

# interNet Services V3.4A

## Administrator Guide

## **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](mailto: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](http://www.cognitas.de)

## **Copyright and Trademarks**

Copyright © Fujitsu Technology Solutions GmbH 2010.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

---

# Contents

<b>1</b>	<b>Preface . . . . .</b>	<b>13</b>
<b>1.1</b>	<b>Target group of the manual . . . . .</b>	<b>13</b>
<b>1.2</b>	<b>Summary of contents . . . . .</b>	<b>14</b>
<b>1.3</b>	<b>Licensing regulations . . . . .</b>	<b>15</b>
<b>1.4</b>	<b>Overview of interNet Services . . . . .</b>	<b>19</b>
<b>1.5</b>	<b>Changes compared to the previous version . . . . .</b>	<b>20</b>
<b>1.6</b>	<b>Notational conventions . . . . .</b>	<b>21</b>
	SDF syntax representation . . . . .	22
<b>1.7</b>	<b>README files . . . . .</b>	<b>31</b>
<b>2</b>	<b>Installing/uninstalling the services without FTP, TELNET and Mail . . . . .</b>	<b>33</b>
<b>2.1</b>	<b>Installation . . . . .</b>	<b>33</b>
<b>2.2</b>	<b>Uninstallation . . . . .</b>	<b>35</b>
<b>2.3</b>	<b>Initial operation . . . . .</b>	<b>35</b>
<b>2.4</b>	<b>Shutting down . . . . .</b>	<b>36</b>
<b>3</b>	<b>Installing FTP and TELNET . . . . .</b>	<b>37</b>
<b>3.1</b>	<b>Installing FTP and TELNET via SDF command . . . . .</b>	<b>38</b>
<b>3.2</b>	<b>Installing and uninstalling FTP and TELNET clients in POSIX . . . . .</b>	<b>47</b>
<b>3.3</b>	<b>Frequently asked questions (FAQ) . . . . .</b>	<b>48</b>

<b>4</b>	<b>FTP configuration and operation</b>	<b>53</b>
<b>4.1</b>	<b>TLS/SSL support on the FTP server</b>	<b>54</b>
4.1.1	Parameterization of TLS/SSL support on the FTP server	54
4.1.2	FTP login commands	56
<b>4.2</b>	<b>Configuration of FTP using the SET-FTP-TELNET-PARAMETERS installation command</b>	<b>57</b>
<b>4.3</b>	<b>Configuring FTP via the option file</b>	<b>69</b>
	-appPrefix   -A	70
	-FTAClevel   -B	71
	-childName   -C	72
	-debugLevel   -D	72
	-serverInfoFile   -E	73
	-childEnterJob   -F	73
	-childJobClass   -J	74
	-FTACuserld   -K	74
	-logonExtension   -L	75
	-maxConn   -N	75
	-timeout   -O	76
	-portNumber   -P	76
	-DSSidLength   -S	77
	-socketTraceLevel   -T	77
	-convSelector   -U	78
	-verbose   -V	78
	-systemExit   -X	79
	-initialChildCmds   -Z	80
	-acctActive	81
	-acctFile	81
	-allowTsosLogin	82
	-defaultFTACsecurityLevel	83
	-disableSiteExecCommand	84
	-disableSizeCommand	85
	-tlsProtocol	86
	-tlsCipherSuite	87
	-tlsRSACertificateFile	93
	-tlsRSAkeyFile	94
	-tlsDSACertificateFile	95
	-tlsDSAkeyFile	96
	-tlsCertificateChainFile	97
	-tlsCAcertificateFile	98
	-tlsAcceptableClientCAFile	99
	-tlsCArevocationFile	100

-tlsVerifyClient . . . . .	101
-tlsVerifyDepth . . . . .	102
-tlsSecureControlConnection . . . . .	103
-tlsSecureDataConnection . . . . .	104
-tlsRandFile . . . . .	105
-tlsUseCryptoHardware . . . . .	106
-tlsOpenSSLlibName . . . . .	107
-sizeCmdTimeLimit . . . . .	107
-cmdBufSize . . . . .	108
<b>4.4 Starting up and shutting down the FTP server . . . . .</b>	<b>109</b>
4.4.1 Starting up the FTP server . . . . .	109
4.4.2 Shutting down the FTP server . . . . .	110
4.4.3 Setting up further FTP server tasks . . . . .	110
4.4.4 Shutdown . . . . .	112
4.4.5 Notes and restrictions for starting up and shutting down servers . . . . .	112
4.4.6 Messages and return codes . . . . .	113
<b>4.5 Notes on installing FTAC functionality . . . . .</b>	<b>114</b>
<b>4.6 Accounting in FTP . . . . .</b>	<b>115</b>
4.6.1 Enabling/disabling accounting and defining the accounting file . . . . .	115
4.6.2 Structure of the accounting records . . . . .	115
<b>4.7 Logging file of the FTP server . . . . .</b>	<b>119</b>
<b>4.8 Displaying the current settings of the FTP server . . . . .</b>	<b>119</b>
<b>4.9 Console interface . . . . .</b>	<b>122</b>
debug - Enable / disable trace to user level . . . . .	122
trace - Enable / disable trace to TCP/IP interface . . . . .	123
shutdown - Shut down service . . . . .	123
acctActive - Enable / disable FTP accounting . . . . .	124
acctFile - Change accounting file for FTP accounting . . . . .	124
RDPROT - Save logging file of FTP servers . . . . .	125
<b>4.10 IPv6 addresses in FTP . . . . .</b>	<b>126</b>
4.10.1 Control connection setup . . . . .	126
4.10.2 Data connection setup . . . . .	126
4.10.3 Proxy mechanism . . . . .	128
4.10.4 Notes on using heterogeneous networks . . . . .	128
<b>4.11 SNMP subagent for FTP . . . . .</b>	<b>129</b>
<b>4.12 FTP exit . . . . .</b>	<b>130</b>
4.12.1 FTP system exit . . . . .	130
4.12.1.1 FTP system exit events . . . . .	131

