNK refers.

-sizeCmdTimeLimit

With the *-sizeCmdTimeLimit* option the execution time of the SIZE command can be limited. The rationale for this is the same as for the *-disableSizeCommand* option (see page 83), except that here the SIZE command is not deactivated completely, but the execution of the command will be aborted when the resource consumption bevomes unreasonably large.

-sizeCmdTimeLimit
<number-of-seconds>

<number-of-seconds>
Time limit in seconds.
Default: 60

-cmdBufSize

The *-cmdBufSize* option enables the size of a buffer which is used to list file names to be changed. In the case of IDs with a very large number of files, the presetting can be too small, which, for example, will result in the client command *ls* or *dir* being aborted with an error. The following rule applies: the buffer must be as large as the output of the corresponding FSTAT command.

-cmdBufSize
<number-of-bytes>

<number-of-bytes>
Buffer size.
Default: 524 288

-TVFS

You can use the *-TVFS* option to enable or disable the TVFS (Trivial Virtual File System).

-TVFS
ON | **OFF**

ON
TVFS is enabled, in other words, DVS and POSIX files can be accessed via a standardized file system similar to POSIX. For further information on the TVFS, see the "FTP client in BS2000" section in the "interNet Services User Guide".
If you use mainly FTP clients with a graphical user interface, you should select ON. If the respective FTP client provides an option to send additional (proprietary) commands to the server at the beginning of the session, you may also enable or disable the TVFS as necessary for the respective FTP session using the BS2000-specific FTP server command *svfs*. For details on *svfs*, see the "FTP servers in BS2000" section in the "interNet Services User Guide".

OFF
TVFS is disabled.
OFF is the default setting because an option is provided here that enables you to use file partial qualifiers as pseudo subdirectories, which meets the needs of human users.

4.4 Starting up and shutting down the FTP server

You can use the commands described below to start up the FTP server.

These commands can also be entered at the operator console.

4.4.1 Starting up the FTP server

The following requirements must be met:

- The /START commands are only permitted under IDs that have the NET-ADMIN privilege.
- The TCPIPAP subsystem must be started before /START-FTP-DEMON or /START-TCP-IP-DEMON.
- The FTP server must be started under the $TSOS ID.
- To start up an FTP server secured by means of FTAC, you must make sure that FTAC is ready for operation when the FTP server is started up and thus, in particular, that the FTAC and FT subsystems have been activated.

The */START* commands for the Enter jobs are as follows:

/START-TCP-IP-DEMON	Enter job for TCP-IP-AP
/START-FTP-DEMON	Enter job for the FTP server

If you want to start FTP and TELNET simultaneously, use the START-TCP-IP-DEMON command.

4.4.2 Shutting down the FTP server

The commands for shutting down the FTP server which are described below are only valid for servers as of interNet Services V3.0.

You shut down the FTP server with the STOP-FTP-DEMON command.

STOP-FTP-DEMON
PORT-NUMBER=*STD-PORT/*ANY/<integer 0..32767>

PORT-NUMBER=
Specifies the port number of the FTP server to be shut down.
Default: The FTP server with default port number 21 is shut down.

PORT-NUMBER=*STD-PORT
Has the same effect as specifying no parameters.

PORT-NUMBER=*ANY
All active FTP servers are shut down.

PORT-NUMBER=<integer 0..32767>
An FTP server with the specified port number is to be shut down.

4.4.3 Setting up further FTP server tasks

It may be desirable in certain situations to operate additional FTP servers, for example as test versions. A copy is made of the standard ENTER jobs SYSENT.TCP-IP-AP.*nnn*.FTPD and SYSENT.TCP-IP-AP.*nnn*.FTPDC for this purpose.

When configuration takes place via an option file, a second option file with correspondingly modified server options must be created. The name of this second option file must be specified in the copy of SYSENT.TCP-IP-AP.*nnn*.FTPD using the *-M* option.

The following table describes the essential changes to be made to Server 2. The changes are highlighted in **bold**.

Server 1	**Server 2**
SYSENT.TCP-IP-AP.nnn.FTPD: /.FTPSR LOGON ... /SYSFILE SYSLST=$TSOS.SYSOUT.TCP-IP-AP.nnn.FTPD ... -B 0 \ -F $TSOS.SYSENT.TCP-IP-AP.nnn.FTPDC\ -E $TSOS.SYSDAT.TCP-IP-AP.nnn.SI\ -K $FTAC \ -J cccccccc\ -X 0\ -O 03600\ -A FTPSR\ -N 0015\ - D 0\ - T 0\ - S 0\ -C *MOD($TSOS.SYSLNK.TCP-IP-AP.nnn,FTPDC,RUN=ADV,PROG=ANY)	*SYSENT.TCP-IP-AP.nnn.FTPD2*: /.FTPS**2** LOGON ... /SYSFILE SYSLST=$TSOS.SYSOUT.TCP-IP-AP.nnn.FTPD**2** ... -B 0 \ -F $TSOS.SYSENT.TCP-IP-AP.nnn.FTPDC**2**\ -E $TSOS.SYSDAT.TCP-IP-AP.nnn.SI**2**\ -K $FTAC \ -J cccccccc\ -X 0\ -O 03600\ -A FTPS**2**\ -N 0015\ - D 0\ - T 0\ - S 0\ -C *MOD($TSOS.SYSLNK.TCP-IP-AP.nnn,FTPDC,RUN=ADV,PROG=ANY) \ **-P *nnn*** (***nnn*** is the port number)
SYSENT.TCP-IP-AP.nnn.FTPDC: /.FTPSR LOGON ... -N\ -E $TSOS.SYSDAT.TCP-IP-AP.nnn.SI\ -D 0\ -T 0	*SYSENT.TCP-IP-AP.nnn.FTPDC**2***: /.FTPS2 LOGON ... -N\ -E $TSOS.SYSDAT.TCP-IP-AP.nnn.SI**2**\ -D 0\ -T 0
Start of server task: /START-FTP-DEMON	Start of second server task: /ENTER-JOB SYSENT.TCP-IP-AP.nnn.FTPD2
Connection setup: /FTP open <host>	Connection setup: /FTP open <host> ***nnn*** (***nnn*** is a port number)

Other changes are possible to the server parameters. The backslash "\" after the server statements is interpreted as a continuation character. No further characters may follow "\".

4.4.4 Shutdown

Servers can still be shut down using the command */INFORM-PROGRAM 'shutdown',*TSN(<tsn>*. *<tsn>* is the TSN of the relevant server task.

4.4.5 Notes and restrictions for starting up and shutting down servers

The following points must be borne in mind when you start up or shut down servers:

- The START-FTP-DEMON command is only effective if the SYSENT.TCP-IP-AP.*nnn*.FTPD start procedure exists.

  If the commands are entered at the console they must also be shareable. This has no negative effect on security since from TCP-IP-AP V5.0 on all server options can be stored in a separate option file which does not need to be shareable (see the section “Configuring FTP via the option file” on page 67).

- The VERSION parameter in the START commands also permits servers with TCP-IP-AP Versions < 5.0 to be started. However, as the servers only log on to the TCPIPAP subsystem after every start as of V5.0, only such servers can be shut down again using the STOP command.

- A maximum of 20 users can be connected to the TCPIPAP subsystem. No more users are permitted owing to the size of internal tables. In practice the maximum number of 20 is, however, sufficient. If the maximum number is exceeded, the server shuts down on startup and issues the following message:

  ```
  "error: too many connections to Subsystem TCPIPAP"
  ```

- If the server was not started under TSOS it shuts down and issues the following message:

  ```
  "error: no privilege to connect to Subsystem TCPIPAP"
  ```

  This can only occur if you attempt to start the server by calling the start procedure explicitly instead of using the START-...-DEMON command, because this would be rejected under an ID other than TSOS.

4.4.6 Messages and return codes

Messages

```
TCP2000    (&00)-COMMAND FOR (&01)-SERVER SUCCESSFULLY COMPLETED.
           &00  =  START or STOP
           &01  =  FTP or TELNET or TCP-IP
```

Meaning
FTP/TELNET-server successfully started/stopped.

Response
<None>

```
TCP2001    STOP-COMMAND FOR (&00)-SERVER HAS NO EFFECT.
           &00  =  FTP or TELNET
```

Meaning
No servers existing.

Response
<None>

```
TCP2003    NO (&00)-SERVER FOR THE GIVEN PORTNUMBER.
           &00  =  FTP or TELNET
```

Meaning
No (&00) server for the given port number.

Response
<None>

```
TCP2004    WAS NOT ABLE TO START PROCEDURE FOR (&00): (&01).
           &00  =  FTP or TELNET or TCP-IP
           &01  =  <start procedure>
```

Meaning
Could not start procedure (&01) for (&00) server.

Response
<None>

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	See message TCP2000
	0	CMD0001	See message TCP2001
	1	TCP2003	See message TCP2003
	32	TCP2004	See message TCP2004
	32	CMD0220	An error occurred in the /CANCEL-JOB command

4.5 Notes on installing FTAC functionality

In addition to describing the FTAC-SUPPORT operand in the SET-FTP-TELNET-PARAMETERS command (see page 40/page 55), this section provides additional information to be noted when using FTAC functionality:

- In order to access the FTAC functionality, you will need to use openFT for BS2000.
- Once FTAC functionality is activated, further FTP access to the TSOS ID via an interactive logon is not possible because an alternative access option is provided in this case by means of FTAC transfer admission (but see the *-allowTsosLogin* option on page 80).
- In order to protect the TSOS password from being illegally intercepted, the input of a wrong LOGON password for the TSOS ID (when FTAC functionality is disabled) is penalized with a time delay, which increases with each further invalid attempt. Following each invalid password attempt, the connection request is rejected after about five seconds. It is only on completion of this time penalty that a successful LOGON attempt can be made, regardless of whether or not the correct password was specified in the mean time. The duration of the time penalty remains invisible to potential "intruders".
- If the FTAC check has been enabled, even the FTAC administrator and the owners of the user IDs involved in an FTP transfer will need to take some actions (e.g. adapt the admission sets), since FTP access could otherwise be locked for many/all user IDs.

  It is therefore important to define an appropriate migration strategy in advance:

  – One option is to select the FTAC check at level 1, i.e. where access is not verified as before by FTAC via an interactive logon, but where access can be additionally controlled via FTAC transfer admissions.

  – A further (or additional) option would be to install two servers in parallel (one with the standard port number 21 and one with some other port number) during a transitional phase, for example (see the section “Setting up further FTP server tasks” on page 99). In this case, only *one* server performs the full FTAC check (level 2).

4.6 Accounting in FTP

FTP accounting records can be collected and output to a file with the aid of FTP accounting (default: SYSDAT.TCP-IP-AP.*nnn*.FTPD.ACCOUNTING).

4.6.1 Enabling/disabling accounting and defining the accounting file

The following options are available for enabling/disabling FTP accounting and defining the accounting file:

- Installation command SET-FTP-TELNET-PARAMETERS (see page 40/page 55)
- Options *-acctActive* and *-acctFile* (see page 79)

You can use the command */INFORM-PROGRAM* to modify these settings during operation in order to enable/disable recording of the accounting records or to change the accounting file:

`/INFORM-PROGRAM 'acctActive: ',*TSN(<tsn>)`	Enable/disable FTP accounting (see page 113)
`/INFORM-PROGRAM 'acctFile: ',*TSN(<tsn>)`	Specify new accounting file (see page 113)

4.6.2 Structure of the accounting records

The structure of the FTP accounting record in the accounting file is based on the structure of the FT accounting record. The individual components of an accounting record are connected to each other in such a way and provided with length and offset information so as to ensure compatibility with older evaluation programs even with the following modifications:

- Individual components enlarged
- Number of variable record extensions increased

The FTP accounting record consists of the following four components:

1. Record description
2. Identification section
3. Basic information
4. Variable information

Description of the components of the FTP accounting record

The individual parts of the record contain the following identifiers:

- Field number: Sequence number of the data field within the written record part
- Offset: Relative distance of the data field to the start of the written record part
- Length: Length of the data field (in bytes)
- Format: Format of the data field

The following abbreviations are used in the description of the components of the FTP accounting record provided below:

A = Alphanumeric (including $, # and @)
B = Binary number
C = Printable characters
F = File name for BS2000
Z = Time specification in the format YYYYMMDDHHMMSS
- = Undefined

Record description section

Field no.	Offset	Length	Format	Meaning
1	0x00	4	A	Record ID “FTP0”
2	0x04	8	-/B	Time stamp of the time-of-day clock
3	0x0C	2	B	Length of the identification section
4	0x0E	2	B	Length of the basic information
5	0x10	4	-	Reserved

Identification section

Field no.	Offset	Length	Format	Meaning
1	0x00	8	A	User ID
2	0x08	8	A	Accounting number
3	0x10	4	Z	TSN of the FTP child

Basic information

Field no.	Offset	Length	Format	Meaning
1	0x00	14	Z	Command reception time
2	0x0E	14	Z	End of transfer time
3	0x1C	1	C	Result of transfer: +: successfully completed -: errored 0: indeterminate
4	0x1D	3	-	Reserved
5	0x20	8	B	Number of bytes from/to disk
6	0x28	8	B	Number of bytes to/from network
7	0x30	4	B	Number of disk accesses
8	0x34	4	B	CPU time used in milliseconds

Explanation of field no. 3

The FTP server cannot always recognize whether or not transfer has failed. Reason: In many cases the end of file transfer is signaled by the data connection clearing down. However, this connection cleardown can also be caused by an error. In such cases the “Result of transfer” is marked with “0” (= indeterminate).

Explanation of field no. 8

The CPU utilization from the end of the preceding transfer (or from the start of the child) to the end of the transfer of the current file is measured here. This includes activities such as setting options, and switching and listing directories. Some activities cannot be assigned to a subsequent file transfer because the connection had been cleared down in the meantime. These activities are incorporated in a special accounting record which is written when a connection finishes. The value X’00’ is entered in this record as the result of the transfer. The “File name” field contains no entry (see the table “Record extension for the file name” on page 107).

Variable information

Field no.	Offset	Length	Format	Meaning
1	0x00	2	B	Number of extensions = 1
2	0x02	2	B	Offset between the record extension for the partner ID and the start of record
3	0x04	2	B	Offset between the record extension for the file name and the start of record
If an offset of 0 is set, the corresponding record extension is not specified.				

Header of the variable section

Field no.	Offset	Length	Format	Meaning
1	0x00	2	A	Extension ID = “PI”
2	0x02	1	B	Extension type = 0x00
3	0x03	1	-	Reserved
4	0x04	2	B	Length of the extension (without ID, type and length field)
5	0x06	1	B	Address type: 1: IPv4 2: IPv6
6	0x07	16	B	IP address (left-justified)
7	0x17	1	-	Reserved
8	0x18	2	B	Length of the partner name
9	0x1A	see field 8	F	Partner name

Record extension for partner identification

Field no.	Offset	Length	Format	Meaning
1	0x00	2	A	Extension ID = “FN”
2	0x02	1	B	Extension type = 0x00
3	0x03	1	-	Reserved
4	0x04	2	B	Length of the file name
5	0x06	see field 4	F	File name

Record extension for the file name

4.7 Logging file of the FTP server

The FTP server logs its outputs in a logging file with the default file name SYSOUT.TCP-IP-AP.*nnn*.FTPD. The logging file always contains the difference entries for the current backup with the */INFORM-PROGRAM* command rdProt (see the section “rdProt-Save logging file of FTP servers” on page 114).

4.8 Displaying the current settings of the FTP server

You can use the BS2000 command SHOW-FTP-TELNET-STATUS to obtain information on the current settings of BS2000 FTP servers.

The following information is output:

- Settings made when the servers were generated
- Current information on the TSN of the server task’ and number of active connections

The servers place the data in auxiliary files with the following names:

- SYSDAT.TCP-IP-AP.*nnn*.FTPD.CONF.<port>

<port> specifies the port number of the relevant server.

These files are deleted when the relevant FTP server is shut down.

SHOW-FTP-TELNET-STATUS
SERVER= *FTP(...)/ ***TELNET**(...) ***FTP**(...) **PORT-NUMBER= *STD-PORT / *ANY** / <integer 0..32767> ***TELNET**(...) **PORT-NUMBER=*STD-PORT / *ANY** / <integer 0..32767> **, INFORMATION= *STD /*ALL** **, OUTPUT=*SYSOUT/*SYSLST**

SERVER=
Name of the server whose data is to be output (FTP or TELNET).

SERVER=*FTP(...)
Output of an FTP server's configuration data. This is the default.

PORT-NUMBER=
Port number of the FTP server whose configuration data is to be output.

PORT-NUMBER=*STD-PORT
Port number 21. This is the default.

PORT-NUMBER=*ANY
Information on all FTP servers currently active is output.

PORT-NUMBER=<integer 0..32767>
Port number of the FTP server whose configuration data is to be output.

INFORMATION=
Type and scope of the information output.

INFORMATION= *STD
Output of a list of servers specified by PORT-NUMBER. This is the default.

INFORMATION= *ALL
Output of all information on all servers specified by PORT-NUMBER.

OUTPUT=
Output medium to which the information is to be written.

OUTPUT=*SYSOUT
Output is to SYSOUT. This is the default.

OUTPUT=*SYSLST
Output is to SYSLST.

Messages and return codes

Messages

TCP9240 COMMAND SHOW-FTP-TELNET-STATUS SUCCESSFULLY COMPLETED.

Meaning
SHOW-FTP-TELNET-STATUS successfully completed.

Response
<None>

TCP9241 ERROR DMS(&00) DURING EXECUTION OF FSTAT.

Meaning
Error in FSTAT on configuration files.

Response
Check whether any servers are active.

TCP9242 COULD NOT OPEN CONFIGURATION FILE (&00).

Meaning
Configuration file (&00) could not be opened.

Response
Check whether the desired server is active.

TCP9243 COULD NOT READ CONFIGURATION FILE (&00).

Meaning
Configuration file (&00) could not be read.

Response
Check file.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	TCP9240	Command successfully completed
	64	TCP9241	See message TCP9241
	64	TCP9242	See message TCP9242
	64	TCP9243	See message TCP9243

4.9 Console interface

Some commands can also be specified by the system operator via the console interface. These commands are introduced with the command */INFORM-PROGRAM*. Commands issued with /INFORM-PROGRAM are used to

– control server traces,
– exit the various server tasks,
– control FTP accounting,
– save the logging files.

The following server commands are possible via the console interface:

Operation	**Brief description**
debug	Enable / disable trace to user level
trace	Enable / disable trace to TCP/IP interface
shutdown	Shut down service
acctActive	Enable / disable FTP accounting
acctFile	Change to new accounting file for FTP accounting
rdProt	Save the FTP server's logging file

The commands for starting up and shutting down the FTP server (see page 98) can also be entered via the console.

debug - Enable / disable trace to user level

/INFORM-PROGRAM
'**debug** <debug-value>',***TSN**(<tsn>)

<tsn>
TSN of the server task for which the trace to the user level is to be activated.

debug <debug-value>
The permitted values are from 0 to 9. Higher values result in the output of more information. A value of 0 means that the trace is disabled. This command is used by the system customer service for error diagnostics. A debug value > 2 is not meaningful.

trace - Enable / disable trace to TCP/IP interface

/INFORM-PROGRAM
'trace <trace value>**'**,***TSN**(<tsn>)

<tsn>
TSN of the server task for which the trace to the TCP/IP interface is to be activated.

trace <trace-value>
The permitted values are from 0 to 9. Higher values result in the output of more information. A value of 0 means that the trace is disabled. This command is used by the system customer service for error diagnostics.

shutdown - Shut down service

/INFORM-PROGRAM
'shutdown',***TSN**(<tsn>)

<tsn>
TSN of the server task to be shut down.

acctActive - Enable / disable FTP accounting

/INFORM-PROGRAM
'**acctActive** {**ON** | **OFF**}',***TSN**(<tsn>)

<tsn>
TSN of the server task.

ON
FTP accounting is enabled.
(Default accounting file: SYSDAT.TCP-IP-AP.*nnn*.FTPD.ACCOUNTING)

OFF
FTP accounting is disabled.

acctFile - Change accounting file for FTP accounting

/INFORM-PROGRAM
'acctFile <accounting-file-name>',***TSN**(<tsn>)

<tsn>
TSN of the FTP server.

<accounting-file-name>
Name of the new accounting file.

The current accounting file is closed and can be evaluated.

rdProt- Save logging file of FTP servers

The *rdProt* command is used to save the logging file of an FTP server under the name of the original logging file (default file name: SYSOUT.TCP-IP-AP.*nnn*.FTPD), extended by a suffix specifying the date and time in the format <MMDD><HHMMSS>. This file always contains the difference entries for the previous backup. Make sure that you do not select a name that is too long for the logging file: *rdProt* fails when the name exceeds the permissible length of 38 characters after the suffix is appended.

/INFORM-PROGRAM
'rdProt ',*TSN(<tsn>)

<tsn>
TSN of the server task.

4.10 IPv6 addresses in FTP

This section deals with the following topics in conjunction with the use of IPv6 addresses in FTP:

- Control connection setup
- Data connection setup in active and passive mode
- Proxy mechanism
- Note on use in heterogeneous networks

4.10.1 Control connection setup

When the control connection is set up using the client command *open*, an IPv6 address can be used which is specified in hexadecimal notation with a colon (:) (see the manual “interNet Services User Guide”).

4.10.2 Data connection setup

The following two FTP protocol commands are available for setting up the data connection with IPv6 addresses:

- EPRT: PORT command extended by IPv6 capability
- EPSV: PASV command extended by IPv6 capability

The following two modes are available for setting up data connections:

- Active mode
- Passive mode

Active mode

Connection setup takes place with an active child and passive client. The child is notified of its own connection endpoint with the aid of the following commands:

- PORT command if the home address is an IPv4 address (IPv4-MAPPED address).
- EPRT command if the home address is an IPv6 address.

PORT a1, a2, a3, a4, p1, p2	ai, i = 1,..,4: IPv4 address; pi, i = 1, 2: port number
EPRT |2|a|p|	2: IPv6 protocol, a: IPv6 address in colon notation, p: port number

Passive mode

Passive mode is selected using the client command *passive* (see the manual "interNet Services User Guide").

The data connection is set up with a passive child and active client. The child sends its IPv4 address including the port number (PASV) or just its port number (EPSV), which the client then combines with the server's IPv6 address known from the control connection.

This process is started by one of the two following commands sent by the client:

- PASV command if the server's IP address for the control connection is an IPv4-MAPPED address.
- EPSV command if the server's IP address for the control connection is an IPv6 address.

227 Entering Passive Mode (a1, a2, a3, a4, p1, p2)	ai, i = 1,..,4: IPv4 address; pi, i = 1, 2: port number (response to PASV)
229 Entering Extended Passive Mode EPRT (|||p|)	p: port number (response to EPSV)

Passive mode is now the default setting in many client implementations and is used to go from a LAN onto the Internet via a firewall. Such firewalls often prevent active connection setup to the user's own LAN.

4.10.3 Proxy mechanism

In the event of data transfer between two FTP servers (referred to as first server and proxy server below) the FTP client sends either a PASV or EPSV command to the proxy server, depending on whether the proxy server can be reached on the control connection via an IPv4 or IPv6 address. The client sends the address information contained in the response to the first server using a PORT or EPRT command.

The following problem can occur here:

- The proxy server has an IPv6 address.
- The first server, however, is a pure IPv4 server which cannot be reached via IPv6 addresses and is consequently still operating with IPv4 FTP.

In order to solve this problem, the proxy server must have at least one IPv4 address in addition to the IPv6 addresses. You must specify this IPv4 address as an IPv4-MAPPED address when you set up the control connection. The PASV command then returns an IPv4 address under which the proxy server can be reached and which the first server can also process.

4.10.4 Notes on using heterogeneous networks

The most important requirement for using an IPv6 FTP in BS2000 is the existence of sockets with IPv6 capability. In BS2000 the SOC6 subsystem has been developed for this purpose. This subsystem is implemented as of SOCKETS(BS2000) Version 2.0A.

Please note the following here:

- If a host only has IPv4 addresses, you can use an FTP server with IPv6 capability provided sockets with IPv6 capability are available. However, no other IPv4 server may run in parallel.
- If a host only has IPv4 and IPv6 addresses, you should only use FTP with IPv6 capability. If you do not do this, an IPv6-capable client could attempt to set up an IPv6 connection, which would then fail.
- If a host only has IPv6 addresses, you must use the FTP with IPv6 capability.

4.11 SNMP subagent for FTP

The FTP server has its own subagent (FTP subagent), which is operated via a management application, the BCAM Manager.

The “SNMP Management for openNet Server and interNet Services” manual contains information on the following topics:

- Handling the BCAM Manager
- Software requirements
- Installation and deinstallation
- Starting up and shutting down the FTP subagent

Interaction between the FTP subagent and FTP server

The FTP server accesses the FTP subagent via the fixed port number 3237. Immediately after starting up, the FTP server reports to the FTP subagent, provided this subagent is also started, and provides it with the following information:

- Port number, under which the FTP subagent can access the FTP server
- Server port number for the control connection to the FTP client

Assuming a server entity with this server port number does not already exist, the FTP server creates the relevant server entry.

Each FTP server writes its two port numbers to the SYSDAT.TCP-IP-AP.*nnn*.SNMP file at startup. If the FTP subagent is only started subsequently, it can check SYSDAT.TCP-IP-AP.*nnn*.SNMP for the currently active FTP server and create the relevant data structures.

If the FTP server is shut down, it deletes its entry from the SYSDAT.TCP-IP-AP.*nnn*.SNMP file.

4.12 FTP exit

The following exits exist for the FTP client and the FTP server:

- FTP system exits
- Exit mechanisms for the FTP server and FTP client

4.12.1 FTP system exit

The FTP system exit enables the operator of a BS2000-FTP server to extend or modify FTP server functions and to add new functions. Such functions may include:

- security checks in addition to the interactive login on accessing the FTP server and for the FTP commands subsequently received,
- logging of verified FTP commands,
- modification of FTP commands.

If the system exit is called from the FTP server, the following actions must be performed, depending on the selected installation mode (see page 39 in the chapter “Installing FTP and TELNET”):

- A system exit level other than 0 must be set in the SDF command SET-FTP-TELNET-PARAMETERS (see page 59) or
- The `-systemExit 1` (see page 77) option must be specified in the option file or
- The *-x 1* option must be specified in the respective enter job.

If FTAC is used at the same time, the system exit is not selected for the subevents FTPLOG and FTPCMD.

In order to protect the BS2000 FTP server, the input of an incorrect LOGON password for the TSOS ID (when FTAC functionality is disabled) is penalized with a time delay, which increases with each further invalid attempt. Following each invalid password entry, the connection request is rejected after about five seconds. It is only on completion of this time that a successful LOGON attempt can be made, regardless of whether or not the correct password was specified in the mean time. The duration of the time penalty remains invisible to potential "intruders".

4.12.1.1 FTP system exit events

The FTP system exit is called for the events described below. Each event is uniquely identified by the main event name and the subevent name. At present, there is only one main event *FTP* for the FTP system exit routine, with the subevents *FTPLOG*, *FTPBYE*, *FTPCMD* and *FTPCMDE*.

When programming exit routines, you should bear in mind that there could be more main events and subevents in future versions of the exit.

Each call to the exit is issued with the port number of the FTP server and a "connection ID". If multiple FTP servers (with different port numbers) are started, a separate behavior can be implemented in the exit routine for each server by means of the port number. The "connection ID" enables all *FTPCMD*, *FTPCMDE*, *FTPBYE* subevents associated with a particular FTP session to be identified, i.e. all events with a certain "connection ID" belong to the last preceding FTPLOG event with the same "connection ID". This enables the exit routine to temporarily save and retrieve information that is needed by the same FTP session in subsequent calls.

Subevent FTPLOG

The subevent *FTPLOG* is called at each FTP login attempt. The exit routine can decide on the basis of the supplied parameters (user ID, account, password, partner system) whether or not this login attempt should be allowed. In addition, the exit routine can log all access attempts and return some parameters (user ID, account, password) changed by using the "Modify call" return code. This can be used, for example, to make the output of interactive access authorizations unnecessary, since the exit routine can map special FTP login passwords, which do not enable interactive access, to actual interactive passwords.

Subevent FTPBYE

The subevent *FTPBYE* is called when an FTP session is ended. The exit routine can then create a closing log of the FTP session, for example, and delete any buffered data, if necessary. In addition to the data that is always supplied, the user id, account, password, partner system and ftp login are always supplied as action classes with this call.

Subevent FTPCMD

The subevent *FTPCMD* is called on receiving certain FTP commands. Not only the name of the FTP (protocol) command is supplied, but also an associated action class (see the table below). Note that the decision to grant or deny permission should preferably be based on the action class instead of the command name to avoid unnecessary updates in case the set of verified FTP commands is changed in future versions.

The command parameter (file or directory name) is also supplied.

Command	Action class
RETR	YAPXREAD
STOR	YAPXWRT
STOU	YAPXWRT
APPE	YAPXWRT
RNFR	YAPXMODA
DELE	YAPXDEL
FILE	YAPXCREA
PWD	YAPXSHDR
XPWD	YAPXSHDR
CWD	YAPXSHDR
XCWD	YAPXSHDR
LIST	YAPXSHDR
NLST	YAPXSHDR
CDUP	YAPXSHDR
XDUP	YAPXSHDR
MKD	YAPXCRDR
XMKD	YAPXCRDR
RMD	YAPXDLDR
XRMD	YAPXDLDR
SIZE	YAPXSHDR
MDTM	YAPXSHDR
MLSD	YAPXSHDR
MLST	YAPXSHDR

The system exit is not called for any of the other FTP commands.

If the exit is caused by a command that results in a write access from the viewpoint of the server, a parameter is used to specify whether this access is to extend, replace or create a new file. In addition, the name of the file or directory on which the command works is also supplied. This name could potentially be modified (by adding a prefix, for example) and returned, and the FTP server could then use only this modified name thereafter. The "Modify call" return code should be used here.

Subevent FTPCMDE

The subevent *FTPCMDE* is called with the same parameters as *FTPCMD* when a command has been processed. It is also established whether the command has been terminated without errors (YAPXCMDR field with the values YAPXOK or YAPXERR).

Register values

On entering the exit routine, registers 4 to 11 are undefined, and the remaining registers have the following values:

R0 = Exit number '023'

R1 = A(YAPXPARL) = FTP system exit parameter list; see below

R2 = A(Task Control Block)

R3 = A(Executive Vector Table)

R12 = A(TPR Program Manager)

R13 = A(Save area of the calling component)

R14 = A(Indirect return)

R15 = A(Exit routine)

Registers 12, 13 and 14 must not be destroyed by the exit routine.

Return information

The exit routine supplies return information to the calling FTP component in register 15 in the following format:

R15 = X'BB0000RR'

BB = Return information of the basic mechanism

00	No error
04	Exit routine not active
08	Invalid call

RR = Return information of the exit routine to the calling FTP component.

The following values are possible:

00	Reject call
04	Accept call
08	Modify call

The YAPSEPA macro can be used to create a DSECT for the FTP system exit parameter list.

Layout of operand list

(macro resolution with MF=D and default values for PREFIX and MACID)

```
[label]    YAPSEPA   MF=D [,PREFIX=Y | <prefix>]
                              _
                          [,MACID=APX | <mac-id>]
                                  ___
```

Example

```
           YAPSEPA MF=D
 1         MFTST MF=D,PREFIX=Y,MACID=APX,ALIGN=F,                          C
 1               DMACID=APX,SUPPORT=(D,C,M,L),DNAME=APXPARL
 2 YAPXPARL DSECT ,
 2               *,##### PREFIX=Y, MACID=APX #####
 1 *   Parameter area
 1 *
 1 YAPXHDR  DS    0XL16                   Header
 1 YAPXVERS DS    F                       Interface version
 1 *   VERSION
 1 YAPXVER1 EQU   1                       Internet Services V1.0
 1 YAPXVER2 EQU   2                       Internet Services V2.0
 1 *
 1 YAPXMCAS DS    F                       Exit main case
 1 *   Exit main case
 1 YAPXFTP  EQU   1                       FTP
 1 *
 1 YAPXSCAS DS    F                       Exit sub case
 1 *   Exit sub case
 1 YAPXFLOG EQU   1                       FTP login
 1 YAPXFCMD EQU   2                       FTP command
 1 YAPXFCDE EQU   3                       FTP command end
 1 YAPXFBYE EQU   4                       FTP logout
 1 *
 1 YAPXRSLT DS    F                       Internally used
 1 *
 1 *
 1 YAPXIND  DS    0XL856                  Input parameters
 1 YAPXPORT DS    F                       FTP server port number
 1 YAPXCNID DS    F                       Connection id
 1 YAPXACT  DS    F                       FTP action class
 1 *   Action class
 1 YAPXWRT  EQU   1                       write
 1 YAPXREAD EQU   2                       read
 1 YAPXSHAT EQU   4                       show attributes
 1 YAPXDEL  EQU   8                       delete
 1 YAPXCREA EQU   16                      create
 1 YAPXMODA EQU   32                      modify attributes
 1 YAPXSHDR EQU   64                      show directory
 1 YAPXMOVE EQU   128                     move
 1 YAPXCRDR EQU   256                     create directory
 1 YAPXDLDR EQU   512                     delete directory
 1 YAPXMDDR EQU   1024                    modify directory
 1 YAPXLOG  EQU   2048                    ftp login
 1 *
 1 YAPXWRMD DS    F                       Write mode
```

```
1 *   Write mode
1 YAPXNEW  EQU   1                        New
1 YAPXREPL EQU   2                        Replace
1 YAPXEXT  EQU   4                        Extend
1 *
1 YAPXPRTN DS    CL256                    Partner system
1 YAPXUSER DS    CL9                      User id
1 YAPXACCT DS    CL9                      Account
1 YAPXPASS DS    CL35                     Password
1 YAPXFILE DS    CL500                    File name
1 YAPXCMD  DS    CL5                      FTP command
1 YAPXRSV1 DS    XL2                      Reserved
1 YAPXCMDR DS    F                        FTP command rc
1 *   Command rc
1 YAPXOK   EQU   0                        Ok
1 YAPXERR  EQU   1                        Error
1 *
1 YAPXRSV2 DS    XL20                     Reserved
1 *
1 *
1 YAPXOUTD DS    0XL576                   Output parameters
1 YAPXUSRO DS    CL9                      User id
1 YAPXACCO DS    CL9                      Account
1 YAPXPASO DS    CL35                     Password
1 YAPXFILO DS    CL500                    File name
1 YAPXRSV3 DS    XL23                     Reserved
1 *
1 YAPX#    EQU   *-YAPXVERS
```

Alternatively, there is also a corresponding C include YAPSEPA.H:

```
#ifndef _YAPSEPA_H
#define _YAPSEPA_H

#if 0
/*****************************************************************
 BEGIN-INTERFACE   YAPSEPA

 TITLE             (/ TCP-IP-AP System exit parameter list /)
 NAME              YAPSEPA.H
 DOMAIN            TCP-IP-AP
 LANGUAGE          C
 COPYRIGHT         (C) Fujitsu Technology Solutions GmbH 2010
                       ALL RIGHTS RESERVED
 COMPILATION-SCOPE USER
 INTERFACE-TYPE    CALL
 RUN-CONTEXT       TU
 PURPOSE           (/ YAPSEPA describes the parameter list for
                      the FTP system exit.
                   /)

 END-INTERFACE     YAPSEPA.
*****************************************************************/
#endif

/* Version                                                      */
/* ENUM version_s                                               */
#define YAPSEPAvers1 1                /* Internet Services V1.0     */
#define YAPSEPAvers2 2                /* Internet Services V2.0     */

/* Exit main case                                               */
/* ENUM mainCase_s                                              */
#define YAPSEPAFTP 1                  /* FTP                        */

/* Exit sub case                                                */
/* ENUM subCase_s                                               */
#define YAPSEPAFTPLOG 1               /* FTP login                  */
#define YAPSEPAFTPCMD 2               /* FTP command                */
#define YAPSEPAFTPCMDE 3              /* FTP command end            */
#define YAPSEPAFTPBYE 4               /* FTP logout                 */

/* Action class                                                 */
/* ENUM action_s                                                */
#define YAPSEPAwrite 1                /* write                      */
#define YAPSEPAread 2                 /* read                       */
#define YAPSEPAshow_attr 4            /* show attributes            */
#define YAPSEPAdelete 8               /* delete                     */
```

```
#define YAPSEPAcreate 16                /* create                       */
#define YAPSEPAmod_attr 32              /* modify attributes            */
#define YAPSEPAshow_dir 64              /* show directory               */
#define YAPSEPAmove 128                 /* move                         */
#define YAPSEPAcre_dir 256              /* create directory             */
#define YAPSEPAdel_dir 512              /* delete directory             */
#define YAPSEPAmod_dir 1024             /* modify directory             */
#define YAPSEPAlogin 2048               /* ftp login                    */

/* Write mode                                                           */
/* ENUM writeMode_s                                                     */
#define YAPSEPAnew 1                    /* New                          */
#define YAPSEPAreplace 2                /* Replace                      */
#define YAPSEPAextend 4                 /* Extend                       */

/* Command rc                                                           */
/* ENUM commandRc_s                                                     */
#define YAPSEPAok 0                     /* Ok                           */
#define YAPSEPAerror 1                  /* Error                        */
/* Parameter area                                                       */
struct YAPSEPA_pl_mdl {

        /* Header                                                       */
        struct {
                unsigned long version;  /* Interface version            */
                unsigned long mainCase; /* Exit main case               */
                unsigned long subCase;  /* Exit sub case                */
                unsigned long result;   /* Internally used              */
        } header;

        /* Input parameters                                             */
        struct {
                unsigned long portNum;  /* FTP server port number       */
                unsigned long connId;   /* Connection id                */
                unsigned long action;   /* FTP action class             */
                unsigned long writeMode;
                                        /* Write mode                   */
                char partner[256];      /* Partner system               */
                char userId[9];         /* User id                      */
                char account[9];        /* Account                      */
                char password[35];      /* Password                     */
                char fileName[500];     /* File name                    */
                char command[5];        /* FTP command                  */
                char reserved_1[2];     /* Reserved                     */
                unsigned long commandRc;
                                        /* FTP command rc               */
                char reserved_2[20];    /* Reserved                     */
        } in_data;
```

```
/* Output parameters                                                  */
        struct {
                char userId[9];         /* User id                    */
                char account[9];        /* Account                    */
                char password[35];      /* Password                   */
                char fileName[500];     /* File name                  */
                char reserved_3[23];    /* Reserved                   */
        } out_data;
};

#endif           /* _YAPSEPA */
```

4.12.2 Exit mechanisms for the FTP server and FTP client

If you want to perform code conversions for the transferred data, where the number of bytes will not be preserved, then XHCS cannot be used. However, there is a special interface that can be used for the FTP server and FTP client to define your own code conversion routines as well as other exit routines, such as routines for "on the fly" data compressions. Such a user-defined exit routine is called once activated before data is sent and after data is received. Please note that with the transfer type ASCII, the exit routine always requires data that has been converted to ASCII.

4.12.2.1 Dummy routines

The exit routine is reloaded automatically from the module library SYSLNK.TCP-IP-AP.*nnn* or SKMLNK.TCP-IP-AP.*nnn* (on x86 systems) when the FTP client program or connection-specific server task is started. To avoid "UNRESOLVED EXTERNAL REFERENCES" problems here, the specified libraries must always contain a module with the entry YAPFEXIT.

The module libraries mentioned are therefore supplied with a dummy routine, which is contained in the LMS element EXITFTP. This routine does not perform any conversion rather logs the transferred data in standard output and otherwise returns it unchanged. You can use the source text for this routine, which can be found in the S element EXITFTP.C in the SYSSRC.TCP-IP-AP.*nnn* library, as the starting point for your own routines. The header file required here can be found in the SYSLIB.TCP-IP-AP.*nnn* library. If you want to create the routine in Assembler, you should use the macro YAPFX. Note that the routine is called in accordance with C conventions.

If you are using your own routine, you should note that only one module with the entry point YAPFEXIT may exist in the module library as otherwise the wrong module may be loaded. In particular, therefore, you should remove the module element EXITFTP supplied.

Because the entry point YAPFEXIT is not selected by default, you have to activate it using an FTP server option or a command.

4.12.2.2 User-defined exit routines

For handling the various exit routines, the callers of the exit routine (FTP client or FTP server) pass the address of a parameter list to the entry point *YAPFEXIT*. The structure *YAPFX_pl_mdl* from this parameter list is supplied in the header file *YAPFX.H* of the SYSLIB.TCP-IP-AP.*nnn* library. The *YAPFEXIT* signature is as follows:

```
void YAPFEXIT (struct YAPFX_pl_mdl*)
```

Structure of *YAPFX_pl_mdl*

The *YAPFX_pl_mdl* structure is defined as follows:

```
/* Return codes                                          */
/* ENUM rc_s                                             */
#define YAPFXok 0                    /* Ok               */
#define YAPFXbufTooShort 1           /* Buffer too short */
#define YAPFXother 255               /* Other error      */

/* Caller                                                */
/* ENUM caller_s                                         */
#define YAPFXclient 1                /* Client           */
#define YAPFXserver 2                /* Server           */


/* Action                                                */
/* ENUM action_s                                         */
#define YAPFXrecv 1                  /* Receive          */
#define YAPFXsend 2                  /* Send             */

/* Action modifier                                       */
/* ENUM actionm_s                                        */
#define YAPFXnone 0                  /* None             */
#define YAPFXfirst 1                 /* First            */
#define YAPFXlast 2                  /* Last             */

/* Parameter area */
struct YAPFX_pl_mdl {

/* Input parameters                                         */
  struct {
    unsigned long caller;          /* Calling instance    */
    unsigned short action;         /* Action              */
    unsigned short actionm;        /* Action modifier     */
    void *selector;                /* Selector            */
    char hostname[33];             /* Host name           */
    char reserved1[7];             /* Reserved            */
    unsigned long portNo;          /* Port number               */
```

```
    unsigned long connId;            /* Connection Id (only server)     */
    void *inBuf;                     /* Address of input buffer         */
    unsigned long inBufLen;          /* Length of data in input buffer  */
    void *outBuf;                    /* Address of output buffer *      */
    unsigned long outBufLen;         /* Length of output buffer         */
  } in_data;

/* Output parameters                                                    */
  struct {
    unsigned long outDataLen;        /* Lengh of data in output buffer  */
    unsigned long rc;                /* Return code */
  } out_data;
};
```

Description of parameters

caller
: Indicates whether the routine was called from the FTP client or FTP server.

action
: Indicates whether the call was made before data was sent (*YAPFXsend*) or after it was received (*YAPFXrecv*).

actionm
: Indicates whether the call is the first or last that belongs to the current transfer (see also the *connId* parameter). Where the files are very small, both flags can also be set at the same time.

selector
: *selector* allows you to choose the desired action from several available actions. A string can be given to the conversion routine for this purpose to which *selector* points. You can then decide freely which strings are used with which meaning. A fixed meaning is only assigned to the strings "*" and "*NONE".

hostName
: Supplies the name of the respective client host (terminated with a null byte).

portNo
: Supplied in the case of *caller* = *YAPFXserver* and specifies the port number under which the calling server offers its services. It is possible to distinguish in this way between several servers started at the same time.

connId
: Where there are several active connections, this option allows the individual connections to be identified by the pair (*portNo*, *connId*). This is necessary, for example, if connection-specific status information is to be stored and then retrieved beyond the

duration of a call. If only the first byte of a string containing two characters is supplied as the last byte in a call, for example, this byte must be buffered until it can be dealt with in the next call.