4.12.2	Exit mechanisms for the FTP server and FTP client . . . . .	139
4.12.2.1	Dummy routines . . . . .	139
4.12.2.2	User-defined exit routines . . . . .	140
4.12.2.3	Enabling / disabling user-defined exit routines . . . . .	143
<b>5</b>	<b>TELNET configuration and operation . . . . .</b>	<b>145</b>
<b>5.1</b>	<b>TLS/SSL support on the TELNET server . . . . .</b>	<b>146</b>
<b>5.2</b>	<b>Configuring TELNET using the SET-FTP-TELNET-PARAMETERS installation command . . . . .</b>	<b>148</b>
<b>5.3</b>	<b>Configuring TELNET using an option file . . . . .</b>	<b>158</b>
5.3.1	Options for general configuration of the TELNET server . . . . .	160
	-A - Specify a name prefix . . . . .	160
	-D - Debug level . . . . .	160
	-E - Specify exit routines . . . . .	161
	-N - Specify the number of connections . . . . .	161
	-P - Specify port number . . . . .	162
	-S - Define terminal name . . . . .	162
	-T - Specify socket trace level . . . . .	163
	-V - Enable/disable verbose . . . . .	163
	-X - Select code tables . . . . .	164
5.3.2	Options for safe use of TELNET with the aid of authentication and encryption . . . . .	165
5.3.3	-Z option - Support of the START-TLS option . . . . .	166
	-Z tls-required . . . . .	167
	-Z RSACertificateFile . . . . .	168
	-Z RSAKeyFile . . . . .	169
	-Z DSACertificateFile . . . . .	170
	-Z DSAKeyFile . . . . .	171
	-Z CACertificateFile . . . . .	172
	-Z CARevocationFile . . . . .	173
	-Z CipherSuite . . . . .	174
	-Z RandFile . . . . .	179
	-Z CertificateChainFile . . . . .	180
	-Z Protocol . . . . .	181
	-Z AcceptableClientCAFile . . . . .	182
	-Z VerifyDepth . . . . .	183
	-Z VerifyClient . . . . .	184
	-Z OpenSSLLibName . . . . .	185
	-Z UseCryptoHardware . . . . .	186
5.3.4	-B option - Enable/disable the AUTHENTICATION option . . . . .	187
5.3.5	Option -H - Enable/disable the ENCRYPTION option . . . . .	188

---

<b>5.4</b>	<b>Starting up and shutting down the TELNET server</b>	<b>189</b>
5.4.1	Starting up the TELNET server	189
5.4.2	Shutting down the TELNET server	189
5.4.3	Shutdown	190
5.4.4	Notes and restrictions for starting up and shutting down servers	190
5.4.5	Messages and return codes	191
<b>5.5</b>	<b>Logging file of TELNET servers</b>	<b>192</b>
<b>5.6</b>	<b>Displaying the current settings of TELNET servers</b>	<b>192</b>
<b>5.7</b>	<b>Console interface</b>	<b>195</b>
	debug - Enable / disable trace to user level	195
	trace - Enable / disable trace to TCP/IP interface	196
	shutdown - Shut down service	196
	RDPROT - Save logging file of TELNET servers	197
<b>5.8</b>	<b>IPv6 addresses in TELNET</b>	<b>197</b>
<b>5.9</b>	<b>TELNET exits</b>	<b>198</b>
5.9.1	DUMMY module	198
5.9.2	Exit routines	199
5.9.3	User-defined exits	202
<b>6</b>	<b>Generating random numbers</b>	<b>205</b>
<b>6.1</b>	<b>Generating random numbers in BS2000/OSD with PRNGD</b>	<b>206</b>
6.1.1	Entropy sources of the BS2000/OSD PRNGD	206
6.1.2	Configuration of the BS2000/OSD PRNGD	207
	poolSize	208
	minimalEntropy	208
	entropyThreshold	209
	seedFile	209
	file	210
	fileInterval	210
	cmd	211
	cmdInterval	212
	bcamInterval	212
6.1.3	GPRBYTE program interface of the BS2000/OSD PRNGD	213
6.1.4	Messages	217
<b>6.2</b>	<b>Random number generation in POSIX with the prngd server daemon</b>	<b>219</b>

<b>7</b>	<b>DNS</b>	<b>221</b>
<b>7.1</b>	<b>Concept of the DNS</b>	<b>223</b>
7.1.1	Development of the DNS	223
7.1.2	DNS name space	224
7.1.3	Storing information in the DNS	227
7.1.4	Format of a DNS message	228
7.1.5	DNS resolver (overview)	229
7.1.6	DNS name server NAMED (overview)	231
7.1.7	DNS security concepts	233
7.1.8	Interaction of the security mechanisms of BCAM and DNS	234
<b>7.2</b>	<b>DNS resolver</b>	<b>235</b>
7.2.1	Installing and uninstalling the DNS resolver	235
7.2.2	Configuring the DNS resolver	238
	nameserver entry	239
	domain entry	240
	search entry	242
	options entry	244
7.2.3	DNS resolver - administration and operation	246
7.2.3.1	Startup and shutdown of the DNS resolver	246
7.2.3.2	Modifying the DNS resolver configuration file	247
7.2.4	DNS resolver - diagnosis and maintenance	248
7.2.4.1	DNS resolver - logging	248
7.2.4.2	DNS resolver - diagnostic options	248
<b>7.3</b>	<b>DNS name server NAMED</b>	<b>249</b>
7.3.1	Installing and uninstalling NAMED	249
7.3.2	Configuring NAMED	253
7.3.2.1	NAMED configuration file named.conf	253
7.3.2.2	NAMED zone data files	254
7.3.2.3	NAMED and security	257
7.3.3	NAMED - administration and operation	259
7.3.3.1	Starting up and shutting down NAMED	259
7.3.3.2	Modifying the zone data files of NAMED	260
7.3.4	NAMED - diagnosis and maintenance	262
7.3.4.1	NAMED - logging	262
7.3.4.2	NAMED - diagnostic options	263
<b>7.4</b>	<b>DNS tools</b>	<b>264</b>
7.4.1	Diagnostic tool dig - examples	265

---

<b>8</b>	<b>NTP</b> . . . . .	<b>269</b>
<b>8.1</b>	<b>NTP concept</b> . . . . .	<b>269</b>
8.1.1	NTP functionality . . . . .	269
8.1.2	Implementing NTP in BS2000/OSD . . . . .	272
<b>8.2</b>	<b>Installing and uninstalling NTP</b> . . . . .	<b>274</b>
8.2.1	Installation . . . . .	274
8.2.2	Uninstallation . . . . .	276
<b>8.3</b>	<b>Startup and shutdown of NTP</b> . . . . .	<b>277</b>
8.3.1	Starting and shutting down NTP . . . . .	277
8.3.2	NTP time synchronization . . . . .	277
8.3.3	Creating the NTP daemon ntpd configuration file . . . . .	279
8.3.4	Startup options of the NTP daemon ntpd . . . . .	291
8.3.5	Setting the date and time via NTP with the ntpdate program . . . . .	293
8.3.6	Generating cryptographic files for NTPv4 authentication using the ntp-keygen program . . . . .	295
<b>8.4</b>	<b>Administration and operation</b> . . . . .	<b>296</b>
8.4.1	Querying the NTP status via command-line options . . . . .	296
8.4.2	Querying the NTP status interactively with commands . . . . .	297
8.4.2.1	Querying the NTP status with internal commands of ntpq . . . . .	298
8.4.2.2	Querying the NTP status with commands for control messages . . . . .	301
<b>8.5</b>	<b>Diagnosis and maintenance of NTP</b> . . . . .	<b>305</b>
8.5.1	Logging function . . . . .	305
8.5.2	Trace functionality of NTP . . . . .	306
8.5.2.1	ntptrace - Trace a chain of NTP servers back to the prevailing clock . . . . .	306
<b>9</b>	<b>OpenSSH</b> . . . . .	<b>307</b>
<b>9.1</b>	<b>Concept of OpenSSH</b> . . . . .	<b>308</b>
9.1.1	Component parts of the OpenSSH protocol suite . . . . .	308
9.1.2	Network security with OpenSSH . . . . .	309
9.1.3	Features of OpenSSH . . . . .	310
<b>9.2</b>	<b>Installing and uninstalling OpenSSH</b> . . . . .	<b>312</b>
9.2.1	Installing OpenSSH . . . . .	312
9.2.2	Uninstalling OpenSSH . . . . .	315
<b>9.3</b>	<b>OpenSSH server daemon sshd</b> . . . . .	<b>316</b>
9.3.1	Configuring the OpenSSH server daemon sshd . . . . .	316
9.3.2	Starting and stopping sshd . . . . .	317
9.3.3	Internal procedure when setting up a connection between sshd and ssh . . . . .	317

---

## Contents

---

9.3.4	Authentication between OpenSSH client ssh and server sshd . . . . .	320
9.3.5	Login process . . . . .	323
9.3.6	Files of the OpenSSH server daemon sshd . . . . .	324
<b>9.4</b>	<b>BS2000/OSD-specific restrictions . . . . .</b>	<b>327</b>
<b>10</b>	<b>Mail servers in POSIX . . . . .</b>	<b>329</b>
<hr/>		
<b>10.1</b>	<b>Overview . . . . .</b>	<b>331</b>
<b>10.2</b>	<b>Functionality . . . . .</b>	<b>332</b>
<b>10.3</b>	<b>Installing and uninstalling the mail servers . . . . .</b>	<b>334</b>
10.3.1	Installing and uninstalling the Postfix server (SMTP server) . . . . .	334
10.3.2	Installing and uninstalling the IMAP and POP3 servers . . . . .	340
<b>10.4</b>	<b>Starting up Mail servers . . . . .</b>	<b>343</b>
10.4.1	Starting up the Postfix server (SMTP server) . . . . .	343
10.4.2	Starting up IMAP and POP3 servers . . . . .	344
10.4.3	TLS/SSL protection of IMAP/POP3 and SMTP connections . . . . .	345
<b>10.5</b>	<b>Operating the Postfix server . . . . .</b>	<b>348</b>
10.5.1	Postfix lookup tables (index files) . . . . .	348
10.5.2	Programs for operating the Postfix mail server . . . . .	352
	postfix - Starting and stopping the Postfix server . . . . .	352
	postconf - Displaying and modifying Postfix configuration parameters . . . . .	353
	postqueue (mailq) - Processing mail queues (as a normal user) . . . . .	354
	postsuper - Processing mail queues (with SYSROOT authorization) . . . . .	355
	postcat - Displaying the contents of messages in the mail queues . . . . .	356
	postmap - Generating and processing index files (Postfix format) . . . . .	357
	postalias - Generating and processing index files (alias format) . . . . .	359
	newaliases - Generating index files (alias format) . . . . .	360
<b>10.6</b>	<b>Migration from Sendmail to Postfix . . . . .</b>	<b>361</b>
<b>11</b>	<b>Mail senders in BS2000/OSD . . . . .</b>	<b>363</b>
<hr/>		
<b>11.1</b>	<b>Installing and uninstalling mail senders . . . . .</b>	<b>363</b>
<b>11.2</b>	<b>Option files . . . . .</b>	<b>363</b>
11.2.1	SYSSSI . . . . .	363
	defaultOptionFileName . . . . .	364
	backendConfigurationFileName . . . . .	365
	senderSuffix . . . . .	365
	useSenderSuffix . . . . .	366

11.2.2	Configuration file for the mail sender backend . . . . .	367
	logFile . . . . .	367
	logLevel . . . . .	368
	logMailContent . . . . .	369
	mailServer . . . . .	370
	mailServerPort . . . . .	370
	mailLogLevel . . . . .	371
	mailLogFile . . . . .	372
	retry Limit . . . . .	372
	tempFilePrefix . . . . .	373
	tlsSecureConnection . . . . .	373
	tlsProtocol . . . . .	374
	tlsCipherSuite . . . . .	375
	tlsCertificateFile . . . . .	380
	tlsKeyFile . . . . .	380
	tlsCACertificateFile . . . . .	381
	tlsCARevocationFile . . . . .	382
	tlsVerifyServer . . . . .	382
	tlsVerifyDepth . . . . .	383
	tlsUseCryptoHardware . . . . .	383
11.3	<b>Mail service commands . . . . .</b>	<b>384</b>
	START-MAIL-SERVICE . . . . .	384
	MODIFY-MAIL-SERVICE-PARAMETER . . . . .	386
	SHOW-MAIL-SERVICE-PARAMETER . . . . .	389
	STOP-MAIL-SERVICE . . . . .	391
11.4	<b>Messages . . . . .</b>	<b>392</b>
	<b>Related publications . . . . .</b>	<b>399</b>
	<b>Index . . . . .</b>	<b>403</b>



---

# 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/OSD
- Mail Reader in BS2000/OSD
- Mail Server in POSIX

## 1.1 Target group of the manual

This Administrator Guide is intended for BS2000/OSD system administrators who want to install and operate interNet Services on BS2000/OSD. Knowledge of the BS2000/OSD 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 PRNGD, 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/OSD 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.