The flags in *actionm* determine how to use this type of buffered information:

– Status information from previous calls must be deleted with *YAPFXfirst*.

– All buffered data must be returned with *YAPFXlast*.

inBuf
Contains the address of the buffer containing the data for conversion (input buffer).

inBufLen
Specifies the length (number of valid bytes) in the receive buffer.

outBuf
Contains the address of the output buffer.

outBuflen
Specifies the length of the output buffer.

outDataLen
Before returning to the caller, the routine must indicate in *outDataLen* the number of valid bytes in the output buffer that thus have to be transferred.

rc
The conversion routine uses *rc* to indicate whether the conversion was terminated successfully.

Writing user-defined FTP exits under POSIX

As the POSIX FTP is created by linking the LLMs FTP and FTPEXIT, you can also work with the FTP exits under POSIX. The interNet Services package contains the dummy module FTPEXIT, which only executes a return.

You write your own FTP exits as follows:

- Create the version of the FTPEXIT you require.
- Link this version to FTP from SYSLNK.TCP-IP-AP.*nnn*.
- Write the resultant LLM to the SINLIB.TCP-IP-AP.*nnn* library under the name FTP.
- Then install POSIX-FTP (see the section “Installing and uninstalling FTP and TELNET clients in POSIX” on page 45).

Results and return codes for exit routines

If the output buffer is too small for the complete conversion, the exit routine must supply the return code YAPFXbufTooShort. The exit routine is called again in this case with the same data but a larger output buffer. This may happen several times until the output buffer is large enough or an internal limit on the part of the caller is exceeded.

In the case of all other errors, the exit routine must supply the return code YAPFXother. The caller then aborts the transfer and initiates a suitable error handling.

4.12.2.3 Enabling / disabling user-defined exit routines

Because user-defined exit routines are not selected by default, they have to be activated before they are used.

Enabling and disabling user-defined exit routines for the FTP server

User-defined exit routines for the FTP server can be activated either in the FTP server or FTP client.

Enabling user-defined exit routines for the FTP server in the FTP server

In order to activate the conversion routines in the FTP server, add the "-U" option to the /START-PROGRAM statement in the Enter file on the FTP server (SYSENT. TCP-IP-AP.*nnn*.FTPD) and then start the FTP server:

```
-U [receive: [<recv-selector>]!][send: [<send-selector>]]
```

The string <recv-selector> or <send-selector> is transferred in the *selector* parameter when the relevant conversion routine is called and may not exceed 32 characters.

Enabling user-defined exit routines for the FTP server in the FTP client

You can activate conversion routines for the FTP server from the FTP client by sending the server a *site exit* command from the client with an operand similar to the -U server option. The conversion routine is only called in this case with these parameters for transfers to this connection. If the FTP client from interNet Services ≥ V2.0 is used, *rexit* ... can also be specified instead of *quote site exit* ....

You should note the following here:

- The *site exit* command has precedence over the -*U* option. In other words, even if the FTP server was started with the -*U* option, the specifications of a *site exit* command sent from the client to the server are valid.
- By sending a *site exit* command with the selector string "*" the general server setting is again set as valid in accordance with the -*U* option.

Disabling user-defined exit routines for the FTP server

The exit routine for the FTP server is disabled for the current connection by sending a *site exit* command with the selector string "*NONE".

Enabling / disabling user-defined exit routines for the FTP client

User-defined conversion routines are enabled and disabled for the FTP client with the *exit* command.

Enabling user-defined exit routines

```
exit [receive:[<recv-selector>]!] [send:[<send-selector>]]
```

A proper selector must be specified for both <recv-selector> and <send-selector>, which may contain at most 32 characters.

Disabling user-defined exit routines

```
exit receive:*NONE or send:*NONE or receive:*NONE!send:*NONE
```

You have to specify `*NONE` for both <recv-selector> and <send-selector>.

5 TELNET configuration and operation

You can configure TELNET using either the SDF command or the option file.

This chapter covers the following topics related to the configuration and operation of TELNET servers:

- Using TLS/SSL to secure the TELNET server (see page 136)
- Configuration of TELNET via SDF command (see page 138) or option file (see page 148)
- Startup and shutdown of TELNET servers (see page 174)
- Saving the logging file of TELNET servers (see page 178)
- Displaying the current settings of the TELNET server (see page 178)
- Specification of commands using the console interface (see page 181)
- Using IPv6 addresses in TELNET (see page 183)
- Exits for TELNET clients and the TELNET server (see page 184)

5.1 TLS/SSL support on the TELNET server

TLS/SSL support on the TELNET server offers a wide range of setting options. You can make these settings as follows:

- With the aid of options which are stored in one or more option files and are interpreted when the TELNET server is started (see the section “Configuring TELNET using an option file” on page 148).
- With the aid of the installation command parameters SET-FTP-TELNET-PARAMETERS (see the section “Configuring TELNET using an option file” on page 148).

Parameterization of TLS/SSL support on the TELNET server

Below you will find an overview of the possible settings for TLS/SSL support on the TELNET server using the options. The individual options correspond to equivalent parameters in the SET-FTP-TELNET-PARAMETERS command.

The following options are available for setting parameters for TLS/SSL support on the TELNET server:

- START-TLS option (*-Z tls-required*, see page 157)

  This option allows you to control TLS support on the TELNET server. A raft of additional options (*-Z* options) is provided here (see the table on page 137). Provisions for authentication are negotiated by SSL to free TELNET of this load.

- AUTHENTICATION option (*-B*, see page 172)

  You use this option to negotiate the provisions for authentication. In BS2000 the AUTHENTICATION option is currently only implemented for TLS/SSL.

  You can control TLS support on the TELNET server using the AUTHENTICATION option implemented for TLS/SSL. To do this you use the same *-Z* options as in the START-TLS option (see the table on page 137).

- ENCRYPTION option (*-H*, page 173)

  You use this option to negotiate the encryption method and the key used. In TELENET, currently only DES64 variants DES_CFB64 and DES_OFB64 are supported.

The START-TLS option and AUTHENTICATION option may not be enabled simultaneously.

The following table lists the options with which you can control TLS/SSL support on the TELNET server in conjunction with the START-TLS / AUTHENTICATION option.

Option	Description	Page
-Z Protocol	Choose SSL protocol versions selectively	167
-Z CipherSuite	Specify cipher suite preference list	164
-Z RSAcertificateFile	Specify file which contains the RSA-based X.509 server certificate in PEM format	158
-Z RSAkeyFile	Specify file which contains the private RSA server key in PEM format	159
-Z DSACertificateFile	Specify file which contains the DSA-based X.509 server certificate in PEM format	160
-Z DSAkeyFile	Specify file which contains the private DSA server key in PEM format	161
-Z CertificateChainFile	Specify file in which all the certificates required for verification of the server certificate can be stored	166
-Z CACertificateFile	Specify file which contains the certificates required for authentication of the TELNET client in PEM format	162
-Z AcceptableClientCAFile	Specify file from which the names of the CAs that the server accepts as signatories of client certificates can be obtained	168
-Z CARevocationFile	Specify file which contains the CRLs of the CAs	163
-Z VerifyClient	Define whether the TELNET client must provide a certificate for server access	170
-Z VerifyDepth	Define verification depth	169
-Z RandFile	Specify file from which the data for initializing the PRNG is read when the server is started	165
-Z OpenSSLlibName	Define the LMS file from which the OpenSSL library should be dynamically loaded	171

5.2 Configuring TELNET using the SET-FTP-TELNET-PARAMETERS installation command

For the full command syntax and the description of the installation operands, see page 40.

SET-FTP-TELNET-**PAR**AMETERS

(...)
, TELNET-**SERVER-PROC= *NO / *CRE**ATE(...)
***CRE**ATE(...)
JOB-NAME= *STD / <name 1..5>
, JOB-CLASS= *STD / <name 1..8>
, CPU-TIME= *STD / <integer 1..32767>
, PRIORITY**= *STD** / <integer 0..255>
, DEBUG= *STD / <integer 0..9>
, TRACE= *STD / <integer 0..10>
, MAXIMUM-**CONN**ECTIONS**= *STD** / <integer 1..900>
, STATION-**ID= *STD** / <integer 0..6>
, ASCII-TABLE**= *STD** / <text 1..8>
, EBCDIC-TABLE**= *STD** / <name 1..8>
, TLS-SUPPORT= *STD / *NO / *PARAMETERS(...)
***PARAMETERS (...)**
OPTION= *STD / *START-TLS / *AUTHENTICATION(...)
***AUTHENTICATION(...)**
DEBUG= *STD / *NO / *YES
, PROTOCOL= *STD / <text 1..80>
, CIPHER-SUITE = *STD / <text 1..80_with-lower-case>
, RSA-CERTIFICATE-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, RSA-KEY-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, DSA-CERTIFICATE-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, DSA-KEY-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, CA-CERTIFICATE-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, CLIENT-CA-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, CERT-CHAIN--FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, CA-REVOCATION-FILE = *STD / *NONE / <filename 1..54_without-generation-version>
, RANDOM-FILE = *STD / <filename 1..54_without-generation-version>
, SSL-LIBRARY = *STD / *NONE / <filename 1..54_without-generation-version>
, VERIFY-CLIENT= *STD / *NONE / *OPTIONAL / *REQUIRE
, VERIFY-DEPTH= *STD / <1..32767>

```
SET-FTP-TELNET-PARAMETERS
        , ENCRYPTION= *STD / *NO / *YES(...)
            *YES(...)
                DEBUG= = *STD / *NO / *YES
                , KEY= <x-text 1..16>
        , SSL-LIBRARY = *STD / *NONE / <filename 1..54_without-generation-version>
        , OPTION-FILE= *STD / <filename 1..54_without-generation-version>
        , SELECTOR= *STD / <text 1..511>
        , PORT-NUMBER= *STD / <integer 0..32767>
        , SERVER-ENTER-FILE= *STD / <filename 1..54_without-generation-version>
        , LOGGING-FILE= *STD / <filename 1..54_without-generation-version>
(...)
```

TELNET-SERVER-PROC=
Parameters for TELNET server.

TELNET-SERVER-PROC=*NO
No installation is to be performed for the TELNET server.

TELNET-SERVER-PROC=*CREATE(...)
The TELNET server is to be configured. The Enter file for starting the TELNET server is created from the parameters entered.

JOB-NAME=
This name is used as a prefix. To construct the terminal name of the current connection, the sequence number is appended to the JOB-NAME. See also the *-A* option on page 150.

JOB-NAME=*STD
Corresponds to the entry TELSR or the value in the installation file.

JOB-NAME=<name 1..5>
Job name used as a prefix.

JOB-CLASS=
The job class in which the server process is to run. Make sure that Enter jobs with the following parameters may be started in this job class:
CPU-LIMIT=*NO, RUN-PRIORITY=120 and START=*IMMEDIATELY.

JOB-CLASS=*STD
Corresponds to the standard job class on the system or the value in the installation file.

JOB-CLASS=<name 1..8>
Name of the job class.

CPU-TIME=
Maximum CPU time available for the server process.

CPU-TIME=*STD
Corresponds to the entry NTL or the value in the installation file.

CPU-TIME=<integer 1..32767>
Maximum CPU time available for the server process.

PRIORITY=
Priority with which the server process is to run.

PRIORITY=*STD
Corresponds to the entry 120 or the value in the installation file.

PRIORITY=<integer 0..255>
Priority with which the server process is to run.

DEBUG=
Trace at user level. See also the *-D* option on page 150.

DEBUG=*STD
Corresponds to the entry 0 or the value in the installation file.

DEBUG=<integer 0..9>
Debug level.

TRACE=
Trace at TCP/IP (socket) interface. See also the *-T* option on page 153.

TRACE=*STD
Corresponds to the entry 0 or the value in the installation file.

TRACE=<integer 0..10>
Socket trace level.

MAXIMUM-CONNECTIONS=
Maximum number of connections that the server is to operate. See also the *-N* option on page 151.

MAXIMUM-CONNECTIONS=*STD
Corresponds to the entry 15 or the value in the installation file.

MAXIMUM-CONNECTIONS=<integer 1..900>
Maximum number of connections that the server is to operate.

STATION-ID=
Number of places with which the name of the remote computer or the job name of the TELNET server is to be taken into account in the terminal name. See also the *-D* option on page 150.

STATION-ID=*STD
Corresponds to the entry 0 or the value in the installation file.

STATION-ID=<integer 0..6>
Number of places with which the name of the remote computer or the job name of the TELNET server is to be taken into account in the terminal name.

ASCII-TABLE=
Standard ASCII code table (ISO 88591). See also the *-X* option on page 154.

ASCII-TABLE=*STD
Corresponds to the entry of 8 blanks or the value in the installation file.

ASCII-TABLE=<text 1..8>
Standard ASCII code table.
A non-empty value for ASCII-TABLE (specified as a command parameter or - if ASCII-TABLE=*STD - as a value in any specified installation file), is only effective if EBCDIC-TABLE (specified as a command parameter or - if EBCDIC-TABLE=*STD - as a value in any specified installation file) also does not contain an empty value.

EBCDIC-TABLE=
Standard EBCDIC code table (EDF041). See also the *-X* option on page 154.

EBCDIC-TABLE=*STD
Corresponds to the value of 8 blanks or the value in the installation file.

EBCDIC-TABLE=<name 1..8>
Standard EBCDIC code table.
A non-empty value for EBCDIC-TABLE (specified as a command parameter or, if EBCDIC-TABLE=*STD is set, as a value in any specified installation file), is only effective if ASCII-TABLE (specified as a command parameter or - if ASCII-TABLE=*STD - as a value in any specified installation file) also does not contain an empty value.

TLS-SUPPORT=
Defines whether the connection is to be secured with TLS/SSL.

TLS-SUPPORT=*STD
Default: *NO.

TLS-SUPPORT=*NO
The TELNET server does not secure the connection with the aid of the TELNET option START-TLS.

TLS-SUPPORT=*PARAMETERS(...)
The TELNET server secures the connection via TLS/SSL.

OPTION=
TELNET option for implementing TLS/SSL. See also the *-Z tls-required* option on page 157.

OPTION=*STD
Default: START-TLS

OPTION=*START-TLS
TLS/SSL is implemented using the TELNET option START-TLS.

OPTION=*AUTHENTICATION(...)
TLS/SSL is implemented using the TELNET option AUTHENTICATION. See also the *-B* option on page 172.

DEBUG=
Switch for the authentication trace.

DEBUG=*STD
Default: *NO

DEBUG=*NO
The authentication trace is not enabled.

DEBUG=*YES
The authentication trace is enabled.

PROTOCOL=
See the *-Z Protocol* option on page 167.

PROTOCOL=*STD
Default: ALL-SSLv2

PROTOCOL=<text 1..80>
Specification of the TLS/SSL protocol to be used.

CIPHER-SUITE=
See the *-Z CipherSuite* option on page 164.

CIPHER-SUITE=*STD
Default: ALL:!EXP:!ADH

CIPHER-SUITE=<text 1..80_with-lower-case>
Specification of the encryption algorithms to be used.

RSA-CERTIFICATE-FILE=
See the *-Z RSAcertificateFile* option on page 158.

RSA-CERTIFICATE-FILE=*STD
Default: *NONE

RSA-CERTIFICATE-FILE=*NONE
No RSA certificate file is specified.

RSA-CERTIFICATE-FILE=<filename 1..54_without-generation-version>
Name of the RSA certificate file.

RSA-KEY-FILE=
See the *-Z RSAkeyFile* option on page 159.

RSA-KEY-FILE=*STD
Default: *NONE

RSA-KEY-FILE=*NONE
No RSA key file is specified.

RSA-KEY-FILE=<filename 1..54_without-generation-version>
Name of the RSA key file.

DSA-CERTIFICATE-FILE=
See the *-Z DSAcertificateFile* option on page 160.

DSA-CERTIFICATE-FILE=*STD
Default: *NONE

DSA-CERTIFICATE-FILE=*NONE
No DSA certificate file is specified.

DSA-CERTIFICATE-FILE=<filename 1..54_without-generation-version>
Name of the DSA certificate file.

DSA-KEY-FILE=
See the *-Z DSAkeyFile* option on page 161.

DSA-KEY-FILE=*STD
Default: *NONE

DSA-KEY-FILE=*NONE
No DSA key file is specified.

DSA-KEY-FILE=<filename 1..54_without-generation-version>
Name of the DSA key file.

CA-CERTIFICATE-FILE=
See the *-Z CACertificateFile* option on page 162.

CA-CERTIFICATE-FILE=*STD
Default: *NONE

CA-CERTIFICATE-FILE=*NONE
No CA certificate file is specified.

CA-CERTIFICATE-FILE=<filename 1..54_without-generation-version>
Name of the CA certificate file.

CLIENT-CA-FILE=
See the *-Z AcceptableClientCAFile* option on page 168.

CLIENT-CA-FILE=*STD
Default: *NONE

CLIENT-CA-FILE=*NONE
No file is specified.

CLIENT-CA-FILE=<filename 1..54_without-generation-version>
Name of the file with the certificates of the accepted CAs.

CERT-CHAIN-FILE=
See the *-Z CertificateChainFile* option on page 166.

CERT-CHAIN-FILE=*STD
Default: *NONE

CERT-CHAIN-FILE=*NONE
No CA certificate chain file is specified.

CERT-CHAIN-FILE=<filename 1..54_without-generation-version>
Name of the CA certificate chain file.

CA-REVOCATION-FILE=
See the *-Z CARevocationFile* option on page 163.

CA-REVOCATION-FILE=*STD
Default: *NONE

CA-REVOCATION-FILE=*NONE
No CA revocation file is specified.

CA-REVOCATION-FILE=<filename 1..54_without-generation-version>
Name of the CA revocation file (CRL).

RANDOM-FILE=
See the *-Z RandFile* option on page 165.

RANDOM-FILE=*STD
Default: SYSDAT.TCP-IP-AP.*nnn*.TELNETD.RAND

RANDOM-FILE=<filename 1..54_without-generation-version>
Name of the file which contains the data for initializing the PRNG.

SSL-LIBRARY=
See the *-Z OpenSSLlibName* option on page 171.

SSL-LIBRARY=*STD
Default: LMS library to which the IMON logical ID SYSLNK refers.

SSL-LIBRARY=*NONE
No LMS library is specified.

SSL-LIBRARY=<filename 1..54_without-generation-version>
Name of the LMS library which contains the OpenSSL dynamically loadable module.

VERIFY-CLIENT=
See the *-Z VerifyClient* option on page 170.

VERIFY-CLIENT=*STD
Default: *NONE

VERIFY-CLIENT=*NONE
No certificate is requested from the TELNET client.

VERIFY-CLIENT=*OPTIONAL
A certificate is requested from the TELNET client. However, if no certificate or only an invalid one is returned, the TELNET client is still granted access.

VERIFY-CLIENT=*REQUIRE
A certificate is requested from the TELNET client. However, if no certificate or only an invalid one is returned, the TELNET client is denied access.

VERIFY-DEPTH=
See the *-Z VerifyDepth* option on page 169.

VERIFY-DEPTH=*STD
Default: 1

VERIFY-DEPTH=<integer 0..32767>
Number of certificates between the client certificate and the certificate which is known to the TELNET server (including the latter).

ENCRYPTION=*STD / *NO / *YES(...)
Defines whether the TELNET server implements encryption for the connection using the ENCRYPTION option. See also the *-H* option on page 173.

ENCRYPTION=*STD
Default: *NO.

ENCRYPTION=*NO
The TELNET server implements no encryption for connections by means of the ENCRYPTION option.

ENCRYPTION=*YES(...)
The TELNET server implements (on principle) encryption for connections by means of the ENCRYPTION option.

ENCRYPTION=*YES(...) may only be specified if TLS-SUPPORT=*PAR(...) is not also specified.

DEBUG=
Switch for the encryption trace.

DEBUG=*STD
Default: *NO

DEBUG=*NO
Encryption trace not enabled.

DEBUG=*YES
Encryption trace enabled.

KEY=<x-text 1..16>
Key to be used for DES 64 encryption.
The key must be specified without quotes and is interpreted as a hexadecimal string.
Example:
The key “TELNET” is E3C5D3D5C5E3 in hexadecimal notation and must be specified as follows:
... ENCRYPTION=*YES(KEY=E3C5D3D5C5E3)

SSL-LIBRARY=
See the *-Z OpenSSLlibName* option on page 171.

SSL-LIBRARY=*STD
Default: LMS library to which the IMON logical ID SYSLNK refers.

SSL-LIBRARY=*NONE
No LMS library is specified.

SSL-LIBRARY=<filename 1..54_without-generation-version>
Name of the LMS library which contains the OpenSSL dynamically loadable module.

OPTION-FILE=
File in which the options are stored.

OPTION-FILE=*STD
Default: SYSDAT.TCP-IP-AP.*nnn*.TELNETD.OPT

OPTION-FILE= <filename 1..54_without-generation-version>
Name of the option file.

SELECTOR=
Selector for TELNET exit routines. See also the *−E* option on page 151.

SELECTOR=*STD
Default: No exits

SELECTOR=<text 1..511>
Specifies the selector for the TELNET exit routines.

PORT-NUMBER=
Port number under which the TELNET server is started. See also the *-P* option on page 152.

PORT-NUMBER=*STD
Default: 23

PORT-NUMBER=<integer 0..32767>
Specifies the port number under which the TELNET server is started.

SERVER-ENTER-FILE=
Name of the start file for the TELNET server.

SERVER-ENTER-FILE=*STD
Default: SYSENT.TCP-IP-AP.*nnn*.TELNETD

SERVER-ENTER-FILE=<filename 1..54_without-generation-version>
Specifies the name of the start file for the TELNET server.

LOGGING-FILE=
Name of the logging file for the TELNET server.

LOGGING-FILE=*STD
Default: SYSOUT.TCP-IP-AP.*nnn*.TELNETD

LOGGING-FILE=<filename 1..54_without-generation-version>
Specifies the name of the logging file for the TELNET server.

5.3 Configuring TELNET using an option file

When the SET-FTP-TELNET-PARAMETERS command is executed (see page 40/page 138), an option file is generated in which the individual TELNET server parameters are stored as options. The default file name of the option file is:

SYSDAT.TCP-IP-AP.*nnn*.TELNETD.OPT

If a file of this name exists in the TELNET server's execution ID, it is interpreted by the TELNET server's start file (default file name: SYSENT.TCP-IP-AP.*nnn*.TELNETD) when the TELNET server is started and the TELNET server is configured accordingly.

You can thus make changes to the TELNET server parameters via the option file without needing to repeat installation using the SET-FTP-TELNET-PARAMETERS command.

If you want to use a file with a different name as the option file, specify the following option in the start file:

–M *option-file-name*

This file is then interpreted in place of any file named SYSDAT.TCP-IP-AP.*nnn*.TELNETD.OPT which may exist.

The following priorities apply for the options:

5. Options specified in the start procedure
6. Options which
   - are included in the option file specified via the -*M* option or,
   - if no -*M* option was specified, which are included in the default option file SYSDAT.TCP-IP-AP.*nnn*.TELNETD.OPT.

Option file and parameter line options

In the option file you can also specify the parameter line options supported in earlier versions. Mixed application of option file and options in the parameter line is also possible. If a particular option is specified both in the parameter line and in the option file, the option specified in the parameter line has priority.

Notation of the options in the option file

The various options must be entered in the option file according to the following rules:

- Any number of options can be contained in a line
- If an option's arguments extend over more than one line, each line that is to be continued must be terminated with the continuation character "\".
- A line beginning with the character "#" in column 1 is ignored when the file is read in.
- No distinction is made between upper and lower case in the option names.

5.3.1 Options for general configuration of the TELNET server

-A - Specify a name prefix

The *-A* option is used to specify a name prefix. This prefix is used to form a terminal name by appending a sequence number. The job started by the server for each client is assigned this prefix as the job name.

-A
<text 1..8>

<text 1..8>
Name prefix.
Default: TELSR

-D - Debug level

The *-D* option is used to specify a debug level which defines whether or which diagnostic information regarding the TELNET server run should be written to the logging file.

-D
<integer 0..2>

<integer 0..2>
Debug level.
The value 0 disables logging. The higher the specified value, the more information is placed in the logging file.
Default: 0

-E - Specify exit routines

The *-E* option enables and disables user-defined exit routines of the TELNET server. The structure of the selector routines is described in the section “Exit routines” on page 185.

-E
<selector-definition 1..511>

<selector-definition 1..511>
Selector definition

-N - Specify the number of connections

The *-N* option is used to specify the number of connections which the TELNET server is to operate.

-N
<integer 1..900>

<integer 1..900>
Number of connections which the server is to operate simultaneously.
Default: 15

-P - Specify port number

The *-P* option is used to specify the port number under which the server can be reached.

-P
<integer 1..1023>

<integer 1..1023>
Port number
Default: 23

-S - Define terminal name

The *-S* option is used to specify the number of digits from the name of the remote computer or name prefix (see option *-A*) of the TELNET server which are included in the terminal name.

-S
<integer 0..6>

<integer 0..6>
Number of digits.
Default: 0

-T - Specify socket trace level

The *-T* option is used to specify the socket trace level. The socket trace level defines whether or which diagnostic information of the TCP-IP (socket) system is to be written to the logging file.

-T
<integer 0..10>

<integer 0..10>
Socket trace level.
The value 0 disables logging. The higher the specified value, the more information is placed in the logging file.
Default: 0

-V - Enable/disable verbose

The *-V* option is the short form of `-D 1` and `-T 1`.

-V

-X - Select code tables

The *-X* option is used to select the valid EBCDIC and ISO code tables. The specifications for the *-X* option are valid only if no contradictory selections were made in the file SYSDAT.TCP-IP-AP.*nnn*.CLIENTS. You must bear in mind here that the TELNET server only supports 7-bit terminals, thus rendering specification of an 8-bit ISO table meaningless.

-X
<ebcdic-code-table>:<iso-code-table>

<ebcdic-code-table>:<iso-code-table>
Valid EBCDIC code table and valid ISO code table.
Default: EDF041: ISO88591

5.3.2 Options for safe use of TELNET with the aid of authentication and encryption

There are three methods of guaranteeing secure operation of TELNET by means of authentication and encryption:

- START-TLS option

  The START-TLS option was implemented exclusively for TLS/SSL. In BS2000 it is supported by the server option *-Z tls-required*.

- “Telnet Authentication Option” (RFC 2941) for negotiating an authentication method

  In BS2000 only TLS/SSL is currently supported. The “Telnet Authentication Option” is selected using the option *-B*. The “Telnet Authentication Option” will possibly gain in importance in the future because it permits a very wide variety of authentication methods to be supported, including Kerberos. In the following, the “Telnet Authentication Option” will be referred to as AUTHENTICATION option.

- “Telnet Data Encryption Option” (RFC 2946) for negotiating a symmetric encryption method and the associated key

- In BS2000 only DES 64 (RFC 2952, RFC 2953) is currently supported. The “Telnet Data Encryption Option” is selected using the server option *-H*. In the following, the “Telnet Data Encryption Option” will be referred to as ENCRYPTION option.

START-TLS option (see page 156), AUTHENTICATION option (see page 172) and ENCRYPTION option (see page 173) are described in detail in the following sections.

In the case of the options below, the equals sign must immediately follow the option name without a space and the option value must immediately follow the equals sign, also without a space.

5.3.3 -Z option - Support of the START-TLS option

This option enables you to control TLS support on the TELNET client. Negotiation of the arrangements for authentication is handled by SSL in this case to relieve the load on TELNET.

You enter the options for using TLS support as follows:

```
-Z <option>
```

Time when the options or changes to the options should become effective

The *-Z OpenSSLlibName* option is only evaluated once during a TELNET session, namely when the OpenSSL library is loaded. All other options become effective after the connection to the TELNET client has been set up.

Description of the -Z options

The -Z options are described below.

The following must be observed here:

- With the exception of the *-Z tls-required* option, all the -Z options can also be used for the AUTHENTICATION option (see page 172) as this operates with TLS/SSL support.
- The *-Z OpenSSLlibName* option is also relevant for the ENCRYPTION option relevant, as only encryption routines from the OpenSSL library are used.
- Simultaneous specification of the ENCRYPTION option (*-H*, see page 173) and either the START-TLS option (*-tls-required*, see page 157) or the AUTHENTICATION option (*-B*, see page 172) results in an error when the TELNET server is started.
- Simultaneous specification of the START-TLS option (*-Z tls-required*) and the AUTHENTICATION option (*-B*) results in an error when the TELNET server is started.

-Z tls-required

The *-Z tls-required* option is used to enable and disable TLS security via the START-TLS option on the TELNET client.

-Z tls-required
[={ **yes** | **no** | optional}]

yes

START-TLS support is enabled. If no TLS security for the connection can be established, the connection is terminated.

optional

START-TLS support is enabled. If no TLS security for the connection can be established, the connection is used without this security.

no

START-TLS support is disabled.

-Z tls-required specified without operands

-Z tls-required = yes applies (START-TLS support is enabled).

-Z tls-required not specified

START-TLS support is not enabled (default).

-Z RSACertificateFile

The *-Z RSACertificateFile* option is used to specify a file which contains the RSA-based X.509 server certificate in PEM format. This file can also contain the private RSA server key. However, generally the certificate and key are stored in different files. In this case the key file is specified using the *-Z RSAKeyFile* option (see page 159).

-Z RSACertificateFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file which contains the RSA-based X.509 server certificate in PEM format.

***NONE**
No file is specified.
*NONE is the default.

-Z RSAKeyFile

The *-Z RSAKeyFile* option is used to specify a file which contains the private RSA server key in PEM format.

If both an X.509 client certificate and a private server key are contained in the same file (see the *-Z RSACertificateFile* option on page 158), the *-Z RSAKeyFile* option need not be specified.

As it should be possible to start up the TELNET server automatically in unattended operation, no passphrase may be entered for the private server key at server startup. You must therefore remove any existing encryption of the private key with a passphrase. In this event, ensure that unauthorized persons cannot access this key.

-Z RSAKeyFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file which contains the private RSA server key.

***NONE**
No separate file is used for the RSA server key.
*NONE is the default.

-Z DSACertificateFile

The *-Z DSACertificateFile* option is used to specify a file which contains the DSA-based X.509 server certificate in PEM format. This file can also contain the private DSA server key. However, generally the certificate and key are stored in different files. In this case the key file is specified using the *-Z DSAKeyFile* option (see page 161).

-Z DSACertificateFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file which contains the X.509 client certificate in PEM format.

***NONE**
No file is specified.
*NONE is the default.

-Z DSAKeyFile

The *-Z DSAKeyFile* option is used to specify a file which contains the private DSA server key in PEM format.

If both an X.509 client certificate and a private server key are contained in the same file (see the *-Z DSACertificateFile* option on page 160), the *-Z DSAKeyFile* option need not be specified.

As it should be possible to start up the TELNET server automatically in unattended operation, no passphrase may be entered for the private server key at server startup. You must therefore remove any existing encryption of the private key with a passphrase. In this event, ensure that unauthorized persons cannot access this key.

-Z DSAKeyFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file which contains the private DSA server key.

***NONE**
No separate file is used for the DSA server key.
*NONE is the default.

-Z CACertificateFile

The *-Z CACertificateFile* option is used to specify a file containing the CA certificates in PEM format which are required for TELNET server authentication. The individual PEM certificates are arranged sequentially in the file.

You can process the file with a text editor of your choice when you wish to add or delete certificates. The individual certificates are registered in the file as follows:

```
-----BEGIN CERTIFICATE-----
< CA certificate in Base64 encoding >
-----END CERTIFICATE-----
```

Text outside these sequences is ignored by the TELNET client and can therefore be used to identify the certificates which, owing to the Base64 encoding, are available in non-readable form.

-Z CACertificateFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file containing the certificates in PEM format which are required for TELNET server authentication.

***NONE**
No file with CA certificates is specified.
*NONE is the default.

-Z CARevocationFile

The *-Z CARevocationFile* option is used to specify a file which contains the CRLs (Certificate Revocation Lists) of the Certificate Authorities (CAs). (Certificates issued by a Certificate Authority can be declared invalid by publication of a Certificate Revocation List (CRL).)

-Z CARevocationFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file which contains the CRLs of the Certificate Authorities.

***NONE**
No file with CRLs is specified.
*NONE is the default.

-Z CipherSuite

The *-Z CipherSuite* option is used to specify a cipher suite preference list. If this option is not specified, a default preference list is used.

-Z CipherSuite
=<specification>

<specification>
Specification of a a cipher suite preference list. For details, see chapter “Specification of a cipher suite preference list” on page 375.
ALL: !EXP: !ADH:!RC4 is the default.

-Z RandFile

The *-Z RandFile* option is used to specify a file from which data is read for initializing the pseudo random numbers generator (PRNG) when the server is started. When the server is shut down, the relevant data from the PRNG is read to this file so that it can be used the next time the server is started.

If several TELNET servers are operated in parallel, a separate file must be defined for each server.

CAUTION!

This file may not be accessible to unauthorized people.

-Z RandFile
=<file-name 1..54>

<file-name 1..54>
Name of the file which contains the data for initializing the PRNG.
Default: SYSDAT.TCP-IP-AP.*nnn*.TELNETD.RAND

-Z CertificateChainFile

The *-Z CertificateChainFile* option is used to specify a file in which all certificates can be stored which are required for verification of the server certificate. The first certificate in this file is the server certificate. The remaining certificates must form an unbroken chain, starting with the certificate of the CA which issued the server certificate, through to the root certificate of a CA which can be verified directly by the FTP client. The certificates in the chain must be sorted in such a way that the root certificate is in last place.

The specified file is only required if the server certificate was issued by a CA that is not known to the TELNET clients and verification can thus not be performed by the TELNET clients without the certificate chain being sent. This mechanism requires that RSA and DSA certificates should not be used simultaneously for the server, as the file is used for both variants.

-Z CertificateChainFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file in which all certificates are stored which are required for verification of the server certificate.

***NONE**
No file is specified.
*NONE is the default.

-Z Protocol

OpenSSL supports Versions 3 of the SSL protocol and also Versions 1, 1.1 and 1.2 of the TLS protocol. Some of these protocols can be activated selectively using the *-tlsProtocol* option.

-Z Protocol
[**+** | **-**] {**SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **All** } ...

+
The protocol specified after this sign is permissible.

If neither “+” nor “-” is specified, this has the same effect as specifying “+”.

-
The protocol specified after this sign is not permissible.

SSLv3
SSL protocol Version 3

Version 3 of the SSL protocol displays some security-related deficiencies and should therefore not be used if possible.

TLSv1
TLS protocol Version 1

TLSv1.1
TLS protocol Version 1.1

TLSv1.2
TLS protocol Version 1.2

ALL
All protocols are to be enabled.

All -SSLv3 is the default.

Example

The specifications `-Z Protocol=TLSv1 TLSv1.1 TLSv1.2` and `-Z Protocol=All -SSLv3` have the same effect as long as no support of the future TLS version 1.3 is added to the TELNET.

-Z AcceptableClientCAFile

When client authentication is enabled, the server notifies the clients upon TLS connection setup of the names of the CAs which it accepts as signatories of client certificates. These name are taken from the certificates in the file specified by the-*Z AcceptableClientCAFile* option. The individual certificates in PEM format are arranged sequentially in this file.

You can process the file with a text editor of your choice when you wish to add or delete certificates. The individual certificates are registered in the file as follows:

```
-----BEGIN CERTIFICATE-----
< CA certificate in Base64 encoding >
-----END CERTIFICATE-----
```

Text outside these sequences is ignored by the TELNET server and can therefore be used to identify the certificates which, owing to the ASN.1/Base64 encoding, are available in non-readable form.

-Z AcceptableClientCAFile
={<file-name 1..54> | ***NONE**}

<file-name 1..54>
Name of the file.

***NONE**
No file is specified.
*NONE is the default.

-Z VerifyDepth

The *-Z VerifyDepth* option is used to define the verification depth, in other words the maximum permissible number of certificates between the TELNET client certificate and the certificate which is known to the TELNET server.

Here you must note the following:

- If the value 1 (default) is specified as the maximum depth, the server certificate must have been signed directly by a CA (Certificate Authority) that the TELNET client knows for it to be accepted.
- If the maximum depth is exceeded, the connection is cleared, unless verification of the TELNET server certificate has been disabled with `-Z VerifyClient=NONE` (see page 170) or `-Z VerifyClient=OPTIONAL`.
- Specifying the depth as 0 is meaningless. In this case only self-signed certificates would be permissible.

-Z VerifyDepth
=<integer 0..32767>

<integer 0..32767>
Maximum permissible number of certificates between the TELNET client certificate and the certificate which is known to the TELNET server.
Default: 1

-Z VerifyClient

The *-tlsVerifyClient* option is used to define whether a TELNET client requires a certificate to access a server.

-Z VerifyClient
={**NONE** | **OPTIONAL** | **REQUIRE**}

NONE
The TELNET server does not request a certificate from the TELNET client. NONE is the default.

OPTIONAL
The TELNET server requests the TELNET client to send its certificate. If the client refuses to do this or supplies an invalid certificate, access is nevertheless allowed.

REQUIRE
The TELNET client must transfer a valid certificate, otherwise access is refused.

-Z OpenSSLlibName

The *-Z OpenSSLlibName* option is used to define the LMS file from which the OpenSSL library should be dynamically loaded. The OpenSSL library is only dynamically loaded if at least one of the options `-Z tls-required`, `-B on` or `-H on` is specified.

It may be necessary to specify a name other than the default name if, for example, the OpenSSL library is also used by other products.

Dynamic loading of the OpenSSL library can be expedited with the aid of DAB using caches. If the OpenSSL library is used jointly by a number of products, the size of the DAB buffer used is reduced.

-Z OpenSSLlibName
=<openssl-libname>

<openssl-libname>
Name of the LMS file from which the OpenSSL library is to be dynamically loaded.
Default: LMS file to which the IMON logical ID SYSLNK refers.

5.3.4 -B option - Enable/disable the AUTHENTICATION option

The *-B* option allows you to enable or disable support of the AUTHENTICATION option with which an authentication method can be negotiated. In BS2000 the AUTHENTICATION option is currently only implemented for TLS/SSL.

You can define the settings in the TELNET server which are required for SSL operation using the *-Z* options described for the START-TLS option (see the section “-Z option - Support of the START-TLS option” on page 156).

The options *-Z tls required* (START-TLS option, see page 157) and *-B* (AUTHENTICATION option) may not be specified simultaneously. If they are, an error message is issued:

```
Both START-TLS and AUTHENTICATION-Option not allowed
```

-B
{**on** | **off** | **debug**}

on
The AUTHENTICATION option is supported.

off
The AUTHENTICATION option is not supported.

debug
The authentication trace is enabled.

5.3.5 -H option - Enable/disable the ENCRYPTION option

With the *-H* option you can enable and disable support of the ENCRYPTION option, which is used to negotiate the encryption method and the key used. Currently only variants DES_CFB64 and DES_OFB64 of DES64 are supported in TELNET. The *-H* option may not be specified at the same time as the option `-Z tls-required` (see page 157) or `-B on` (see page 172). If it is, the following error message is issued:

```
error: SSL and encryption opton
```

Only encryption routines from the OpenSSL library are used. If an OpenSSL library with a name different from the default file name (SYSLNK.TCP-IP-AP.*nnn*) is to be used, you can specify this using the *-Z OpenSSLlibName* option (see page 171).

-H
{**on** | **off** | **debug** | **key=**{<c-string 1..8> | <x-string 1..16>}

on
The ENCRYPTION option is supported.

off
The ENCRYPTION option is not supported.

debug
The encryption trace is enabled.

key={<c-string 1..8> | <x-string 1..16>}
Encryption key for DES

– Note that no distinction is made between the key for encryption and the key for decryption. The TELNET client and TELNET server use the same key. In the case of the TELNET server this means that this key applies for all TELNET clients.

– When `-H on` is specified, the specification `-H key=...` is always required, otherwise the following error message is issued:

```
Error: Encryption on and no Encryption Key!
```

5.4 Starting up and shutting down the TELNET server

You can use the commands described below to start up the TELNET server.

These commands can also be entered at the operator console.

5.4.1 Starting up the TELNET server

The */START* commands for the Enter jobs are as follows:

/START-TCP-IP-DEMON	Enter Job for TCP-IP-AP
/START-TELNET-DEMON	Enter Job for the TELNET server

The /START commands are only permissible under IDs that have the NET-ADMIN privilege.

If you want to start FTP and TELNET simultaneously, use the START-TCP-IP-DEMON command.

5.4.2 Shutting down the TELNET server

The commands for shutting down the TELNET server which are described below are only valid for servers as of interNet Services V3.0A.

5.4.2.1 Shutting down the TELNET server via STOP-TELNET-DEMON

You shut down the TELNET server with the STOP-TELNET-DEMON command.

STOP-TELNET-DEMON
PORT-NUMBER=*STD-PORT/*ANY/<integer 0..32767>

PORT-NUMBER=
Specifies the port number of the TELNET server to be shut down.
Default: The TELNET server with default port number 23 is shut down.

PORT-NUMBER=*STD-PORT
Has the same effect as specifying no parameters.