## 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](mailto: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 routines 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.
 */
```

## 1.4 Overview of interNet Services

interNet Services V3.4A requires the following software:

- BS2000/OSD  $\geq$  V6.0A
- openNet Server  $\geq$  V3.0

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 V3.4A and relevant to this manual. Innovations with respect to interNet Services V3.2 and interNet Services V3.3A are listed. (The documentation for interNet Services V3.3A consists of manual supplements in TXT format for the User and System Administrator Guides.)

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

### Supplementary and revised chapters

- The NTP chapter has been revised because of the migration to NTPv4.
- Mail sender:  
Additional options in the SYSSSI file and the configuration file for the mail sender backend plus additional messages for the subsystem.

### README files

The README files listed below were included in the manual:

- MAIL (BS2000/OSD) V3.2A
- TCP-IP-AP V5.1A

### The following sections are no longer included:

- The section "Installing FTP and TELNET via FHS masks" has been omitted as this installation method is no longer supported.
- The section "NAMED process models" has been omitted as the thread process model is no longer supported.

## 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 alternatives 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 *.	<b>HELP-SDF</b>
<b>UPPERCASE LETTERS</b> in boldface	Uppercase letters printed in boldface denote guaranteed or suggested abbreviations of keywords.	<b>SCREEN-STEPS = *NO</b>
=	The equals sign connects an operand name with the associated operand values.	<b>GUIDANCE-MODE = *YES</b>
< >	Angle brackets denote variables whose range of values is described by data types and suffixes (see Tables 2 and 3).	<b>GUIDANCE-MODE = *NO</b>
<u>Underscoring</u>	Underscoring denotes the default value of an operand.	<b>SYNTAX-FILE = &lt;filename 1..54&gt;</b>
/	A slash serves to separate alternative operand values.	<b>GUIDANCE-MODE = *NO</b>
(...)	Parentheses denote operand values that initiate a structure.	<b>NEXT-FIELD = *NO / *YES</b>
[ ]	Square brackets denote operand values which introduce a structure and are optional. The subsequent structure can be specified without the initiating operand value.	<b>,UNGUIDED-DIALOG = *YES (...) / *NO</b>
Indentation	Indentation indicates that the operand is dependent on a higher-ranking operand.	<b>SELECT = [*BY-ATTRIBUTES](...)</b>
		<b>GUIDED-DIALOG = *YES (...)</b> *YES(...)   <b>SCREEN-STEPS = *NO / *YES</b>

Table 1: Metasyntax

(part 1 of 2)



## Data types

Data type	Character set	Special rules
alphanumeric-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	<p>Input format:</p> $[:cat:][\$user.] \left\{ \begin{array}{l} \text{file} \\ \text{file(no)} \\ \text{group} \end{array} \right\}$ $\text{group} \left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\}$ <p>: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.</p> <p>\$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.</p> <p>\$. (special case)                      system default ID</p> <p>file                      file or job variable name;                      may be split into a number of partial names                      using a period as a delimiter:                      name<sub>1</sub>[.name<sub>2</sub>[...]]                      name<sub>i</sub> 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.</p>

Table 2: Data types

(part 2 of 6)

Data type	Character set	Special rules
filename (contd.)		<p>#file (special case)  @file (special case)  # or @ used as the first character indicates temporary files or job variables, depending on system generation.</p> <p>file(no)  tape file name  no: version number;  character set is A...Z, 0...9, \$, #, @.  Parentheses must be specified.</p> <p>group  name of a file generation group  (character set: as for "file")</p> <p>group <math>\left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\}</math></p> <p>(*abs)  absolute generation number (1-9999);  * and parentheses must be specified.</p> <p>(+rel)  (-rel)  relative generation number (0-99);  sign and parentheses must be specified.</p>
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	<p>Input format: [:cat:][\$user.][partname.]</p> <p>:cat: see filename \$user. see filename</p> <p>partname optional entry of the initial part of a name common to a number of files or file generation groups in the form: name<sub>1</sub>. [name<sub>2</sub>. [...]] name<sub>i</sub> (see filename). The final character of “partname” must be a period. At least one of the parts :cat:, \$user. or partname must be specified.</p>
posix-filename	A...Z 0...9 special characters	<p>String with a length of up to 255 characters; consists of either one or two periods or of alphanumeric 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 alternative data types are permitted, separators are used, or the first character is a ?, ! or ^. A distinction is made between uppercase and lowercase.</p>
posix-pathname	A...Z 0...9 special characters structure identifier: slash	<p>Input format: [/]part<sub>1</sub>/.../part<sub>n</sub> where part<sub>i</sub> is a posix-filename; max. 1023 characters; must be enclosed within single quotes if alternative data types are permitted, separators are used, or the first character is a ?, ! or ^.</p>

Table 2: Data types

(part 4 of 6)

Data type	Character set	Special rules
product-version	A...Z 0...9 period single quote	<p>Input format: <code>[[C]']][V][m]m.naso[']</code></p> <div style="text-align: right; margin-right: 20px;"> <math display="block">\begin{array}{c}   \\   \\   \\ \hline \text{correction status} \\ \text{release status} \end{array}</math> </div> <p>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.</p>
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	<p>Time-of-day entry:</p> <p>Input format: <math>\left. \begin{array}{l} \text{hh:mm:ss} \\ \text{hh:mm} \\ \text{hh} \end{array} \right\}</math></p> <p>hh: hours mm: minutes ss: seconds <math>\left. \begin{array}{l} \\ \\ \end{array} \right\}</math> Leading zeros may be omitted</p>
vsn	<p>a) A...Z 0...9</p> <p>b) A...Z 0...9 \$, #, @</p>	<p>a) Input format: pvsid.sequence-no max. 6 characters pvsid: 2-4 characters; PUB must not be entered sequence-no: 1-3 characters</p> <p>b) Max. 6 characters; PUB may be prefixed, but must not be followed by \$, #, @.</p>

Table 2: Data types

(part 5 of 6)

<b>Data type</b>	<b>Character set</b>	<b>Special rules</b>
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												
<i>x..y unit</i>	<p>With data type “integer”: interval specification</p> <p><i>x</i> minimum value permitted for “integer”. <i>x</i> is an (optionally signed) integer.</p> <p><i>y</i> maximum value permitted for “integer”. <i>y</i> is an (optionally signed) integer.</p> <p><i>unit</i> with “integer” only: additional units. The following units may be specified:</p> <table style="margin-left: 40px;"> <tr> <td><i>days</i></td> <td><i>byte</i></td> </tr> <tr> <td><i>hours</i></td> <td><i>2Kbyte</i></td> </tr> <tr> <td><i>minutes</i></td> <td><i>4Kbyte</i></td> </tr> <tr> <td><i>seconds</i></td> <td><i>Mbyte</i></td> </tr> <tr> <td>milliseconds</td> <td></td> </tr> </table>	<i>days</i>	<i>byte</i>	<i>hours</i>	<i>2Kbyte</i>	<i>minutes</i>	<i>4Kbyte</i>	<i>seconds</i>	<i>Mbyte</i>	milliseconds			
<i>days</i>	<i>byte</i>												
<i>hours</i>	<i>2Kbyte</i>												
<i>minutes</i>	<i>4Kbyte</i>												
<i>seconds</i>	<i>Mbyte</i>												
milliseconds													
<i>x..y special</i>	<p>With the other data types: length specification</p> <p>For data types <i>catid</i>, <i>date</i>, <i>device</i>, <i>product-version</i>, <i>time</i> and <i>vsn</i> the length specification is not displayed.</p> <p><i>x</i> minimum length for the operand value; <i>x</i> is an integer.</p> <p><i>y</i> maximum length for the operand value; <i>y</i> is an integer.</p> <p><i>x=y</i> the length of the operand value must be precisely <i>x</i>.</p> <p><i>special</i> 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:</p> <table style="margin-left: 40px;"> <tr> <td><i>arithm-expr</i></td> <td>arithmetic expression (SDF-P)</td> </tr> <tr> <td><i>bool-expr</i></td> <td>logical expression (SDF-P)</td> </tr> <tr> <td><i>string-expr</i></td> <td>string expression (SDF-P)</td> </tr> <tr> <td><i>expr</i></td> <td>freely selectable expression (SDF-P)</td> </tr> <tr> <td><i>cond-expr</i></td> <td>conditional expression (JV)</td> </tr> <tr> <td><i>symbol</i></td> <td>CSECT or entry name (BLS)</td> </tr> </table>	<i>arithm-expr</i>	arithmetic expression (SDF-P)	<i>bool-expr</i>	logical expression (SDF-P)	<i>string-expr</i>	string expression (SDF-P)	<i>expr</i>	freely selectable expression (SDF-P)	<i>cond-expr</i>	conditional expression (JV)	<i>symbol</i>	CSECT or entry name (BLS)
<i>arithm-expr</i>	arithmetic expression (SDF-P)												
<i>bool-expr</i>	logical expression (SDF-P)												
<i>string-expr</i>	string expression (SDF-P)												
<i>expr</i>	freely selectable expression (SDF-P)												
<i>cond-expr</i>	conditional expression (JV)												
<i>symbol</i>	CSECT or entry name (BLS)												

Table 3: Data type suffixes

## 1.7 README files

Information on any functional changes and additions to the current product version described in this manual can be found in the product-specific README files. You will find the README files on your BS2000/OSD computer under the following file names:

```
SYSRME.TCP-IP-SV.nnn.E  
SYSRME.TCP-IP-AP.nnn.E  
SYSRME.MAIL.nnn.E.
```

*nnn* stands for the particular TCP-IP-SV, TCP-IP-AP or MAIL version.

The user ID under which the README files are cataloged can be obtained from your system administrator. You can view a README file using the /SHOW-FILE command or an editor and print it out on a standard printer using the following command:

```
/PRINT-DOCUMENT <filename>,LINE-SPACING=*BY-EBCDIC-CONTROL
```



---

## 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 V3.4A software package must be installed component by component. The DNS, NAMED, PRNGD, 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 V3.4A 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 OPENSSSH

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 */var/adm/messages* message file. 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*
- For PRNGD: */etc/init.d/TCP-IP-SV.prngd 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*
- For PRNGD: */etc/init.d/TCP-IP-SV.prngd 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*
- For PRNGD: */etc/init.d/TCP-IP-SV.prngd 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 (PRNGD, NAMED, OPENSSSH) 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 39](#)), 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 69](#) or [page 158](#))
- Creation of ENTER files for the FTP and TELNET daemons
- Management of installation parameters in the installation file



TCP-IP-AP V5.0 or higher must be reinstalled. Files created by an older version of the installation program may not be used.

You can use the option files of the FTP and TELNET servers (see [page 69](#) and [page 158](#)) to modify the configuration of the FTP and TELNET servers without needing to restart installation with the aid of the SET-FTP-TELNET-PARAMETERS command.

Storing of the installation parameters in the installation file SYSDAT.TCP-IP-AP.*mm*.INSTALL no longer takes place for the new parameters supported as of interNetServices V3.0A.

**SET-FTP-TELNET-PARAMETERS**

**INPUT=** \*NO / \***INSTALLATION-FILE**(...)

\***INSTALLATION-FILE**(...)

**INSTALLATION-FILE=** \*STD / <filename 1..54\_without-generation-version>

    , **PARAMETER-RECORD=** \*STD / <name 1..8>

, **BS2000-VERSION=** \*STD / <alphanum-name 3..3>

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

\***CREATE**(...)

**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-CONNECTIONS=** \*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-LEVEL=** \*STD / <integer 0..3>

    , **FTAC-SUPPORT=** \*STD / \*NO / \***YES**(...)

        \***YES**(...)

**LEVEL=** \*STD / <integer 1..2>

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

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

            , **SERVER-INFORMATION-FILE=** \*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>

**SET-FTP-TELNET-PARAMETERS**

```
, CA-REVOCAION-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
, USE-CRYPTO-HARDWARE = *STD / *NO / *SERVER / *CHILD / *ALL
, 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-PARAMETERS**

, TELNET-SERVER-PROC= \*NO / \*CREATE(...)

\*CREATE(...)