PORT-NUMBER=*ANY
All active TELNET servers are shut down.

PORT-NUMBER=<integer 0..32767>
A TELNET server with the specified port number is to be shut down.

5.4.2.2 Shutting down the TELNET server via Shutdown

Servers can still be shut down using the command
*/INFORM-PROGRAM 'shutdown',*TSN(<tsn>*. <*tsn*> is the TSN of the relevant server task.

5.4.3 Notes and restrictions for starting up and shutting down servers

The following points must be borne in mind when you start up or shut down servers:

- The START-TELNET-DEMON command is only effective if start procedure SYSENT.TCP-IP-AP.*nnn*.TELNETD exists.

  If the command is entered at the console it must also be shareable. This has no negative effect on security since from TCP-IP-AP V5.0 on all server options can be stored in a separate option file which does not need to be shareable (see the section “Configuring TELNET using an option file” on page 148).

- The VERSION parameter in the START commands also permits servers with TCP-IP-AP Versions < 5.0 to be started. However, as the servers only log on to the TCPIPAP subsystem after every start as of V5.0, only such servers can be shut down again using the STOP command.

- A maximum of 20 users can be connected to the TCPIPAP subsystem. No more users are permitted owing to the size of internal tables. In practice the maximum number of 20 is, however, sufficient. If the maximum number is exceeded, the server shuts down on startup and issues the following message:

  ```
  "error: too many connections to Subsystem TCPIPAP"
  ```

- If the server was not started under TSOS it shuts down and issues the following message:

  ```
  "error: no privilege to connect to Subsystem TCPIPAP"
  ```

  This can only occur if you attempt to start the server by calling the start procedure explicitly instead of using the START-...-DEMON command, because this would be rejected under an ID other than TSOS.

5.4.4 Messages and return codes

Messages

```
TCP2000    (&00)-COMMAND FOR (&01)-SERVER SUCCESSFULLY COMPLETED.
           &00  =  START or STOP
           &01  =  FTP or TELNET or TCP-IP
```

Meaning
FTP/TELNET-server successfully started/stopped.

Response
<None>

```
TCP2001    STOP-COMMAND FOR (&00)-SERVER HAS NO EFFECT.
           &00  =  FTP or TELNET
```

Meaning
No servers existing.

Response
<None>

```
TCP2003    NO (&00)-SERVER FOR THE GIVEN PORTNUMBER.
           &00  =  FTP or TELNET
```

Meaning
No (&00)-server for the given port number.

Response
<None>

```
TCP2004    WAS NOT ABLE TO START PROCEDURE FOR (&00): (&01).
           &00  =  FTP or TELNET or TCP-IP
           &01  =  <start procedure>
```

Meaning
Could not start procedure (&01) for (&00)-server.

Response
<None>

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	See message TCP2000
	0	CMD0001	See message TCP2001
	1	TCP2003	See message TCP2003
	32	TCP2004	See message TCP2004
	32	CMD0220	An error occurred in the /CANCEL-JOB command

5.5 Logging file of TELNET servers

TELNET servers log their outputs in a logging file with the default file name SYSOUT.TCP-IP-AP.*nnn*.TELNETD. The logging file always contains the difference entries for the current backup with the */INFORM-PROGRAM* command *rdProt* (see the section “rdProt- Save logging file of TELNET servers” on page 183).

5.6 Displaying the current settings of TELNET servers

You can use the BS2000 command SHOW-FTP-TELNET-STATUS to obtain information on the current settings of BS2000 TELNET servers.

The following information is output:

- Settings made when the servers were generated
- Current information on the server task’s TSN and number of active connections

The servers place the data in auxiliary files with the following names:

- SYSDAT.TCP-IP-AP.*nnn*.TELNETD.CONF.<port>

  <port> specifies the port number of the relevant server.

These files are deleted when the server is shut down.

SHOW-FTP-TELNET-STATUS
SERVER= *FTP(...)/ ***TELNET**(...) ***FTP**(...) **PORT-NUMBER= *STD-PORT / ****ANY** / <integer 0..32767> ***TELNET**(...) **PORT-NUMBER=*STD-PORT / ****ANY** / <integer 0..32767> **, INFORMATION= *STD /**ALL** **, OUTPUT=*SYSOUT/**SYSLST**

SERVER=
Name of the server whose data is to be output (FTP or TELNET).

SERVER=*TELNET(...)
Output of a TELNET server’s configuration data.

PORT-NUMBER=
Port number of the TELNET server whose configuration data is to be output.

PORT-NUMBER=*STD-PORT
Port number 23. This is the default.

PORT-NUMBER=*ANY
Information on all TELNET servers currently active is output.

PORT-NUMBER=< integer 0..32767>
Port number of the TELNET server whose configuration data is to be output.

INFORMATION=
Type and scope of the information output.

INFORMATION= *STD
Output of a list of servers specified by PORT-NUMBER. This is the default.

INFORMATION= *ALL
Output of all information on all servers specified by PORT-NUMBER.

OUTPUT=
Output medium to which the information is to be written.

OUTPUT=*SYSOUT
Output is to SYSOUT. This is the default.

OUTPUT=*SYSLST
Output is to SYSLST.

Messages

```
TCP9240   COMMAND SHOW-FTP-TELNET-STATUS SUCCESSFULLY COMPLETED.
```

Meaning
SHOW-FTP-TELNET-STATUS successfully completed.

Response
<None>

```
TCP9241   ERROR DMS(&00) DURING EXECUTION OF FSTAT.
```

Meaning
Error during execution of FSTAT on configuration files.

Response
Check whether any servers are active.

```
TCP9242   COULD NOT OPEN CONFIGURATION FILE (&00).
```

Meaning
Desired server not active or configuration file (&00) locked.

Response
Check whether the desired server is active.

TCP9243 COULD NOT READ CONFIGURATION FILE (&00).

Meaning
Configuration file (&00) could not be read.

Response
Check file.

Command return codes

(SC2)	SC1	Maincode	Meaning
	0	TCP9240	Command successfully completed
	64	TCP9241	See message TCP9241
	64	TCP9242	See message TCP9242
	64	TCP9243	See message TCP9243

5.7 Console interface

Some commands can also be specified by the system operator via the console interface. These commands are introduced with the command */INFORM-PROGRAM*. Commands issued with /*INFORM-PROGRAM*. are used to

– control server traces,
– exit the various server tasks,
– save the logging files.

The following server commands are possible via the console interface:

Operation	**Brief description**
debug	Enable / disable trace to user level
trace	Enable / disable trace to TCP/IP interface
shutdown	Shut down service
rdProt	Save the TELNET server's logging file

The commands for starting up and shutting down the TELNET servers (see page 174) can also be entered via the console.

debug - Enable / disable trace to user level

/INFORM-PROGRAM
'debug <debug-value>**',*TSN**(<tsn>)

<tsn>
TSN of the server task for which the trace to the user level is to be activated.

debug <debug-value>
The permitted values are from 0 to 9. Higher values result in the output of more information. A value of 0 means that the trace is disabled. This command is used by the system customer service for error diagnostics. A debug value > 2 is not meaningful.

trace - Enable / disable trace to TCP/IP interface

/INFORM-PROGRAM
'trace <trace-value>**',*TSN**(<tsn>)

<tsn>
TSN of the server task for which the trace to the TCP/IP interface is to be activated.

trace <trace-value>
The permitted values are from 0 to 10. Higher values result in the output of more information. A value of 0 means that the trace is disabled. This command is used by the system customer service for error diagnostics.

shutdown - Shut down service

/INFORM-PROGRAM
'shutdown',***TSN**(<tsn>)

<tsn>
TSN of the server task to be shut down.

rdProt- Save logging file of TELNET servers

The *rdProt* command is used to save the logging file of a TELNET server under the name of the original logging file (default file name: SYSOUT.TCP-IP-AP.*nnn*.TELNETD), extended by a suffix specifying the date and time in the format <MMDD><HHMMSS>. This file always contains the difference entries for the previous backup. Make sure that you do not select a name that is too long for the logging file: *rdProt* fails when the name exceeds the permissible length after the suffix is appended.

/INFORM-PROGRAM
'rdProt ',*TSN(<tsn>**)**

<tsn>
TSN of the server task.

5.8 IPv6 addresses in TELNET

When setting up the control connection using the client command *open* you can also use an IPv6 address which you must specify in hexadecimal notation with a colon (:) (see the manual “interNet Services User Guide”).

5.9 TELNET exits

The following exits exist for the TELNET client and the TELNET server:

- Exit before starting the interactive application (Open Exit)
- Exit before terminating the interactive application (Close Exit)
- Exit before sending data
- Exit after receiving data

The exits for starting and terminating the interactive application can only be set on the TELNET server.

5.9.1 DUMMY module

TELNET exits are handled in the reloaded EXITTEL.C module. EXITTEL.C is provided as a support and has just one entry point in *YAPTEXIT*. The only EXITTEL.C action is the return with the return value 0xff. This indicates that no user-defined exit routines have been defined as yet.

In order to define your own exits, create a module containing the entry point YAPTEXIT and replace the EXITTEL module in the SYSLNK.TCP-IP-AP.*nnn* library (SRMLNK.TCP-IP-AP.*nnn*) with the module you have created.

5.9.2 Exit routines

For handling the various exits, the callers of the exit routine (TELNET client or TELNET server) pass the address of a parameter list to the entry point YAPTEXIT. The *yaptx* structure from this parameter list is supplied in SYSLIB.TCP-IP-AP.*nnn* under the name *yaptx.h*.

Structure of *yaptx*

The *yaptx* structure is defined as follows:

```
struct yaptx {
        int     caller;         /* Aufrufer: S=Server, C=Client           */
#define client  0x01            /* X'01' Client                           */
#define server  0x02            /* X'02' Server                           */
        int     action          /* main action code                       */
#define telrecv 0x01            /* action = receive                       */
#define telsend 0x02            /* action = send                          */
#define telopen 0x03            /* action = open connection               */
#define telclos 0x04            /* action = close connection              */
        char   *selector;       /* Selector                               */
        int     portno;         /* Port-Nummer (Server-Port)              */
        char    hname[128+1];   /* Hostname von Server / Client           */
        char    reserved[3];    /* Reserved                               */
        int     connid;         /* identifies uniquely client-server      */
                                /* connection. Only for Server            */
        char   *inbuf;          /* Address of input buffer                */
        int     inbuflen;       /* Length of input buffer                 */
        char   *outbuf;         /* Address of output buffer               */
        int     outbuflen;      /* Length of output buffer                */
        int     outDatalen;     /* Length of output data                  */
```

Description of parameters

The following parameters are transferred:

- Caller of exit routine (*caller*)
- Type of entry (*action*)
- Printable character string for identifying the exit routine (*selector*)
- Connection identifier (*connid*)
- Host name (*hname*)
  - of the addressed TELNET server (for TELNET clients)
  - of the calling TELNET server (for TELNET servers)

- Port number (*portno*)
  - of the addressed TELNET server (for TELNET clients)
  - of the addressed TELNET client (for TELNET servers)
- For send/receive exits: *inbuf*, *inbuflen*, *outbuf*, *outbufLen*, *outDataLen*
  outDataLen must be supplied by the exit itself.

caller
Specifies whether the exit routine was called by the TELNET client or TELNET server.

action
Specifies the type of entry (*receive*, *send*, *open connection*, *close connection*)

selector
Printable character string for selecting the required exit routine. The character string is terminated with \0.

portno
Port number of the server or client:

- On TELNET clients, *portno* specifies the port number of the addressed TELNET server.
- On TELNET servers, *portno* specifies the port number of the addressed TELNET client.

hname[128+1]
Host name of server or client:

- On TELNET clients, *hname* specifies the host name of the addressed TELNET server.
- On TELNET servers, *hname* specifies the host name of the addressed TELNET client.

inBuf
Contains the address of the buffer containing the data to be converted (input buffer). Only relevant for send/receive exits.

inBufLen
Specifies the length (number of valid bytes) of the input buffer. Only relevant for send/receive exits.

outBuf
Contains the address of the output buffer. Only relevant for send/receive exits.

outBuflen
Specifies the length of the output buffer. Only relevant for send/receive exits.

outDataLen

Before returning to the caller, the routine must specify in *outDataLen* the number of valid bytes in the output buffer that must therefore be sent. Only relevant for send/receive exits.

connid

Connection identifier. The Connection Identifier uniquely identifies the client/server connection in the case of a fixed TELNET server. If several servers are active, the pair (*portno*, *connid*) uniquely identifies the connection between client and server and thus the TELNET client (see also the *connid* parameter). This is important, for example, for code conversions if two bytes are mapped to one byte. The exit routine must buffer the first byte in this case for the conversion to be able to perform the conversion when the second byte is received. *connid* ensures the unique assignment of the buffer to the respective connection.

Results and return values of exit routines

Exit routines must always supply a return value. Every exit for which no user-defined exit routine exists, must return the value 0xff as a return value.

Results and return values for *send* and *receive*

Following orderly execution, *outbuf* contains the converted code and *outDataLen* the length of the converted code. Return value: 0.

If an exit determines that there is not enough memory for a code conversion, the return value -2 must be returned.

The only query with *inbuf* = 0 is whether or not a relevant exit was defined:

- If a relevant exit is defined, the return value must be 0.
- If no relevant exit is defined, the return value must be 0xff.

Results and return values for *open* and *close*

open

If the exit permits access to the server, the return value is 0, otherwise it must be -1.

close

close always returns the return value 0.

5.9.3 User-defined exits

For exit handling with your own procedures, you have to add the relevant code to the EXITTEL.C module.

If you define an exit for *send* or *receive*, you are also responsible for handling all special characters. In particular, you must ensure that

- a 0xff character is duplicated for sending,
- two successive 0xff characters are converted to a single 0xff for sending.

Activating user-defined server exits on the TELNET server

You can activate the code created in this way on the TELNET server with the -e option:

```
-e
[open:<selector1>!][close:<selector2>!][receive:<selector3>!][send:<selector4
>]
```

If you specify * for <selector>, the exit mechanism is started without the more precise specification of <selector>.

Activating user-defined server exits on the TELNET client

You can activate server exits for *send* and *receive* on the TELNET client with the *rexit* (remote exit) command. This restricts the effect of *rexit*, however, to the connection of this special client to the server.

The *rexit* command has the following syntax:

```
rexit [receive:<selector1>][<send:<selector2>]
```

The following cases can be distinguished:

- Server options are specified:

  If no *rexit* command was specified for a connection between the server and client, the setting of the *-e* server option applies.

  If there are entries for an exit routine both in the *rexit* command and in the *-e* server options, the specifications of the *rexit* command apply. Subsequent specification of * for <selector1> or <selector2> in the *rexit* command causes a reset to the entry in the server option. A blank entry for <selector1> or <selector2> disables the exit.

- No server options are specified:

  The specification of * for <selector> in the *rexit* command activates the exit routine. More precise differentiation of the handling type as enabled by the specification of <selector>, is not provided for in this case. A blank entry for <selector1> or <selector2> disables the exit.

Activating user-defined client exits on the TELNET client

User-defined exit routines for the client can be set on the client with the *exit* command:

```
exit [receive:<selector1>][<send:<selector2>]
```

If * is specified for <selector1> or <selector2>, the YAPTEXIT entry is selected but without the more precise specification of *selector*. If no entry is made for <selector>, no exit routine will be selected. A blank entry for <selector1> or <selector2> disables the exit.

Examples

1. Clients on the host *host* should not be given access to the TELNET server.
   This can be specified with the following user-defined module with the YAPTEXIT entry:

```
YAPTEXIT (struct yaptx *exparam)
{
   switch (exparam->action) {
      case telopen:
         if (strcmp(exparam->hname,"host") == 0)
            return(-1);
         break;
      default;
         break;
   }
}
```

   This module must replace the EXITTEL module in the SYSLNK.TCP-IP-AP.*nnn* library and be activated with the `-e open:*` option. The external reference YAPTEXIT is resolved from the user-defined module at runtime.

2. Two different user-defined open exits are defined for a TELNET server:

   – If a client logs on from *host1*, *exit1* should be selected.
   – If a client logs on from *host2*, *exit2* should be selected.

```
YAPTEXIT (struct yaptx *exparam)
{
   if ((exparam->action) == telopen)
      if (strcmp(exparam.hname,"host1")==0)
         exit1(exparam)
      else
         if ((strcmp(exparam.hname,"host2")==0)
            exit2(exparam);
}
```

   The relevant option is: `-e open;*`

3. Two different code conversions should be defined for the data that are sent from the client to the server. For example, x‘0102’ should be converted therefore to x‘0a’ or x‘0a0b’. In the example below, the code conversion routines are called *proc1* or *proc2*. The recognition strings are "proc1" or "proc2".

```
YAPTEXIT (struct yaptx *exparam)
{
   if ((exparam->caller) == client) && (exparam->action == telsend))
      if (strcmp(exparam->selector,"proc1")==0)
         proc1(exparam);
      else
         if ((strcmp(exparam->selector,"proc2")==0)
            proc2(exparam);
}
```

   The relevant commands for enabling this exit are:

   `exit send:proc1` or `exit send:proc2`

User-defined TELNET exits under POSIX

The POSIX TELNET is created by linking the LLMs TELPOSIX and EXITTEL. You can thus also work with the TELNET exits under POSIX.

The interNet Services package contains the dummy module EXITTEL, which only executes a return.

You write your own TELNET exits as follows:

- Create the version of the EXITTEL you require.
- Link this version to TELPOSIX from SYSLNK.TCP-IP-AP.*nnn*
- Write the resultant LLM to SINLIB.TCP-IP-AP.*nnn* under the name TELNET.
- Then install POSIX-TELNET (see the section “Installing and uninstalling FTP and TELNET clients in POSIX” on page 45).

6 Generating random numbers

This chapter describes:

- Generating random numbers in BS2000 with PRNGD
- Generating random numbers in POSIX

6.1 Generating random numbers in BS2000 with PRNGD

Almost all cryptographic methods require strong random numbers. If attackers can predict the numbers the random number generator will provide sufficiently well, they can find the keys that are used relatively easily.

To avoid every cryptographic application from having to implement procedures for producing random numbers, BS2000 provides a central random number generator. The BS2000 random number generator PRNGD (**P**seudo **R**andom **N**umber **G**enerator **D**aemon) is implemented as a pseudo random number generator. This prevents blocking and thus offers no openings for “Denial of Services (DoS)” attacks.

To improve access to data of the operating system kernel, the BS2000 PRNGD is implemented as a TPR subsystem. This is a major advantage in comparison with application-specific random number generators that do not run in TPR. In addition, this also ensures that the PRNGD is better protected against access by potential attackers. The user interface is implemented via an SVC.

The entropy sources, configuration and user interface of the BS2000 PRNGD are described below.

6.1.1 Entropy sources of the BS2000 PRNGD

The entropy sources used by the BS2000 PRNGD include:

- access times to specific, frequently used files
- outputs of commands relating to data that changes dynamically, such as SHOW-USER-STATUS with various parameters, or outputs of the SHOW-DAB-CACHING command if DAB is used on the system in automatic operation

You can define the files and commands to be used in a configuration file. To do this you specify for the commands what relative entropy content a particular output has so as to place more emphasis on commands with little explanatory (in other words unchanging) text and outputs that vary greatly. This permits adaptations to local conditions. In contrast, access times to files play no part in entropy accounting.

You can also specify the intervals at which the time stamps for the files are to be queried and the commands are to be executed. Outside the initialization phase, only one of the files is queried and one of the commands executed.

One further important entropy source is network traffic. Here the BS2000 PRNGD accesses the network traffic data collected by BCAM (number of bytes and packets etc. sent and received on a connection). However, configuration for this is neither necessary nor possible.

6.1.2 Configuration of the BS2000 PRNGD

You configure the BS2000 PRNGD with the aid of the options in a configuration file (the default name of which is SYSSSI.PRNGD.*nnn* and can be changed using the IMON logical ID SYSSSI). This is interpreted when the subsystem is started. The various options must be entered in the option file according to the following rules:

- Each option must be in a separate line
- If an option's arguments extend over more than one line, each line that is to be continued must be terminated with the continuation character "\".
- A line beginning with the character "#" in column 1 is ignored when the file is read in.
- The option names are not case-sensitive.

The various options are described below.

poolSize

The *poolSize* option is used to specify the size (in bytes) of the entropy pool to be used.

poolSize
<size>

<size>
Size of the entropy pool to be used (in bytes).
Default: 4096

Generally the default value is sufficient. Only in the event of (temporary) high requirements for random numbers can it make sense from the security viewpoint to use higher numbers. If numbers lower than 1024 are specified, 1024 is used.

minimalEntropy

The *minimalEntropy* option is used to specify how high (in bytes) the entropy content of the entropy pool must be before access to the random number generator is permitted.

minimalEntropy
<value>

<value>
Value for the entropy content of the entropy pool (in bytes) as of which access is permitted to the random number generator.
Default: 256

Once this value has been exceeded random numbers are always supplied, even when this minimum value is not reached. Generally the default value is sufficient. If values below 256 are specified, 256 is used.

entropyThreshold

The *entropyThreshold* option is used to specify a threshold (in bytes) for filling the entropy pool: If the entropy content of the pool falls below the threshold value, entropy is collected at short intervals by collecting command output and the BCAM-internal data until the threshold value is once again exceeded. During this time the *cmdInterval* and *bcamInterval* options are ignored.

entropyThreshold
<value>

<value>
Threshold value for filling the entropy pool (in bytes).
Default: 1024

seedFile

The *seedFile* option specifies the file from which the initialization values for the entropy pool of the PRNGD are read when the subsystem is started.

After is has been successfully read in the entropy pool is mixed. Random numbers taken from the pool are read back in. This ensures that the same initial values are not used at the next startup if the subsystem could not be terminated correctly. If the subsystem is terminated correctly, data is also written back into the file beforehand.

In all cases the entropy pool is not written directly into the file. Instead, as many random bytes are taken from the PRNGD and written to the file as the size of the entropy pool requires. This limits damage if unauthorized people access this file.

seedFile
<file-name>

<file-name>
Name of the file from which the initialization values are read.
Default: SYSDAT.PRNGD.*nnn*.SEED

file

The *file* option is used to specify files whose access times the PRNGD uses as an entropy source. You can specify this option as often as you wish. The files specified are entered in a list which is processed cyclically at intervals defined by the *fileInterval* option. Multiple specification of the same file is also possible. This can, for example, make sense in the case of files which are accessed much more frequently than the other specified files.

The entropy collected by this mechanism plays no part in entropy accounting.

file
<file-name>

<file-name>
Name of the file whose access times are used as an entropy source.

fileInterval

The *fileInterval* option is used to specify the (minimum) time intervals at which a file is taken from a list defined with the *file* option and its last access time written to the entropy pool.

fileInterval
<time-interval>

<time-interval>
Time interval in seconds.

cmd

The *cmd* option is used to specify an SDF command whose SYSOUT outputs are to be used as an entropy source. You can specify this option as often as you wish. The commands specified here are entered in a list which is processed cyclically at intervals defined by the *cmdInterval* option.

Multiple specification of the same command is possible. This can, for example, make sense for commands whose SYSOUT output changes considerably more or more frequently than that of the other commands specified.

cmd
<sdf-command-name> <entropy-rate>

<sdf-command-name>
: Name of the SDF command whose SYSOUT outputs are to be used as the entropy source.

<entropy-rate>
: Estimated relative entropy content of the SYSOUT output of the command specified under <sdf-command-name>.

Example

In this example it is assumed that a line of a SYSOUT output of the SDF command SHOW-USER-STATUS INFORMATION=*PROGRAM is 40 bytes long and that there are 16 different, equally probable variants of each of these output lines. The latter means that a task executes one of 16 given programs with equal probability.

In this case each of these lines contains half a byte of entropy and the entropy rate is correspondingly 0.5/40 = 0.0125. In practice it is much more difficult to determine how many different outputs can occur. Furthermore, these outputs are rarely equally probable. However, specific, plausible assumptions regarding the number and probability of various variants of an output line (see above) can be made to estimate the entropy rate and, as required, to reduce this by a safety factor.

cmdInterval

The *cmdInterval* option is used to specify the (minimum) time intervals at which a command is taken from the list defined with the *cmd* option and its SYSOUT output is added to the entropy pool.

cmdInterval
<time-interval>

<time-interval>
Time interval in seconds.
Default: 49

bcamInterval

The *bcamInterval* option is used to specify the (minimum) time intervals at which the BCAM-internal data is added to the entropy pool.

bcamInterval
<time-interval>

<time-interval>
Time interval in seconds.
Default: 49

6.1.3 GPRBYTE program interface of the BS2000 PRNGD

The GPRBYTE program interface of the BS2000 PRNGD is available for the languages C/C++ and ASSEMBLER.

GPRBYTE is the interface of the NLKRES96 routine in the GPRBYTE module of the DSSM subsystem PRNGD. The NLKRES96 routine supplies the pseudo random numbers for the calling program.

Compatibility

The PRNGD program interface is source- and object-compatible as of PRNGD Version 1.0.

Entry name(s) and SVC number(s)

SVC 16 (decimal)

Macro type

The following MF values are supported for macro generation (see the “Executive Macros” manual):

ASSEMBLER MF = { C | D | E | L | M }

The ASSEMBLER macro and the C/C++ include file are in the LMS library SYSLIB.PRNGD.*nnn*.

Macro syntax

	Operation	**Operands**
<Marker>	GPRBYTE	DATAADR = <data-buffer-address> ,NUM_BYT = <integer 1..255> / <number-of-bytes> [,MODE = *NON_BLOCKING]

The MF and PARAM operands are supported in accordance with the convention.

Operand description

DATAADR = <data-buffer-address>
Pointer (char*) to the memory area to which the random numbers are to be written.

NUM_BYT = { <integer 1..255> | <number-of-bytes> }
Number of random bytes which are to be written to the memory area.
The number of random bytes can be specified as follows:

<integer 1..255>
Integer between 1 and 255

<number-of-bytes>
Variable of the data type integer

MODE = *NON_BLOCKING
Operating mode of the random number generator.
Currently only the non-blocking mode is supported, in other words the random numbers are to be supplied by a generator which is not blocked by lack of entropy.

Special language features

Language	Language-specific operand
ASSEMBLER	[PREFIX = { G | <name> }] [,MACID = { PRB | <name>}] [,EQUATES = { YES | NO }]
C	SVC-#: 16, UNIT = 430, FUNCTION = 1, VERSION = 1

Return codes

Return code			Identifier		Meaning
Subcode		Maincode			
2	1				
0	00	0000	successful,	ASS: GPRBSUCC	No error detected
0	20	0001	int_error,	ASS: GPRBINTE	Internal error
0	01	0002	parameter_error,	ASS: GPRBPARE	Parameter error
0	40	0003	buffer_invalid,	ASS: GPRBBUFE	Buffer too small or not allocated
0	40	0004	too_many_bytes,	ASS: GPRBTOOM	More than 255 bytes requested
0	80	0005	prngd_not_ready,	ASS: GPRBNRDY	Random number generator does not have enough entropy
0	80	0006	timeout,	ASS: GPRBTOUT	Random number generator temporarily unavailable

C programming example

```
#include <stdio.h>
#include "FHDR.H"
#include "GPRBYTE.H"

main(int argc, char *argv[])
{
  char randomBytes[128];
  struct GPRBYTE_pl_mdl param;
  enum {UNIT = 430, FUNCTION = 1, VERSION = 1, dataLen = 32};

  FHDR_SET_RC_NIL(param.hdr);
  FHDR_MOD_IFID(param.hdr, UNIT, FUNCTION, VERSION);
  param.in_data.mode = GPRBYTEnon_blocking;
  param.in_data.buffer = &randomBytes;
  param.in_data.num_bytes = dataLen;
  GPRBYTEC(param);
  if (param.hdr.FHDR_RC_MAINCODE == GPRBYTEsuccessful) {
    int i;
    printf("GPRBYTEC called successfully\nData: ");
    for (i = 0; i < dataLen; i++)
      printf("%02X", randomBytes[i]);
    printf("\n");
  }
  else
    printf("Error in call of GPRBYTEC: %08X\n", param.hdr.FHDR_RC_NBR);
}
```

6.1.4 Messages

The PRNGD message code has the format `GPRnnnn`.

Output of messages using /HELP-MSG-INFORMATION

The BS2000 command `/HELP-MSG-INFORMATION MSG-ID=GPRnnnn` allows you to query the meaning and response texts for a message in ongoing operation.

Output of messages on the internet

You can also find the messages on our manual server using the HTML application in the place of the former “System Messages” manual (*http://manuals.ts.fujitsu.com*) and on the DVD “BS2000 SoftBooks”.

6.2 Random number generation in POSIX

As already announced, the POSIX *prngd* daemon is no longer provided. POSIX programs can address the BS2000 PRNGD either directly via the GPRBYTE interface or can use the device file */dev/urandom*, which also obtains the random numbers from BS2000 PRNGD.

/dev/urandom is particularly suitable for scripts, which can access it using the *dd* command, for example.

7 DNS

This chapter is based on the “BIND9 Administrator Reference Manual” of the Internet Software Consortium. The copyright for this Administrator Guide is owned by the Internet Software Consortium. This description is restricted to the parts relevant to BS2000. At the relevant places in the manual, for example syntax descriptions, the “BIND9 Administrator Reference Manual” of the Internet Software Consortium is referred to as the current versions are described only there. You will find the "BIND9 Administrator Reference Manual" of the Internet Software Consortium on your server in the directory */opt/TCP-IP-SV/dns-named/readme* or under *http://www.isc.org/files/arm97.pdf* in the internet.

The DNS (**D**omain **N**ame **S**ervice) is a TCP/IP protocol for the Application Layer that enables TCP/IP application programs to translate the symbolic computer names that are normally used there into their associated IP addresses. The network-wide assignment of computer names to IP addresses is implemented by the DNS with the aid of a distributed database that is made available to all those who need this information on the network. TCP/IP application programs access the DNS functionality via socket functions like *gethostbyname()* and *gethostbyaddr()* (see the manuals “SOCKETS(BS2000)” and “SOCKETS/XTI for POSIX”).

The definition of DNS which is currently valid is based on the RFCs (Request for Comments). These standards are defined by the Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG). Comprehensive information on the RFCs is available on the IETF home page: *http://www.ietf.org/rfc/*

When this manual went to print the following RFCs provided the basis:

- Standards:
  RFC974, RFC1034, RFC1035
- Proposed standards:
  RFC2181, RFC2308, RFC1995, RFC1996, RFC2136, RFC2845, RFC2671, RFC2672, RFC2930, RFC2931, RFC3007, RFC3645
- Proposed standardss for DNS security:
  RFC3225, RFC3833, RFC4033, RFC4034, RFC4035
- Proposed standards, still under development
  RFC1886, RFC2065, RFC2137
- Other Important RFCs About DNS Implementation:
  RFC1535, RFC1536, RFC1982, RFC4074
- Resource Record types:
  RFC1183, RFC1706, RFC2168, RFC1876, RFC2052, RFC2163, RFC2230, RFC2536, RFC2537, RFC2538, RFC2539, RFC2540, RFC2782, RFC2915, RFC3110, RFC3123, RFC3596, RFC3597
- DNS and the Internet:
  RFC1101, RFC1123, RFC1591, RFC2317, RFC2826, RFC2929
- DNS operations:
  RFC1033, RFC1537, RFC1912, RFC2010, RFC2219
- Other DNS-related RFCs:
  RFC1464, RFC1713, RFC1794, RFC2240, RFC2345, RFC2352, RFC3071, RFC3258, RFC3901
- Obsolete and unimplemented Experimental RRs:
  RFC1712, RFC2673, RFC2874

7.1 Concept of the DNS

The Domain Name Service is a distributed, replicated database with servers (DNS servers) and clients (resolvers) in which data from many different DNS servers is maintained. The resolvers, by contrast, have no local database and must issue DNS requests to one or more DNS servers to receive the required information.

7.1.1 Development of the DNS

Applications use IP addresses to set up connections (TCP) and for the datagram traffic (UDP) to partner computers. IP addresses are, however, not very user-friendly, so a naming system was developed in the early days of the internet to enable important computers to be identified by name. Each computer in the TCP/IP network can be assigned one or more freely selectable name(s). The names are independent of the IP address(es) of the computer, but can be mapped to an IP address via functions of the Application Programming Interface (API).

The assignment between computer names and the IP addresses of the computers with which communication is desired is implemented via a host file:

- in Unix systems: */etc/hosts*
- in BS2000 systems: BCAM host file

The host file contains a computer name and its associated IP address in each line.

In the early days of the internet, a central host file had to be administered manually by the Network Information Center (NIC) and copied to all computers on the internet at regular intervals. All new computers were entered into this host file by the network administrator of the NIC and were known to the other computers only after the file was redistributed via FTP.

Due to the fast growth of the internet, this procedure was soon rendered impractical, since the continuously expanding host files needed to be updated and redistributed too often. The use of database technology offered the best solution to this problem.

7.1.2 DNS name space

The DNS name space is structured hierarchically in the form of a tree and divided into different domain levels. It has a root called the root domain, which serves as the anchor for all search transactions within the DNS name space, and several subtrees, which may form independent administration units as so-called zones.

Structure of the DNS name space

The figure 1 illustrates the domain structure of the DNS name space.

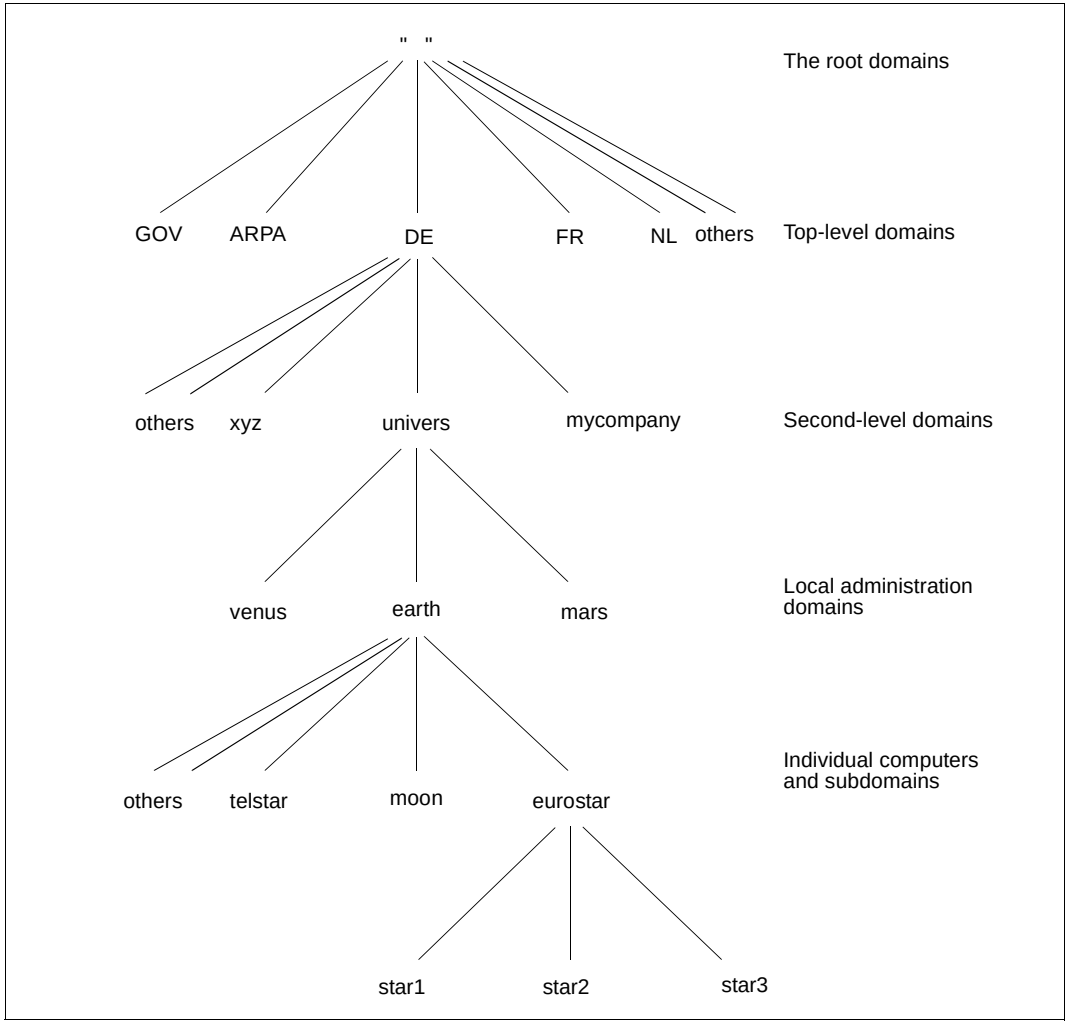


Figure 1: Domain structure of the DNS name space

There are DNS servers at each level of the DNS name space. A DNS server is a computer that performs the following tasks:

- managing information on servers at the next lower name level, and
- mapping of symbolic names to addresses in cases where no subordinate DNS servers for those names exist

The domains of the DNS name space have the following features:

- Root domain

  The root domain is at the very top of the DNS hierarchy and is managed by ICANN. In the root domain, ICANN maintains root domain DNS servers which, in turn, manage the information on the DNS servers of the next lower level.

- Top-level domains

  The top-level domains are in the level below the root domain.

  These are the top-level domains in the United States:

aero	Air-transport industry
biz	Businesses
com	Commercial organizations
coop	Cooperatives
edu	Educational institutions
gov	American government institutions
info	Unrestricted use
int	International organizations
mil	American military institutions
museum	Museums
name	For registration by individuals
net	Network organizations
org	Non-commercial organizations
pro	Accountants, lawyers, physicians, and other professionals

  The top-level domains outside the USA are organized on a country basis. The ISO country codes are used as the names of the countries concerned, e.g. DE for Germany, FR for France, etc. When a new network is registered, the NIC assigns it to the appropriate domain.

- Second-level domains

  Each of the top-level domains branches into several second-level domains. The organizations based on this level nominate domain administrators who are responsible for managing the name servers of their networks. The NIC additionally nominates a central technical office to manage and coordinate general administration issues across domains.

- Local administration domains

  The local administration domains are on the level below the second-level domains of the DNS hierarchy. These domains are administered independently by network providers at their own responsibility. The sizes of local administration domains differ greatly. There are some which consist of only one computer, while others include several computers and even additional DNS servers. A local domain can also have further administration domains (so-called subdomains) subordinate to it.

The *in-addr.arpa* domain is a special domain that is used for converting IP addresses to DNS names. IP addresses are entered in the *in-addr.arpa* domain in reversed decimal dotted notation.

The DNS concept places no restrictions on converting DNS names to IP addresses, and vice versa. It is therefore also possible to assign multiple DNS names to one IP address or multiple IP addresses to one DNS name.

Zones

A zone defines a part of the DNS name space that is administered by a master server (primary name server). A zone is not restricted to an administration domain and may also include some or all subordinate domains.

Zone data file are files from where the name servers load their zone data.

7.1.3 Storing information in the DNS

All DNS information is only stored within the DNS server. This information is maintained in the form of resource records (RRs). Resource records are coded in ASCII and contain the following details:

– Owner of the information
– RR type
– Class (e.g. IN for internet)
– Validity duration
– Data

The format of a Resource Record is described in detail on page 212.

The most important RR types include:

SOA (Start of Authority)

The SOA RR specifies the start of the zone serviced by the DNS name server on which the SOA RR is stored. A system file may contain only one SOA RR per zone. The zone ends at the start of the next zone, i.e. when a new SOA Resource Record is specified.

NS (Name Server)

The NS RR specifies the name of the server responsible for a specific DNS domain.

A (Address)

The Address RR defines the IP address assigned to a DNS name. There should be a separate Address RR for each IP address of a computer.

AAAA (quad A)

The AAAA RR defines the IPv6 address which is assigned to a DNS name. There should be a separate AAAA RR for each IPv6 address of a computer.

PTR (Domain Name Pointer)

The PTR RR defines special names as pointers to other names in the domain. PTR RRs are primarily used in *in-addr.arpa* data records to map addresses (the special names) to computer names. PTR names should be unique within a zone.

MX (Mail Exchanger)

The MX RR defines the IP address assigned to a DNS name.

CNAME (Canonical Name)

The CNAME RR can be used to assign an alternative name to a standard host name. The alternative name must not be specified in the name field of other Resource Records. Changes in the standard host name remain transparent to all application programs using the alternative name, i.e. the application programs need not be updated on changing the standard host name.

Information on further Resource Record types is available, for example, in RFC1035. Comprehensive information on the RFCs is available on the home page of the Internet Engineering Task Force (IETF): *http://www.ietf.org*

7.1.4 Format of a DNS message

The same format is used in the DNS for queries and responses (see figure 2).

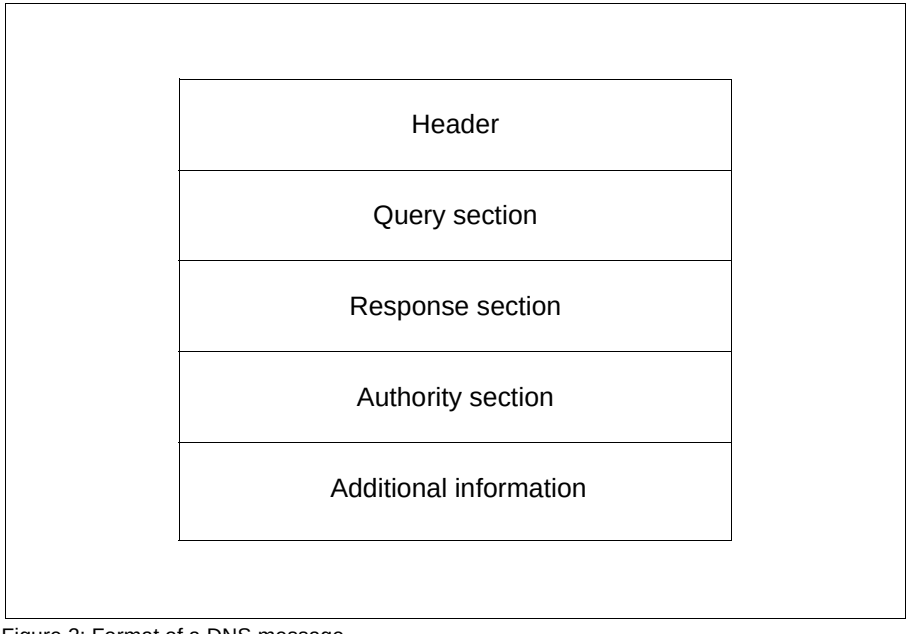


Figure 2: Format of a DNS message

The individual sections of a DNS message have the following significance:

- The header has a fixed length and contains metadata on the DNS message, including, among other things, the identification, the type of message (query or response) and the length of the following variable sections.

- The Query section specifies the information to be found.
- The Response section contains the found information in the form of a variable number of Resource Records.
- The Authority section contains RRs with the names of other name servers in case the name server contacted earlier could not return a response.
- The lowest section may include additional information in the form of RRs.

7.1.5 DNS resolver (overview)

The DNS resolver handles the requests for resolving DNS domain names into IP addresses which application programs direct to the DNS Name Server.

Access to the DNS resolver

Application programs gain access to the resolver via the socket functions linked to the application, such as the functions *gethostbyname()* and *gethostbyaddr()* which are used for IPv4, and the functions *getaddrinfo()*, *getnameinfo()*, *getipnodebyaddr()* and *getipnodebyname()* which are used for IPv4 and IPv6:

gethostbyname()	Supplies the associated IPv4 address for a host name.
gethostbyaddr()	Supplies the associated host name for an IPv4 address.
getaddrinfo()	Supplies information on host names, host addresses and service independently of the protocol.
getnameinfo()	Supplies the name of the communications partner independently of the protocol.
getipnodebyaddr()	Supplies information on host names independently of the protocol.
getipnodebyname()	Supplies information on host addresses independently of the protocol.

Normally the resolver functions are provided in a library which is supplied with the products “SOCKETS(BS2000)” and “SOCKETS/XTI for POSIX”. The resolver functions used there are described in the manuals "SOCKETS(BS2000)” and “SOCKETS/XTI for POSIX”.

In POSIX it is also possible to include the complete resolver functionality in the application via the library. In this case the resolver functions are described in the online documentation on DNS. To access the online documentation you must install the DNS development modules (see page 221).

Resolution of the requests directed to the DNS resolver

The requests directed to the DNS resolver can be resolved in various ways:

- Via the resolver provided with SOCKETS(BS2000) using the configuration file in the BS2000 file system. This configuration file has the name $TSOS.SYSDAT.SOCKETS.*nnn*.SOC6.RESOLV (up to and including openNet Server V3.0) or SYSDAT.LWRESD.*nnn*.RESOLV.CONF. (*nnn* specifies the version.)

  If you wish to use the DNS resolver functionality via SOCKETS(BS2000), the subsystem must be installed and running. Further information on this can be found in the “SOCKETS(BS2000)” manual.

- With the resolver daemon provided via interNet Services DNS using the configuration file */etc/resolv.conf* in the POSIX file system.

  If you wish to use the DNS resolver functionality via interNet Services DNS, the DNS resolver daemon must be installed and running. Further information on this can be found in the section “Installing and uninstalling the DNS resolver” on page 219.

- By linking the resolver library using the configuration file */etc/resolv.conf* in the POSIX file system (only possible in POSIX).

  Here you do not need to start either the SOCKETS(BS2000) subsystem or the DNS resolver daemon. The configuration file */etc/resolv.conf* in the POSIX file system must, however, be configured correctly.

- By querying the BCAM tables.

Order of the queries when resolving the resolver requests

How or in what order the queries occur is determined by the program platform (BS2000 or POSIX) and how the application is linked:

- Sockets application in BS2000:

  The resolver functions initially use the internal Sockets resolver or, in Sockets 2.2 and higher, the LWRESD which runs in Native BS2000. Only if these supply no result is name/address conversion attempted with the entries from the BCAM tables. The DNS resolver daemon of interNet Services is never addressed.

- Sockets application in POSIX (default):

  – When you are using the pure IPv4 functions *gethostbyname()* and *gethostbyaddr()* in a Sockets application in POSIX, first name or address conversion is attempted using the entries in the BCAM tables.

    If a name or address cannot be determined by a BCAM request, the DNS resolver daemon of interNet Services DNS is called.

If the daemon supplies no result, the resolver functionality of BS2000 Sockets is called, i.e. LWRESD is called in Sockets 2.2 or higher.

Only if this conversion attempt fails do the *gethostbyname()* and *gethostbyaddr()* functions return an error.

– When you are using the DNS resolver functions *getaddrinfo()*, *getnameinfo()*, *getipnodebyaddr()*, *getipnodebyname()*, first the resolver functionality of BS2000 Sockets is used.

  If a name or address cannot be determined hereby, name or address conversion is attempted using the entries in the BCAM tables.

  Only if this conversion attempt also fails do the *getaddrinfo()*, *getnameinfo()*, *getipnodebyaddr()*, *getipnodebyname()* functions return an error.

- Sockets application in POSIX with linked resolver library of interNet Services DNS:

  By including the resolver library of interNet Services DNS results in the file */etc/resolv.conf* being accessed exclusively in the POSIX file system for all calls (IPv4 and IPv6) when the application runs.

  A prerequisite for production of the application is that the DNS development modules are installed (see page 221).

  When the application is linked, the following link sequence must be observed:

  1. Resolver library of interNet Services DNS: */usr/lib/libbind.a*
  2. SOCKETS/XTI library: */usr/lib/libsocket.a*

7.1.6 DNS name server NAMED (overview)

NAMED is the DNS name server in BS2000. NAMED enables both recursive and iterative queries as well as caching and can be optionally configured as:

– a master server,
– a slave server,
– a "caching-only" server,
– a forwarding server.

Recursive and iterative queries

On receiving a query of another name server or the resolver, NAMED first tries to answer the query on the basis of its own database.

If this does not succeed, NAMED behaves as described below, depending on the configuration:

- **recursive** queries (default case)

  In the case of a recursive query, the name server performs all activities required to answer that query. Here recursive means that the name server repeats the same basic procedure (sending queries to a remote name server and following links) until it receives the answer it is looking for.

  The resolver sends a recursive query to a name server to obtain information on a particular domain name. The name server should then supply the requested data or return an error message reporting that

  – the data of the requested type is not available or
  – the specified domain name does not exist.

  With a recursive query the name server cannot simply refer the resolver to another name server but must itself contact the next name server it knows.

- **iterative** queries
  The name server rejects a query if it cannot be answered locally, i.e. with information from its own database. The source of the query must then use some other name server to get an answer for the query.

caching

NAMED stores all incoming information during the iterative process in a local cache, which extends the database and reduces the answering time. If too many queries are required, the cache can grow very large.

Master server and slave server

DNS name servers administer zones, if they are authoritative for the zone. As a rule, the DNS within a zone is not implemented on a single NAMED server, but on a group of such servers. To enable failover security, apart from the master server (primary server), at least one additional server should be set up as a slave server (secondary server), which manages a copy of the master server data. NAMED can be configured both as a master server and as a slave server.

Changes in the zone data are always made on the master server. The Master name server for a zone reads the zone data from a file on its host, the zone data file. The slave server, which always receives its data only from the master server, queries the master server periodically to check whether its own copy of the master server data needs to be updated. The transfer of master server data to the slave server is known as a zone transfer.

If desired, the master server can also be configured to actively notify all slave servers when the zone data is changed. These slave servers can then initiate the zone transfer. A slave server that only takes over NS Resource Records from the master server is known as a stub server.

During a zone transfer, the slave servers can save the transferred data to backup files. If the backup files are not available on starting up a slave server, the slave server in question requests the current data from the master server.

Zone data should be transferred regularly even if it has not been modified. This enables the data to be accessed even if the master server is not available.

A NAMED server may be used in different zones as a master and/or slave server.

Forwarding servers and forwarders

If NAMED is configured as a forwarding server, it forwards all queries that it cannot answer with its own database to special name servers (called "forwarders") for recursive processing. These forwarders then try to obtain the required answers in an iterative process. If the forwarders cannot return the final information, the forwarding server will try to independently answer the query again, depending on its configuration, or simply abort the process.

Forwarding-only mode

In this mode a name server that uses a forwarder does not inquire at other name servers if the forwarder supplies no result.

Caching-only servers

NAMED can also be configured as a "caching-only" server. A caching-only server does not maintain its own committed database, but requests information from other authorized name servers. This information is then saved in the cache of the caching-only server.

Views

The *view* statement enables a NAMED server to be configured in such a way that it handles queries differently depending on the sender address.

7.1.7 DNS security concepts

TSIG (Transaction SIGnatures) is a key-based security mechanism. It is suitable for securing communication between two servers. As an additional security measure TSIG uses shared secrets. TSIG is, for example, useful for the dynamic update.

The DNSSEC (DNS SECurity) extensions are also key-based. They use encryption with a public key.

7.1.8 Interaction of the security mechanisms of BCAM and DNS

If conflicts occur between the established security mechanisms and the new options of name or address conversion resulting from the use of DNS, the internal system security mechanisms always have priority.

BCAM offers two methods of allowing communication relationships between partners:

- Communication relationships are possible to any partners, even if they are not known to the BCAM transport system.

  In this case, using the addresses determined by the DNS cannot cause conflicts with the BCAM transport system security mechanisms, since these are explicitly disabled.

- Communication relationships are only possible to partners known to the BCAM transport system.

  In this case, the following situation could occur:

  An application may receive the address of a partner system from the DNS, but may still be unable to set up a connection to it, since this partner system is not known to the BCAM transport system, and communication is not permitted for security reasons.

  If required, a remedy is provided in this situation with the transport system enhancement BCAM-DNS-ACCESS which was introduced with openNet Server V3.1.

7.2 DNS resolver

This section provides information on the following topics:

- Installing and uninstalling the DNS resolver
- Configuring the DNS resolver
- Administration and operation of the DNS resolver
- Diagnosis and maintenance for the DNS resolver

7.2.1 Installing and uninstalling the DNS resolver

In this context please also consult the Release Notice supplied with the product interNet Services.

The different components of the interNet Services software package are installed as a POSIX program package by the POSIX installation program (see the manual “POSIX Basics for Users and System Administrators”). You will find more detailed information on the installation of the components in the section “Installation” on page 35.

For the installation and operation of the TCP-IP-SV component, the PLAM library SINLIB.TCP-IP-SV.*nnn*.DNS must be shareable. During installation of DNS, the installation path can be defined in the POSIX file system. The DNS resolver is installed by default in the */opt/TCP-IP-SV/dns* directory.

Installing the DNS resolver

After successful installation of the DNS components the relevant commands and files are contained in the directories */usr/bin*, */usr/sbin* and */etc* and in the installation directory */opt/TCP-IP-SV/dns*. In the table on page 219, <instdir> stands for the installation directory */opt/TCP-IP-SV/dns*.

Directory	File/Link	Description
/etc/default	TCP-IP-SV.dns	Parameter file for the start/stop procedure
	TCP-IP-SV.dns.default	Default for TCP-IP-SV.dns
/etc/init.d/	TCP-IP-SV.dns	Start/stop procedure
/etc/rc0.d/	K66TCP-IP-SV.dns	Symbolic link to */etc/init.d/TCP-IP-SV.dns* (to terminate when POSIX shuts down)

[1] Only available if the DNS development modules are installed (see page 221).

Directory	File/Link	Description
/etc/rc2.d/	S70TCP-IP-SV.dns	Symbolic link to */etc/init.d/TCP-IP-SV.dns* (to start when POSIX starts)
/usr/bin/	dig	Symbolic link to *<instdir>/bin/dig*, diagnostic tool for DNS
	host	Symbolic link to *<instdir>/bin/host*, host names DNS lookup tool
	nslookup	Symbolic link to *<instdir>/bin/nslookup*, lookup tool for DNS
/usr/lib/	libbind.a	Symbolic link to *<instdir>/lib/libbind.a*[1)]
<instdir>/bin/	dig	Diagnostic tool for DNS
	host	host names DNS lookup tool
	nslookup	Lookup tool for DNS
/usr/sbin/	in.dnsd	Symbolic link to *<instdir>/sbin/dnsd*
<instdir>/etc/	install.TCP-IP-SV-DEV.*nnn*.DNS	Installation script for the DNS development modules
	remove.TCP-IP-SV-DEV.*nnn*.DNS	Uninstallation script for the DNS development modules
	rc.d/dnsd	Start/stop script for dnsd
<instdir>/lib/	libbind.a	Resolver library[1)]
<instdir>/readme/	*	man pages for the DNS tools and for the resolver library[1)] in HTML and text format
<instdir>/sbin/	dnsd	DNS resolver daemon

[1)] Only available if the DNS development modules are installed (see page 221).

After successful installation the other activities that are required are executed under the POSIX shell in an ID with POSIX root authorization. For this purpose the POSIX shell is started with the BS2000 command START-POSIX-SHELL.

After interNet Services has been installed, the configuration files specific to the DNS components must be adapted to individual requirements before interNet Services is started up.

Uninstalling the DNS resolver

The uninstallation of the interNet Services components is carried out using the POSIX installation program under the TSOS ID in the same way as installation. During uninstallation, the active DNS resolver daemon is searched for and terminated. The termination of daemons that are still active is logged in the syslog file */var/adm/syslog*. All files, links and procedures of the DNS resolver are then deleted.

In the course of uninstallation the configuration file */etc/default/TCP-IP-SV.dns* is saved in the */etc/tcpipsv* directory if it was changed in comparison to the default file */etc/default/TCP-IP-SV.dns.default*. If a backup file with the same name already exists, it is first renamed by the date being appended to it. If installation takes place again, the backup copy can be used as a template for generating the new configuration file. In this context please ensure that the backup copy has the required status.

Installing/uninstalling DNS development modules

With the DNS development modules additional files are provided which are not required for operation of the DNS resolver daemon. These comprise the resolver library with the associated man pages which are needed if a Sockets application is to be produced in POSIX with a linked resolver library (see the table on page 219).

The DNS development modules are made available in the POSIX file system with the following shell script:

/opt/TCP-IP-SV/dns/etc/install.TCP-IP-SV-DEV.nnn.DNS

When the DNS development modules are no longer required, they can be removed from the POSIX file system using the following shell script:

/opt/TCP-IP-SV/dns/etc/remove.TCP-IP-SV-DEV.nnn.DNS

7.2.2 Configuring the DNS resolver

The start procedure */etc/init.d/dnsd* is parameterized by the configuration file */etc/default/TCP-IP-SV.dns*. The job name, account and job class under which the daemon is to run can be set. You can also define whether the daemon is started automatically when POSIX starts.

Operation of the DNS resolver daemon is controlled by the configuration file */etc/resolv.conf*. The configuration file can contain several entries which specify the information relevant to the DNS resolver.

The following types of entry exist:

- *nameserver* entry
- *domain* entry
- *search* entry
- *options* entry

Each entry is contained in a separate line in the configuration file and begins with the keyword (nameserver, domain etc.) in column 1. The keyword is followed by the entry's parameters, each separated by a blank. The last parameter may not be followed by a blank. The keywords of the */etc/resolv.conf* entries are expected in lower case.

Lines in */etc/resolv.conf* which begin with a semicolon (;) are interpreted as comment lines.

nameserver entry

```
nameserver ip_address
```

The *nameserver* entry is used to inform the DNS resolver daemon of the IP address of a DNS server. The */etc/resolv.conf* configuration file may contain a maximum of three *nameserver* entries.

When the DNS resolver daemon processes a DNS request, the DNS server defined with the first *nameserver* entry is queried first. If the first DNS server does not reply, the DNS server defined with the second *nameserver* entry is queried, if such an entry exists. If the second server also does not reply, the process is repeated with the third name server, and so on. If the DNS server defined by the third *nameserver* also does not reply, you should repeat the DNS request with all DNS servers until the maximum number of retries (see the *options* entry on page 228, *attempts* parameter) is reached.

domain entry

domain domain

The *domain* entry defines a default domain for the requests of the DNS resolver daemon to the DNS name server(s).

The following must be observed here:

- A maximum of one *domain* entry can be valid in the configuration file */etc/resolv.conf*. If the configuration file contains several *domain* entries but no *search* entries, the last *domain* entry applies.
- *domain* and *search* entries invalidate each other, so if there are several *domain* and *search* entries in the configuration file, the *domain* entry and *search* entry which apply are those which are not followed by a further *domain* or *search* entry.

The function of the *domain* entry depends on whether or not the DNS name specified in the DNS request contains a dot (.).

The following description applies for the default setting of the *ndots* parameter in the *options* entry (see page 228).

- Function of the *domain* entry if the DNS name contains a dot:

  If the DNS server cannot reply successfully to a DNS request for a name *name*, the DNS resolver daemon sends another DNS request with *name.domain* to the DNS server. If this DNS request also fails, the DNS resolver daemon sends a further DNS request with the higher-level domain to the DNS server. This procedure can be continued up to a DNS request in which the appended domain is one level below the top level.

Example

The following example shows the requests generated by the DNS resolver daemon to a DNS server.

Entry in the configuration file */etc/resolv.conf*:

```
domain mch.fj.example
```

Name specified by the user:

```
my.host
```

Name generated by the DNS resolver daemon for requests to a DNS server:

1. `my.host`
2. `my.host.mch`.fj.example
3. `my.host`.fj.example

- Function of the *domain* entry if the specified DNS name does not include a dot:

  If a DNS request for a name *name.domain* cannot be successfully answered by the DNS server, the DNS resolver daemon issues a new DNS request with *name* to the DNS server.

  Example

  The following example shows the requests to a DNS server generated by the DNS resolver daemon.

  Entry in the configuration file */etc/resolv.conf*:

  ```
  domain mch.fj.example
  ```

  Name specified by the user:

  ```
  myhost
  ```

  Name generated by the DNS resolver daemon for requests to a DNS server:

  1. `myhost.mch`.fj.example
  2. `myhost`.fj.example
  3. `myhost`

search entry

```
search domain1[ domain2[ .... [ domain6]]]
```

The *search* entry defines for the resolver daemon several domain names for the requests to the DNS name server(s).

The following must be observed here:

- A maximum of one *search* entry can be valid in the configuration file */etc/resolv.conf*. If the configuration file contains several *search* entries but no *domain* entries, the last *search* entry applies.
- *domain* and *search* entries invalidate each other, so if there are several *domain* and *search* entries in the configuration file, the *domain* entry and *search* entry which apply are those which are not followed by a further *domain* or *search* entry.

The function of the *search* entry depends on whether or not the DNS name specified in the DNS request contains a dot (.).

- Function of the *search* entry if the specified DNS name includes a dot:

  If a DNS request for a specific *name* cannot be successfully answered by the DNS server, the DNS resolver daemon issues a new DNS request with *name.domain1* to the DNS server.

  If even this DNS request cannot be successfully answered by the DNS server, the DNS resolver repeats the DNS request with the next domain name *domainx* (x = 2 - 6) defined in the *search* entry. This is repeated until either the DNS request is successfully answered by the DNS server or no further domain names are defined in the *search* entry.

Example

The following example shows the requests generated by the DNS resolver daemon to a DNS server:

Entry in the */etc/resolv.conf* configuration file:

```
search mch.fj.example fj.example
```

Name specified by the user: `my.host`

Names generated by the DNS resolver daemon for requests to a DNS server:

1. `my.host`
2. `my.host.mch`.fj.example
3. `my.host`.fj.example

- Function of the *search* entry if the specified DNS name does not include a dot:

  A DNS request with *name.domain1* is initially issued to the DNS server. If this DNS request is not successfully answered by the DNS server, the DNS resolver daemon then repeats the request using the next domain name *domainx* (x = 2 to 6) defined in the *search* entry. This is repeated until either the DNS request is successfully answered by the DNS server or no further domain names are defined in the *search* entry.

  If no DNS request has been successfully answered by the DNS server after all domain names defined in the *search* statement have been processed, a DNS request is issued with *name*.

  Example

  The following example shows the requests generated by the DNS resolver daemon to a DNS server.

  Entry in the */etc/resolv.conf*: configuration file:

  ```
  search mch.fj.example fj.example
  ```

  Name specified by the user:

  ```
  myhost
  ```

  Names generated by the DNS resolver daemon for requests to a DNS server:

  1. `myhost.mch`.fj.example
  2. `myhost`.fj.example
  3. `myhost`

options entry

```
options option [option] ...
```

The *options* entry is used to define the behavior of a number of resolver routines by the values specified for *option*.

The following values can be specified for *option*:

debug
: This activates the diagnostic mechanism of the DNS resolver daemon, and diagnostic messages are written to the */var/adm/syslog* file.

ndots:*n*
: Specifies the lower threshold value for the number of dots (“.”) which a name transferred in a DNS request must contain for a first absolute (“as-is”) request to the DNS name server to be executed.
The default is `ndots:1`. Thus if a name in a request to the DNS name server contains at least one dot, this name is first of all treated as an absolute name in a request before it is supplemented by domain names (see also the *domain* entry on page 224).

attempts:*n*
: Specifies the maximum number of connection attempts which can be made to each DNS name server per DNS request. The specification `attempts:0` has the same effect as the specification `attempts:1`.
The default is `attempts:4`

timeout:*n*
: Specifies the start timeout value for a retransmission in seconds. The previous timeout value is doubled with every attempt until the maximum number of connection attempts (see the *attempts* parameter) has been made. The specification `timeout:0` has the same effect as the specification `timeout:1`.
The default is `timeout:5`.
When the default values for *attempts* and *timeout* are used, the overall timeout value for each DNS server is as follows:
5+10+20+40=75 seconds.

rotate
: The requests are sent in time slicing mode to the DNS name server generated with the aid of the *nameserver* entry. Thus instead of letting all clients send their first request to the first DNS server listed, the request volume is distributed evenly over all the listed DNS name servers.

no_tld_query
: Causes the DNS resolver not to search for the name of the top level domain, in other words for a name which contains no dots (“.”). Using this option does not prevent the DNS resolver from applying the rules in accordance with a *domain* or *search* entry.

7.2.3 DNS resolver - administration and operation

This section provides information on the following topics:

- Starting up and shutting down the DNS resolver
- Modifying the configuration of the DNS resolver

7.2.3.1 Startup and shutdown of the DNS resolver

The DNS resolver is automatically started up when the POSIX subsystem is started and shut down when POSIX is shut down if it is activated by AUTOSTART='yes' in */etc/default/TCP-IP-SV.dns*.

Startup call for the DNS resolver

The startup call for the DNS resolver daemon is:

```
/etc/init.d/TCP-IP-SV.dns start
```

Initially a check is made to see whether AUTOSTART is set to 'yes'. If it is not, no further action is executed. If it is set to 'yes', a check is made to see whether a daemon has already been started. If this is the case, only a corresponding message is output, and if it is not the case the daemon is started.

If the daemon is to be started in any case, irrespective of AUTOSTART, the *mstart* option must be used instead of *start*.

The *dstart* option enables the daemon to be executed in the foreground for debugging purposes; the debugging options in the *TCP-IP-SV.dns* script may need to be adjusted.

Restart call for the DNS resolver

There is also a restart call offered for the DNS resolver daemon. This call is needed whenever a modified configuration file is to be read in during a session.

The restart calls for the DNS resolver is:

```
/etc/init.d/TCP-IP-SV.dns restart
```

A check is made during the restart procedure run to determine if the daemon concerned has been started. If no active daemon is found, a normal new startup is executed.

Shutting down the DNS resolver

The following call is available to stop the DNS resolver daemon:

```
/etc/init.d/TCP-IP-SV.dns stop
```

The shutdown only applies until the POSIX subsystem is terminated. If an automatic restart is to be prevented when the POSIX subsystem is restarted, the AUTOSTART variable in */etc/default/TCP-IP-SV.dn*s must be set to 'no' or 'never'.

7.2.3.2 Modifying the DNS resolver configuration file

The DNS resolver daemon */etc/resolv.conf* configuration file can be modified in the POSIX shell under $TSOS or $SYSROOT using EDT. The syntax of the entries must be stringently adhered to when editing the configuration file; otherwise, DNS queries may produce undesirable results.

Changes in the configuration file only take effect after a DNS resolver daemon new start or restart (see page 230).

7.2.4 DNS resolver - diagnosis and maintenance

This section describes logging functionality and diagnostic options for DNS.

7.2.4.1 DNS resolver - logging

The DNS resolver stores its logging information in the */var/adm/syslog* file. These entries have the following basic format:

```
Jan 22 15:44:58 LOG_NOTICE syslog[799]: dnsdamon: <message text>
```

The date and time output is followed by a keyword denoting the message classification. The DNS resolver daemon only uses the LOG_NOTICE keyword for its logging messages.

The classification is followed by the ID as a system message (syslog) with information on the relevant process ID (PID) [pid]. If no valid PID is available for the message output (e.g. the message issued by the daemon start procedure), an empty parenthesis expression is output. The name of the daemon (in this case *dnsdamon*) is output after the PID, enclosed in colons (:). The actual message text then follows.

No further logging messages are output after the *stopdns* command is entered, i.e. when the DNS resolver daemon terminates or is no longer running.

7.2.4.2 DNS resolver - diagnostic options

You activate the DNS debugging mechanism with the aid of the *options* entry (`options debug`, see page 228) in the configuration file */etc/resolv.conf*. The diagnostic messages of the DNS resolver daemon are then logged in the */var/adm/syslog* file.

7.3 DNS name server NAMED

This section provides information on the following topics:

- Installing and uninstalling NAMED
- Configuring NAMED
- Administration and operation of NAMED
- Diagnosis and maintenance of NAMED

7.3.1 Installing and uninstalling NAMED

In this context please also consult the Release Notice supplied with the product interNet Services.

The different components of the interNet Services software package are installed as a POSIX program package by the POSIX installation program (see the manual “POSIX Basics for Users and System Administrators”). You will find more detailed information on the installation of the components in the section “Installation” on page 35.

For the installation and operation of the TCP-IP-SV component, the PLAM library SINLIB.TCP-IP-SV.*nnn*.NAMED must be shareable. During installation of NAMED, the installation path can be defined in the POSIX file system. The default directory for installation is */opt/TCP-IP-SV/dns-named*.

Installing NAMED

After successful installation of the NAMED component, the relevant commands and files are contained in the directories */usr/sbin*, */usr/bin* and */etc* and in the installation directory */opt/TCP-IP-SV/dns-named*. In the table on page 233, <instdir> stands for the installation directory */opt/TCP-IP-SV/dns-named*.

Directory	File	Description
/etc/	named.conf	DNS configuration file of the name server
/etc/default	TCP-IP-SV.named	Parameter file for the start/stop procedure
	TCP-IP-SV.named.default	Default for TCP-IP-SV.named
/etc/init.d/	TCP-IP-SV.named	Start/stop procedure

[1)] Only available if the NAMED development modules are installed (see page 236).

Directory	File	Description
/etc/rc0.d/	K64TCP-IP-SV.named	Symbolic link to */etc/init.d/TCP-IP-SV.named* to terminate when POSIX shuts down
/etc/rc2.d/	S72TCP-IP-SV.named	Symbolic link to */etc/init.d/TCP-IP-SV.named* to start when POSIX starts
/usr/bin/	nsupdate	Symbolic link to *<instdir>/bin/nsupdate*
/usr/lib/	liblwres.a libisccfg.a libisccc.a libisc.a libdns.a	Additional DNS libraries not required for running NAMED [1)]
/usr/sbin/	in.named	Symbolic link to *<instdir>/sbin/named*
	named-checkconf	Symbolic link to *<instdir>/sbin/named-checkconf*
	named-checkzone	Symbolic link to *<instdir>/sbin/named-checkzone*
	rndc	Symbolic link to *<instdir>/sbin/rndc*
	rndc-confgen	Symbolic link to *<instdir>/sbin/rndc-confgen*
	dnssec-keygen	Symbolic link to *<instdir>/sbin/dnssec-keygen*
	dnssec-signzone	Symbolic link to *<instdir>/sbin/dnssec-signzone*
<instdir>/etc/	install.TCP-IP-SV-DEV.*nnn*.NAMED	Installation script for the development modules
	remove.TCP-IP-SV-DEV.*nnn*.NAMED	Deinstallation script for the development modules
<instdir>/include/	*	Include files[1)]
<instdir>/readme/	bind9arm.pdf	Bind9 Administrator Reference Manual
	*	man pages in HTML and text format

[1)] Only available if the NAMED development modules are installed (see page 236).

Directory	File	Description
<instdir>/sbin/	named	DNS name server daemon
	named-checkconf	Syntax check of *named.conf* file
	named-checkzone	Syntax check of a zone data file
	rndc	Remote control program for name server daemon
	rndc-confgen	Generation program for *rndc* configuration file
	dnssec-dsfromkey	Generates the "Delegation Signer" resource record.
	dnssec-keygen	Key generation
	dnssec-revoke	Revokes a key pair
	dnssec-settime	Program for setting and displaying key-specific time specifications
	dnssec-signzone	Assigns a zone

[1)] Only available if the NAMED development modules are installed (see page 236).

After successful installation the other activities that are required are executed under the POSIX shell in an ID with POSIX root authorization. For this purpose the POSIX shell is started with the BS2000 command START-POSIX-SHELL.

After interNet Services has been installed, the configuration and system files specific to the DNS components must be adapted to individual requirements before interNet Services is started up.

Uninstalling NAMED

The uninstallation of the interNet Services components is carried out using the POSIX installation program under the TSOS ID in the same way as installation.

Before you remove the NAMED components using the POSIX-INSTALLER, you should use the script *<instdir>/etc/remove.TCP-IP-SV-DEV.nnn.NAMED* to remove the development modules (if installed).

During uninstallation, the active DNS NAMED daemon is searched for and terminated. The termination of daemons that are still active is logged in the syslog file */var/adm/syslog*. All files, links and procedures of DNS NAMED are then deleted.

During uninstallation, the */etc/named.conf* and */etc/default/TCP-IP-SV.named* configuration files are saved. If a file with the same name but with different content already exists, it is renamed by the date being appended to it. In the event of reinstallation, the backup copy is copied to the corresponding directory again. Ensure that the backup copy has the desired status.

Installing/uninstalling NAMED development modules

With the NAMED development modules, additional files are provided which are not required for operation of the NAMED. These comprise Bind9 libraries which are used internally, plus the associated include files, which are provided by default with Bind9 but are only needed if application development is to be based on these Bind9 libraries (see the table on page 233).

The NAMED development modules are made available in the POSIX file system with the following shell script:

/opt/TCP-IP-SV/dns-named/etc/install.TCP-IP-SV-DEV.nnn.NAMED

When the NAMED development modules are no longer required, they can be removed from the POSIX file system using the following shell script:

/opt/TCP-IP-SV/dns-named/etc/remove.TCP-IP-SV-DEV.nnn.NAMED

7.3.2 Configuring NAMED

This section provides information on the following topics:

– NAMED configuration file *named.conf*
– NAMED and security
– NAMED process models

7.3.2.1 NAMED configuration file named.conf

The operation of the DNS name server daemon is controlled by means of the configuration file */etc/named.conf*. The syntax of the configuration file *named.conf* is described in the “BIND9 Administrator Reference Manual” of the Internet Software Consortium.

Example

Structure of a *named.conf* file

```
options {
   directory "/var/named";
};

logging {
   channel my_security_channel {file "my_security_file"; severity info; };
   category security { my_security_channel; default syslog; };
   category cname { null; };
};

zone "test1.mch.fj.example" IN {
   type master;
   file "masterzone";
};

zone "test2.mch.fj.example" IN {
   type slave;
   file "slavezone";
   masters { 155.90.80.1; };
};

zone "." in {
   type hint;
   file "named.cache";
};

zone "60.155.in-addr.arpa" IN {
   type master;
   file "arpafile";
```

```
};

zone "0.0.127.in-addr.arpa" in {
   type master;
   file "loopback";
};
```

7.3.2.2 NAMED zone data files

Apart from general configuration settings, the file */etc/named.conf* also contains information on the zones to be set up. The actual zone data is stored in local system files as Resource Records and read from these system files on starting the server daemon.

The syntax of the local zone data files is described in the “BIND9 Administrator Reference Manual” of the Internet Software Consortium.

All zone data files must be created in the SRR (Standard Resource Record) format, where each line of the file contains one Resource Record (RR).

Example

Structure of the *masterzone* file

```
test1.mch.fj.example.  IN  SOA  host1.test1.mch.fj.example.  postmaster (
                       1 ; serial
                       10800 ;    Refresh after 3 hours
                       3600 ;     Retry after 1 hour
                       604800 ;   Expire after 1 week
                       86400) ;   Minimum TTL of 1 day
;
;  Name servers
;
test1.mch.fj.example.  IN    NS     host1.test1.mch.fj.example.
                         IN    NS     host2.test1.mch.fj.example.
;
;  Addresses
;
host1             IN     A        155.60.70.1
host2             IN     A        155.60.70.2
host3             IN     A        155.60.70.3
```

```
loopback          IN     A        127.0.0.1
;  Aliases
;
alias1            IN    CNAME        host1.test1.mch.fj.example.
;
;   Mail Exchanger
;
test1.mch.fj.example.    IN    MX    0      host1.test1.mch.fj.example.
test1.mch.fj.example.    IN    MX    10     host3.test1.mch.fj.example.
```

Example

Structure of the *arpafile* file

```
60.155.in_addr.arpa.    IN   SOA   host1.test1.mch.fj.example.postmaster (
                                   postmaster (
                        1       ; serial
                        10800   ; Refresh after 3 hours
                        3600    ; Retry after 1 hour
                        604800  ;  Expire after 1 week
                        86400)  ;  Minimum TTL of 1 day
;
;  Name servers
;
60.155.in_addr.arpa.     IN  NS     host1.test1.mch.fj.example.
                         IN  NS     host2.test1.mch.fj.example.
;
;  Addresses
;
1.70            IN         PTR     host1.test1.mch.fj.example.
2.70            IN         PTR     host2.test1.mch.fj.example.
3.70            IN         PTR     host3.test1.mch.fj.example.
```

Example

Structure of the *loopback* file

```
0.0.127.in_addr.arpa.  IN  SOA  host1.test1.mch.fj.example.
                                 postmaster (
                       10800   ; Refresh after 3 hours
                       3600    ; Retry after 1 hour
                       604800  ; Expire after 1 week
                       86400)  ; Minimum TTL of 1 day
;
;  Name servers
;
0.0.127.in_addr.arpa.  IN  NS     host1.test1.mch.fj.example.
;
;  Addresses
;
1.0.0.127.in_addr.arpa.  IN  PTR    localhost.
```

7.3.2.3 NAMED and security

When NAMED is configured, the following options allow security aspects to be taken into account when data is accessed:

- The *allow-query* option of the *options* statement enables the authorization to send requests to the name server to be restricted to individual hosts.
- The *allow-transfer* option of the *options* statement enables the authorization to receive zone data from the name server to be restricted to individual hosts.
- The *allow-update* option of the *zone* statement enables the option of dynamic data update to be restricted to individual hosts on a zone-specific basis.

TSIG

A further security mechanism is provided by the Transaction SIGnatures (TSIG). These support server-to-server communication, including zone transfer, notify and recursive queries.

TSIG is key-based and is applied to communication between two DNS name servers. TSIG first generates a key (automatically or manually) which the two servers share. Transfer and use of the key is controlled by entries in the servers' configuration files.

A detailed description of the function of TSIG is provided in the "BIND9 Administrator Reference Manual" of the Internet Software Consortium.

DNSSEC

DNS security (DNSSEC) extensions permit cryptographic authentication of the DNS information. They are defined in RFC 2535.

DNSSEC uses public keys for encryption. This enables zone administrators to sign the zone data digitally and authenticate themselves. Communication must be established between the administrators of the parent zone and/or the child zone in order to transfer the keys and signatures.

DNSSEC provides the following tools among other things:

- *dnssec-keygen* for key generation
- *dnssec-signzone* for signing a zone

A detailed description of the function of DNSSEC is provided in the “BIND9 Administrator Reference Manual” of the Internet Software Consortium.

Executing NAMED without root authorization

By default NAMED is executed with root authorization. To prevent an intruder gaining full access to the file system or being able to execute commands under the root ID in the event of any security problems with the software, it is also possible to run NAMED without root authorization. The */etc/default/TCP-IP-SV.named* parameter *USERID* enables you to run NAMED under any user ID. We recommend that you specify the user ID with the lowest privilege for *USERID*.

The user ID defined in *USERID* must be assigned read and write permission for the working files for which the entries *directory* and *pidfile* were agreed in the configuration file *named.conf*.

7.3.3 NAMED - administration and operation

Administration and operation of NAMED comprise the following tasks:

- Starting up and shutting down NAMED
- Modifying NAMED zone data

7.3.3.1 Starting up and shutting down NAMED

NAMED is automatically started up when the POSIX subsystem is started and shut down when POSIX is shut down if it is activated by AUTOSTART='yes' in */etc/default/TCP-IP-SV.dns*.

Startup call for NAMED

The startup call for the NAMED daemon is:

```
/etc/init.d/TCP-IP-SV.named start
```

Initially a check is made to see whether AUTOSTART is set to 'yes'. If it is not, no further action is executed. If it is set to 'yes', a check is made to see whether a daemon has already been started. If this is the case, only a corresponding message is output, and if it is not the case the daemon is started.

If the daemon is to be started in any case, irrespective of AUTOSTART, the *mstart* option must be used instead of *start*.

The *dstart* option enables the daemon to be executed in the foreground for debugging purposes; the debugging options in the *TCP-IP-SV.named* script may need to be adjusted.

With the *USERID* parameter in the */etc/default/TCP-IP-SV.named file* you define whether NAMED is to run without root authorization (see page 242).

Restart call for NAMED

A restart call is offered for NAMED. This is necessary when a modified configuration file is to be read in during ongoing operations.

The restart call for NAMED is:

```
/etc/init.d/TCP-IP-SV.named restart
```

When the restart procedure is executed, a check is made to see whether the corresponding daemon has been started. If no active daemon is found, a normal restart takes place.

Shutting down NAMED

The following call is available to stop the NAMED daemon:

```
/etc/init.d/TCP-IP-SV.named stop
```

CAUTION!

The shutdown only applies until the POSIX subsystem is terminated. If an automatic restart is to be prevented when the POSIX subsystem is restarted, the AUTOSTART variable in */etc/default/TCP-IP-SV.named* must be set to 'no' or 'never'.

7.3.3.2 Modifying the zone data files of NAMED

If desired, the configuration of the NAMED server can be changed outside the current server session by modifying the NAMED zone data files and starting NAMED again.

In addition, it is also possible to change the configuration via so-called dynamic updates. In this case, Resource Records can be dynamically added, deleted or modified during the current server session.This feature will play an important role in the future, especially in connection with DHCP.

Dynamic Update

Dynamic update is the term used for the ability under certain specified conditions to add, modify or delete records or RRsets in the master zone files. Dynamic update is fully described in RFC 2136.

Dynamic update is enabled on a zone-by-zone basis, by including an *allow-update* or *update-policy* clause in the *zone* statement.

Updating of secure zones (zones using DNSSEC) follows RFC 3007 and subsequent RFCs: SIG and NXT records affected by updates are automatically regenerated by the server using an online zone key. Update authorization is based on transaction signatures and an explicit server policy.

The journal file

All changes made to a zone using dynamic update are stored in the zone's journal file. This file is automatically created by the server when the first dynamic update takes place. The name of the journal file is formed by appending the extension *.jnl* to the name of the corresponding zone file. The journal file is in a binary format and should not be edited manually.

The server will also occasionally write ("dump") the complete contents of the updated zone to its zone file. This is not done immediately after each dynamic update, because that would be too slow when a large zone is updated frequently. Instead, the dump is delayed by 15 minutes, allowing additional updates to take place.

When a server is restarted after a shutdown or crash, it will replay the journal file to incorporate into the zone any updates that took place after the last zone dump.

Changes that result from incoming incremental zone transfers are also journaled in a similar way.

The zone files of dynamic zones cannot normally be edited by hand because they are not guaranteed to contain the most recent dynamic changes - those are only in the journal file. The only way to ensure that the zone file of a dynamic zone is up to date is to run *rndc stop*.

If you have to make changes to a dynamic zone manually, the following procedure will work:

- shut down the server using *rndc stop* ().

  sending a signal or using *rndc halt* is not sufficient

- wait for the server to exit
- remove the journal file,
- edit the zone file,
- and restart the server.

CAUTION!

Removing the journal file is necessary because the manual edits will not be present in the journal, rendering it inconsistent with the contents of the zone file.

7.3.4 NAMED - diagnosis and maintenance

This section describes logging functionality and diagnostic options for NAMED.

7.3.4.1 NAMED - logging

The DNS name server NAMED enters its logging information in the */var/adm/syslog* file. In the configuration file */etc/named.conf* you can use the *logging* function to define the categories for which logging is to be performed, in other words on which channels the logging information is to be output.

By default, all messages with a severity from "Info" to "Critical" are stored via SYSLOG in the file */var/adm/syslog*. Messages of the categories "Packet" and "Eventlib" are exceptions. These messages as well as all debug messages are stored in the file *named.run* in the startup directory of the NAMED daemon.

The following information is contained in the entries of the NAMED daemon in the logging file *var/adm/syslog*:

– Date and system time
– Keyword for classification, which corresponds to the message priority (severity)
– Name of the NAMED daemon and process ID (PID)
– Logging category
– Actual meaning

Example

```
Jan 04 09:42:06 LOG_INFO named[1064]: load: master zone "test.mch.fj.example"
(IN) loaded (serial 36)
```

The following information is contained in the entries of the NAMED daemon in the *named.run* file:

– Date and system time
– Logging category
– Actual meaning

Example

```
30-Jan-2010 11:14:12.574 load: info: master zone "test.mch.fj.example" (IN)
loaded (serial 36)
```

Further information on logging is provided in the section “Configuring NAMED” on page 237.

7.3.4.2 NAMED - diagnostic options

All activities of NAMED are logged in debug mode. You can enable this mode by setting the environment variable *DEBUGNAMED* or via the *rndc* tool. The volume of diagnostic information depends on the debug level. The higher the level, the more detailed the messages. You can transfer the level directly in the environment variable *DEBUGNAMED* or set it with the *rndc* tool. You can also disable debug mode with the *rndc* tool.

A further diagnostic option is provided by the database dump. You can use the *rndc* tool to have the data in the cache and the root data output to file. You can also enable query logging with the *rndc* tool.

The *rndc* tool is described in the “BIND9 Administrator Reference Manual” of the Internet Software Consortium.

7.4 DNS tools

A number of tools are available for diagnostics, administration and monitoring of NAMED. These tools are listed below. You will find a description of them on your server under */opt/TCP-IP-SV/dns /readme* or under */opt/TCP-IP-SV/dns-named/readme*.