**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-CONNECTIONS= \*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>**

      , **USE-CRYPTO-HARDWARE= \*STD / \*NO / \*YES**

  , **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-PARAMETERS**

```

, 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 / *CREATE
, INSTALLATION-FILE= *UNCHANGED / *NEW-FILE(...) / *UPDATE-FILE(...)
    / *DELETE-PARAMETER-RECORD(...)

*NEW-FILE(...)
    | FILE-NAME= *STD / <filename 1..54_without-generation-version>
    | , PARAMETER-RECORD= *STD / <name 1..8>

*UPDATE-FILE(...)
    | FILE-NAME= *STD / <filename 1..54_without-generation-version>
    | , PARAMETER-RECORD= *STD / <name 1..8>

*DELETE-PARAMETER-RECORD(...)
    | FILE-NAME= *STD / <filename 1..54_without-generation-version>
    | , PARAMETER-RECORD = *STD / <name 1..8>

```

## Shared operands of FTP and TELNET



For a description of the FTP operands, see [page 57](#).

For a description of the TELNET operands, see [page 148](#).

### INPUT

Determines whether or not an existing installation file may be used to preset the values.

#### **INPUT=\*NO**

No installation file is used.

#### **INPUT=\*INSTALLATION-FILE(...)**

The default values are already in an existing installation file.

##### **INSTALLATION-FILE= \*STD**

Corresponds to the entry SYSDAT.TCP-IP-AP.*nnn*.INSTALL.

##### **INSTALLATION-FILE=<filename 1..54\_without-generation-version>**

Specifies the name of the installation file to be used.

##### **PARAMETER-RECORD= \*STD**

Corresponds to the entry TCP.

##### **PARAMETER-RECORD=<name 1..8>**

Specifies the name of the required installation in the specified installation file.

### **BS2000-VERSION=**

Operating system version for which the installation is being performed.

#### **BS2000-VERSION=\*STD**

Corresponds to the current version or the value in the installation file, if specified.

#### **BS2000-VERSION=<alphanum-name 3..3>**

Three-digit version name.

### **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.

### **INSTALLATION-FILE=**

Parameter for managing the installation file.

#### **INSTALLATION-FILE=\*UNCHANGED**

There is no write access to an installation file.

**INSTALLATION-FILE=\*NEW-FILE(...)**

A new installation file is created and any specified parameters are stored in it.

**FILE-NAME=\*STD/<filename 1..54\_without-generation-version>**

Name of the new installation file to be created

Default setting: SYSDAT.TCP-IP-AP.*nnn*.INSTALL

**PARAMETER-RECORD=\*STD/<name 1..8>**

Specifies the name of the required installation in the specified installation file.

Default: TCP

**INSTALLATION-FILE=\*UPDATE-FILE(...)**

The data entered is stored in the specified installation file under the specified element name.

**FILE-NAME=\*STD/<filename 1..54\_without-generation-version>**

Name of the installation file

Default setting: SYSDAT.TCP-IP-AP.*nnn*.INSTALL

**PARAMETER-RECORD=\*STD/<name 1..8>**

Specifies the name of the installation to be stored.

Default setting: TCP

**INSTALLATION-FILE=\*DELETE-PARAMETER-RECORD(...)**

The installation with the specified element name will be deleted in the specified installation file.

**FILE-NAME=\*STD/<filename 1..54\_without-generation-version>**

Name of the installation file

Default setting: SYSDAT.TCP-IP-AP.*nnn*.INSTALL

**PARAMETER-RECORD=\*STD/<name 1..8>**

Specifies the name of the installation to be deleted.

Default: TCP

**Return codes of SET-FTP-TELNET-PARAMETERS commands**

SC2 SC1	Main code	Message
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)

<b>SC2 SC1</b>	<b>Main code</b>	<b>Message</b>
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)



### 3.3 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/OSD is:

```
SYSDAT.SOCKETS.mnn.SOC6.RESOLV or
SYSDAT.LWRESD.mnn.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/OSD-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 80](#))

- **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/OSD 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/OSD) 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 console command:

```
/INTR <tsn ftpserver>,debug 2
```

Repeat the login and, using the console command

```
/INTR <tsn ftpserver>,rdprot
```

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/OSD-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/OSD 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 `site <command>` and `site exec <command>`. What is the difference?

**Answer:**

- With `quote <command>` you send FTP commands that conform with the standard to the server.
- With `site <command>` you send BS2000/OSD-specific (“proprietary”) commands to the server which are not defined in the standard. These commands include *ftyp*, *cmod*, *modc*, *file*, *setc*, *sfil*.
- With `site exec <command>` you send BS2000/OSD 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/OSD. 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/OSD FTP server my non-BS2000/OSD 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/OSD 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:

- Using TLS/SSL to secure the FTP server (see [page 54](#))
- Configuration of FTP via SDF command (see [page 57](#)) or option file (see [page 69](#))
- Startup and shutdown of FTP servers (see [page 109](#))
- Notes on the installation and use of the FTAC functionality (see [page 114](#))
- Enabling/disabling accounting and specifying the accounting file (see [page 115](#))
- Saving the logging file of FTP servers (see [page 119](#))
- Displaying the current settings of the FTP server (see [page 119](#))
- Specification of commands using the console interface (see [page 122](#))
- Using IPv6 addresses in FTP (see [page 126](#))
- Notes on the use of the SNMP subagent for FTP (see [page 129](#))
- Exits for FTP clients and the FTP server (see [page 130](#))

## 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 69](#)).
- 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 57](#)).

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	<a href="#">86</a>
-tlsCipherSuite	Specify cipher suite preference list	<a href="#">87</a>
-tlsRSACertificateFile	Specify file which contains the RSA-based X.509 server certificate in PEM format	<a href="#">93</a>
-tlsRSAkeyFile	Specify file which contains the private RSA server key in PEM format	<a href="#">94</a>
-tlsDSACertificateFile	Specify file which contains the DSA-based X.509 server certificate in PEM format	<a href="#">95</a>
-tlsDSAkeyFile	Specify file which contains the private DSA server key in PEM format	<a href="#">96</a>
-tlsCertificateChainFile	Specify file in which all the certificates required for verification of the server certificate can be stored	<a href="#">97</a>
-tlsCAcertificateFile	Specify file which contains the certificates required for authentication of the FTP client in PEM format	<a href="#">98</a>
-tlsAcceptableClientCAFile	Specify file from which the names of the CAs that the server accepts as signatories of client certificates can be obtained	<a href="#">99</a>
-tlsCArevocationFile	Specify file which contains the CRLs of the CAs	<a href="#">100</a>
-tlsVerifyClient	Define whether the FTP client must provide a certificate for server access	<a href="#">101</a>
-tlsVerifyDepth	Define verification depth	<a href="#">102</a>
-tlsSecureControlConnection	Define whether the control connection from the FTP client to the FTP server is to be secured with TLS	<a href="#">103</a>
-tlsSecureDataConnection	Define whether the data connection from the FTP client to the server child is to be secured with TLS	<a href="#">104</a>
-tlsRandFile	Specify file from which the data for initializing the PRNG is read when the server is started	<a href="#">105</a>
-tlsUseCryptoHardware	Define whether crypto hardware is to be used for calculating crypto algorithms	<a href="#">106</a>
-tlsOpenSSLlibName	Define the LMS file from which the OpenSSL library should be dynamically loaded	<a href="#">107</a>

## 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: Yes, cipher: 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: on, but cleared
Private data channel: on
Cipher: DES-CBC3-SHA (168 bits)
```

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

**i** For the full command syntax and the description of the installation operands, see [page 38](#).

### SET-FTP-TELNET-PARAMETERS

(...)