Diagnostic tools

- **dig** (domain information groper)

  Command line tool for retrieving information on the domain name servers

- **host**

  Command line tool for querying Internet host names

- **nslookup**

  Command line tool for querying domain name servers on the Internet. *nslookup* will not be developed any further. In future the *dig* tool should be used instead.

Administration tools

- **rndc** (remote name daemon control)

  Tool for monitoring name server operation

- **rndc-confgen**

  Utility for generating the *rndc.conf* file

A detailed description of these tools is provided in the “BIND9 Administrator Reference Manual” of the Internet Software Consortium.

7.4.1 Diagnostic tool dig - examples

A number of examples of how to work with the diagnostic tool dig is provided below.

Querying the address for a name (server address from resolv.conf)

```
# dig ts.fujitsu.com

; <<>> DiG 9.7.1 <<>> ts.fujitsu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27805
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ts.fujitsu.com.                        IN      A

;; ANSWER SECTION:
ts.fujitsu.com.         86400   IN      A       217.115.66.11

;; Query time: 2 msec
;; SERVER: 123.123.123.123#53(123.123.123.123)
;; WHEN: Wed Nov 24 15:52:42 2010
;; MSG SIZE  rcvd: 48
```

Querying the name for an address (server address from resolv.conf)

```
# dig -x 217.115.66.11

; <<>> DiG 9.7.1 <<>> -x 217.115.66.11
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47949
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;11.66.115.217.in-addr.arpa.    IN      PTR

;; ANSWER SECTION:
11.66.115.217.in-addr.arpa. 202 IN      PTR     ts.fujitsu.com.

;; Query time: 7 msec
;; SERVER: 123.123.123.123#53(123.123.123.123)
;; WHEN: Wed Nov 24 15:56:31 2010
;; MSG SIZE  rcvd: 72
```

Querying the name server of a DNS domain (server address from resolv.conf)

```
# dig ts.fujitsu.com ns

; <<>> DiG 9.7.1 <<>> ts.fujitsu.com ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46920
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 5

;; QUESTION SECTION:
;ts.fujitsu.com.                        IN      NS

;; ANSWER SECTION:
ts.fujitsu.com.         86400   IN      NS      ns2.klute-thiemann.de.
ts.fujitsu.com.         86400   IN      NS      ns3.klute-thiemann.de.
ts.fujitsu.com.         86400   IN      NS      ns4.klute-thiemann.de.
ts.fujitsu.com.         86400   IN      NS      ns1.ts.fujitsu.com.
ts.fujitsu.com.         86400   IN      NS      ns1.klute-thiemann.de.

;; ADDITIONAL SECTION:
ns2.klute-thiemann.de.  77674   IN      A       82.139.223.161
ns3.klute-thiemann.de.  78837   IN      A       81.92.4.138
ns4.klute-thiemann.de.  78837   IN      A       217.160.130.182
ns1.ts.fujitsu.com.     86400   IN      A       80.70.172.154
ns1.klute-thiemann.de.  77674   IN      A       217.194.235.1

;; Query time: 10 msec
;; SERVER: 123.123.123.123#53(123.123.123.123)
;; WHEN: Wed Nov 24 15:57:20 2010
;; MSG SIZE  rcvd: 219
```

Querying all entries for a name (server address from resolv.conf)

```
# dig www.mycompany.com any
```

```
# dig ts.fujitsu.com any

; <<>> DiG 9.7.1 <<>> ts.fujitsu.com any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49007
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 11, AUTHORITY: 0, ADDITIONAL: 6

;; QUESTION SECTION:
;ts.fujitsu.com.                        IN      ANY

;; ANSWER SECTION:
ts.fujitsu.com.         86400   IN      A       217.115.66.11
ts.fujitsu.com.         86400   IN      NS      ns3.klute-thiemann.de.
ts.fujitsu.com.         86400   IN      NS      ns4.klute-thiemann.de.
ts.fujitsu.com.         86400   IN      NS      ns1.ts.fujitsu.com.
ts.fujitsu.com.         86400   IN      NS      ns1.klute-thiemann.de.
ts.fujitsu.com.         86400   IN      NS      ns2.klute-thiemann.de.
ts.fujitsu.com.         900     IN      SOA     ns1.ts.fujitsu.com.
dns.ts.fujitsu.com. 2373 86400 3600 777600 86400
ts.fujitsu.com.         86400   IN      MX      50 dgate30.ts.fujitsu.com.
ts.fujitsu.com.         86400   IN      MX      10 dgate10.ts.fujitsu.com.
ts.fujitsu.com.         86400   IN      MX      10 dgate20.ts.fujitsu.com.
ts.fujitsu.com.         86400   IN      TXT     "v=spf1 +mx
include:spf.ts.fujitsu.com include:spf2.ts.fujitsu.com
include:spf3.ts.fujitsu.com include:_spf.muc.ec-messenger.com ~all"

;; ADDITIONAL SECTION:
ns3.klute-thiemann.de.  78792   IN      A       81.92.4.138
ns4.klute-thiemann.de.  78792   IN      A       217.160.130.182
ns1.ts.fujitsu.com.     86400   IN      A       80.70.172.154
ns1.klute-thiemann.de.  77629   IN      A       217.194.235.1
ns2.klute-thiemann.de.  77629   IN      A       82.139.223.161
dgate30.ts.fujitsu.com. 86400   IN      A       195.127.188.205

;; Query time: 10 msec
;; SERVER: 123.123.123.123#53(123.123.123.123)
;; WHEN: Wed Nov 24 15:58:04 2010
;; MSG SIZE  rcvd: 508
```

Example of zone transfer (with specification of an explicit server address)

```
# dig @123.123.123.123 mydom.de axfr

; <<>> DiG 9.7.1 <<>> @123.123.129.15 dom1.sq axfr
; (1 server found)
;; global options: +cmd
mydom.de.               604800  IN      SOA    mydom-ns.mydom.de.
root.mydom.de. 42 172800 1
4400 3628800 604800
mydom.de.               604800  IN      MX 0   mx-host0.mydom.de.
mydom.de.               604800  IN      MX 1   mx-host1.mydom.de.
mydom.de.               604800  IN      MX 2   mx-host2.mydom.de.
mydom.de.               604800  IN      NS     mydom-ns.mydom.de.
mydom-ns.mydom.de.      604800  IN      A      123.123.123.101
mydom1.mydom.de.        604800  IN      NS     mydom1-ns.mydom1.mydom.de.
mydom1-ns.mydom1.mydom.de. 604800 IN    A      123.123.123.102
mydom2.mydom.de.        604800  IN      NS     mydom2-ns.mydom2.mydom.de.
mydom2-ns.mydom2.mydom.de. 604800 IN    A      123.123.123.103
myhost.mydom.de.        604800  IN      A      123.123.123.201
myhost.mydom.de.        604800  IN      A      123.123.123.202
myhost.mydom.de.        604800  IN      A      123.123.226.203
myhost.mydom.de.        604800  IN      AAAA
3ffe:1:1000:2000:abc:abcd:abcd:abc1
myhost.mydom.de.        604800  IN      AAAA
3ffe:1:1000:2000:abc:abcd:abcd:abc2
myhost.mydom.de.        604800  IN      AAAA
3ffe:1:1001:3000:bcd:bcde:bcde:bcd2
myhost-alias.mydom.de.  604800  IN      CNAME  myhost.mydom.de.
myhost1.mydom.de.       604800  IN      A      123.123.123.201
myhost1-aaaa.mydom.de.  604800  IN      AAAA
3ffe:1:1000:2000:abc:abcd:abcd:abc1
...
...
3628800 604800
;; Query time: 9 msec
;; SERVER: 123.123.129.15#53(123.123.129.15)
;; WHEN: Wed Nov 24 15:59:31 2010
;; XFR size: 69 records (messages 1, bytes 2056)
```

8 NTP

The Network **T**ime **P**rotocol Version 3 is documented in RFCs 5905-5908.

8.1 NTP concept

The Network Time Protocol is based on the client/server principle. It uses highly-developed methods to provide the time to systems on LANs and/or WANs with an accuracy within the millisecond range.

8.1.1 NTP functionality

NTP offers the following functionality:

- It allows a reference time (Universal Coordinated Time, UTC) to be distributed within a network, and
- also coordinates the clocks within networks of any size.

In order to feed the UTC time into the network, it is recommended that the network computer on which NTP is installed be equipped with a radio-controlled hardware clock which receives a time signal generated by an atomic clock. With its DCF77 time signal, the authoritative source for disseminating time in Germany is the Federal Institute for Physical Technology in Braunschweig (Physikalisch-Technische Bundesanstalt in Braunschweig).

A computer equipped in this way synchronizes its own system clock with the external hardware clock, thus becoming the primary time server on the network. The precision of this time server is determined by the resolution of its own system clock.

Time servers have hierarchical relationships. A secondary time server receives its time via the network from a primary time server. The position of this time server in the hierarchy is identified by a number called the stratum. A primary time server has a stratum of 1, a secondary time server has a stratum of 2, etc. The higher the stratum, the further the time server is from the primary time source and the higher the probability that this time server is inaccurate.

In simple terms, time is synchronized between the client and the server as follows:

1. The client sends an NTP message to the server as a datagram.
2. The server replaces the IP addresses and some of the message fields in the arriving datagram and sends it back.
3. The server sends the modified datagram back to the client.

The result of this process is four timestamps from which the following two measurements can then be calculated:

- the time the NTP message spent travelling through the network (delay)
- the time difference between the clocks of the two computers (offset)

Both measurements are approximated. The offset contains an average delay, i.e. NTP assumes that the arrival route and the return route of the NTP packets are the same length. Any deviations therefore cause errors in the calculation of the offset. In LANs, the round trip delay is only a fraction of the offset, so the offset calculation is very precise. However, this does not apply to WANs.

To minimize the effect of runtime fluctuations, the calculated offset and delay values are put through an additional filtering procedure. Of the last eight received NTP messages of a time server, the message with the lowest delay determines the current offset value. An additional measurement is also calculated from the same list - the dispersion. This is the weighted mean value of the offset deviations of the last eight NTP messages from the current offset value. The offset values with a low delay value are given a greater weight than those with large delay values. Dispersion is the measurement used to evaluate the quality of a time server.

A client/server configuration is entirely dependent on the availability of its components. In NTP, two major techniques are used to prevent any possible sources of failure:

- Redundancy:

  Each time server or each client is in contact with at least three other time servers of the same or a higher stratum. If a computer’s current synchronization source fails, another time server automatically takes over.

- Selection:

  A selection mechanism is applied, in which the best time server of those available in the list is chosen as the current synchronization source.

  The selection criterion are:

  – stratum (the lower, the better)
  – delay (the smaller, the better)
  – dispersion (the smaller, the better)

(Time) servers and clients are differentiated according to who receives the correct time from whom. The client requests an NTP message with the already described timestamps from the server and uses this to synchronize its own clock if the server proves to be the best available. In the NTP protocol, the five different operating states in which a time server operates are defined in terms of "associations" between time servers.

Association	**Host1**	**Host2**
Peer to Peer	Symmetrically active	Symmetrically passive
Client to Server	Client	Server
	Server	Client
Broadcast	Client	Broadcast server
	Broadcast server	Client

Symmetrical associations (Peer to Peer) between time servers use the full range of functions of the NTP protocol. In particular, a peer host receives status information about the corresponding peer and is prepared, if necessary, to use it as a synchronization source. In client/server associations, by contrast, the roles are fixed. The server supplies the client with the precise time, but is never ready to synchronize with the client. Symmetrical associations are usually between time servers with low stratum values, whereas client/server associations are normally between time servers with high stratum values. Broadcast associations can be used in LANs in which a server provides several clients with NTP messages in the broadcast procedure.

Not every client requiring the time service should be synchronized directly with the stratum-1 server(s), as this would overload the server and prevent it from providing a precise time. It is better to distribute the time among a group of selected stratum-2 servers.

8.1.2 Implementing NTP in BS2000

BS2000 can use the NTP functionality both as a client and as a server.

The *adjtime* function, which is offered as a POSIX interface and as a privileged TPR interface, enables the system time in BS2000 to be modified by a specific value.

The system time is available to the user via the following interfaces:

- GTIME (both in TU and in TPR)
- GDATE
- via the runtime routines of high-level languages

The time-of-day register (TODR) is not adapted as part of one of these synchronizations.

The actual adaptation of the system time is carried out asynchronously with the *adjtime* call in small steps. The change is made by accelerating or delaying the system clock for a certain time, depending on whether the correction value is positive or negative. The technique involved ensures that two consecutive calls for the current time still receive monotonous time values and not jumps in time.

The process of synchronizing the system time by an absolute value of *n* seconds lasts 4**n* seconds. NTP performs its synchronization in BS2000 every 64 seconds with the current offset value.

The time of the *adjtime* call is determined by the following factors:

- Current deviation between the time shown on the own system and the accurate NTP server time. If the deviation is small, the polling interval is increased, thus reducing the synchronization frequency.
- The set *minpoll* value, which specifies the minimum polling interval. The memory requirements of the *adjtime* call must be taken into account here.
- For each *adjtime* call, 8 KB of class-4 memory and KB of class-3 memory are required. The memory areas involved are occupied for 15 minutes each.

  This results in the following requirement for class-4 memory:

  Requirement = (900 / *sync_interval*) * 8 KB

  where *sync_interval* specifies the interval (in seconds) between two synchronization operations initiated by NTP.

When *adjtime* is called, it terminates any previous *adjtime* call that may be running. The functionality of *adjtime* can be requested by several privileged users in BS2000. An internal system of preferences establishes whose synchronization jobs are executed, and whose are not. The priority remains in effect throughout the existence of a higher-priority instance.

The NTP server logs in with its first *adjtime* call and is implicitly logged off by terminating its execution task.

Details on the priority rules are provided in the BS2000 manual " System Administration", chapter "System time administration".

The time server with the highest priority is preferred by BS2000; all other time servers are then ignored. Since this is not checked by BS2000, the system administrator must make suitable organizational arrangements.

By default the */var/adm/ntp.log* and */var/adm/syslog* files are used for logging.

NTP can remain in use even during seasonal time switchovers.

NTP programs

The table below contains an overview of the programs used for starting the NTP daemon and for controlling the NTP functionality:

Program	Function	Refer to
ntpd	NTP daemon	page 261
ntpq	Querying NTP status	page 280
ntpdate	Setting date and time	page 277
ntptrace	Tracing NTP servers	page 290
ntpdc	Querying NTP status (specifically)	
sntp	Client for simplified (SNTP) protocol	
ntp-keygen	Generating public and private keys	page 279

8.2 Installing and uninstalling NTP

Please read the Release Notice supplied with the product in addition to this chapter.

8.2.1 Installation

The different components of the interNet Services software package are installed as a POSIX program package by the POSIX installation program (see the manual “POSIX Basics for Users and System Administrators”). You will find more detailed information on the installation of the components in the section “Installation” on page 35.

For the installation and operation of the TCP-IP-SV component NTP, the PLAM library SINLIB.TCP-IP-SV.*nnn*.NTP must be shareable.

- Install NTP under the TSOS ID with the POSIX installation command because privileged actions also need to be executed:

  ```
  /START-POSIX-INSTALLATION
  ```

- Select the following for installation / specify the following:

Function[1]:	Install packages on POSIX
IMON support ?	Y
name of product	TCP-IP-SV
package of product	NTP
…	

  [1] *version of product* is determined by the highest version in IMON.

- You can change the installation path which is displayed as the procedure continues. However, you are recommended to retain the default */opt/TCP-IP-SV/ntp* as all path names in this document are based on this default setting.

  The POSIX installation program then checks that the /opt file system has sufficient storage space and inquires whether the NTP daemon should later be started automatically each time POSIX starts.

- If you change the default setting "no" to "yes", you can undo this decision later by editing the */etc/default/TCP-IP-SV.ntp* file (AUTOSTART variable).

  After you have answered the question, the files from the PLAM library are imported to the POSIX file system.

Any further activities required after successful installation are carried out in the POSIX shell under a user ID with POSIX root authorization. The POSIX shell is started with the BS2000 START-POSIX-SHELL command for this purpose. The relevant commands and files are in the */etc* and */var* directories and the installation directory */opt/TCP-IP-SV/ntp*. In the table below *<instdir>* stands for the installation directory */opt/TCP-IP-SV/ntp*.

Dirctory	File	Description
<instdir>/etc/rc.d	ntpd	Start/stop script
<instdir>/readme	<diverse>	Original NTP documentation in various file formats
<instdir>/sbin	ntpd	NTP daemon
	ntpdate	NTP program
	ntpdc	NTP program
	ntp-keygen	NTP program
	ntp-wait	Script for synchronization with *ntpd*
	ntptrace	Script for displaying the NTP server chain
	ntpq	NTP program
	sntp	NTP program
	start-ntp-daemon	Auxiliary routine for starting ntpd, cannot be called directly
	ntp.conf.default	Default NTP configuration file
	ntp.drift	Drift file
/etc/default	TCP-IP-SV.ntp	Parameter file for the start/stop script
	TCP-IP-SV.ntp. default	Default parameter file for the start/stop script
/etc/init.d	TCP-IP-SV.ntp	Start/stop script for ntpd
/etc/rc0.d	K17TCP-IP-SV.ntp	Symbolic link to the start/stop script
/etc/rc2.d	S97TCP-IP-SV.ntp	Symbolic link to the start/stop script
/var/run	ntp.pid	File with the pid of ntpd
/var/adm	ntpd_startup.log	Start messages of ntpd
	ntp.log	Logfile of ntpd

After installing the NTP interNet Service and before starting the NTP daemon, the NTP configuration file of the individual components must be customized to individual requirements (see page 263). However, this is only necessary if no backup of the NTP configuration file from a previous installation could be restored.

If the POSIX subsystem is stopped and then restarted, the interNet Services daemons activated after installation are also started automatically.

8.2.2 Uninstallation

The uninstallation of the interNet Services components is carried out using the POSIX installation program under the TSOS ID in the same way as installation. During uninstallation, the active NTP daemon is searched for and terminated. The termination of daemons that are still active is logged in the syslog file */var/adm/syslog*. All files, links and procedures of the NTP are then deleted.

During uninstallation, the */etc/resolv.conf* configuration file and the */etc/default/TCP-IP-SV.ntp* parameter file for the start/stop script are saved in the */etc/tcpipsv/ntp* directory. In the event of reinstallation, the backup copies are copied to the directories */etc* and/or */etc/default* again. Ensure that the backup copy has the current status.

8.3 Startup and shutdown of NTP

This section provides information on the following topics:

- Startup and shutdown of NTP
- NTP time synchronization (set date and time)
- Creating the configuration file of the NTP daemon

8.3.1 Starting and shutting down NTP

If you want to start or shut down NTP manually, under SYSROOT or TSOS use the following script call

```
/etc/init.d/TCP-IP-SV.ntp start
```

or

```
/etc/init.d/TCP-IP-SV.ntp stop
```

This type of startup is possible only if the AUTOSTART variable is set to 'yes' in */etc/default/TCP-IP-SV.ntp*. If, on the other hand, you want to shut down NTP by terminating the POSIX subsystem, set the AUTOSTART variable to 'no' or 'never'.

If the daemon is to be started in any case, irrespective of AUTOSTART, the *mstart* option must be used instead of *start*.

The *dstart* option enables the daemon to be executed in the foreground for debugging purposes; the debugging options in the *TCP-IP-SV.dns* script may need to be adjusted.

Use the script call below to stop and restart any running NTP daemon:

```
/etc/init.d/TCP-IP-SV.ntp restart
```

8.3.2 NTP time synchronization

In BS2000, NTP time synchronization can either be performed continuously using the NTP daemon *ntpd* or in a single operation using the *ntpdate* program. (In the long term the *ntpdate* program is to be replaced by calling *ntpd* with the *-q* option. It is consequently recommendable to use *ntpd*.) How you set the date and time with the *ntpdate* program is described in the section “Setting the date and time via NTP with the *ntpdate* program” on page 277.

ntpd and *ntpdate* both use port number 123. Only one of the applications can therefore run at any given time.

Addressing clocks

In BS2000-POSIX only the system clock of the local computer can be configured as a reference clock. However, if possible the system clock should not be used as the clock, even if it is synchronized with external clocks via the SKP or the carrier system. The reason for this is that in this case synchronization is only accurate to the second. Instead, by means of NTP the same external clocks or partner computers should if possible be used as are used by the carrier system's SKP.

If BS2000 is part of an XCS network (see manual „HIPLEX MSCF“), you should nevertheless configure the system clock as an additional clock using the *server* configuration statement. In doing so, the system clock is assigned stratum 5 by default, which can be increased using the *fudge* statement. If the configured NTP servers can no longer be reached, for example, the local NTP server would remain on the last stratum value reached if the system clock is not configured. If a system clock is configured to stratum 5 as the clock, the local NTP server drops to stratum 6, which means that other time synchronization mechanisms may be used (especially those of the XCS network).

Further information is provided in the chapter "System time administration" in the manual "BS2000 OSD/BC - System Administration".

Clocks are addressed analogously to partner hosts by using pseudo IP addresses, which are invalid as normal IP addresses. The following section describes how clocks are configured using the *server* and *fudge* configuration statements.

8.3.3 Creating the NTP daemon ntpd configuration file

The default name for the configuration file of the NTP daemon *ntpd* is */etc/ntp.conf*. The format of the configuration file for *ntpd* is similar to that of other OSD/POSIX configuration files. Comments are introduced with the # character and continue to the end of the line. Blank lines are ignored.

Configuration statements consist of a keyword, followed by a list of arguments which are separated by spaces. No distinction is made between uppercase and lowercase letters for the keywords of the commands. A statement may not extend over multiple lines. The most important configuration statements are described below.

A complete description is available, for example, in the HTML files under */opt/TCP-IP-SV/ntp/readme/TCP-IP-SV.ntp/html*.

When *ntpd* is used as a broadcast client, no configuration statements are required.

server statement

The *server* statement sets the local *ntpd* daemon to "client" mode for the specified server. In this mode, the local *ntpd* is synchronized by the remote NTP server, but not vice versa.

```
server host_addr [ autokey | key][ version #][ prefer][ minpoll minpoll][
maxpoll maxpoll]
```

---

host_addr
: Host address in dotted notation, which specifies the server for which the local *ntpd* daemon is to be set to client mode.

  Clocks are addressed analogously to partner hosts using pseudo IP addresses of the form `127.127.`*t*`.`*u*, which are invalid as normal IP addresses. *t* identifies the type of clock, and *u* is type-specific. The local system clock is specified with the pseudo IP address `127.127.1.0` and is the only clock possible in BS2000.

autokey
: NTP packets are sent and received on an authenticated basis in accordance with the autokey schema. This option and the *key* option are mutually exclusive.

key
NTP packets are sent and received on an authenticated basis using a schema which is based on symmetrical keys. *key* specifies the key identifier, the range of values extending from 1 through 65534. This option and the *autokey* option are mutually exclusive.

version *#*
Version number for outgoing NTP packets. Possible values for *#* are 1, 2, 3 or 4.
Default: `version 4`

prefer
This option can be used to mark the host specified with *host_addr* as "preferred" and thus select it for synchronization over other candidates that have the same values.

minpoll
Integer that specifies the minimum polling interval for NTP messages as $2^{minpoll}$ seconds.
Default: 6 (64 sec)
Values permitted in BS2000: ≥ 6 (64 sec)

maxpoll
Integer that specifies the maximum polling interval for NTP messages as $2^{minpoll}$ seconds.
Default: 10 (1024 sec)
Values permitted in BS2000: up to 14 (approx. 4 hours, 30 minutes)

Example (client mode)

```
# ntp.conf
#
#
server 127.127.1.0              #own clock
server 172.24.4.121 prefer      #time server
server 172.25.113.36            #unix machine
server 172.25.109.118           #linux pc
```

restrict statement

The *restrict* statement enables access control.

```
restrict address [mask numeric_mask] [flag]
```

---

address
: If *address* is be entered in "decimal dotted notation" then it specifies the IP address of a network, a subnetwork or an individual host. If the *mask* clause is not specified, *numeraddress* refers to the IP address of a single host. Alternatively *address* can specify a host's valid DNS name.

numeric_mask
: *numeric_mask* must be entered in "decimal dotted notation" and must specify a subnet mask.
Default: 255.255.255.255

flag
: The *flag* specification always implies a restriction on access rights. A *restrict* statement without the *flag* entry thus allows unrestricted access rights.

  flag can have the following values (you will find a complete list, for example, in the HTML documentation supplied):

  ignore
  : Packets arriving from hosts specified in the *restrict* statement are ignored. Neither *ntpq*/*ntpdc* queries nor time server polls are answered.

  noserve
  : All NTP packets with a mode not equal to 6 or 7 are ignored. Time service is rejected, but *ntpq*/*ntpdc* queries are allowed.

  notrust
  : The local *ntpd* daemon does not perform any synchronization due to NTP packets received from hosts specified in the *restrict* statement.

Example

```
# ntp.conf
#
#
server   127.127.1.0               #own clock
server   172.24.4.121 prefer       #time server
server   172.25.24.12              #VM2
server   172.25.109.118            #linux pc
#
# access control:
# by default, ignore all packets
#
restrict default ignore
#
# don't trust servers on local net
#
restrict 172.25.0.0 mask 255.255.0.0 notrust
#
# the above defined servers are unrestricted
#
restrict 127.127.1.0               #own clock
restrict 172.24.4.121              #time server
restrict 172.25.24.12              #VM2
restrict 172.25.109.118            #linux pc
```

fudge statement

Clocks can be configured via the *server* statement (see page 263). A further statement for configuring a clock is the *fudge* statement. The *fudge* statement must immediately follow the *server* statement that addresses the clock.

```
fudge 127.127.t.u[ stratum stratum]
```

---

t
: identifies the type of clock.

u
: is type-specific. The local system clock is specified with the pseudo IP address 127.127.1.0 and is the only clock possible in BS2000.

stratum
: Integer that specifies the number of stations via which the local *ntpd* daemon can obtain a high-precision timestamp from an external clock. *stratum* can be significant in the configuration of the local system clock. *stratum* servers of stratum 1 have direct access to external time signals. Servers which receive their timestamps from stratum-1 servers are stratum-2 servers, and so on.

  The value of *stratum* is the major criterion used by *ntpd* to select one server from several as the client. The local system clock is also a "server" for *ntpd*. The default value for its stratum is 5. By specifying a lower or higher value for its stratum in the *fudge* statement, you can thus assign the local system clock a higher or lower preference than the other servers.

  Example (server mode with own clock)

```
# ntp.conf
#
#
server 127.127.1.0           #own clock
fudge  127.127.1.0 stratum 1
```

peer statement

The *peer* statement places the local *ntpd* in “symmetrically active” mode as opposed to the remote server. In this mode:

- the local server can be synchronized by the remote server, and
- the remote server can be synchronized by the local server.

This setting is meaningful in a network of servers in which, for example, either the local or the remote server may have the better time source, depending on the load.

```
peer host_addr [ autokey | key][ version #][ prefer][ minpoll minpoll]
     [ maxpoll maxpoll]
```

---

host_addr
Host address in dotted notation, which specifies the server for which the local *ntpd* daemon is to be set to "symmetrically active" mode. Alternatively the host’s DNS name can also be used.

autokey
NTP packets are sent and received on an authenticated basis in accordance with the autokey schema. This option and the *key* option are mutually exclusive.

key
NTP packets are sent and received on an authenticated basis using a schema which is based on symmetrical keys. *key* specifies the key identifier, the range of values extending from 1 through 65534. This option and the *autokey* option are mutually exclusive.

version *#*
Version number for outgoing NTP packets. The possible values for *#* are 1, 2, 3 or 4.
Default: `version 4`

prefer
This option can be used to mark the host specified with *host_addr* as "preferred" and thus select it for synchronization over other candidates that have the same values.

minpoll
Integer that specifies the minimum polling interval for NTP messages as $2^{minpoll}$ seconds.
Default: 6 (64 sec)
Values permitted in BS2000: ≥ 6 (64 sec)

maxpoll

Integer that specifies the maximum polling interval for NTP messages as $2^{minpoll}$ seconds.
Default: 10 (1024 sec)
Values permitted in BS2000: up to 14 (approx. 4 hours, 30 minutes)

broadcast statement

The *broadcast* statement places the local server *ntpd* in broadcast mode. In this mode, the local server periodically sends broadcast messages to clients with a specific address.

```
broadcast host_addr [ autokey | key] [ version #][ minpoll minpoll]
          [ maxpoll maxpoll]
```

host_addr
: Host address, in dotted notation, which specifies the clients to which the local *ntpd* daemon is to send broadcast messages periodically. The host address can be the IP broadcast address of a local interface or a multicast address. When the host address is a multicast address, the messages are sent to all local interfaces. The IANA has assigned the multicast group addresses IPv4 224.0.1.1 and IPv6 ff05::101 (site local) exclusively to NTP. You can, however, also use other conflict-free addresses.

autokey
: NTP packets are sent and received on an authenticated basis in accordance with the autokey schema. This option and the *key* option are mutually exclusive.

key
: NTP packets are sent and received on an authenticated basis using a schema which is based on symmetrical keys. *key* specifies the key identifier, the range of values extending from 1 through 65534. This option and the *autokey* option are mutually exclusive.

version *#*
: Version number for outgoing NTP packets. The possible values for # are 1, 2, 3 or 4.
Default: `version 4`

minpoll
: Integer that specifies the minimum polling interval for NTP messages as $2^{minpoll}$ seconds.
Default: 6 (64 sec)
Values permitted in BS2000: ≥ 6 (64 sec)

maxpoll
: Integer that specifies the maximum polling interval for NTP messages as $2^{minpoll}$ seconds.
Default: 10 (1024 sec)
Values permitted in BS2000: up to 14 (approx. 4 hours, 30 minutes)

broadcastclient statement

The *broadcastclient* statement instructs the local *ntpd* daemon to synchronize the local system time using incoming NTP broadcasts. In order to calculate the local system time, the message runtime with respect to the server is required. To measure this, a brief data exchange is initiated with the broadcast server when the first broadcast message is received, unless the *broadcastdelay* statement was used to specify a message runtime.

```
broadcastclient
```

broadcastdelay statement

The *ntpd* daemon usually balances out message runtimes between a broadcast server and the client automatically by using a message protocol specially intended for this purpose. If this calibration fails, the *ntpd* uses a default value of 0.004 seconds. The *broadcastdelay* statement can be used to define some other value. The *broadcastdelay* statement enables you to suppress the execution of this protocol and instead to define a value explicitly.

```
broadcastdelay delaytime
```

delaytime
: Specifies the value, in seconds, to offset the message runtime between the broadcast server and client.
Values between 0.003 and 0.007 seconds are appropriate for Ethernet.

driftfile statement

The *driftfile* statement specifies the file containing the precision drift of the local oscillator. This value is used by the local *ntpd* daemon to calculate an adjustment for local frequency fluctuations.

```
driftfile filename
```

---

filename

Specifies the file containing the precision drift of the local oscillator. The file is updated by the current drift value being written to a temporary file which is then renamed to replace the old file. The *ntpd* therefore requires write authorization for the directory containing the file, and you should avoid file links.

If the *filename* file or the *driftfile* statement does not exist, the drift is initially assumed to be 0. The local NTP daemon *ntpd* then calculates the drift and saves it as a floating-point value in multiples of 10^{-6} (parts per million, ppm). This saved value is used with a restart.

controlkey statement

The *controlkey* statement specifies a key ID for use with the *ntpq* program which employs the standard protocol defined in RFC 1305.

```
controlkey key
```

---

key

A key ID in the range from 1 through 65534. This key must be specified as trusted using the *trustedkey* statement.

requestkey statement

The *requestkey* statement specifies a key ID for use with the *ntpdc* program, which employs a proprietary protocol.

```
requestkey key
```

---

key
A key ID in the range from 1 through 65534. This key must be specified as trusted using the *trustedkey* statement.

trustedkey statement

The *trustedkey* statement specifies the key IDs which are to be trusted for the purpose of authenticating NTP peers by means of symmetrical cryptography or for use by *ntpq* and *ntpdc*. The authentication mechanisms require the local and remote servers to use the same key and the same key ID for this purpose. However, different keys can be used with different servers.

```
trustedkey key [...]
```

---

key
A key ID in the range from 1 through 65534.

keys statement

The *keys* statement specifies the complete path of the MD5 key file. This file contains the keys and key IDs which are used by *ntpd*, *ntpq* and *ntpdc* when they work with symmetrical keys.

```
keys keyfile
```

---

keyfile
Complete path of the key file.

keysdir statement

The *keysdir* statement specifies the default path name for autokey keys, parameters and certificates.

```
keysdir path
```

---

path

Path name of the directory containing the autokey files with cryptographic keys, parameters and certificates.

8.3.4 Startup options of the NTP daemon ntpd

ntpd is the NTP daemon process which manages the local time of an OSD/POSIX system, sometimes in conjunction with internet standard servers.

ntpd
[-bdgq] [-c <conffile>] [-D <level>] [-f <driftfile>] [-k <keyfile>] [-l <logfile>] [-p <pidfile>] [-r <broadcastdelay>]

You should normally start the NTP daemon *ntpd* with the */etc/init.d/TCP-IP-SV.ntp* script or with the */etc/rc2.d/S97TCP-IP-SV.ntp* link which points to this script. The script includes the */etc/default/TCP-IP-SV.ntp* file in order to perform preallocations for particular shell variables (which define, among other things, job class, job name, account, *conffile*, *driftfile*, *logfile* and *pidfile*) and calls the */opt/TCP-IP-SV/ntp/sbin/start-ntp-daemon* program.

The latter generates and starts an ENTER job which calls the */opt/TCP-IP-SV/ntp/etc/rc.d/ntpd* script, which then calls the *ntpd*. Changes to the options should therefore be made in */etc/default/TCP-IP-SV.ntp*.

The configuration parameters for *ntpd* are read from a configuration file (default: */etc/ntp.conf*) when the daemon is started. If *ntpd* is used only as a broadcast client, a configuration file is not needed provided the NTP daemon is started using the *ntpd -b* command and not using the */etc/init.d/TCP-IP-SV.ntp* script. Internal variables and/or configuration parameters
of *ntpd* can be displayed with the *ntpq* program.

The local *ntpd* can be configured to one of the following modes with respect to the remote hosts:

– Symmetrically active/passive
– Client
– Broadcast client

A broadcast client automatically recognizes remote servers and calculates time corrections on the basis of the message runtimes between the server and the client. Broadcast clients can be completely configured by means of parameters in the command line.

Options

Further information on the options described below and those which are not listed is available, for example, in the HTML documentation.

-b

ntpd is a broadcast client, i.e. it receives NTP broadcasts and synchronizes the local time accordingly.

-d
Starts *ntpd* in debugging mode. This option can be specified more than once for more extensive debugging messages. If the <logfile> option (see below) is not set, all debugging messages are written to the */var/adm/syslog* file.

-g
Sets *ntpd* to always correct the local system time on the basis of a received timestamp. Without this option, the local system time is corrected only if it differs from the received timestamp by no more than 1000 seconds. You can use this option with the *-q* and *-x* options.

-q
ntpd is terminated after the system clock has been set for the first time. This behavior emulates the *ntpdate* program which will be scrapped in the future. You can use this option with the *-g* and *-x* options.

-c <conffile>
Refers to the name of the configuration file.
Default: */etc/ntp.conf*

-D <level>
Sets the test mode level directly.

-f <driftfile>
Specifies the name of the drift file (see page 272). This has the same effect as the *driftfile* statement.

-k <keyfile>
Specifies the file with the symmetrical keys. This has the same effect as the *keys* statement.

-l <logfile>
Specifies the name of the log file for *syslog* messages.

-p <pidfile>
Specifies the file in which *ntpd* stores its process ID.

-r <broadcastdelay>
ntpd usually balances out message runtimes between a broadcast server and the client automatically. If this is not possible, you can use this option to specify an estimated message runtime.

8.3.5 Setting the date and time via NTP with the ntpdate program

The *ntpdate* program sets the local date and time. *ntpdate* determines the correct date and time by polling the NTP server on the specified server or servers.

ntpdate
[-ds] [-o <version#>] [-p <zeitstempel_anzahl>] [-t <timeout>] [-a key] <server >

ntpdate must be executed on the local computer by the POSIX administrator. Several timestamps are received from each of the specified servers; the most suitable is then chosen with the help of the NTP algorithms for selecting an NTP clock.

ntpdate can be included in a startup script to synchronize the clock during POSIX-START and/or executed regularly via *cron*. Using at least three to four servers rather than one produces better results and reduces the susceptibility of servers to functional failures.

ntpdate can only be executed when the *ntpd* daemon is not running on the same computer.

The *ntpdate* program is to be discontinued in the future. You should therefore consider whether you should switch to using *ntpd* with the *-q* option in good time.

Options

Further information on the options described below and those which are not listed is available, for example, in the HTML documentation.

-d
This option can be used to simulate an *ntpdate* action, i.e. without actually carrying it out. Furthermore, it also provides debugging information.

-s
This option instructs *ntpdate* to log the actions via the *syslog* function (output in */var/adm/syslog*), instead of using the standard output. This option is useful when *ntpdate* is executed by *cron*.

-o <version#>
This option can be used to specify a different version number than the default value of 4 for use in the transmitted NTP packets.
The possible values are 1, 2, 3 or 4.

-p <number_of_timstamps>
This option can be used to set the number of timestamps received by *ntpdate* to values from 1 to 8. The default value is 4.

-t <timeout>
This option can be used to specify the time to wait for an answer. The timeout is rounded up or down to a multiple of 0.2 seconds.
The default value is 1 second.

-a <key>
Activates the authentication function and specifies the key indicator which is to be used for authentication. The keys and the key indicators must match in the client and server key files.

<server>
Specifies one or more computers from which the timestamp is to be requested.

8.3.6 Generating cryptographic files for NTPv4 authentication using the ntp-keygen program

The *ntp-keygen* program generates files with cryptographic material which are used for the NTPv4 authentication mechanisms. It can generate message digest keys which are employed in symmetrical encryption. It also generates host keys, signature keys, certificates and identity keys which are used in "autokey" cryptography with public keys.

You will find a description of the call parameters, for example, in the HTML documentation.

8.4 Administration and operation

The administration and operation of NTP essentially involves querying the NTP status with the NTP program *ntpq*.

The NTP program *ntpq* can be used to query the current status of NTP servers via control messages. It can be optionally executed in interactive mode or controlled via command-line options. When executing *ntpq* interactively with commands, a distinction must be made between the internal commands of *ntpq* and the commands for control messages.

8.4.1 Querying the NTP status via command-line options

ntpq
[-inp][-d][-c <command>] [<server>] [...]

If one or more statements are specified on the command line, each of these statements is sent to the NTP servers running either on the host specified in the command line arguments or, by default, on the local host (*localhost*). If none of the statements is specified, *ntpq* attempts to read commands from the standard input and execute them on the NTP server that is running on the first host specified in the command line or, by default (if no other host is specified), on the local host.

ntpq uses special packets to communicate with the NTP server and can therefore also be used to query any compatible NTP servers on the network.

The command-line options are described below. If a command-line option other than -i or -n is specified, the queries involved are sent directly to the named hosts. Otherwise, *ntpq* tries to read commands from the standard input interactively.

Options

-i
Places *ntpq* in interactive mode. Commands are read from the standard input.

-n
Outputs all host addresses in dotted notation, instead of converting them to host names.

-p
Prints a list of partners known to the server (NTP partner hosts and clocks) as well as an overview of their statuses. This is the equivalent of the interactive command *peers*.

-d
Activates the output of debugging data.

-c <command>[<host>] ...
The following argument is interpreted as a command and added to the list of commands to be executed on the specified hosts. The *-c* option can be specified any number of times.

<server>
Specifies the NTP server for which the current status is to be determined.

8.4.2 Querying the NTP status interactively with commands

The commands for interactive queries consist of a keyword, followed by 0 to 4 arguments. You only have to specify as many characters of the keyword as needed to uniquely identify the command. The output generated by a command is normally written to the standard output; however, you can optionally have the output of individual commands piped to the file *filename* by appending >*filename* to the command.

The following distinction must be made between commands:

- commands that are executed internally by the *ntpq* program itself (*ntpq*-internal commands).
- commands that cause NTP requests to be sent to an NTP server.

8.4.2.1 Querying the NTP status with internal commands of ntpq

The following commands are executed by the *ntpq* program itself.

Show information on command keywords

?[keyword]

A “?” on its own outputs a list of all the command keywords known to *ntpq*. A “?” followed by a command keyword provides information on the function and syntax of the command.

Specify a time limit for answers to server queries

---

timeout milliseconds

The *timeout* command specifies a timeout for answers to server queries (in milliseconds). Since *ntpq* repeats each query on exceeding the timeout, the total waiting period is twice the length of the specified timeout.
Default: 5000 milliseconds.

Specify host name for subsequent queries

---

host hostname

The *host* command specifies the host to which subsequent queries are to be sent. The *hostname* can be specified as either a computer name or a numerical address.

Show host names with ntpq output

```
hostnames yes | hostnames no
```

If *hostname yes* is specified, the host names are shown in the *ntpq* output. If *hostname no* is specified, numerical addresses are shown instead of the host names.

Default: *hostname yes* (unless changed with the *-n* switch in the command line).

Show unformatted answers of the remote server

```
raw
```

The answers of the remote server to the query commands are shown as received. The only formatting/interpretation carried out on the data is the conversion of non-ASCII data to a readable format.

Show formatted answers to query commands

```
cooked
```

The answers to the query commands are interpreted, and the values of the variables recognized by the server are converted to a user-friendly format. Variables that cannot be interpreted by *ntpq* are identified with a “?”.

Set version number

```
ntpversion 1 | ntpversion 2 | ntpversion 3 | ntpversion 4
```

Sets the version number to be used by *ntpq* in NTP packets. Please note that mode 6 control messages (and, of course, the corresponding mode) do not exist in versions prior to NTP Version 2.
Default: `ntpversion 2`

Enable debugging function

```
debug more | debug less | debug off
```

Enables or disables the internal debugging function of the query command.

debug more
: Increments the debug level by 1, thus returning more diagnostic information.

debug less
: Reduces the debug level by 1, thus returning less diagnostic information.

debug off
: Turns off the debugging function.

Exits ntpq

```
quit
```

Exits the *ntpq* program.

8.4.2.2 Querying the NTP status with commands for control messages

Each partner that is known to a server has a 16-bit association identifier consisting of an integer. NTP control messages with partner variables must use this association identifier to identify which partner the values belong to.

By means of commands for control messages, one or more NTP messages are sent to the server, and the returned data is output in partially formatted form. Most of the commands output a single message and also expect a single answer. One exception is the *peers* command, which sends a preprogrammed sequence of messages.

Request and display association IDs and partner statuses

```
associations
```

The *associations* command requests a list of the association IDs and partner statuses for the partners of the queried server, and outputs this list in columns. The first column contains a sequential number for the associations (starting with 1); the second column contains the actual association ID returned by the server, and the third column contains the status word for the partner. These are followed by several columns with data that was decoded from the status word. The data returned by the *associations* command is buffered internally in *ntpq*.

Send request to read a status

```
pstatus assocID
```

The *pstatus* command sends a request to the server to read the status for the specified association ID (*assocID*). The returned names and values of the partner variables are output.

Request a list of the clock variables of the server

```
clockvar[ assocID] [name[,name...]]
-----------------------------------------------------------------------
cv[ assocID] [name[,name...]]
```

The *clockvar* command, or *cv* for short, requests a list of the clock variables of the server. Servers on which a clock is configured react positively to this request. *assocID* specifies the association ID. If the association ID is not specified or is specified as “0”, the request for variables applies to the internal NTP system clock of the server. Otherwise, the association ID is interpreted as the partner association ID of a clock, and the related variables are displayed. If no variable list is specified, the server returns a standard list of variables and values.

Request a list of the server’s peers, including additional information

```
peers
```

The *peers* command requests a list of the server’s partners, including an overview of the status of each partner. The information in the overview includes:

– the address of the remote partner,
– the reference ID (0.0.0.0 if the reference ID is unknown),
– the stratum of the remote partner,
– the type of partner (*local*, *unicast* or *broadcast*),
– an indication of when the last packet arrived,
– the polling interval (in seconds),
– the accessibility register (as an octal number),
– the current message runtime,
– the drift and variance of the partner (both in milliseconds).

The character in the left margin indicates the significance of this partner in the clock selection process.

The meanings of the codes are as follows:

sp
deleted because of a high stratum and/or negative plausibility checks

x
time is invalid since the drift is too large

.
selected from the end of the candidate list

-
deleted from the cluster algorithms

+
added to the final selection group

#
selected for synchronization, but the synchronization distance exceeds the maximum

*****
selected for synchronization

o
selected for synchronization; external clock

The host field may contain either a host name, an IP address or the name of a clock.
If *hostnames no* (see also page 283) is specified, only IP addresses are displayed.

Specifying a key ID for authenticating configuration requests

---

keyid keyid

The *keyid* command specifies a key ID for authenticating configuration requests. The key number specified must match a key number which was configured on the server for this purpose.

Having the password inquired for authenticating configuration requests

passwd

The *passwd* command inquires a password which must match a key that was configured on the server for this purpose.

8.5 Diagnosis and maintenance of NTP

The following sections provide you with information on the logging function and trace functionality of NTP.

8.5.1 Logging function

The NTP components store their logging information in the */var/adm/syslog* file.

The ntp entries of this logging file have the following basic format:

```
<date> <time> keyword ntpd[pid]: <message text>
```

The date and time is followed by a keyword that classifies the message.

The NTP uses the keywords LOG_INFO, LOG_NOTICE, LOG_DEBUG and LOG_ERR.

The classification is followed by the process name (here *ntpd*) with information on the relevant process ID (PID). If no valid PID is available for the message output (e.g. for a message output during the daemon startup procedure), an empty parenthesis expression is displayed. The actual message text then follows.

Furthermore, logging entries are by default written to the */var/adm/ntp.log* file.

8.5.2 Trace functionality of NTP

The trace functionality of NTP is provided by the Perl script *ntptrace*. Perl must be installed to enable this functionality to be used.

8.5.2.1 ntptrace - Trace a chain of NTP servers back to the prevailing clock

ntptrace
-n -m <maxhosts> [<server>]

ntptrace determines the clock to which a specific NTP server refers for its time and traces the chain of NTP servers back to the prevailing clock. If no arguments are specified, the command begins with *localhost*.

Example of an *ntptrace* output:

```
% ntptrace
localhost: stratum 4, offset 0.0019529, synch distance 0.144135
server2.bozo.com: stratum 2, offset 0.0124263, synch distance 0.115784
usndh.edu: stratum 1, offset 0.0019298, synch distance 0.020993, refid 'WWVB'
```

The fields (from left to right) in each line are: the host name, the host stratum, the time offset between this host and the local host (as calculated by *ntptrace*; this value is therefore not always 0 for *localhost*), the host's synchronization distance and (only for servers of stratum 1) the ID of the external clock. All times are specified in seconds. (The synchronization distance is a measure of the message runtime to the partner.)

Options

-n
IP addresses are specified instead of the host names. This may be necessary if no name server is running.

-m <maxhosts>
Specifies a maximum length for tracing the chain of NTP servers.
Default: 99

<server>
Specifies the server from which the chain to the prevailing clock should be traced back.
Default: *localhost*

9 OpenSSH

This chapter is based on the manual pages (man pages) of OpenSSH and describes the OpenSSH server. The OpenSSH client is described in the interNet Services User Guide.

The description provided here has been shortened to contain only the parts relevant to BS2000. At certain places in the chapter, such as in the descriptions of options, the OpenSSH man pages are referred to as only these contain the description of the most up-to-date versions. You can find the OpenSSH man pages on the Internet at *http://www.openssh.org/manual.html* or, after you have installed the component OpenSSH on your server,

- under *<installationpath>/readme/TCP-IP-SV.openssh/html/* as an HTML file,
- under *<installationpath>/readme/TCP-IP-SV.openssh/pdf/* as a PDF file,
- under *<installationpath>/readme/TCP-IP-SV.openssh/text/* as a text file.

The default installation path is: */opt/TCP-IP-SV/openssh*

When the “OpenSSH man pages” are referred to in the course of this chapter, these sources are meant. The man pages supplied with the product should preferably be used as these man pages contain the BS2000-specific adaptations (changed path names, extended functionality, etc.).

SSH (**S**ecure **Sh**ell) is a cryptographic protocol for performing the following tasks:

- Login on a remote computer
- Interactive / non-interactive command execution on a remote computer
- File transfer between different computers in a network

SSH designates not just the protocol itself but also concrete implementations.

Programs such as *telnet*, *rlogin*, *rsh* and *rcp* do cover the range of tasks mentioned, but they have considerable security gaps. Thus, for example, all the communication, including passwords, is generally transferred unencrypted.

SSH guarantees cryptographically secure communication over insecure networks and offers comprehensive security through

- reliable, mutual authentication of the communication partners,
- integrity and confidentiality of the data exchanged.

One of the intentions in developing SSH was to replace the r utilities *rlogin*, *rsh* and *rcp*. SSH is available in the protocol versions SSH 1 and SSH 2.

OpenSSH is a free version of SSH (i.e. there is no license fee). OpenSSH porting in accordance with POSIX is based on the OpenSSH V7.3 manual at the time of publication. As the SSH protocol versions 1.3 and 1.5 are only partially supported by this OpenSSH version and this partial support will cease entirely in the foreseeable future, the description below is based on circumstances that apply only when using SSH version 2.

9.1 Concept of OpenSSH

OpenSSH is the secure alternative to the r utilities *rlogin*, *rcp* and *rhs*, and the programs *telnet* and *ftp*. In contrast to the aforementioned programs, OpenSSH encrypts all network traffic (including passwords) and thus prevents eavesdropping, connection hijacking, and other attacks at network level. Furthermore, OpenSSH supports a raft of tunneling variants and a wide range of authentication methods.

9.1.1 Component parts of the OpenSSH protocol suite

The OpenSSH protocol suite comprises the following programs and commands:

- On the server side: Server program *sshd* (see page 300)
- On the client side (see the interNet Services User Guide):
  - Client program *ssh* or *slogin*: replaces *rlogin* and *telnet*.
  - *scp*: replaces *rcp*.
  - *sftp*: replaces *ftp*.
- Administration utilities (see the interNet Services User Guide):
  - *ssh-agent*
  - *ssh-add*
  - *ssh-keygen*
  - *ssh-keyscan*

To permit secure communication of Windows systems with BS2000 via SSH, the open source software PuTTY, for example, provides the functionality of the client side of OpenSSH.

9.1.2 Network security with OpenSSH

OpenSSH protects against the following threats to network security:

- IP spoofing

  In the event of IP spoofing a remote computer sends packets with a counterfeited sender address. OpenSSH even provides protection against a spoofer in the local network who claims to be your router for outgoing messages.

- DNS spoofing

  In the event of DNS spoofing an attacker falsifies the Resource Records (RR) in the DNS Name Server.

- Connection hijacking
- Eavesdropping

  Unauthorized tapping into unencrypted passwords and other plaintext messages.

- Data corruption

OpenSSH protects against unauthorized reading and data corruption by encrypting the network traffic. OpenSSH prevents IP spoofing and DNS spoofing by authenticating the communication partners.

Thus an attacker who has obtained control over the network can only force the disconnection of OpenSSH. However, the attacker cannot

- decrypt messages,
- intercept messages and read them in again,
- engage in connection hijacking.

9.1.3 Features of OpenSSH

OpenSSH is characterized by the following features:

- Strong encryption
- Automatic and transparent encryption
- Strong authentication
- Interoperability
- Transmission of binary data and data compression
- Agent forwarding
- TCP forwarding

Strong encryption

OpenSSH supports the encryption algorithms AES, ChaCha20 and 3DES; the support of Blowfish, Cast128 and Arcfour will be presumably omitted soon.

- AES is a high-speed block encryptor. AES satisfies the US Federal Information Processing Standard (FIPS) Advanced Encryption Standard and was developed as a replacement for DES.
- 3DES is a tried and tested encryption algorithm for strong encryption. 3DES is now showing signs of weaknesses because of its short 64 bit block length and will therefore no longer be supported by OpenSSH in the medium term.
- ChaCha20 is a fast stream encryptor to replace Arcfour, which has fallen into disrepute due to security issues.

Automatic and transparent encryption

By default, encryption of all communication between the OpenSSH client and the OpenSSH server is performed automatically and transparently. A symmetrical encryption method is used for this purpose, for example AES oder ChaCha20.

Strong authentication

Authentication of the OpenSSH server to the OpenSSH client is based on the asymmetrical encryption algorithms RSA, DSA , ECDSA and Ed25519. Several methods are available for authenticating the OpenSSH client to the OpenSSH server (see page 303).

Transmission of binary data and data compression

Transmission of binary data via the network is supported. Optional data compression before encryption enhances the performance when transmitting over low-speed network connections.

Agent forwarding

In the case of agent forwarding the authentication agent (see the interNet Services User Guide) which runs on your local computer administers your authentication keys (RSA/DSA/ECDSA/Ed25519). OpenSSH can automatically forward the connection to the authentication agent via any network connection. The authentication keys then only need to be kept on your local computer, but not on any other computer in the network.

Port forwarding (TCP forwarding)

Port forwarding makes insecure TCP/IP connections secure by forwarding (tunnelling) TCP/IP connections to a remote computer using an encrypted protocol. Port forwarding implements mapping of a local port on the client computer onto a port on the remote computer.

9.2 Installing and uninstalling OpenSSH

In addition to this chapter, please also refer to the Release Notice supplied with the product interNet Services.

9.2.1 Installing OpenSSH

The different components of the interNet Services software package are installed as a POSIX program package by the POSIX installation program (see the manual “POSIX Basics for Users and System Administrators”).

For the installation and operation of the OpenSSH suite, the PLAM library SINLIB.TCP-IP-SV.*nnn*.OPENSSH must be shareable.

```
/START-POSIX-INSTALLATION
```

Function: Install POSIX program packages (IMON support: Y)
Product name: TCP-IP-SV
Package name: OPENSSH

If you install OpenSSH using the POSIX installation program, you will first be queried about the installation path <*installationpath*>. It is recommendable to use the default setting */opt/TCP-IP-SV/openssh*. If you specify a different path name, the path */opt/TCP-IP-SV/openssh* is created as the symbolic link to the path name specified by you when the post-installation script is executed.

The POSIX installation program then checks whether the */opt* file system has sufficient memory and reads all files from the PLAM library into the POSIX file system.

Executing the post-installation script

After all files have been read in, a post-installation script is started automatically which handles the computer-specific setup of the OpenSSH components.

In detail, the post-installation script performs the following activities and logs their execution on the console:

1. The directories */etc/tcpipsv/openssh*, */opt/SMAWPlus/etc* and */usr/local/etc* are searched for SSH host key files of an earlier installation of TCP-IP-SV:openssh. The first host key file (if one exists) of each directory is transferred to the */etc/ssh/* directory.

   The following pairs of host key files (private/public keys) are searched for:

   - *ssh_host_rsa_key* and *ssh_host_rsa_key.pub* (RSA)
   - *ssh_host_dsa_key* and *ssh_host_dsa_key.pub* (DSA)
   - *ssh_host_ecdsa_key* and *ssh_host_ecdsa_key.pub* (ECDSA)
   - *ssh_host_ed25519_key* and *ssh_host_ed25519_key.pub* (Ed25519)

2. The */etc/tcpipsv/openssh* directory is searched for the *ssh* and *sshd* configuration files (*ssh_config*, *sshd_config*) of an earlier installation of TCP-IP-SV:openssh. If no configuration files can be found, the configuration file supplied with the product is copied.
3. The */var/run* directory, if it does not exist, is generated as it is required for the process ID file of the OpenSSH server *sshd*.
4. If for installation you specified an installation path other than the default installation path */opt/TCP-IP-SV/openssh*, the default installation path will be changed in the startup scripts to the installation you have specified, and */opt/TCP-IP-SV/openssh* will be set up as the symbolic link to this installation path.
5. If not all host key files were found in step 1), these are now generated. The *ssh-keygen* utility (see the interNet Services User Guide) is called up to four times for this purpose (for RSA, DSA, ECDSA and Ed25519) to generate a non-repeatable, random host key.
6. If no POSIX group with the group ID 22 exists, this is now generated and entered in the */etc/group* file with the group name “sshd”.

7. If no user ID SYSSSHD exists as yet, the post-installation script now creates this. In addition, an address space limit of 32 MB is defined, and a POSIX user ID 22 and a POSIX group ID 22 assigned. For this purpose the post-installation script issues the following commands:

```
/ADD-USER USER-ID=SYSSSHD,ADDR-SP-LIM=32,-
/         ACC-ATT=*P(ACC=SYSACC,CPU-LIM=*MAX,POSIX=*YES),-
/         MAIL-ADDR='Privilege Separation user id for OpenSSH',-
/         LOGON-PASSWORD=${PASSWORD}
/SET-JOB-STEP
/MOD-USER USER-ID=SYSSSHD,ADDR-SP-LIM=32,-
/         ACC-ATT=*M(ACC=SYSACC,CPU-LIM=*MAX,POSIX=*YES),-
/         LOGON-PASSWORD=${PASSWORD}
/MOD-POS-USER-ATTR USER-ID=SYSSSHD,USER-N=22,GROUP-N=22,-
/         DIRECTORY='/var/empty',-
/         PROGRAM='/bin/false',-
/         COMM='Privilege Separation user id for OpenSSH'
```

8. In the startup file */etc/profile* of the POSIX shell the PATH specification is extended to the <*installationpath*>*/bin* directory (and <*installationpath*>*/sbin* for the user with the user ID 0 (SYSROOT or TSOS)). If a C shell (*csh* or *tcsh*) is installed, an analogous procedure is followed using the i*csh* startup file (*/etc/.login* or */etc/csh.login*).
9. If ADDRESS-SPACE-LIMIT < 32 MB is specified for SYSROOT, a warning is issued on the console.
10. If no POSIX-RLOGIN-DEFAULT is set for SYSROOT, a corresponding error message is issued on the console.
11. If no general read permission exists for the installation library (e.g. *SINLIB.TCP-IP-SV.nnn.OPENSSH*), a corresponding error message is issued on the console.

12. If no serious errors are detected, depending on the response to the installation query AutoStartOpenSSH the general POSIX start script for OpenSSH is called which starts the *sshd* daemon under SYROOT. (To call the POSIX start script: `/etc/init.d/TCP-IP-SV.openssh start`)

    The `/etc/init.d/TCP-IP-SV.openssh start` script automatically determines the maximum permissible values for JOB-CLASS, ACCOUNT, START and CPU LIMIT for SYSROOT. Using these values the script starts an ENTER job with the job name SSHLOGIN.

    The *sshd* daemon logs important messages in the */var/adm/syslog* file via the syslog mechanism. In addition, problems which occur when the ENTER job is started are recorded in the */var/adm/opensshd_startup.log* file.

The *sshd* daemon does not need to be configured individually and can thus be started automatically.

9.2.2 Uninstalling OpenSSH

When you begin uninstalling OpenSSH a pre-remove script is executed which saves the modified configuration data (host key files, *modified* configuration files) in the */etc/tcpipsv/openssh/* directory. Unmodified configuration files are not saved. From this directory the files mentioned are once more copied into the configuration directory */etc/ssh/* when installation takes place again.

If you do not wish to use the configuration data of the old installation as a basis when you reinstall OpenSSH at some future time, you must completely remove the content of the */etc/tcpipsv/openssh/* directory:

```
# rm -fr /etc/tcpipsv/openssh
```

9.3 OpenSSH server daemon sshd

sshd (OpenSSH Daemon) is the daemon program for OpenSSH.

9.3.1 Configuring the OpenSSH server daemon sshd

sshd can be configured optionally

- with the aid of command line arguments which you specify when calling *sshd* (see page 301),
- with the aid of a configuration file.

The values specified as command line arguments have priority over the corresponding values in the configuration file.

Configuration file of sshd

By default *sshd* reads its configuration data from the */etc/ssh/sshd_config* file. If you want to use a different configuration file, specify this with the parameter `-f` in the command line when you start *sshd*.

If a SIGHUP hangup signal is received, *sshd* reads the configuration file again and restarts. Here *sshd* is executed under the same name under which it was started, e.g. */opt/TCP-IP-SV/openssh/sbin/sshd*.

In POSIX a restart will only be successful if no *sshd* session is currently active. (An active *sshd* session occupies the TCP/IP port and prevents the restart.)

Syntax of the configuration file

The configuration file of *sshd* must comply with the following syntax:

- In each line the file contains a pair comprising keyword and associated argument or associated argument list:
  - For keywords no distinction is made between upper and lower case.
  - Arguments are case-sensitive.
- Empty lines and lines beginning with “#” are interpreted as comments.

A detailed description of the configuration options is provided in the OpenSSH man pages.

9.3.2 Starting and stopping sshd

By default, *sshd* is started in POSIX as follows:

```
/etc/init.d/TCP-IP-SV.openssh start
```

```
/etc/init.d/TCP-IP-SV.openssh stop
```

In this startup script the *sshd* call is issued with the parameter `-f <config_file>`. This specifies the configuration file from which the *sshd* configuration data is read. After startup, *sshd* waits at port 22 for connection requests from the OpenSSH client.

Alternatively you start *sshd* under SYSROOT with the following command:

```
sshd [-46DdeiqTt] [-b bits] [-C connection_spec] [-c host_cert_file]
     [-E log_file] [-f config_file] [-g login_grace_time]
     [-h host_key_file] [-k key_gen_time] [-o option] [-p port]
     [-u len]
```

A detailed description of the operands is provided in the OpenSSH man pages.

For each incoming connection request *sshd* generates a new child process. These *sshd* child processes implement key exchange, encryption, authentication, command execution and data exchange.

9.3.3 Internal procedure when setting up a connection between sshd and ssh

To enable *sshd* to receive connection requests from *ssh*, the following requirements must be met for *sshd*:

- A host key (RSA, DSA, ECDSA or Ed25519) exists on the computer on which *sshd* is running (server computer). The host key is key pair consisting of a private key and a public key and identifies the server. This requirement is already satisfied by installation in POSIX.

The following activities are performed for every *ssh* connection request:

1. *sshd* sends the public part of its host key and a list of the encryption algorithms it supports to *ssh*. The encryption methods that can be used at present are AES, ChaCha20 or 3DES
2. *ssh* checks whether it recognizes *sshd* by looking to see if a public key is stored for the respective system in the user's file *$HOME/.ssh/known_hosts* or in the file provided centrally by the system administrator */etc/ssh/ssh_known_hosts*, and if this is the case, whether it matches the host key type sent from *sshd* (RSA/DSA/ECDSA/Ed25519):

- If a public key is stored for the respective system and it is identical to the public host key sent from *sshd*, proceed to step 3).
- If a key is stored, but it is not identical to the host key sent from *sshd*, there are two possible causes:

  a) The *sshd* host key has just been generated (harmless variation).

  b) Somebody is currently attempting an active "man-in-the-middle" attack (MITM attack) on the SSH connection.

  In this situation, *ssh* displays a relevant warning and you should only dismiss this if you are sufficiently certain that a) is true.

  If you dismiss the warning, then *ssh* proceeds as though no key is stored.
- If the host key is not contained in these files, *ssh* calculates a fingerprint from the public host key sent from *sshd*. If *ssh* is configured accordingly (*VerifyHostKeyDNS Yes*), *ssh* fetches the host key fingerprint from DNS (if it is stored there) and compares it with the fingerprint calculated. Otherwise, *ssh* displays the calculated fingerprint and asks the user if it is OK.
  - If the two fingerprints do not match, the user answers the question with "no" and *ssh* terminates the connection.
  - If the answer is "yes", *ssh* enters the host key in the file *$HOME/.ssh/known_hosts* so that it is subsequently known and the change of host key, which may signal a potential MITM attack, is noted. This only applies, however, if the *StrictHost-KeyChecking* option in the configuration file *ssh_config* is not set to *yes*. Otherwise, you must add the correct contents to the file *$HOME/.ssh/known_hosts* manually.
    For a detailed description of the *StrictHostKeyChecking* option, see the OpenSSH man pages.

3. *ssh* and *sshd* exchange various data. These files are used, on the one hand, to perform a Diffie-Hellman key exchange to obtain one shared session key. On the other hand, *sshd* also sends data, which it has signed with its private host key. *ssh* can use this signature to determine whether *sshd* is really in possession of the private host key belonging to the public host key transmitted.
4. Both sides use the session key for encrypting all communication in the current session. The rest of the session is encrypted using a symmetrical encryption method. Currently AES, ChaCha20 oder 3DES is used essentially for this purpose. *ssh* selects the encryption algorithm from the list which it received from *sshd* together with the host key (see step 2).

   The session key is also used for ensuring data integrity using a MAC method (message authentication code), which *ssh* also selects from the list provided by *sshd* in step 1. Such methods include *HMAC-SHA2*, *HMAC-SHA1*, or *UMAC* in their various forms.

5. The client and *sshd* now conduct an authentication dialog in which the client proves its identity and authorization to *sshd*.

   Details on client authentication can be found in the section “Authentication between OpenSSH client ssh and server sshd” on page 303.

   sshd authenticates itself implicitly to the client as it can only ascertain the session key generated and encrypted by the client if it knows its private RSA keys.

9.3.4 Authentication between OpenSSH client ssh and server sshd

When using the default value for the *PreferredAuthentication* option in the client configuration file *ssh_config* (see „InterNet Services User Guide“), the OpenSSH client executes the following authentication methods one after the other:

1. host based authentication
2. Public key authentication
3. Password authentication

The methods are applied one after the other until a method has successfully provided authentication or until all methods have failed.

Public key authentication permits the use of RSA, DSA, ECDSA and Ed25519 algorithms. The OpenSSH client signs the session ID (togehter with other data) with its private key *($HOME/.ssh/id_rsa, $HOME/.ssh/id_dsa, $HOME/.ssh/id_ecdsa or $HOME/.ssh/id_ed25519*) and sends the result to the OpenSSH server. The server checks whether the corresponding public key is contained in the *<user home>/.ssh/authorized_keys* file. *<user home>* is the home directory of the user with whose user ID the *ssh* caller wishes to log in. If yes, the server accepts the connection.

The OpenSSH client *ssh* authenticates the server by checking whether a public key is stored for the respective system in the user's file *$HOME/.ssh/known_hosts* or in the file provided centrally by the system administrator */etc/ssh/ssh_known_hosts* and, if this is the case, whether it matches the host key type sent from *sshd* (RSA/DSA/ECDSA/Ed25519).

The *StrictHostKeyChecking* option in the configuration file *ssh_config* controls the behavior of the client in the event that no suitable entry is found in the *known_hosts* files:

– If *no* is returned, the previously unknown host key is entered in *$HOME/.ssh/known_hosts* without requesting confirmation.

– If *ask* is returned, the user is asked whether the host key is to be entered.

– If *yes* is returned, the host key is never entered by the client but must be entered in the respective *known_hosts* file by the user or system administrator instead.

A detailed description of the *StrictHostKeyChecking* options is provided in the OpenSSH man pages.

9.3.5 Login process

After a user has logged in successfully, *sshd* performs the following activities:

1. Depending on whether the user has logged in on a user terminal (tty), *sshd* proceeds as follows:
   - If the user has logged in on a tty and has not entered a command, *sshd* outputs the time of the last login and the contents of the */etc/motd* file. However, a prerequisite here is that the output was not suppressed by means of an option in the *sshd* configuration file or by means of `$HOME/.hushlogin`.
   - If the user has logged in on a tty, *sshd* logs the time of the login.
2. *sshd* checks whether the */etc/nologin* file exists. If the file exists, *sshd* prints out its contents. If the user who logs in does not have root authorization, *sshd* terminates.
3. *sshd* switches to execution mode with normal user privileges.
4. *sshd* sets up a basic runtime environment.
5. *sshd* reads the *$HOME/.ssh/environment* file if this exists and users are permitted to set their environment variables. For information on this see the `PermitUserEnvironment` option in the *sshd* configuration file.
6. *sshd* switches to the user's home directory.

7. If the client is operating in an X11 environment and transfers a valid *$DISPLAY* variable,
   - the user-specific command *$HOME/.ssh/rc* is called, the X11 authentication parameters being transferred via *stdin*, or
   - the global *xauth* program is called. As no X11 environment is available for POSIX, this call fails.
8. *sshd* executes the user shell or the user command.

9.3.6 Files of the OpenSSH server daemon sshd

In addition to the configuration file *sshd_config* (see page 300), the OpenSSH daemon *sshd* uses further files, some of which are described below. A complete overview of all files used by *sshd* can be found in the OpenSSH man pages.

$HOME /.ssh/authorized_keys

This file contains a list of all user public keys which are permitted for RSA authentication (see page 303). The file must be readable for a user with root authorization and should not be accessible for other users.

Using the *AuthorizedKeyFile* option in the *sshd_config* file (see page 300) you can specify another file to handle this function.

A detailed description of the syntax and options of the *$HOME/.ssh/authorized_keys* file is provided in the OpenSSH man pages.

/etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_dsa_key
/etc/ssh/ssh_host_ecdsa_key
/etc/ssh/ssh_host_ed25519_key

These files contain the private sections of the host keys and may only be owned by users with root authorization. Only users with root authorization may read these files. The files may not be accessible to anyone else. Note that *sshd* cannot be started if these files are accessible for the group or for everyone.

/etc/ssh/ssh_host_rsa_key.pub,
/etc/ssh/ssh_host_dsa_key.pub
/etc/ssh/ssh_host_ecdsa_key.pub
/etc/ssh/ssh_host_ed25519_key.pub

These files contain the public sections of the host keys and should be available for all to read, but it should only be possible for users with root authorization to overwrite them. The public sections of the host keys stored in the files should match the corresponding private sections of the host keys in the files */etc/ssh/ssh_host_rsa_key*, */etc/ssh/ssh_host_dsa_key*, */etc/ssh/ssh_host_ecdsa_key* and */etc/ssh/ssh_host_ed25519_key*.

These files perform no major functions. They merely simplify application for the users by enabling users to copy their contents directly into the *ssh_known_hosts* files.

The files are generated automatically during installation using *ssh-keygen* (see the interNet Services User Guide).

/etc/ssh/moduli

This file contains Diffie-Hellman groups which are used for the “Diffie-Hellman Group Exchange”. The file format is described in moduli (5) in the OpenSSH man pages.

/var/empty

chroot directory which is used by *sshd* during privilege separation in the pre-authentication phase. The directory should not contain any files and must be owned by a user with root authorization. The directory may not be owned by a group or a user without root authorization.

/var/run/sshd.pid

This file contains the process ID of the *sshd* which listens for connection requests at the port. If multiple *sshd* daemons are listening at various ports simultaneously, the file contains the ID of the last daemon started. The content of this file is not confidential and may be read by all.

/etc/nologin

If this file exists, *sshd* only permits users with root authorization to log in. The content of the file is shown to everyone who attempts to log in, the login attempts of users without root authorization being rejected. It should be possible for everyone to read the file.

/etc/hosts.allow, */etc/hosts.deny*

This file defines access controls which are performed by TCP wrappers. Further information is available under hosts_access (5) in the OpenSSH man pages.

9.4 BS2000-specific restrictions

When working with OpenSSH in a BS2000 environment, the special aspects described below must be borne in mind.

Use of a user's own resolver library instead of the BCAM host name

To resolve host names, OpenSSH(BS2000) uses neither the BCAM host tables nor the resolver on the BS2000 side which is configured in the *$TSOS.SYSDAT.SOCKETS.nnn.SOC6.RESOLV* or *$TSOS.SYSDAT.LWRESD.nnn.RESOLV.CONF* file, but the DNS resolver library of TCP-IP-SV:DNS(BS2000) (see chapter "DNS" on page 205). As a result the *sshd* daemon is independent of the BCAM version used and behaves in the same way as applications such as TCP-IP-SV:DNS(BS2000) and APACHE(BS2000). Prerequisites here are that the */etc/resolv.conf* file exists and contains the address of at least one valid DNS name server.

When TCP-IP-SV:OPENSSH is installed, a check is made to see whether the */etc/resolv.conf* file exists. The option is also available of taking over any existing configuration from *$TSOS.SYSDAT.LWRESD.nnn.RESOLV.CONF* or *$TSOS.SYSDAT.SOCKETS.nnn.SOC6.RESOLV* to a newly created */etc/resolv.conf*.

The complete DNS name and the complete IPv4 or IPv6 address are also entered in the *lastlog* file so that the place where the remote login took place can be ascertained. Here the host name is used in the notation in which it was supplied by the name server, i.e. normally in lower case. This behavior is compatible with other Unix platforms, but differs from the behavior of the POSIX *rlogin*. The POSIX rlogin always uses BCAM names in upper case.

In the case of *ssh* connections or *ssh-keyscan* calls from POSIX, the behavior described also means that host names which are not entered in the DNS resolver library of TCP-IP-SV:DNS(BS2000) but are entered in the BCAM host table are invisible for *ssh* and *ssh-keyscan* and thus invalid.

In this case you must therefore

- specify the associated IP addresses or
- use a name entered in the DNS resolver library of TCP-IP-SV:DNS(BS2000) or
- enter the BCAM names "manually" in the */etc/hosts* file.

Prompt when the password is empty

Normal Unix systems do not request a password if they use *login* or *slogin* to log into an ID without a password. However, the POSIX *rlogin* requests a password to be entered even for an ID without a password. This behavior does not result in increased security, though, because it is at the same time possible to issue an *rsh* command without a password for the same ID.

Here OpenSSH behaves like the other Unix systems and does not ask for a empty password. However, as a login to IDs without a password is, by default, blocked in OpenSSH, in this case you must set the *PermitEmptyPasswords* directive in the configuration file */etc/ssh/sshd_config* to “yes”.

Upper/lower case in the user name

Unlike in Unix operating system, no distinction is made between upper and lower case in BS2000 and BS2000 POSIX. Thus in BS2000 and OSD/POSIX the user “Username” can log on as “username”, “USERNAME” or “uSeRnAmE”. The name of the user who has logged on is recorded in the */var/adm/utmp* file. You can use the `who` command to have the user name displayed.

Whereas rlogin enters the user name in upper case, OpenSSH specifies the user name in lower case (as is usual in Unix operating systems).

10 Mail servers in POSIX

Sending and receiving electronic mail (e-mail) are among the most important services on the Internet. Here the role of the post offices is taken over by the mail servers, which are also called Mail Transfer Agents (MTA). Mail servers handle transfer of e-mails over the network and ensure they are delivered to mailboxes.

Mail User Agents (MUAs) offer user-friendly interfaces for performing the following tasks:

- Writing and sending e-mails
- Accessing the mailboxes
- Presenting and processing the electronic mail received

Man pages

This chapter is in part based on the manual pages (man pages) for Postfix and IMAP/POP3. The man pages which provide further information are referred to at the appropriate places in the chapter, e.g. in the description of programs and parameters/options. You can find the most up-to-date man pages on the Internet at:

- *www.postfix.org/* (Postfix)
- *www.washington.edu/imap/* (IMAP/POP3)

In addition, the Postfix and IMAP/POP3 man pages will be available on your server after you have installed these components.

The Postfix man pages can be found on your server

- under <*installation-path*>*/readme/ MAIL.postfix/html/* as HTML files,
- under <*installation-path*>*/readme/ MAIL.postfix/pdf/* as PDF files,
- under <*installation-path*>*/readme/ MAIL.postfix/text/* as text files.

(Standard installation path for Postfix: */opt/MAIL/postfix/*)

The IMAP/POP3 man pages can be found on your server

- under <*installation-path*>*/readme/ MAIL.imap/html/* as HTML files,
- under <*installation-path*>*/readme/ MAIL.imap/text/* as text files.

(Standard installation path for IMAP/POP3: */opt/MAIL/imap/*)

Where the Postfix or IMAP/POP3 man pages are referred to in the course of this chapter, these sources are meant. You should by preference use the man pages supplied with the product as these relate to the software version contained in the product.

10.1 Overview

The electronic mail service on the Internet is based on the Simple Mail Transfer Protocol (SMTP), which is defined in RFC 5321. Originally only pure text messages could be transferred, but today the MIME mechanism (Multipurpose Internet Mail Extensions, RFC 2045 through 2049) enables a wide range of formats, e.g. images, to be transmitted. Mail servers which handle the electronic mail service on the basis of the SMTP protocol are also called SMTP servers.

The SMTP server used by the interNet Services in BS2000 is the product Postfix Version 3.1.2. This version applies for the creation time of this manual. Rebasing to a newer version can take place in the context of correction packages. This Open Source SMTP server created by Wietse Venema is characterized in particular by high performance, simple manageabilty, and a high degree of security. In addition, the partial compatibility of Postfix to the Sendmail™ program contained in the interNet Value Edition guarantees simple migration from Sendmail to Postfix. Further details on this are provided in the section “Migration from Sendmail to Postfix” on page 342.

The following protocols are used by the Mail User Agents to access mailboxes on a remote server:

- Internet Mail Access Protocol (IMAP, RFC 2060 etc.)
- Post Office Protocol Version 3 (POP3, RFC 1939)

The IMAP and POP3 servers supplied with interNet Services support support the IPv6 and TLS protocols (see page 325).

To permit e-mails to be sent from BS2000, the interNet Services delivery package contains a simple User Agent, the Mail Sender (see the interNetServices User Guide). To permit automatic processing of received e-mails, interNet Services contains a Mail Reader on an IMAP/POP3 basis. If POSIX-SH is installed, POSIX also contains a simple, local User Agent (Mailx) for sending and processing text messages in BS2000.

10.2 Functionality

An SMTP server or Mail Transfer Agent (MTA) is a mail server for transferring e-mails on the Internet using the Simple Mail Transfer Protocol (SMTP). Here the SMTP server can function as a mail relay or mail end system. A further major function provided by the SMTP server is the option of creating mailing lists and forwarding requests with the aid of aliases. Aliases permit the user part of a local receiver address to be replaced by one or more receiver addresses.

The SMTP server uses the Domain Name Service (DNS) to select a suitable route to an end system.

The SMTP server receives messages either from another SMTP server or from a Mail User Agent (MUA). On the basis of the message's receiver address(es), the message is transferred via a TCP/SMTP connection to another SMTP server and/or a local Mail Delivery Agent (MDA).

Here the local MDA, which stores the messages in special files (mailboxes), is particularly important. The MUA can be used to further process (read, forward, sort, save, reply to, delete) the messages contained in the local mailboxes. In addition, the MUA enables messages to be created and transferred to the local SMTP server for delivery. Some MUAs can also transfer messages to a remote SMTP server via a TCP/SMTP connection.

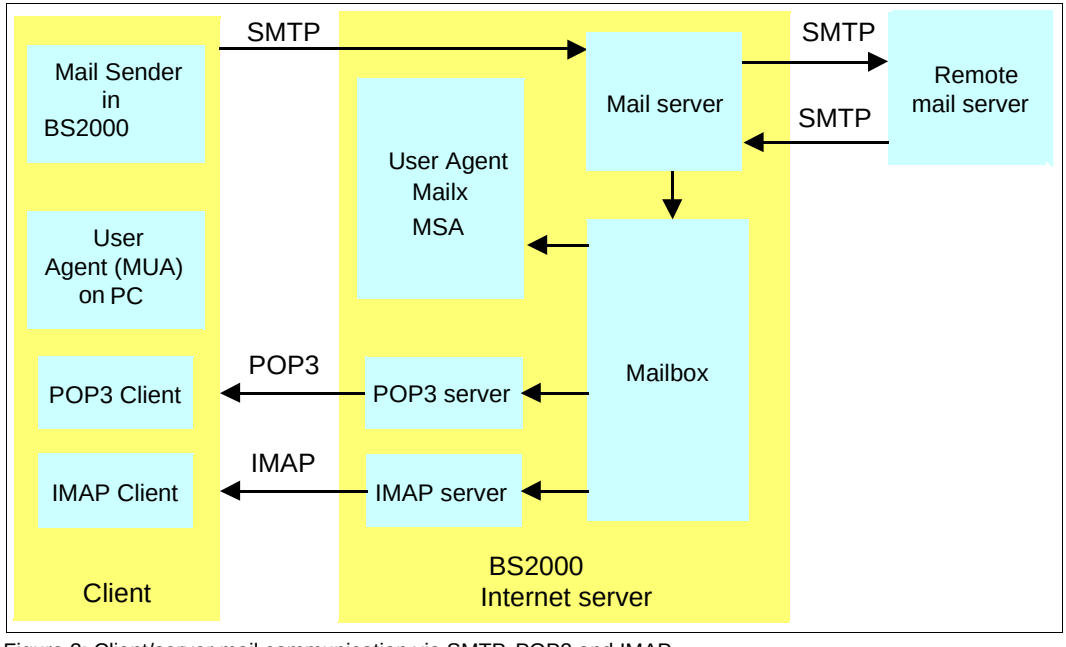


Figure 3: Client/server mail communication via SMTP, POP3 and IMAP

The IMAP and POP3 mechanisms enable a client system, especially if it is not equipped with an SMTP server (MTA), to access the mailboxes of the SMTP server system. For this purpose, an IMAP and a POP3 server are installed on the SMTP server system. For example, clients (e.g. Outlook or Mozilla/Thunderbird) which run on a remote system can then read and delete messages in the server system's mailboxes.

In the case of remote User Agents you must ensure that an account number for accounting a POSIX Remote Login Session exists for the users (BS2000 command /ADD-USER or /MODIFY-USER-ATTRIBUTES POSIX-RLOGIN-DEFAULT=*YES).

IMAP and POP3 differ in the following ways:

- With IMAP, the e-mails remain on the server, where they are also backed up.
- With POP3, the e-mails are by default downloaded to the client and stored there.

Format of the messages

The format of the messages is defined in RFC 822. Messages consist of the message header and the message text. The header and text are separated by precisely one blank line.

Both the message header and the message text consist of readable characters in ASCII format. The message header comprises multiple declarations which in principle are contained in one line, but to enable them to be read more easily can be distributed over several lines. The declarations consist of a name and a text part whose format is defined by the name. Depending on the name, but also on other declarations, a particular type of declaration can occur precisely once, not at all, at most once, or as often as required in a message header.

Most names are defined in RFC 5322. RFC 1522 defines some extensions. Names beginning with "X" are used for private extensions.

The MIME mechanism enables not only pure text but also binary data to be included in the message text. The MIME mechanism defines additional message header declarations and is described in the RFCs 2045 through 2049.

10.3 Installing and uninstalling the mail servers

This section describes installation and uninstallation of the SMPT server (Postfix server) and of the IMAP and POP3 servers.

By default, the mail servers are assigned the following port numbers:

Port		Protocol	Explanation
25	tcp/udp	SMTP	Simple Mail Transfer Protocol
110	tcp/udp	POP3	Post Office Protocol - Version 3
143	tcp/udp	IMAP	Internet Message Access Protocol
993	tcp/udp	IMAPS	IMAP via TLS/SSL
995	tcp/udp	POP3S	POP3 via TLS/SSL

10.3.1 Installing and uninstalling the Postfix server (SMTP server)

You install the Postfix server as a POSIX program package using the POSIX installation program (see the manual “POSIX Basics for Users and System Administrators”).

Installation requirements

To permit the installation and operation of the Postfix server, the PLAM library SINLIB.MAIL.*nnn*.POSTFIX must be available and shareable. The POSIX subsystem must have been started.

The Postfix server requires correctly functioning DNS functionality to provide error-free operation. As the server can be started automatically after installation has been completed, you should ensure that the DNS functionality is functioning correctly before beginning installation. You do this through correct configuration of the */etc/resolv.conf* file and the DNS servers referenced there (see section “Configuring the DNS resolver” on page 222).

When MAIL:POSTFIX is installed, a check is made to see whether the */etc/resolv.conf* file exists. The option is also available of taking over any existing configuration from *$TSOS.SYSDAT.LWRESD.nnn.RESOLV.CONF* or *$TSOS.SYSDAT.SOCKETS.nnn.SOC6.RESOLV* to a newly create */etc/resolv.conf*.

If the DNS functionality is not functioning correctly, installation of the Postfix server can be delayed because of a DNS timeout, resulting in errored behavior of the Postfix server.