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

\*CREATE(...)

**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-CONNECTIONS= \*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-LEVEL= \*STD** / <integer 0..3>

    , **FTAC-SUPPORT= \*STD** / \*NO / \*YES(...)

        \*YES(...)

**LEVEL= \*STD** / <integer 1..2>

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

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

            , **SERVER-INFORMATION-FILE= \*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-PARAMETERS

```

, CA-REVOCAION-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
, USE-CRYPTO-HARDWARE = *STD / *NO / *SERVER / *CHILD / *ALL
,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

```

(...)

**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 procedure 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 70](#).

**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 parameters TIME=NTL and PRIORITY=(120,EXPRESS) may be started in this job class.

**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 72](#).

**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 77](#).

**TRACE=\*STD**

Corresponds to the entry 0 or the value in the installation file.

**TRACE=<integer 0..9>**

Socket trace level.

**MAXIMUM-CONNECTIONS=**

Maximum number of connections that the server is to operate. See also the *-maxConn* option on [page 75](#).

**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 77](#).

**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 75](#).

**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 75](#).

**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 76](#).

**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 79](#).

**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 82](#) and the *-FTAClevel* option on [page 71](#).

**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 74](#).

**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 73](#).

**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 73](#).

**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 74](#).

**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 86](#).

**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 87](#).

**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 93](#).

**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 94](#).

**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 95](#).

**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 96](#).

**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 98](#).**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 99](#).**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 97](#).**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-REVOCAATION-FILE=**See the *-tlsCArevocationFile* option on [page 100](#).**CA-REVOCAATION-FILE=\*STD**

Default: \*NONE

**CA-REVOCACTION-FILE=\*NONE**

No CA revocation file is specified.

**CA-REVOCACTION-FILE=<filename 1..54\_without-generation-version>**

Name of the CA revocation file.

**RANDOM-FILE=**

See the *-tlsRandFile* option on [page 105](#).

**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 107](#).

**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 101](#).

**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 102](#).

**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 103](#).

**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 104](#).

**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.

**USE-CRYPTO-HARDWARE=**

See the *-tlsUseCryptoHardware* option on [page 106](#).

**USE-CRYPTO-HARDWARE=\*STD**

Default: \*NO

**USE-CRYPTO-HARDWARE=\*NO**

No crypto hardware is used.

**USE-CRYPTO-HARDWARE=\*SERVER**

Crypto hardware is used by the FTP server process.

**USE-CRYPTO-HARDWARE=\*CHILD**

Crypto hardware is used by the FTP child processes.

**USE-CRYPTO-HARDWARE=\*ALL**

Crypto hardware is used by all FTP server processes.

**ACCOUNTING=**

Specifies whether FTP accounting records are to be collected. See also the *-acctActive* and *-acctFile* options on [page 81](#).

**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 78](#).

**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 80](#).

**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 76](#).

**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 82](#).

**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.

## 4.3 Configuring FTP via the option file

When the SET-FTP-TELNET-PARAMETERS command is executed (see [page 38](#)) 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.

<b>-appPrefix</b> <b>-A</b>
<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.

<b>-FTAClevel</b>
<b>-B</b>
<b>0   1   2</b>

**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 82](#)).

**-childName | -C**

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

<b>-childName</b>
<b>-C</b>
<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.

<b>-debugLevel</b>
<b>-D</b>
<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.

<b>-serverInfoFile</b>
<b>-E</b>
<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.

<b>-childEnterJob</b>
<b>-F</b>
<file-name 1..54>

<file-name 1..54>

Name of the enter file.

Default: SYSDAT.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.

<b>-childJobClass</b> <b>-J</b>
<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/OSD user ID but with an FTAC transfer admission.

<b>-FTACuserid</b> <b>-K</b>
<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.

<b>-logonExtension</b>
<b>-L</b>
<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.

<b>-maxConn</b>
<b>-N</b>
<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.

<b>-timeout</b>
<b>-O</b>
<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.

<b>-portNumber</b>
<b>-P</b>
<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.

<b>-DSSidLength</b>
<b>-S</b>
<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.

<b>-socketTraceLevel</b>
<b>-T</b>
<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 139](#).

<b>-convSelector</b>
<b>-U</b>
<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*.

<b>-verbose</b>
<b>-V</b>

**-systemExit | -X**

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

<b>-systemExit</b>
<b>-X</b>
<b>0   1</b>

**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/OSD 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/OSD proprietary command *file*. The BS2000/OSD 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.

<b>-initialChildCmds</b> <b>-Z</b>
<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).

<b>-acctActive</b>
<b>ON   OFF</b>

### **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.

<b>-acctFile</b>
<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 71](#)).

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.

<b>-allowTsosLogin</b>
<b>NO   TLS   YES</b>

### **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.

<b>-defaultFTACsecurityLevel</b>
<integer 1..100>

&lt;integer 1..100&gt;

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/OSD 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 71](#)), 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.

<b>-disableSiteExecCommand</b>
<b>NO   YES</b>

**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, it 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.

<b>-disableSizeCommand</b>
<b>NO   YES</b>

### **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 2 and 3 of the SSL protocol and also Version 1 of the TLS protocol. Some of these protocols can be activated selectively using the *-tlsProtocol* option.

<b>-tlsProtocol</b>
[+   -] { <b>SSLv2</b>   <b>SSLv3</b>   <b>TLSv1</b>   <b>ALL</b> } ...

+

The protocol specified after this sign is permissible.

-

The protocol specified after this sign is not permissible.



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

### **SSLv2**

SSL protocol Version 3.