Starting installation

The Postfix server must run in privileged mode. You must therefore install the Postfix server using the POSIX installation command under the SYSROOT or TSOS ID (UID=0, GID=0):

- Start installation: `/START-POSIX-INSTALLATION`
- Select the following function: `Install packages on POSIX`
- Specify the following values in the screen which is then displayed:

IMON support?:	`Y`
name of product:	`MAIL`
package of product:	`POSTFIX`

You can modify the installation path displayed for the installation procedure. However, it is advisable to keep to the default */opt/MAIL/postfix*.

If another version of the interNet Value Edition delivery package is installed, the POSIX installation program issues a corresponding error message.

Postfix installation is fundamentally also possible when interNet Value Edition is already installed. However, in this case you must ensure that no *sendmail* daemon is active which contends with the Postfix server with regard to port number assignment. Generally this concerns port number 25.

- Abort Postfix installation by pressing the K2 key if you want to uninstall interNet Value Edition before installing Postfix.
- Continue Postfix installation by pressing the send key if you do not want to uninstall interNet Value Edition beforehand.

The POSIX installation program then checks whether the */opt* file system has sufficient memory space and reads all the files from the PLAM library into the POSIX file system or generates references to PLAM library members. After all files have been read in, the POSIX installation program automatically starts a post-installation script which handles computer-specific configuration of the Postfix server.

Executing the post-installation script

The post-installation script performs the following steps:

1. First the post-installation script checks whether a POSTFIX user ID already exists.

   If this is the case and the two conditions below are also satisfied, an error message is issued and execution of the post-installation script is aborted:

   – POSIX User Number ≤ Default POSIX User Number

   – POSIX Group Number ≤ Default POSIX Group Number

2. If the POSTFIX user ID does not yet exist, the post-installation script sets up this user ID with the following attributes:

   – Random password
   – BS2000 account: SYSACC
   – Address space limit: 32 MB

   Under POSIX this ID is assigned the home directory */var/empty* and the login shell */bin/false* as a login option is neither required nor desirable for the POSTFIX ID.

3. The post-installation script sets up a POSIX group MAILDROP.

4. If the NOBODY user ID does not yet exist, the post-installation script sets up this user ID with the following attributes:

   – Random password
   – BS2000 account: SYSACC
   – Address space limit: 32 MB

   Under POSIX this ID is assigned the home directory */var/empty* and the login shell */bin/false* as a login option is neither required nor desirable for the NOBODY ID.

5. In the startup file */etc/profile* of the POSIX shell, the PATH specification for the user with the user number 0 (SYSROOT or TSOS) is extended by the directories <*installation-path*>/*bin* and <*installation-path*>/*sbin*. If a C shell (*csh* or *tcsh*) is installed, the post-installation script proceeds analogously with the *csh* startup file (*/etc/.login* or */etc/csh.login*).

6. If the default job class for batch jobs of the SYSROOT ID has a JOB-CLASS-LIMIT < 20, a warning is issued.

7. If ADDRESS-SPACE-LIMIT < 32 MB applies for SYSROOT, a warning is issued.

8. If ADDRESS-SPACE-LIMIT < 32 MB applies for POSTFIX, a warning is issued.

9. If no POSIX-RLOGIN-DEFAULT is set for POSTFIX, a corresponding error message is output.

10. If SYSROOT has no read permission for the installation library (e.g. *SINLIB.MAIL.nnn.POSTFIX*), a corresponding error message is output.
11. The post-installation script attempts to determine the system's domain name by inspecting the DNS Resolver configuration files */etc/resolv.conf* and *$TSOS.SYSDAT.SOCKETS.nnn.SOC6.RESOLV* or *$TSOS.SYSDAT.LWRESD.nnn.RESOLV.CONF* .
12. The post-installation script sets the following parameters in the Postfix configuration file */etc/postfix/main.cf* by calling *postconf -e*: *mailq_path*, *newaliases_path*, *daemon_directory*, *readme_directory*, *html_directory*, *command_directory*, *sendmail_path*, *manpage_directory*, *mail_owner*, *setgid_group*, *config_directory* and *queue_directory*. If a domain was found in step 11, the *mydomain* parameter is also set.
13. The post-installation script calls Postfix's own post-installation script */etc/postfix/post-install*.
14. If initial installation of Postfix is concerned, i.e. no configuration backup directory */etc/postfix.sav* for a previous Postfix installation exists, the post-installation script checks whether a */etc/mail/aliases* or */etc/aliases* file exists. If one of these files exists, the post-installation script copies the first file found to */etc/postfix/aliases* and enters the following parameter definitions in the */etc/postfix/main.cf* file:

    ```
    alias_maps = hash:/etc/postfix/aliases
    alias_database = hash:/etc/postfix/aliases
    ```

15. If no serious errors are detected, the post-installation script uses */etc/init.d/MAIL.postfix start* (depending on the answer to the installation query AutoStart-Postfix) to call the general POSIX start script for Postfix, and this starts the Postfix server under SYSROOT.

    The */etc/init.d/MAIL.postfix* script automatically determines the maximum values for JOB-CLASS, ACCOUNT, START and MAXIMUM CPU LIMIT which are permissible for SYSROOT. The script uses these values to start an enter job with the job name POSTFIX.

    The Postfix server logs important messages in the */var/adm/syslog* file using the *syslog* mechanism. In addition, the Postfix server records problems which occur when the enter job starts in the *var/adm/postfix_startup.log* file.

Completing installation

After successful installation, perform the possiblyremaining activities which are required in the POSIX shell under the TSOS ID. To do this, start the POSIX shell under the TSOS ID using the BS2000 command `/START-POSIX-SHELL`.

Files and directories used by the Postfix server

The table below lists the most important files and directories used by the Postfix server. If you use an installation path which differs from the standard Postfix installation path, the path names specified in the table must be adjusted accordingly.

Name	Type	Explanation
/opt/MAIL	Directory	
/opt/MAIL/postfix	Directory	Standard Postfix installation directory
/opt/MAIL/postfix/bin	Directory	Links to Sendmail compatibility program
/opt/MAIL/postfix/bin/mailq	Link	Outputs the mail queue
/opt/MAIL/postfix/bin/newaliases	Link	Updates the alias index file
/opt/MAIL/postfix/libexec/postfix	Directory	Daemon programs
/opt/MAIL/postfix/libexex/postfix/local	Program	Local Delivery Agent
/opt/MAIL/postfix/libexec/postfix/master	Program	Central administration program
/opt/MAIL/postfix/libexec/postfix/pickup	Program	Processes mail selected locally
/opt/MAIL/postfix/libexec/postfix/qmgr	Program	Queue management
/opt/MAIL/postfix/libexec/postfix/smtpd	Program	Receipt of mails using SMTP
/opt/MAIL/postfix/libexec/postfix/smtp	Program	Forwards mails using SMTP
/opt/MAIL/postfix/readme	Directory	Online documentation
/opt/MAIL/postfix/sbin	Directory	Management programs
/opt/MAIL/postfix/sbin/postalias	Program	Generates/inquires the alias index file
/opt/MAIL/postfix/sbin/postcat	Program	Outputs files in queues
/opt/MAIL/postfix/sbin/postconf	Program	Displays/modifies Postfix parameters
/opt/MAIL/postfix/sbin/postdrop	Program	Writes mails in the maildrop directory for delivery by Postfix
/opt/MAIL/postfix/sbin/postfix	Program	Starts/stops the Postfix system
/opt/MAIL/postfix/sbin/postmap	Program	Generates/inquires the index files
/opt/MAIL/postfix/sbin/postqueue	Program	Queue management by the system administrator
/opt/MAIL/postfix/sbin/postsuper	Program	Grants the system administrator super-user access to queues
/opt/MAIL/postfix/sbin/sendmail	Program	Sendmail compatibility program
/opt/MAIL/postfix/share	Directory	Online documentation in the form of man pages
/etc/postfix	Directory	Directory for Postfix configuration files
/etc/postfix/master.cf	Text file	Configuration for master daemon

Name	Type	Explanation
/etc/postfix/main.cf	Text file	Central Postfix configuration file
/var/spool/postfix	Directory	Directory for queues and lock files
/var/mail	Directory	Directory for mailboxes
/var/mail/USER	Mailbox file	Mailbox for the user USER
/etc/rc0.d/K17MAIL.postfix	Link	Link to start/stop script
/etc/rc2.d/S97MAIL.postfix	Link	Link to start/stop script
/etc/init.d/MAIL.postfix	Script	Start/stop script

Uninstalling the Postfix server (SMTP server)

Uninstall the Postfix server using the POSIX installation command under the SYSROOT or TSOS ID (UID=0, GID=0):

- Start uninstallation: `/START-POSIX-INSTALLATION`
- Select the following function: `Delete packages from POSIX`
- In the subsequent screen display mark the line containing the following values and activate your selection by pressing the send key:

```
Product              Version Package
...                  ...     ...
MAIL                 nnn     POSTFIX
```

- Follow the further instructions issued by the tool.

If Postfix configuration files have been modified since the time installation took place, these files are saved in the */etc/postfix.sav* directory. When POSIX is installed again later, the files in the backup directory */etc/postfix.sav* are used instead of the standard configuration files supplied.

10.3.2 Installing and uninstalling the IMAP and POP3 servers

You install IMAP and POP3 servers as a POSIX program package using the POSIX installation program (see the manual “POSIX Basics for Users and System Administrators”).

Installation requirements

To permit the installation and operation of IMAP and POP3 servers, the PLAM library SINLIB.MAIL.033.IMAP must be available, shareable and read-only. The POSIX subsystem must have been started.

Starting installation

IMAP and POP3 servers must run in privileged mode. You must therefore install IMAP and POP3 servers using the POSIX installation command under the SYSROOT or TSOS ID (UID=0, GID=0):

- Start installation: `/START-POSIX-INSTALLATION`
- Select the following function: `Install packages on POSIX`
- Specify the following values in the screen which is then displayed:

IMON support?:	`Y`
name of product:	`MAIL`
package of product:	`IMAP`

You can modify the installation path displayed for the installation procedure. However, it is advisable to keep to the default */opt/MAIL/imap*.

If another version of the interNet Value Edition delivery package is installed, the POSIX installation program issues a corresponding error message.

IMAP and POP3 installation is fundamentally also possible when interNet Value Edition is already installed. The IMAP and POP3 daemon programs are then replaced by the current versions.

- Abort IMAP/POP3 installation by pressing the K2 key if you want to uninstall interNet Value Edition before installing IMAP/POP3.
- Continue IMAP/POP3 installation by pressing the send key if you do not want to uninstall interNet Value Edition beforehand.

The POSIX installation program then checks whether the */opt* file system has sufficient memory space and reads all the files from the PLAM library into the POSIX file system or generates references to PLAM library members.

Files and directories used by the IMAP and POP3 servers

The table below lists the most important files and directories used by the IMAP and POP3 servers. If you use an installation path which differs from the standard Postfix installation path, the path names specified in the table must be adjusted accordingly.

Name	Type	Explanation
/opt/MAIL	Directory	
/opt/MAIL/imap	Directory	Standard IMAP installation directory
/opt/MAIL/imap/readme	Directory	Online documentation
/opt/MAIL/imap/sbin	Directory	Daemon programs
/opt/MAIL/imap/sbin/imapd	Program	IMAP daemon
/opt/MAIL/imap/sbin/ipop3d	Program	POP3 daemon
/opt/MAIL/imap/share	Directory	Online documentation in the form of man pages
/usr/sbin/in.imapd	Link	Link to IMAP daemon
/usr/sbin/in.ipop3d	Link	Link to POP3 daemon
/etc/imap	Directory	Directory for IMAP/POP3 files (X.509 certificates, private keys)
/etc/imap/MAKE.CERT.sh	Script	Script for generating X.509 certificate requests and X.509 test certificates
/etc/imap/certs	Directory	Directory for X.509 certificates
/etc/imap/private	Directory	Directory for private keys
/etc/inet	Directory	Directory for configuration files (in particular *inetd*)
/etc/inet/services	Text file	Assignment of service names to port numbers
/etc/inet/inetd.conf	Text file	Configuration file for *inetd*
/var/mail	Directory	Directory for mailboxes
/var/mail/USER	Mailbox file	Mailbox for the user USER

Uninstalling the IMAP server and POP3 server

Uninstall the IMAP and POP3 servers using the POSIX installation command under the SYSROOT or TSOS ID (UID=0, GID=0):

- Start uninstallation: `/START-POSIX-INSTALLATION`
- Select the following function: `Delete packages from POSIX`
- In the subsequent screen display mark the line containing the following values and confirm your selection by pressing the send key:

```
Product              Version Package=IMAP
...                  ...     ...
MAIL                  nnn     IMAP
```

- Follow the further instructions issued by the tool.

10.4 Starting up Mail servers

This section deals with the following topics:

- Starting up the Postfix server (SMTP server)
- Starting up the IMAP and POP3 servers
- Use of TLS/SSL-secured connections by Postfix, IMAP and POP3 servers

10.4.1 Starting up the Postfix server (SMTP server)

After the Postfix server has been installed you can adjust the configuration files */etc/postfix/master.cf* and */etc/postfix/main.cf*.

Generally the Postfix server is ready to operate even without any modification to the configuration files as the installation program attempts to determine system-specific parameters, such as the system's domain name, automatically via system calls and by inspecting the DNS Resolver's configuration file */etc/resolv.conf* and to save these parameters in the configuration file *main.cf*. You should therefore store the system's domain name (e.g. *systemx.mycompany.com*) and any DNS aliases on the DNS server concerned before performing installation. To do this, you must configure the */etc/resolv.conf* file accordingly (see the section "Configuring the DNS resolver" on page 222).

If it was possible to perform automatic Postfix configuration successfully, the Postfix server is started automatically depending on the answer to the installation query AutoStartPostfix after installation has been completed.

In the standard configuration the Postfix server operates merely as a mail end system, but not as a mail relay. If the Postfix server is also to be used as a mail relay, you must adapt the Postfix configuration file */etc/mail/postfix/main.cf* accordingly. More detailed information on this can be found in the man pages on the Postfix server and in the documentation on the topic of Postfix which is referred to in "Related publications".

CAUTION!

Proceed particularly carefully when you configure the Postfix server as a mail relay because a mail relay which can be accessed openly from the Internet very quickly attracts mail spammers. Among other things, this can result in your Internet access being blocked by the Internet service provider.

10.4.2 Starting up IMAP and POP3 servers

In contrast to the Postfix server, IMAP and POP3 servers do not run permanently as independent daemons but are started up for the associated connection by the *inetd* daemon at connection setup. There is thus one dedicated IMAP or POP3 daemon per IMAP and POP3 connection. In the event of heavy parallel use this can result in considerable demands on resources.

To permit the *inetd* daemon to start the individual IMAP and POP3 daemons, the *inetd* daemon must be configured appropriately. In the standard case the *inetd* daemon is configured automatically during IMAP installation.

If problems should occur when you are operating IMAP or POP3 servers, you should proceed in accordance with the description below to check whether the *inetd* daemon has been configured correctly:

- In the */etc/inet/services* file check the entries which assign the service names to the port numbers of the IMAP and POP3 services.

  The following entries are required:

```
pop-3           110/tcp             # Post Office V3
pop-3s          995/tcp             # SSL secured Post Office V3
imap            143/tcp             # IMAP
imaps           993/tcp             # SSL secured IMAP
```

- Ensure that a symbolic link exists from `/etc/services` to `/etc/inet/services`.
- In the *inetd* configuration file */etc/inet/inetd.conf* check the entries for the individual service names.

  The following entries are required:

```
pop-3    stream tcp  nowait SYSROOT /usr/sbin/in.ipop3d in.ipop3d
pop-3s   stream tcp  nowait SYSROOT /usr/sbin/in.ipop3d in.ipop3d
imap     stream tcp  nowait SYSROOT /usr/sbin/in.imapd in.imapd
imaps    stream tcp  nowait SYSROOT /usr/sbin/in.imapd in.imapd
```

- If you have modified the *inetd* configuration file */etc/inet/inetd.conf* "manually", you must inform the *inetd* daemon of the modification in the configuration file using the following command:

```
kill -HUP <process id of the inetd daemon>
```

IMAP and POP3 servers have no configuration file of their own. Furthermore, IMAP and POP3 servers only support the standard Mbox format for the user mailboxes, but not the Maildir format supported as an alternative by the Postfix server.

10.4.3 TLS/SSL protection of IMAP/POP3 and SMTP connections

On both the IMAP and POP3 servers and also on the Postfix server (SMTP server) the connections can be secured with the aid of TLS/SSL. This is particularly important for IMAP and POP3 connections as passwords would otherwise be transferred unencrypted.

X.509 certificates are used in conjunction with SSL. An X.509 certificate contains all the information needed to identify the server or client and also the certificate owner's public key. Certificates are issued by a central authority, the Certificate Authority (CA), after the identity of the organization named in the certificate and of an authorized representative have been checked. To apply for an X.509 certificate from a CA you require a Certificate Signing Request (CSR) which you can generate with the */etc/imap/MAKE.CERT.sh* script. Furthermore, this script generates a test certificate which you can use to perform tests of the TLS/SSL functionality in advance. As this test certificate is generated with the aid of a publicly known CA key and consequently offers no effective protection, it may not be used for productive operation.

Detailed information on the topic of TLS/SSL is provided in the interNet Services User Guide.

TLS/SSL protection of IMAP and POP3 connections

You store the test certificate and the private key at the appropriate positions for the IMAP server in the directory tree (*/etc/imap/certs/in.imapd.pem* and */etc/imap/private/in.imapd.pem*). You store the certificate of the test CA under */etc/imap/certs/cacert.pem*. Explicit configuration settings are not possible with the IMAP and POP3 servers. The CSR can be found under */etc/imap/imapd-csr.pem*.

Note that the private key will later also be used together with the official certificate in productive operation. Consequently you must ensure that the private key is always protected against unauthorized access. During installation the access rights for the */etc/imap/private* directory are thus restricted to the super user (SYSROOT, TSOS).

During installation, suitable links are automatically created for the POP2 server which refer to the aforementioned certificate and the private key for the IMAP server. This enables the private key and the certificate to be used both by the IMAP and the POP3 server.

As soon as you have received the certificate from an official CA you must store this under */etc/imap/certs/in.imapd.pem*, thus overwriting the test certificate which you no longer require.

An excerpt from an */etc/imap/MAKE.CERT.sh* run is shown on the next page. User inputs are highlighted with **bold print**.

```
# /etc/imap/MAKE.CERT.sh
1. Country Name              (2 letter code) [DE]:◂
2. State or Province Name    (full name)     [Bavaria]: Bayern◂
3. Locality Name             (eg, city)      [Munich]: Muenchen◂
4. Organization Name         (eg, company)   [Manufacturer, Ltd]:
   Fujitsu Technology Solutions GmbH◂
5. Organizational Unit Name (eg, section)    [Marketing]: Internet Services◂
6. Common Name               (eg, FQDN)      [www.manufacturer.com]:
   www.ts.fujitsu.com◂
7. Email Address             (eg, name@FQDN) [info@manufacturer.com]:
   info@ts.fujitsu.com◂
8. Certificate Validity      (days)          [365]: 730◂
   Certificate Version (1 or 3)                [3]:◂
9. subjectAltName:dNSName    (eg, FQDN)      [www.ts.fujitsu.com]:◂
   Generating certificate, please wait...
   Done

Subject: C=DE, ST=Bayern, L=Muenchen, O=Fujitsu Technology Solutions GmbH, OU=Internet
Services, CN=www.ts.fujitsu.com/emailAddress=info@ts.fujitsu.com

The private key for IMAP has been created as /etc/imap/private/in.imapd.pem.
The certifcate for IMAP has been created as /etc/imap/certs/in.imapd.pem.
The used CA certificate has been saved as /etc/imap/certs/cacert.pem.

For using certificate  and  key also
 for POP3 we make appropriate links to the created files:
Creating  link  /etc/imap/private/in.ipop3d.pem  to  file
/etc/imap/private/in.imapd.pem.
Creating link /etc/imap/certs/in.ipop3d.pem to file /etc/imap/certs/in.imapd.pem.

WARNING: Use this certificate only for testing, not for production!
#
#
```

TLS/SSL protection of SMTP connections (Postfix)

SMTP connections can only be secured between two adjacent Mail servers (Mail Transfer Agents, MTAs) on the communication path. End-to-end security cannot be guaranteed with TLS/SSL. In particular, the e-mails are buffered on the MTAs in unencrypted format. TLS/SSL can, however, be used for implementing partially open mail relays. The group of users of a mail relay can be restricted to the authorized persons via TLS/SSL authentication. Similarly, in cases where the e-mail is sent directly to the recipient, TLS/SSL enables mail encryption to be implemented which is transparent for e-mail senders and recipients.

Unlike mail encryption (S/MIME, PGP), TLS/SSL also protects both the e-mail content on the respective transmission path and the meta data. You should therefore always use the TLS/SSL protection of the SMTP traffic if possible.

If you use TLS/SSL in conjunction with Postfix, you must set the corresponding parameters in the Postfix configuration file */etc/postfix/main.cf*. Here you must distinguish between use of TLS/SSL for incoming and outgoing connections:

- In the case of incoming connections the parameter names begin with the prefix “smtpd_”.
- In the case of outgoing connections the parameter names begin with the prefix “smtp_”.

Further information on the TLS/SSL parameters available is provided in the */etc/postfix/sample-tls.cf* file.

Minimum configuration for TLS/SSL protection with incoming SMTP connections

The additional parameters for a minimum configuration for TLS/SSL protection of an incoming SMTP connection are shown below. The TLS/SSL configuration also uses the IMAP/POP3 certificate as the Postfix server certificate.

```
smtpd_tls_cert_file = /etc/imap/certs/in.imapd.pem
smtpd_tls_key_file = /etc/imap/private/in.imapd.pem
smtpd_use_tls = yes
```

Minimum configuration for TLS/SSL protection with outgoing SMTP connections

The additional parameters for a minimum configuration for TLS/SSL protection of an outgoing SMTP connection are shown below. The TLS/SSL configuration also uses the IMAP/POP3 certificate as the Postfix Client certificate.

```
smtp_tls_cert_file = /etc/imap/certs/in.imapd.pem
smtp_tls_key_file = /etc/imap/private/in.imapd.pem
smtp_tls_CAfile = /etc/imap/certs/trusted-certs.pem
smtp_use_tls = yes
```

The */etc/imap/certs/trusted-certs.pem* file (the name is freely selectable) must contain certificates of all CAs which are to be trusted. In test operation to a server whose server certificate was generated with the aid of the *MAKE.CERT.sh* script you must enter the contents of the

/etc/imap/certs/cacert.pem file in the */etc/imap/certs/trusted-certs.pem* file. After you switch over to productive operation you must replace this test CA certificate by the CA certificate of the CA which issued the productive server certificate.

10.5 Operating the Postfix server

This section deals with the following topics:

- Postfix lookup tables
- Important programs for operating the Postfix mail server

10.5.1 Postfix lookup tables (index files)

Postfix uses a number of lookup tables to permit efficient searching for information on address substitution, access control, etc. In a lookup table for address substitution, for instance, the old address is the search criterion, while the new address is assigned to the search criterion as the search result. In a lookup table for access control, for example, all local e-mail recipients which Postfix accepts can be listed. Here it is only important for Postfix to know whether the search criterion exists. The search result as such is of no importance.

The information provided via the lookup tables is initially entered in ordinary text files by the administrator of the Postfix mail server. However, for high-performance access to this information Postfix does not use the text files (source files) created by the administrator, but index files which the administrator generates from the source files with the aid of the *postmap* (see page 338) or *postalias* (see page 340) program.

postalias differs from *postmap* in the format required for *postalias* source files (alias files) which, owing to its compatibility with the Sendmail server, deviates slightly from the source file format of the other Postfix source files for lookup tables (see page 330).

Two different types of index files

The BS2000 ported version of Postfix supports two different types of index files:

- *hash* files: Here the search is based on hash tables and algorithms.
- *btree* files: Here the search is based on Balanced-Tree (B-Tree) structures and algorithms.

Normally it is advisable to use the *hash* format. The use of *btree* files can make sense if performance problems occur with very large lookup tables in *hash* format.
Storage of lookup tables in external databases or directories such as NIS, MySQL or LDAP which is fundamentally possible with Postfix is currently not supported by the BS2000 ported version.

General Postfix format for postmap source files

An entry (logical line) in a *postmap* source file has the following syntax:

```
Key White space Value
```

Key
: Search criterion

White space
: White space consists of a sequence of blanks and/or tabulator characters. White space must contain at least one blank or tabulator character.

Value
: Information which is assigned to the search criterion.

In addition, the following rules apply for the format of a logical line:

- A logical line can extend over several text lines.
- A logical line starts with a text which is different from the white space.
- A line which begins with white space continues a logical line.
- Empty lines and lines which - possibly after a white space - begin with “#” are ignored by *postmap*.

Format for postalias source files (alias files)

With the aid of aliases the user part of a local receiver address can be replaced by one or more receiver addresses.

You can enter the following as receiver addresses:

- Local addresses
- Remote addresses
- Programs to which messages are transferred
- Files in which the messages are saved

The aliases are defined by entries in an alias file. The default file name of the alias file is */etc/postfix/aliases*.

An alias entry (alias definition) has the following syntax:

`Name: Value`$_1$`, Value`$_2$`, Value`$_3$`, ..., Value`$_n$

Name
: User part of a local receiver address. If the user part of a local receiver address contains special characters such as “@” or blanks, it must be enclosed in double quotes (""). The value of the user part of a local receiver address is always stored in lower case.

The alias file must always contain the entries for the local names “postmaster” and ”MAILER-DAEMON” (requirement of RFC 822).

$Value_1$, $Value_2$, $Value_3$, ..., $Value_n$
: One or more receiver addresses.

  The following can be specified as $Value_i$, i = 1 ... n:

- Local name (e.g. sysroot)
- Remote address (e.g. user@domain.com)
- */File*

  In this case the message is by default appended to the end of the *File* file. The *allow_mail_to_ files* parameter in the Postfix configuration file *main.cf* enables you to suppress this behavior.

- |*Program*

  In this case the message is by default transferred via a pipe to a program call *Program*. The *allow_mail_to_commands* parameter in the *main.cf* file enables you to suppress this behavior. If the program call contains special characters (in particular blanks), you must enclose it in double quotes ("").

- `:include:`*File*

  File is a text file which allows you to define mailing lists, for example. The e-mails are sent to the recipients listed in these mailing lists. Blanks and lines which begin with “#” are ignored. All other lines in *File* have the same syntax as the right-hand sides of alias entries, i.e. `Value`$_1$`, Valuet`$_2$`, ..., Value`$_n$.

  With `:include:`*File*, entries to be appended to files (|*File*) or to be transferred to program calls via a pipe (|*Program*) are ignored unless otherwise specified. The *main.cf* parameters *allow_mail_to_ files* and *allow_mail_to_commands* enable you to activate the appending of entries to files or the transfer of entries to program calls as the default option.

- owner-*aliasname*: *owner_mail_address*

  You use this entry to define an owner or administrator for the mailing list *aliasname* who has the mail address *owner_mail_address*. In the event of an error (undeliverable e-mail) the notification is then sent to *owner_mail_address* and not to the sender of the mail since the owner / administrator of the mailing list can generally react to delivery problems better than the sender of the mail.

  If, for example, a mailing list with the name “dbadmin” is defined, the following entry defines the mail address (here: “owner-mail-address”) of the mailing list’s owner:

  ```
  owner-dbadmin: owner-mail-address
  ```

If at the end of alias conversion it is determined that the message is to be delivered to a local user, Postfix checks whether this user’s home directory contains a *.forward* file. If this is the case, a further stage of alias conversion is performed in which the *.forward* file is interpreted according to the same rules as an alias file.

Example

An example of an alias file is shown below.

```
# Alias file
# Forward error messages to postmaster and append them
# to logging file /var/adm/mailerr
MAILER-DAEMON:   postmaster, /var/adm/mailerr
# Forward postmaster to system administrator ID
postmaster:      sysroot
# Forward messages addressed to system administrator
# to real person
sysroot:         mueller
# Program for automatic mail answering
auto-test:       |"/home/rwk/auto-test -i 3"
# Mailing list
dbadmin:         :include:/home/admin/db-admins
owner-dbadmin:   postmaster
```

This `owner-dbadmin` alias means that errors, e.g. undeliverable mails to `dbadmin`, are not sent to the orignial sender but to `postmaster`.

Content of */home/admin/db-admins*:

```
amueller@firma.example
bmeier@firma.example
dhuber@firma.example
/home/admin/db-admins.maillog
```

10.5.2 Programs for operating the Postfix mail server

The most important programs for operating the Postfix mail server are described below. The description covers the main parameters and options for these programs. A complete list of all the programs available for operating the mail services and their parameters/options is provided in the man pages for Postfix.

postfix - Starting and stopping the Postfix server

You use the *postfix* program to start and stop the Postfix mail server. In addition, after the configuration files */etc/postfix/master.cf* and */etc/postfix/main.cf* have been modified you can use *postfix* to read these files in again. You do not need to stop the Postfix mail server explicitly to do this.

On account of special POSIX features you should not call the *postfix* program directly, but use the encapsulation script */etc/init.d/MAIL.postfix* for the *postfix* call. Before Postfix starts, this script regenerates, among other things, all the index files defined in the configuration file *main.cf*, thus preventing a divergence of source and index files.

/etc/init.d/MAIL.postfix
{ **start** | **stop** | **reload** }

start
Starts the Postfix mail server.

stop
Stops the Postfix mail server.

reload
Reads the (modified) configuration files in again.

postconf - Displaying and modifying Postfix configuration parameters

The *postconf* program enables you to modify the values of the Postfix configuration parameters or have them displayed on the screen. This applies both for parameters which are set specifically in the configuration file */etc/postfix/main.cf* and for those preset by default.

postconf
[**-d**] [**-n**] [**-e**]

No parameter specified
: Shows all currently valid parameter values.

-d
: Shows all default parameter values.

-n
: Shows all currently valid parameter values which differ from the associated default value.

-e
: Sets Postfix configuration parameters.
The *-e* option makes sense above all for automatic setting of parameters via a shell script. Otherwise you can also process the configuration file */etc/postfix/main.cf* with any editor (e.g. EDT).

postqueue (mailq) - Processing mail queues (as a normal user)

With normal user authorization, the *postqueue* program enables you to execute operations on the mail queues which are organized as files in various subdirectories.

Each e-mail which the Postfix server receives for local delivery or forwarding is initially buffered in subdirectories of */var/spool/postfix*, e.g. */var/spool/postfix/incoming* or */var/spool/postfix/active*.

The e-mails placed in the mail queues are handled as follows:

- As soon as an e-mail has been successfully delivered or forwarded, it is removed from the mail queues.
- If an e-mail cannot initially be forwarded or delivered, e.g. because the mail server cannot be reached, it is once more placed in a mail queue. Another attempt to deliver it is only made after a wait time.

The *mailq* program is supported only because of its compatibility with the Sendmail mail server and provides the same functionality as `postqueue -p`.

As users with normal user authorization do not have the *sbin* directory in their path, they must specify the complete path name.

[**/opt/MAIL/postfix/sbin/**]**postqueue**
{ **-p** | **-f** | **-s** <site>}

-p
Lists the contents of the mail queues in a presentation which is familiar from the Sendmail program *mailq*.

-f
Causes the Postfix server to make an attempt (possibly a repeated attempt) to deliver all e-mails in the mail queues. This is, for example, useful after connection problems have been remedied which have temporarily prevented the forwarding of e-mails. Instead of waiting for the wait time for a renewed delivery attempt to elapse you can thus initiate the immediate delivery of all waiting messages.

-s <site>
Causes the immediate delivery of all e-mails intended for <site> which are contained in the mail queues.

postsuper - Processing mail queues (with SYSROOT authorization)

The *postsuper* program enables you to execute operations on the mail queues for which SYSROOT authorization is required.

postsuper
[**-p**] [**-s**] [**-d** <queue-id>] [**-h** <queue-id>] [**-H <**queue-id>] [<directory> ...]

-p
Deletes old temporary files which are left over after a system or software crash.

-s
Checks and repairs the structure of the mail queues. You are urgently recommended to perform this operation before each Postfix startup.

-d <queue-id>
In the mail queue(s) specified by <directory> ..., deletes an e-mail with the queue ID <queue-id>. If you specify the value ALL for <queue-id>, all e-mails in the specified mail queue(s) are deleted.
Default value for <directory> ... : hold, incoming, active, deferred

-h <queue-id>
In the mail queue(s) specified by <directory> ..., places an e-mail with the queue ID <queue-id> in the hold status. This e-mail will then initially not be delivered or forwarded. If you specify the value ALL for <queue-id>, all e-mails in the specified mail queue(s) are placed in the hold status.
Default value for <directory> ... : incoming, active, deferred

-H <queue-id>
In the mail queue(s) specified by <directory> ..., terminates the hold status for an e-mail with the queue ID <queue-id>. If you specify the value ALL for <queue-id>, the hold status is terminated for all e-mails in the specified mail queue(s).
Default value for <directory> ... : hold

<directory> ...
Specifies one or more mail queue directories.

postcat - Displaying the contents of messages in the mail queues

The *postcat* program enables the administrator to have the contents of individual messages (e-mails in the mail queues) displayed in a readable format.

[**/opt/MAIL/postfix/sbin/**]**postcat**
[**-vq**] [**-c** <config_dir>] [<file> ...]

-v
Activates detailed logging for debugging purposes.

-q
In this case you only need to specify the queue ID of the e-mail to be displayed as <file>. *postcat* then automatically determines the complete path name, i.e. the mail queue which contains <file>.

-c <config_dir>
The Postfix configuration file *main.cf* is located in the <config_dir> directory instead of the standard configuration directory.

<file> ...
Name(s) of the file(s)/e-mails whose contents are to be displayed:

– If you specify the `-q` option you need only specify the queue ID(s) of the message(s) to be output.

– If you do not specify the `-q` option, you must specify the complete path name(s) of the message(s).

Example

An e-mail with the queue ID 457106EA05 is contained in the *deferred* queue */var/spool/postfix/deferred/4/*. A file with the name */var/spool/postfix/deferred/4/457106EA05* thus exists.

You can have the contents of the e-mail 457106EA05 displayed as follows:

- `postcat /var/spool/postfix/deferred/4/457106EA05`

  Here you must specify the complete path name as the argument.

- `postcat -q 457106EA05`

  With the `-q` option it is sufficient if you specify the queue ID of the e-mail to be displayed.

postmap - Generating and processing index files (Postfix format)

Postfix uses index files (lookup tables, see page 329). The *postmap* program offers the following functionality for processing lookup tables:

- Generating index files from text files
- Displaying entries for a specific key value (index search)
- Adding entries to an index file
- Removing entries from an index file

The BS2000 ported version of Postfix supports the *hash* and *btree* formats for index files.

postmap
[**-q** <key>] [**-d** <key>] [**-i**] [**hash:**|**btree:**]<path-name> ...

-q <key>
Searches the entry for the key value <key> and outputs the first assigned value.

-d <key>
Deletes the entry with the key value <key>.

-i
Reads entries from the standard input and inserts these in the index file <path-name>.db.

hash: | btree:
Specifies the type of index file to be generated (`hash` or `btree`).

<path-name>
Name of the file for which the associated index file <path-name>.db is to be generated.

Example

A lookup table *canonical* which maps BS2000 user names (up to 8 characters long) onto e-mail addresses with the format *first-name.family-name* contains the following entries:

```
maier Georg.Maier
mueller Elisabeth.Mueller
```

You generate the associated index file using the following command (*hash* type) */etc/postfix/canonical.db*:

```
postmap hash:/etc/postfix/canonical
```

The command

```
postmap -q maier /etc/postfix/canonical
```

then supplies the following output for the key “maier” : `Georg.Maier`

The following command deletes the entry with the key “mueller” in the index file */etc/postfix/canonical.db*:

```
postmap -d mueller /etc/postfix/canonical
```

Note that the original text file */etc/postfix/canonical* is not changed.

postalias - Generating and processing index files (alias format)

The *postalias* program differs from the *postmap* program only in the format of the input files (see page 330).

postalias offers the following functionality for processing lookup tables:

- Generating index files from text files
- Displaying entries for a specific key value (index search)
- Adding entries to an index file
- Removing entries from an index file

The BS2000 ported version of *postalias* supports the *hash* and *btree* formats for index files.

postalias
[**-q** <key>] [**-d** <key>] [**-i** <key>] [**hash:**|**btree:**]<path-name> ...

-q <key>
Searches the entry for the key value <key> and outputs the first assigned value.

-d <key>
Deletes the entry with the key value <key>.

-i
Reads entries from the standard input and inserts these in the index file <path-name>.db.

hash: | btree:
Specifies the type of index file to be generated (`hash` or `btree`).

<path-name>
Name of the file for which the associated index file <path-name>.db is to be generated.

Example

This example is based on the alias file from the example on page 332.

You generate the associated index file (*hash* type) with the following command */etc/postfix/aliases.db*:

```
postalias hash:/etc/postfix/aliases
```

The command

```
postalias -q postmaster /etc/postfix/aliases
```

then supplies the following output for the key “postmaster”: `sysroot`

The following command deletes the entry with the key “sysroot” in the index file *//etc/postfix/aliases.db*:

```
postalias -d sysroot /etc/postfix/aliases
```

Note that the original text file */etc/postfix/aliases* is not changed.

newaliases - Generating index files (alias format)

The *newaliases* program generates the associated index files for all files named in the configuration parameter *alias_database*. The *alias_database* parameter is defined in the configuration file *main.cf*.

The *newaliases* program is supported for reasons of compatibility with the Sendmail mail server.

Example

The *alias_database* parameter is. for example, set as follows in the configuration file *main.cf*:

```
alias_database = hash:/etc/postfix/aliases
```

In this case you generate the index file */etc/postfix/aliases.db* through the program call *newaliases*.

10.6 Migration from Sendmail to Postfix

Migration from Sendmail to Postfix requires relatively little effort as the Postfix mail server uses the Sendmail alias file format and the compatibility programs *sendmail*, *mailq* and *newaliases* are also available.

Two alternative procedures are available for migrating from the Sendmail server to the Postfix mail server:

- Installing the Postfix mail server only after the Sendmail server has been uninstalled.
- Installing the Postfix mail server in parallel to the existing Sendmail server installation.

Installing the Postfix mail server after uninstalling the Sendmail server

If you uninstall the Sendmail server before you install the Postfix mail server, you must ensure that the e-mails contained in the Sendmail server's mail queue (default: */var/mqueue*) are delivered before Sendmail is uninstalled. This is required because Postfix uses a queue system which is not compatible with the Sendmail queue system.

Proceed as follows:

► Stop the *sendmail* daemon.

► Use the following command to start a Sendmail run to empty the Sendmail mail queue:

/usr/sbin/sendmail -q

► If e-mails are still left in the Sendmail mail queue: As required, make further attempts to deliver the remaining e-mails or delete them.

► Start installation of the Postfix mail server (see page 315).

Installing the Postfix mail server in parallel to the existing Sendmail server installation

If clearing of the Sendmail mail queue is delayed, for example because mail servers have failed temporarily, it is advisable to initially leave the Sendmail installation as it is and to install a Postfix mail server in parallel.

Proceed as follows:

- ► Start installation of the Postfix mail server (see page 315).

  If the Postfix installation routine finds an installed Sendmail, it issues a warning.

- ► Press the send key.

  By doing so you ignore the warning and Postfix installation is continued.

  The Postfix installation routine saves the *sendmail* program of the Sendmail server under the path */usr/sbin/sendmail.renamed.by.MAIL:postfix* and replaces it by the Postfix variant.

  If the Postfix installation routine finds an */etc/mail/aliases* or */etc/aliases* file (e.g. from an earlier Sendmail installation), it copies this file into the Postfix configuration directory. When installation has been completed the script call */etc/init.d/MAIL.postfix start* is used to call the *newaliases* program automatically in order to generate the associated index file.

- ► Use the following command to start a Sendmail run to empty the Sendmail mail queue:

  /usr/sbin/sendmail.renamed.by.MAIL:postfix -q

Before Postfix installation the */usr/sbin/mailq* and */usr/sbin/newaliases* calls were links on the Sendmail server to the Sendmail program. However, after successful Postfix installation the Postfix variant of the Sendmail program is started with these calls.

If after Postfix has been installed you still wish to use the Sendmail variants of the *mailq* and *newaliases* programs which operate on Sendmail files and queues, call the Sendmail program as follows with the suitable options selected:

- */usr/sbin/sendmail.renamed.by.MAIL:postfix -bi* for the *newaliases* functionality
- */usr/sbin/sendmail.renamed.by.MAIL:postfix -bp* for the *mailq* functionality

Postfix offers compatible support for the user-specific alias mechanism which is based on the *.forward* file contained in the home directory of the user concerned. As a result, no modifications to these files are required when migrating from Sendmail to Postfix.

11 Mail senders in BS2000

11.1 Installing and uninstalling mail senders

After the installation of the product files, a further installation step is required:

The address of the mail server (i.e. its DNS name or IP or IPv6 address) must be set using the *mailServer* option in the configuration file for the mail sender backend (see the section “mailServer” on page 352).

If there are problems, you should set the *logLevel* option in the configuration file in order to store diagnostic information in the logging file specified by means of *logFile* (see the sections “logFile” on page 349 and “logLevel” on page 350).

11.2 Option files

11.2.1 SYSSSI

The *SYSSSI* file specifies different parameters that are used by parts of the mail sender. The default name of the option file is:

`$.SYSSSI.MAIL.`*nnn*`.MAILCLNT`

Alternatively, it has the same name under the corresponding installation ID.

You can change the file name using the IMON logical ID SYSSSI.MAILCLNT.

Notation of the options in the option file

The various options must be entered in the option file in compliance with the following rules:

- Each option must be entered in a separate line.
- If the arguments of an option extend over several lines, each line to be continued must be concluded with a backslash (\), indicating that the line is to be continued.
- A line that begins with the number sign (#) in column 1 is ignored when the file is read in.
- No distinction is drawn between uppercase and lowercase for the option names.
- Unless otherwise specified, no distinction is drawn between uppercase and lowercase for the option values.

The available options are listed below:

defaultOptionFileName

The file name specified in the *defaultOptionFileName* option is used when the SEND-MAIL command is called with the default value *STD in the USER-OPTION-FILE operand.

defaultOptionFileName
<filename 1..54>

<filename 1..54>
Name of the file to be called by the SEND-MAIL command.
Default: SYSDAT.MAIL.*nnn*.USER.OPT

backendConfigurationFileName

The *backendConfigurationFileName* option specifies the name of the configuration file for the mail sender backend (see the section “Configuration file for the mail sender backend” on page 349).

backendConfigurationFileName
<filename 1..54>

<filename 1..54>
Name of the configuration file.
Default: $.SYSDAT.MAIL.*nnn*.SERVICE.OPT

senderSuffix

When the *useSenderSuffix* option (see page 348) is set appropriately, then the suffix specified with the *senderSuffix* option is used for the creation of the sender address, if this address hasn't been already specified otherwise. The address will be created through concatination of BS2000 user ID and suffix.

senderSuffix
<suffix 1..128>

<suffix 1..128>
For getting a valid mail address the suffix must contain a @ character. The typical form for this suffix will be @<fully qualified domain name>, e.g. @fujitsu.com, but it may be advisable to prepend a character string for distinction, e.g. as with .BS2000@fujitsu.com. In the case of the user id TSOS the sender address TSOS.BS2000@fujitsu.com would be constructed. If the address built this way is not a valid e-mail address of the corresponding BS2000 user, then the generated bounce mails in case of mail delivery problems will get lost. Therefore one should contact the mail server administration for e.g. defining this mail address as an alias for a valid e-mail address.

useSenderSuffix

This option specifies whether the sender address shall be generated automatically if need be.

useSenderSuffix
NO | **YES**

NO
: No automatic generation of the sender address will be done. This is the default.

YES
: When additionally the *senderSuffix* option (see page 347) is specified, then the sender address will be generated automatically in case it hasn't been specified otherwise. With this the users are not forced to create an user option file or to specify the sender address with every mail sending order.

11.2.2 Configuration file for the mail sender backend

The mail server backend is a TU task and:

- Receives an e-mail order from ASTI.
- Generates from this order a correctly formatted e-mail in accordance with MIME and S/MIME.
- Sends this e-mail to a mail server over an SMTP connection that may be secured by means of TLS/SSL.

The configuration options required for this are stored in a configuration file. The default file name is specified in the *backendConfigurationFileName* option in the *SYSSSI* file (see the section “SYSSSI” on page 345). This can be overwritten using the commands START-MAIL-SERVICE and MODIFY-MAIL-SERVICE-PARAMETER.

The available options are listed below:

logFile

The *logFile* option specifies the name of the logging file in which the diagnostic messages are stored. The *logLevel* option (see below) specifies which messages are to be written in the file.

logFile
<filename 1..54>

<filename 1..54>
Name of the logging file.
Default: $.SYSDAT.MAIL.*nnn*.LOG

logLevel

The *logLevel* option specifies the priority of the diagnostic messages to be logged.

logLevel
<selector> <level>

<selector>
Specifies the functional set of messages to which the log level applies. The following values are possible:

ASTI	Messages relating to order handling in ASTI
SMTP	Messages relating to e-mail transfer to the SMTP server
TLS	Messages relating to the TLS/SSL protection of the connection to the SMTP server
SMIME	Messages relating to the S/MIME signing and/or encryption of e-mails
OTHER	All other messages
ALL	All messages

<level>
Specifies that all messages are to be logged with at least the specified priority. The possible values, in descending order of priority, are:

ALERT
CRITICAL
ERROR
WARNING
NOTICE
INFO
DEBUG
NONE, which suppresses the logging of all messages

The *logLevel* option can be used more than once.

Example

```
logLevel ALL WARNING
logLevel SMTP CRITICAL
```

This specifies that all messages with the priority WARNING are to be logged. The exception are SMTP messages, which are only to be logged when their priority is CRITICAL or higher.

Default: ALL CRITICAL

logMailContent

The *logMailContent* option specifies whether, for example, in ASTI logging at the highest level (“logLevel ASTI DEBUG”), the contents of e-mails (subject lines, e-mails and attachments) are to be logged as well. The logging of e-mail contents greatly increases the size of the logging files and causes data protection problems. It should therefore only be enabled when absolutely necessary and when precautions are taken to prevent the contents of e-mails from falling into the hands of people not authorized to read them.

logMailContent
NO / **YES**

NO
The contents of e-mails are not logged. This is the default.

YES
The contents of e-mails are logged when the logging level is sufficiently high.

mailServer

The *mailServer* option specifies the mail server to which the e-mail is to be forwarded.

mailServer
<DNS-name | IP-address | IPv6-address>

<DNS-name | IP-address | IPv6-address>
DNS name, IP address or IPv6 address of the mail server to which the e-mail is to be forwarded.

mailServerPort

The mail server is generally addressed via port 25. The *mailServerPort* may specify a different port for the mail server.

mailServerPort
<port-number>

<port-number>
Port number of the mail server.
Default: 25

mailLogLevel

The *mailLogLevel* option specifies which data is to be logged.

mailLogLevel
0 | **1** | **2**

0

No logging (default).

1

Simple logging.
In this case, the following data is logged:

- Time
- ASTI order ID
- Sender addresses (“From”), both envelope and header
- Recipient addresses (“To”), both envelope and header
- Copy recipient addresses (“Cc”) (header)
- Success status of the transfer to the SMTP server, including the OK message of the mail server. Depending on the mail server software, this message may contain part of the message ID.

2

Advanced logging.
In addition to the data logged in simple logging, the subject header is logged. Since the subject header generally contains contents of the e-mail, data protection aspects have to be considered.

mailLogFile

The *mailLogLevel* option specifies the name of the e-mail logging file. Every e-mail send order sent to the service is entered in this file. The information written to the file depends on the *mailLogLevel* option (see page 353).

mailLogFile
<filename 1..54>

<filename 1..54>
Name of the e-mail logging file.
Default: $.SYSDAT.MAIL.*nnn*.MAILLOG

maxQueueLifeTime

The *maxQueueLifeTime* option defines the maximum life expectancy of an e-mail, during which a failed e-mail dispatch is repeated.

maxQueueLifeTime
<lifetime>[s | m | h | d]

<lifetime>
Default setting: 5d

If a unit is not specified, the *<lifetime>* specified is in days. If a unit is specified (*s* for second, *m* for minute, *h* for hour, *d* for day), it must immediately follow *<lifetime>*, in other words, without a space.

The *retryLimit* option becomes invalid with the introduction of *maxQueueLifeTime*.

retryLimit

This option is no longer supported in future.

Please use the *maxQueueLifeTime* option instead, see page 354.

smtpReadMaxWaitTime

The *smtpReadMaxWaitTime* option defines how long the mail sender backend has to wait for an answer form the SMTP servers is applicable. If a mail send request is canceled due to an exceeded wait time then it will be repeated after a specific time, just as in case of error messages of the SMTP servers which indicate a temporary existing problem, see options *smtpRetryTimeBase* and *smtpRetryTimeMaxExp*.

The commands MODIFY-MAIL-SERVICE-PARAMETER, SHOW-MAIL-SERVICE-PARAMETER and STOP-MAIL-SERVICE which are communicating with the mail server backend and the communication of the backend with the SNTP server are serialized. Hence, it is preferable to limit the waiting state time of the backed - which may be caused by SMTP server problems- in order to process these commands as immediately as possible. On the other hand, this limitation should not be too rigorous because an overloaded SMTP server would be loaded still more due to the canceled and repeated transfers.

A waiting time in the digit minute range may be bi a good compromise in general.

smtpReadMaxWaitTime
<time>[s | m | h | d]

<time>
Maximum waiting time
Default: 5m

If a unit is not specified, the *<time>* specified is in minutes. If a unit is specified (*s* for second, *m* for minute, *h* for hour, *d* for day), it must immediately follow *<time>*, in other words, without a space. If 0 is specified then the waiting time not limited.

smtpRetryTimeBase

The *smtpRetryTimeBase* option defines the time base used for determining the time after which a renewed mail dispatch attempt will be started when a mail dispatch has been failed. See *smtpRetryTimeMaxExp* option for details.

smtpRetryTimeBase
<value>[s | m | h | d]

<value>
time base
Default: 15m

If a unit is not specified, the <*value*> specified is in minutes. If a unit is specified (*s* for second, *m* for minute, *h* for hour, *d* for day), it must immediately follow <*value*>, in other words, without a space.

smtpRetryTimeMaxExp

The *smtpRetryTimeMaxExp* option limits the increase in waiting time between two repeated mail dispatch attempts. The waiting time normally doubles with every failed dispatch attempt in order to restrict CPU usage caused by the dispatch attempts during persistent dispatch problems. After doubling *smtpRetryTimeMaxExp*, the waiting time remains at the value reached.

smtpRetryTimeMaxExp
<value>

<value>
Default setting: 6

If errors occur while trying to connect to the SMTP mail server, double *smtpRetryTimeBase* is continually used as the waiting time until a renewed delivery attempt is made.

If the error occurs later in the SMTP dialog, which could possibly mean that the problem is not a general server problem that is quickly noticed, but a mail-specific problem that is often only noticed after some time, then the waiting time between two dispatch attempts (beginning with *smtpRetryTimeBase*) doubles with every attempt until doubling of the *smtpRetryTimeMaxExp* is reached. The default maximum waiting time between two delivery attempts is therefore as follows:

Maximum waiting time = $smtpRetryTimeBase * 2^{smtpRetryTimeMaxExp} = 15m * 2^6 = 960m = 16h$

Examples

1. Mail server not accessible

   Renewed delivery attempt after 30 min= 2 * 15m

2. Connection setup to mail server possible; mail-specific error:

   – Renewed delivery attempt after 15 min= $15m * 2^0$
   – Renewed delivery attempt after 30 min= $15m * 2^1$
   – Renewed delivery attempt after 1 h = $15m * 2^2$
   – Renewed delivery attempt after 2 h = $15m * 2^3$
   – Renewed delivery attempt after 4 h = $15m * 2^4$
   – Renewed delivery attempt after 8 h = $15m * 2^5$
   – All further delivery attempts after 16 h = $15m * 2^6$

   until *maxQueueLifeTime* is reached (default: 5 days).

When reducing the *smtpRetryTimeBase*, the value for *smtpRetryTimeMaxExp* should be increased at the same time; otherwise repeated frequent delivery attempts overload the CPU.

tempFilePrefix

Temporary files are used for signing or encryption with S/MIME. The names of these temporary files consist of a prefix and different suffixes. The prefix is specified in the *tempFilePrefix* option.

This option allows you to stored the files on a pubset and/or under a user ID other than the default pubset or TSOS ID. To this end, a CAT ID or user ID must be specified in the prefix.

tempFilePrefix
<prefix>

<prefix>
Prefix for the name of the temporary files.
Default: #SYSDAT.MAIL.SMIME-TMP

tlsSecureConnection

The *tlsSecureConnection* option specifies whether the SMTP connection to the SMTP server is to be secured with TLS.

tlsSecureConnection
NONE | **OPTIONAL | REQUIRE**

NONE
The SMTP connection is never secured (default).

OPTIONAL
The SMTP connection is secured if the SMTP server supports it. If not, an unsecured connection is used.

REQUIRE
The SMTP connection is closed if the SMTP server does not support security.

tlsProtocol

OpenSSL supports Versions 3 of the SSL protocol and also Versions 1, 1.1 and 1.2 of the TLS protocol. Some of these protocols can be activated selectively using the *-tlsProtocol* option.

-tlsProtocol
[**+** | **-**] {**SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **All** } ...

+
The protocol specified after this sign is permissible.

If neither “+” nor “-” is specified, this has the same effect as specifying “+”.

-
The protocol specified after this sign is not permissible.

SSLv3
SSL protocol Version 3

Version 3 of the SSL protocol displays some security-related deficiencies and should therefore not be used if possible.

TLSv1
TLS protocol Version 1

TLSv1.1
TLS protocol Version 1.1

TLSv1.2
TLS protocol Version 1.2

ALL
All protocols are to be enabled.

All -SSLv3 is the default.

Example

The specifications `–tlsProtocol TLSv1 TLSv1.1 TLSv1.2` and `–tlsProtocol All –SSLv3` have the same effect as long as no support of the future TLS version 1.3 is added to the Mail sender.

tlsCipherSuite

You can use the *tlsCipherSuite* option to specify a list of preferred encryption methods. If this option is not specified, a default list of preferences is used.

tlsCipherSuite
<specification>

<specification>
Specification in a list of preferred encryption methods see chapter “Specification of a cipher suite preference list” on page 375
Default: ALL: !EXP: !ADH:!RC4

tlsCertificateFile

The *tlsCertificateFile* option allows you to specify a file that contains the X.509 client certificate in PEM format. This file can also contain the private client key. Generally, however, the certificate and key are stored in separate files. In this case, the key file is specified using the *tlsKeyFile* option (see below).

tlsCertificateFile
<filename 1..54> | ***NONE**

<filename 1..54>
Name of the file that contains the X.509 client certificate in PEM format.

***NONE**
No file with certificates is used (default).

tlsKeyFile

The *tlsKeyFile* option allows you to specify a file that contains the private client key in PEM format.

If both the certificate and private key are contained in the same file (see the *tlsCertificateFile* option above), the *tlsKeyFile* option does not have to be specified.

The client key cannot be secured with a passphrase because it is not possible to enter a passphrase at startup of the service task.

tlsKeyFile
<filename 1..54> | ***NONE**

<filename 1..54>
Name of the file that contains the private client key.
Default: file name specified by means of *tlsCertificateFile*

***NONE**
A separate file is not used for the client key.

tlsCACertificateFile

The *tlsCACertificateFile* option allows you to specify a file that contains in PEM format the CA certificates that are required for authentication on the server. The individual PEM certificates are arranged sequentially in the file.

To add or delete certificates, you can edit the file in a text editor. The certificates are entered in the file as follows:

```
-----BEGIN CERTIFICATE-----
< CA certificate in Base64 coding >
-----END CERTIFICATE-----
```

Any text outside these sequences is ignored and can therefore be used to identify the certificates that are not available in readable form because of ASN.1/Base64 coding.

tlsCACertificateFile
<filename 1..54> | ***NONE**

<filename 1..54>
: Name of the file that contains the certificates in PEM format that are required for authentication on the server.

***NONE**
: No file is specified (default).

tlsCARevocationFile

The *tlsCARevocationFile* option allows you to specify a file that contains the CRLs (certificate revocation lists) of the certificate authorities (CA). Certificates issued by a certificate authority can be declared invalid by the publication of a certificate revocation list (CRL).

tlsCARevocationFile
<filename 1..54> | ***NONE**

<filename 1..54>
Name of the file that contains the CRLs of the certificate authorities.

***NONE**
No file with CRLs is specified (default).

tlsVerifyServer

The *tlsVerifyServer* option allows you to specify whether a server certificate has to be verified.

tlsVerifyServer
YES | **NO**

YES
The certificate must be verified (default).

NO
The certificate does not have to be verified. In this case, there is a danger that an attacker (“man in the middle”) will be able to tap the data transfer on the connection between the mail sender backend and mail server unnoticed.

tlsVerifyDepth

The *tlsVerifyDepth* option allows you to specify the verification depth (i.e. the maximum permissible number of certificates between the server certificate and the certificate known to the service task).

- The default for the maximum depth is 1. In this case, if it is to be accepted, the server certificate must have been issued directly by a certificate authority (CA) known to the service task.
- If the maximum depth is exceeded, the connection is aborted provided the verification of the server certificate is not disabled by means of *tlsVerifyServer* (see page 364).
- It makes no sense to specify a depth of 0. In this case, only self-signed certificates would be permissible.

tlsVerifyDepth
<depth>

<depth>
Number of maximum permissible certificates between the server certificate and the certificate known to the service task.
Default: 1

11.3 Mail service commands

START-MAIL-SERVICE

The START-MAIL-SERVICE command starts the ASTI service task that processes the SMTP dialog with the mail server.

Domain
SYSTEM-MANAGEMENT

Privileges
TSOS

START-MAIL-SERVICE
ORDER-LIMIT = **255** / <integer 1..32000> **,CONFIGURATION-FILE** = ***STD** / <filename 1..54 without-gen>

Operand description

ORDER-LIMIT = 255 / <integer 1..32000>
This operand specifies how many orders can be active at any one time.

CONFIGURATION-FILE = *STD / <filename 1..54 without-gen>
Specifies the file that contains the configuration options for the service task. You will find the contents of the file in the section “Configuration file for the mail sender backend” on page 349.

CONFIGURATION-FILE = *STD
The file name specified for the *backendConfigurationFileName* option in the *SYSSSI* file is used for the querying of the configuration options (see the section “SYSSSI” on page 345).

CONFIGURATION-FILE = <filename 1..54 without-gen>
Specifies the file that contains the configuration options.

Return codes

(SC2) SC1	Maincode	Meaning
0	CMD0001	No error
1	CMD0202	Syntax error
64	CMD0216	The user has no authorization for the command
64	YML0120	The ASTI subsystem is not available
64	YML0130	The configuration cannot be read or does not exist
64	YML0132	ASTI reports an error
64	YML0134	The ASTI service is already running
64	YML0136	The program for the mail sender backend cannot be found

MODIFY-MAIL-SERVICE-PARAMETER

You use the MODIFY-MAIL-SERVICE-PARAMETER command to change different parameters of the ASTI service task. In particular, you can change the file that contains the configuration for the mail sender backend. You can also specify separate parameters for logging settings and assign values that differ from the contents of the configuration file. In this way, it is possible to change the logging files or the set of logging events dynamically without the need to change the configuration file.

Since the command works with the service task, the completion of the execution of the command may be delayed when the service task is lagging behind due to previous orders that have to be processed.

Domain
SYSTEM-MANAGEMENT

Privileges
TSOS

MODIFY-MAIL-SERVICE-PARAMETER

CONFIGURATION-FILE = ***UNCHANGED** / <filename 1..54 without-gen>

,MAIL-LOG-FILE = ***UNCHANGED** / <filename 1..54 without-gen>

,TRACE-FILE = ***UNCHANGED** / <filename 1..54 without-gen>

,ASTI-TRACE-LEVEL = ***UNCHANGED** / ***NONE** / ***ALERT** / ***CRITICAL** / ***ERROR** / ***WARNING** / ***NOTICE** / ***INFO** / ***DEBUG**

,SMTP-TRACE-LEVEL = ***UNCHANGED** / ***NONE** / ***ALERT** / ***CRITICAL** / ***ERROR** / ***WARNING** / ***NOTICE** / ***INFO** / ***DEBUG**

,TLS-TRACE-LEVEL = ***UNCHANGED** / ***NONE** / ***ALERT** / ***CRITICAL** / ***ERROR** / ***WARNING** / ***NOTICE** / ***INFO** / ***DEBUG**

,SMIME-TRACE-LEVEL = ***UNCHANGED** / ***NONE** / ***ALERT** / ***CRITICAL** / ***ERROR** / ***WARNING** / ***NOTICE** / ***INFO** / ***DEBUG**

,OTHER-TRACE-LEVEL = ***UNCHANGED** / ***NONE** / ***ALERT** / ***CRITICAL** / ***ERROR** / ***WARNING** / ***NOTICE** / ***INFO** / ***DEBUG**

Operand description

CONFIGURATION-FILE = *UNCHANGED / <filename 1..54 without-gen>
Specifies the file that contains the configuration options for the service task. You will find the contents of the file in the section “Configuration file for the mail sender backend” on page 349.

CONFIGURATION-FILE = *UNCHANGED
The name of the configuration file remains unchanged.

CONFIGURATION-FILE = <filename 1..54 without-gen>
Specifies the file that contains the configuration options for the service task. If the name of the configuration currently being used is specified, this file is read in again and any changes made to the file become effective.

MAIL-LOG-FILE = *UNCHANGED / <filename 1..54 without-gen>
Specifies the name of the file in which the mail send result is to be logged (see the *mailLogFile* option on page 354). It is not possible to access the file when the mail service is running. To open the file, the assignment must therefore be made (temporarily) to another file.

MAIL-LOG-FILE = *UNCHANGED
The name of the mail logging file remains unchanged.

MAIL-LOG-FILE = <filename 1..54 without-gen>
Specifies the name of the file in which the mail send result is to be logged.

TRACE-FILE = *UNCHANGED / <filename 1..54 without-gen>
Specifies the name of the file in which diagnostic messages are to be logged (see the *logFile* option on page 349). It is not possible to access the file when the mail service is running. To open the file, the assignment must therefore be made (temporarily) to another file.

TRACE-FILE = *UNCHANGED
The name of the logging file remains unchanged.

TRACE-FILE = <filename 1..54 without-gen>
Specifies the name of the file in which diagnostic messages are to be logged.

ASTI-TRACE-LEVEL = *UNCHANGED / *NONE / *ALERT / *CRITICAL / *ERROR / *WARNING / *NOTICE / *INFO / *DEBUG
Specifies the filter level for the logging of diagnostic messages relating to ASTI actions (see the *logLevel* option on page 350).

SMTP-TRACE-LEVEL = *UNCHANGED / *NONE / *ALERT / *CRITICAL / *ERROR / *WARNING / *NOTICE / *INFO / *DEBUG
Specifies the filter level for the logging of diagnostic messages relating to the SMTP protocol (see the *logLevel* option on page 350).

TLS-TRACE-LEVEL = *UNCHANGED / *NONE / *ALERT / *CRITICAL / *ERROR / *WARNING / *NOTICE / *INFO / *DEBUG
Specifies the filter level for the logging of diagnostic messages relating to the TLS protocol (see the *logLevel* option on page 350).

SMIME-TRACE-LEVEL = *UNCHANGED / *NONE / *ALERT / *CRITICAL / *ERROR / *WARNING / *NOTICE / *INFO / *DEBUG
Specifies the filter level for the logging of diagnostic messages relating to S/MIME actions (see the *logLevel* option on page 350).

OTHER-TRACE-LEVEL = *UNCHANGED / *NONE / *ALERT / *CRITICAL / *ERROR / *WARNING / *NOTICE / *INFO / *DEBUG
Specifies the filter level for the logging of diagnostic messages relating to other actions (see the *logLevel* option on page 350).

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	64	CMD0216	The user has no authorization for the command
	32	CMD0220	Internal error
	64	YML0120	The ASTI subsystem is not available
	64	YML0130	The configuration file cannot be read or does not exist
	64	YML0131	Error accessing file
	64	YML0140	The mail service is not running
	64	YML0148	The maximum order limit has been reached
	32	YML0176	Unexpected ASTI error

SHOW-MAIL-SERVICE-PARAMETER

The SHOW-MAIL-SERVICE-PARAMETER command outputs the current settings of the mail service parameters.

In some cases (when a MODIFY-MAIL-SERVICE-PARAMETER command has not been issued beforehand), the command works together with the service task. The completion of the execution of the command may therefore be delayed when the service task is lagging behind due to previous orders that have to be processed.

Domain
SYSTEM-MANAGEMENT

Privileges
TSOS

SHOW-MAIL-SERVICE-PARAMETER

Example

```
/SHOW-MAIL-SERVICE-PARAMETER
%  Configuration file:    :DEFL:$TSOSDEFL.SYSDAT.MAIL.033.SERVICE.OPT
%  Mail log file:         $.SYSDAT.MAIL.033.MAILLOG
%  Trace file:            $.SYSDAT.MAIL.033.LOG
%  Trace level
%    ASTI:                *ERROR
%    SMTP:                *DEBUG
%    TLS:                 *ERROR
%    SMIME:               *INFO
%    OTHER:               *ERROR
```

Return codes

(SC2) SC1	Maincode	Meaning
0	CMD0001	No error
1	CMD0202	Syntax error
64	CMD0216	The user has no authorization for the command
32	CMD0220	Internal error
64	YML0120	The ASTI subsystem is not available
64	YML0140	The mail service is not running
128	YML0148	The maximum order limit has been reached
32	YML0176	Unexpected ASTI error

STOP-MAIL-SERVICE

The STOP-MAIL-SERVICE command stops the ASTI service task, which processes the SMTP dialog with the mail server.

Domain
SYSTEM-MANAGEMENT

Privileges
TSOS

STOP-MAIL-SERVICE

Return codes

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	No error
	1	CMD0202	Syntax error
	64	CMD0216	The user has no authorization for the command
	64	YML0120	The ASTI subsystem is not available

11.4 Messages

The Mail sender message code has the format `YMLnnnn`.

Output of messages using /HELP-MSG-INFORMATION

The BS2000 command `/HELP-MSG-INFORMATION MSG-ID=YMLnnnn` allows you to query the meaning and response texts for a message in ongoing operation.

Output of messages on the internet

You can also find the messages on our manual server using the HTML application in the place of the former “System Messages” manual (*http://manuals.ts.fujitsu.com*) and on the DVD “BS2000 SoftBooks”.

12 Specification of a cipher suite preference list

The specification consists of one or more cipher mnemonics which are separated by a colon (:).

A cipher mnemonic can take the following forms:

- A cipher mnemonic can consist of a single cipher suite such as DES-CBC-SHA.
- A cipher mnemonic can represent:
  - a list of cipher suites which contain a particular algorithm
  - cipher suites of a particular type

  For example, SHA1 represents all cipher suites which use the digest algorithm SHA1, and SSLv3 represents all SSL Version-3 algorithms.
- Lists of cipher suites can be combined to form a single cipher mnemonic with the aid of the “+” character. This is then interpreted as a logical AND operation. Thus SHA1+DES represents all cipher suites which contain the SHA1 and DES algorithms.
- Each cipher mnemonic can optionally be prefixed by one of the characters “!”, “-” or “+”:
  - If the prefix is “!”, the relevant cipher suites are permanently deleted from the preference list. Subsequently these no longer appear in the preference list even when they are specified explicitly.
  - If the prefix is “-”, the relevant cipher suites are deleted from the preference list, but some or all of them can be added again by means of subsequent options.
  - If the prefix is “+”, the relevant cipher suites are moved to the end of the preference list. This means that no cipher suites are added to the preference list, but only existing ones moved.
  - If none of the three characters “!”, “-” or “+” is prefixed, the cipher mnemonic is interpreted as a list of cipher suites which is appended to the current preference list. If this includes a cipher suite which is already contained in the current preference list, it is ignored. It is not moved to the end of the preference list.
- The cipher mnemonic @STRENGTH can be added at any position in order to sort the current preference list according to the length of the encryption key.

Permissible cipher mnemonics

The permissible cipher mnemonics are described below.

ALL
: All cipher suites with the exception of the eNULL ciphers. The latter must be enabled explicitly.

HIGH
: Cipher suites with key lengths greater than 128 bits.

MEDIUM
: Cipher suites with a key length of 128 bits or cipher suites downgraded due to other reasons.

LOW
: Cipher suites with key lengths of 64 or 56 bits, except Export cipher suites.

EXP, EXPORT
: Export encryption algorithms including 40- and 56-bit algorithms.

EXPORT40
: 40-bit Export encryption algorithms.

EXPORT56
: 56-bit Export encryption algorithms.

eNULL, NULL
: “NULL” encryption algorithms, in other words those without encryption. As these offer no encryption and thus present a security risk, they are by default disabled and, if required, must be specified explicitly.

aNULL
: Cipher suites without authentication. This means at present the anonymous Diffie-Hellman algorithms. These algorithms are vulnerable to “man in the middle” attacks, and you are consequently advised not to use them.

kRSA, RSA
: Cipher suites with RSA key exchange.

aRSA
: Cipher suites with RSA authentication, in other words the certificates contain RSA keys.

aDSS, DSS
: Cipher suites with DSS authentication, in other words the certificates contain DSS keys.

TLSv1.2, TLSv1.0, SSLv3
: TLSv1.2, TLSv1.0 or SSLv3 cipher suites.

  Note: There exist no TLSv1.1.specific cipher suites.

DH
: Cipher suites with Diffie-Hellman key exchange, including anonymous exchange.

ADH
: Cipher suites with anonymous Diffie-Hellman key exchange.

kEDH, kDHE
: Cipher suites with ephemeral Diffie-Hellmann key negotiation including anonymous suites.

kEECDH, kECDHE
: Cipher suites with ephemeral Elliptic Curve Diffie-Hellmann key negotiation including anonymous suites.

EECDH, ECDHE
: Cipher suites with ephemeral Elliptic Curve Diffie-Hellmann key negotiation without anonymous suites.

AECDH
: Anonymous Cipher suites with Elliptic Curve Diffie-Hellmann key negotiation.

ECDH
: Cipher suites with Elliptic Curve Diffie-Hellmann key negotiation including anonymous, ephemeral and fixed ECDH.

AES128, AES256, AES
: Cipher suites with AES encryption (key length of 128 or 256 bits or one of them).

3DES
: Cipher suites with Triple DES encryption.

DES
: Cipher suites with DES encryption (no Triple DES).

RC4
: Cipher suites with RC4 encryption.

RC2
: Cipher suites with RC2 encryption.

MD5
: Cipher suites with MD5 hash function.

SHA1, SHA
: Cipher suites with SHA1 hash function.

As it is just a matter of time until feasible attacks on SHA1 appear, you should switch as soon as possible to cipher suites that use the hash functions SHA256 or SHA384, for example. However, this usually also implies switching to TLS protocol version 1.2.

SHA256, SHA384
: Cipher suites using the SHA256 and SHA384 hash function respectively for the MAC (message authentication code) computation. In the case of cipher suites using AESGCM and hence AEAD (Authenticated Encryption with Associated Data) as the MAC method, the SHA256 and SHA384 respectively in the name has a different meaning.

aECDSA
: Cipher suites using ECDSA authentication, in other words, the certificates contain ECDSA keys.

AESGCM
: Cipher suites using AES in "Galois Counter Mode (GCM)". These cipher suites are only supported by TLSv1.2.

CAMELLIA128, CAMELLIA256, CAMELLIA
: Cipher suites that use 128 bit CAMELLIA, 256 bit CAMELLIA or either 128 or 256 bit CAMELLIA.

The selecting effect of a preference list specification can be checked with the SHOW.CIPHERLIST procedure (see “InterNet Services User Guide“)

The available cipher suites are listed in the table below.

Name	Version	Key exchange	Authentication	Encryption	Digest	Export
DHE-RSA-AES256-SHA	SSLv3	DH	RSA	AES(256)	SHA1	
DHE-DSS-AES256-SHA	SSLv3	DH	DSS	AES(256)	SHA1	
AES256-SHA	SSLv3	RSA	RSA	AES(256)	SHA1	
DHE-RSA-AES128-SHA	SSLv3	DH	RSA	AES(128)	SHA1	
DHE-DSS-AES128-SHA	SSLv3	DH	DSS	AES(128)	SHA1	
AES-128-SHA	SSLv3	RSA	RSA	AES(128)	SHA1	
DHE-DSS-RC4-SHA	SSLv3	DH	DSS	RC4(128)	SHA1	
EDH-RSA-DES-CBC3-SHA	SSLv3	DH	RSA	3DES(168)	SHA1	
EDH-DSS-DES-CBC3-SHA	SSLv3	DH	DSS	3DES(168)	SHA1	
DES-CBC3-SHA	SSLv3	RSA	RSA	3DES(168)	SHA1	

Available cipher suites

Name	Version	Key exchange	Authentication	Encryption	Digest	Export
RC4-SHA	SSLv3	RSA	RSA	RC4(128)	SHA1	
RC4-MD5	SSLv3	RSA	RSA	RC4(128)	MD5	
EDH-RSA-DES-CBC-SHA	SSLv3	DH	RSA	DES(56)	SHA1	
EDH-DSS-DES-CBC-SHA	SSLv3	DH	DSS	DES(56)	SHA1	
DES-CBC-SHA	SSLv3	RSA	RSA	DES(56)	SHA1	
DES-CBC3-MD5	SSLv2	RSA	RSA	3DES(168)	MD5	
RC2-CBC-MD5	SSLv2	RSA	RSA	RC2(128)	MD5	
RC4-MD5	SSLv2	RSA	RSA	RC4(128)	MD5	
RC4-64-MD5	SSLv2	RSA	RSA	RC4(64)	MD5	
DES-CBC-MD5	SSLv2	RSA	RSA	DES(56)	MD5	
EXP1024-DHE-DSS-RC4-SHA	SSLv3	DH(1024)	DSS	RC4(56)	SHA1	export
EXP1024-RC4-SHA	SSLv3	RSA(1024)	RSA	RC4(56)	SHA1	export
EXP1024-DHE-DSS-DES-CBC-SHA	SSLv3	DH(1024)	DSS	DES(56)	SHA1	export
EXP1024-DES-CBC-SHA	SSLv3	RSA(1024)	RSA	DES(56)	SHA1	export
EXP1024-RC2-CBC-MD5	SSLv3	RSA(1024)	RSA	RC2(56)	MD5	export
EXP1024-RC4-MD5	SSLv3	RSA(1024)	RSA	RC4(56)	MD5	export
EXP-EDH-RSA-DES-CBC-SHA	SSLv3	DH(512)	RSA	DES(40)	SHA1	export
EXP-EDH-DSS-DES-CBC-SHA	SSLv3	DH(512)	DSS	DES(40)	SHA1	export
EXP-DES-CBC-SHA	SSLv3	RSA(512)	RSA	DES(40)	SHA1	export
EXP-RC2-CBC-MD5	SSLv3	RSA(512)	RSA	RC2(40)	MD5	export
EXP-RC4-MD5	SSLv3	RSA(512)	RSA	RC4(40)	MD5	export
EXP-RC2-CBC-MD5	SSLv2	RSA(512)	RSA	RC2(40)	MD5	export
EXP-RC4-MD5	SSLv2	RSA(512)	RSA	RC4(40)	MD5	export
ADH-AES256-SHA	SSLv3	DH	none	AES(256)	SHA1	
ADH-AES128-SHA	SSLv3	DH	none	AES(128)	SHA1	
ADH-DES-CBC3-SHA	SSLv3	DH	none	3DES(168)	SHA1	
ADH-DES-CBC-SHA	SSLv3	DH	none	DES(56)	SHA1	
ADH-RC4-MD5	SSLv3	DH	none	RC4(128)	MD5	
EXP-ADH-DES-CBC-SHA	SSLv3	DH(512)	none	DES(40)	SHA1	export
EXP-ADH-RC4-MD5	SSLv3	DH(512)	none	RC4(40)	MD5	export
NULL-SHA	SSLv3	RSA	RSA	none	SHA1	
NULL-MD5	SSLv3	RSA	RSA	none	MD5	

Available cipher suites

Name	Version	Key exchange	Authentication	Encryption	Digest	Export
ECDHE-ECDSA-AES128-SHA	SSLv3	ECDH	ECDSA	AES(128)	SHA1	
ECDHE-ECDSA-AES256-SHA	SSLv3	ECDH	ECDSA	AES(256)	SHA1	
ECDHE-ECDSA-DES-CBC3-SHA	SSLv3	ECDH	ECDSA	3DES(168)	SHA1	
ECDHE-ECDSA-RC4-SHA	SSLv3	ECDH	ECDSA	RC4(128)	SHA1	
ECDHE-ECDSA-NULL-SHA	SSLv3	ECDH	ECDSA	none	SHA1	
ECDHE-RSA-AES256-SHA	SSLv3	ECDH	RSA	AES(256)	SHA1	
ECDHE-RSA-AES128-SHA	SSLv3	ECDH	RSA	AES(128)	SHA1	
ECDHE-RSA-DES-CBC3-SHA	SSLv3	ECDH	RSA	3DES(168)	SHA1	
ECDHE-RSA-RC4-SHA	SSLv3	ECDH	RSA	RC4(128)	SHA1	
ECDHE-RSA-NULL-SHA	SSLv3	ECDH	RSA	none	SHA1	
AECDH-AES256-SHA	SSLv3	ECDH	none	AES(256)	SHA1	
AECDH-AES128-SHA	SSLv3	ECDH	none	AES(128)	SHA1	
AECDH-DES-CBC3-SHA	SSLv3	ECDH	none	3DES(168)	SHA1	
AECDH-RC4-SHA	SSLv3	ECDH	none	RC4(128)	SHA1	
AECDH-NULL-SHA	SSLv3	ECDH	none	none	SHA1	
DHE-RSA-CAMELLIA256-SHA	SSLv3	DH	RSA	Camellia(256)	SHA1	
DHE-RSA-CAMELLIA128-SHA	SSLv3	DH	RSA	Camellia(128)	SHA1	
DHE-DSS-CAMELLIA256-SHA	SSLv3	DH	DSS	Camellia(256)	SHA1	
DHE-DSS-CAMELLIA128-SHA	SSLv3	DH	DSS	Camellia(128)	SHA1	
CAMELLIA256-SHA	SSLv3	RSA	RSA	Camellia(256)	SHA1	
CAMELLIA128-SHA	SSLv3	RSA	RSA	Camellia(128)	SHA1	
ADH-CAMELLIA256-SHA	SSLv3	DH	none	Camellia(256)	SHA1	
ADH-CAMELLIA128-SHA	SSLv3	DH	none	Camellia(128)	SHA1	
ECDHE-ECDSA-AES256-GCM-SHA384	TLSv1.2	ECDH	ECDSA	AESGCM(256)	AEAD	
ECDHE-ECDSA-AES128-GCM-SHA256	TLSv1.2	ECDH	ECDSA	AESGCM(128)	AEAD	
ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	ECDH	RSA	AESGCM(256)	AEAD	
ECDHE-RSA-AES128-GCM-SHA256	TLSv1.2	ECDH	RSA	AESGCM(128)	AEAD	
AES256-GCM-SHA384	TLSv1.2	RSA	RSA	AESGCM(256)	AEAD	
AES128-GCM-SHA256	TLSv1.2	RSA	RSA	AESGCM(128)	AEAD	
DHE-RSA-AES256-GCM-SHA384	TLSv1.2	DH	RSA	AESGCM(256)	AEAD	
DHE-RSA-AES128-GCM-SHA256	TLSv1.2	DH	RSA	AESGCM(128)	AEAD	
DHE-DSS-AES356-GCM-SHA384	TLSv1.2	DH	DSS	AESGCM(256)	AEAD	

Available cipher suites

Name	Version	Key exchange	Authentication	Encryption	Digest	Export
DHE-DSS-AES128-GCM-SHA256	TLSv1.2	DH	DSS	AESGCM(128)	AEAD	
ADH-AES256-GCM-SHA384	TLSv1.2	DH	none	AESGCM(256)	AEAD	
ADH-AES128-GCM-SHA256	TLSv1.2	DH	none	AESGCM(128)	AEAD	
ECDHE-ECDSA-AES256-SHA384	TLSvv1.2	ECDH	ECDSA	AES(256)	SHA384	
ECDHE-ECDSA-AES128-SHA256	TLSv1.2	ECDH	ECDSA	AES(128)	SHA256	
ECDHE-RSA-AES256-SHA384	TLSv1.2	ECDH	RSA	AES(256)	SHA384	
ECDHE-RSA-AES128-SHA256	TLSv1.2	ECDH	RSA	AES(128)	SHA256	
DHE-RSA-AES256-SHA256	TLSv1.2	DH	RSA	AES(256)	SHA256	
DHE-RSA-AES128-SHA256	TLSv1.2	DH	RSA	AES(128)	SHA256	
DHE-DSS-AES256-SHA256	TLSv1.2	DH	DSS	AES(256)	SHA256	
DHE-DSS-AES128-SHA256	TLSv1.2	DH	DSS	AES(128)	SHA256	
AES256-SHA256	TLSv1.2	RSA	RSA	AES(256)	SHA256	
AES128-SHA256	TLSv1.2	RSA	RSA	AES(128)	SHA256	
ADH-AES256-SHA256	TLSv1.2	DH	none	AES(256)	SHA256	
ADH-AES128-SHA256	TLSv1.2	DH	none	AES(128)	SHA256	

Available cipher suites

Related publications

The manuals are available as online manuals, see *http://manuals.ts.fujitsu.com*, or in printed form which must be paid and ordered separately at *http://manualshop.ts.fujitsu.com*.

openNet Server
BCAM
User Guide

interNet Services
User Guide

openNet Server
IPv6 Introduction and Conversion Guide, Stage 1
User Guide

openNet Server
IPSec
User Guide

CRYPT
Security with Cryptography
User Guide

openNet Server, interNet Services (BS2000/OSD)
SNMP Management for openNet Server and interNet Services
User Guide

SOCKETS(BS2000)
SOCKETS for BS2000/OSD
User Guide

POSIX (BS2000/OSD)
SOCKETS/XTI for POSIX
User Guide

CMX (BS2000)
Communication Method in BS2000
User Guide

SNMP Management V5.0
SNMP Management for BS2000/OSD
User Guide

SNMP Management V6.0
SNMP Management for BS2000/OSD
User Guide

C Library Functions (BS2000)
for POSIX Applications
Reference Manual

XHCS (BS2000/OSD)
8-Bit Code Processing in BS2000/OSD
User Guide

BS2000
User Commands (ISP Format)
User Guide

BS2000 OSD/BC
Commands
User Guide

BS2000 OSD/BC
Executive Macros
User Guide

IMON (BS2000/OSD)
Installation Monitor
User Guide

BS2000 OSD/BC
System Administration
User Guide

Additional publications

SSL and TLS
Designing and Building Secure Systems
by Eric Rescorla
ISBN 0-201-61598-3

Contents
Detailed description of SSL and TLS and of the application environment

Secrets & Lies
Digital Security in a Networked World
by Bruce Schneier
ISBN 0-471-25311-1

Contents
Overview of IT security

Postfix
The Definitive Guide
by Kyle D. Dent
ISBN 0-596-00212-2

Contents
Relatively compact and up-to-date introduction to configuring and using the Postfix mail server.

http://www.oreilly.de/catalog/postfix/index.html

Postfix
by Richard Blum
ISBN 0-672-32114-9

Contents
Detailed description of how to configure and use the Postfix mail server. The May 2001 edition is no longer quite up-to-date, and there may be slight differences as far as BS2000/OSD porting concerned. In addition, the focus is somewhat Linux-oriented. Nevertheless, the level of detail makes it a noteworthy book.

The Book of Postfix
State-of-the-Art Message Transport
by Ralf Hildebrandt and Patrick Koetter
ISBN 1-59327-001-1

Contents
Extensive treatment of the issues of mail transport constraints and mail filtering. Of the books listed here, this one contains the most detailed description of TLS use. Theory-oriented chapters are complemented by detailed practical examples (generally on the basis of Linux systems).

http://www.postfix-book.com/index.html

RFCs

Comprehensive information on the Requests for Comments (RFCs) is available on the Internet Engineering Task Force's (IETF) home page:

www.ietf.org

Index

N

Q

R

T

U

V