Version 2 of the SSL protocol displays some security-related deficiencies and should therefore not be used if possible.

### **SSLv3**

SSL protocol Version 3.

### **TLSv1**

TLS protocol Version 1.

### **ALL**

All protocols are to be enabled.

ALL -SSLv2 is the default.

### *Example*

The specifications `-tlsProtocol SSLv3 TLSv1` and `-tlsProtocol ALL -SSLv2` have the same effect.

## **-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.

<b>-tlsCipherSuite</b>
<specification>

<specification>

Specification of a cipher suite preference list (for details, see the description of the specification structure below).

ALL: !EXP: !ADH is the default.

### **Description of the specification structure**

The specification consists of one or more cipher mnemonics which are separated by a colon (:).

A cipher mnemonic can have 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 suite 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 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 using 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.

#### DEFAULT

The following cipher list is the default:

```
ALL : !ADH : RC4+RSA : +SSLv2 : @STRENGTH
```

If it is used, this cipher mnemonic must be at the start of the specification.

#### 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. As 3DES is interpreted with a length of 168 bits (instead of with 112 bits as many cryptographers do), it belongs to this suite class.

#### MEDIUM

Cipher suites with a key length of 128 bits.

#### 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.

**kEDH**

Cipher suites with ephemeral Diffie-Hellmann key negotiation.

**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, SSLv3, SSLv2**

TLSv1, SSLv3 and SSLv2 cipher suites. The TLSv1 suites and the SSLv3 suites are identical.

**DH**

Cipher suites with Diffie-Hellman key exchange, including anonymous exchange.

**ADH**

Cipher suites with anonymous Diffie-Hellman key exchange.

**AES**

Cipher suites with AES encryption (key length of 128 and 256 bits).

**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.

**CAMELLIA**

Cipher suites with CAMELLIA encryption.

**SEED**

Cipher suites with SEED encryption.

**MD5**

Cipher suites with MD5 hash function.

**SHA1, SHA**

Cipher suites with SHA1 hash function.

The available cipher suites are listed in the table below.

<b>Name</b>	<b>Version</b>	<b>Key exchange</b>	<b>Authentification</b>	<b>Encryption</b>	<b>Digest</b>	<b>Export</b>
DHE-RSA-AES256-SHA	SSLv3	DH	RSA	AES(256)	SHA1	
DHE-DSS-AES256-SHA	SSLv3	DH	DSS	AES(256)	SHA1	
DHE-RSA-CAMELLIA256-SHA	SSLv3	DH	RSA	CAMELLIA (256)	SHA1	
DHE-DSS-CAMELLIA256-SHA	SSLv3	DH	DSS	CAMELLIA (256)	SHA1	
AES256-SHA	SSLv3	RSA	RSA	AES(256)	SHA1	
CAMELLIA256-SHA	SSLv3	RSA	RSA	CAMELLIA(256)	SHA1	
DHE-RSA-AES128-SHA	SSLv3	DH	RSA	AES(128)	SHA1	
DHE-DSS-AES128-SHA	SSLv3	DH	DSS	AES(128)	SHA1	
DHE-RSA-SEED-SHA	SSLv3	DH	RSA	SEED	SHA1	
DHE-DSS-SEED-SHA	SSLv3	DH	DSS	SEED	SHA1	
DHE-RSA-CAMELLIA128-SHA	SSLv3	DH	RSA	CAMELLIA (128)	SHA1	
DHE-DSS-CAMELLIA128-SHA	SSLv3	DH	DSS	CAMELLIA (128)	SHA1	
AES-128-SHA	SSLv3	RSA	RSA	AES(128)	SHA1	
SEED-SHA	SSLv3	RSA	RSA	SEED	SHA1	

Available cipher suites

Name	Version	Key exchange	Authentication	Encryption	Digest	Export
CAMELLIA128-SHA	SSLv3	RSA	RSA	CAMELLIA (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	
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-CAMELLIA256-SHA	SSLv3	DH	none	CAMELLIA (256)	SHA1	

Available cipher suites

<b>Name</b>	<b>Version</b>	<b>Key exchange</b>	<b>Authenti- cation</b>	<b>Encryption</b>	<b>Digest</b>	<b>Export</b>
ADH-AES128-SHA	SSLv3	DH	none	AES(128)	SHA1	
ADH-SEED-SHA	SSLv3	DH	none	SEED	SHA1	
ADH-CAMELLIA128-SHA	SSLv3	DH	none	CAMELLIA (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

**-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 94](#)).

<b>-tlsRSACertificateFile</b>
<file-name 1..54>   <b>*NONE</b>

<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 93](#)), 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.

<b>-tlsRSAkeyFile</b>
<file-name 1..54>   <b>*NONE</b>

<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 96](#)).

<b>-tlsDSACertificateFile</b>
<file-name 1..54>   <b>*NONE</b>

<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 95](#)), 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.

<b>-tlsDSAkeyFile</b>
<file-name 1..54>   <b>*NONE</b>

<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.

<b>-tlsCertificateChainFile</b>
<file-name 1..54>   <b>*NONE</b>

<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.

<b>-tlsCAcertificateFile</b>
<file-name 1..54>   <b>*NONE</b>

<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.

<b>-tlsAcceptableClientCAFile</b>
<file-name 1..54>   <b>*NONE</b>

<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).)

<b>-tlsCArevocationFile</b>
file-name 1..54>   <b>*NONE</b>

<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.

<b>-tlsVerifyClient</b>
<b><u>NONE</u>   OPTIONAL   REQUIRE</b>

### **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 101](#)) or *-tlsVerifyClient OPTIONAL*.
- Specifying the depth as 0 is meaningless. In this case only self-signed certificates would be permissible.

<b>-tlsVerifyDepth</b>
<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.

<b>-tlsSecureControlConnection</b>
<b><u>NONE</u>   OPTIONAL   REQUIRE</b>

### **NONE**

The control connection is never secured, in other words a corresponding *AUTH* command (see [page 56](#)) 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 103](#)) than for *-tlsSecureDataConnection*. Consequently *-tlsSecureControlConnection* will, if required, automatically be raised to the same value as *-tlsSecureDataConnection*.

<b>-tlsSecureDataConnectionr</b>
<b><u>NONE</u>   OPTIONAL   REQUIRE</b>

### **NONE**

The control connection is never secured, in other words a corresponding *PROT* command (see [page 56](#)) 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.

<b>-tlsRandFile</b>
<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

## **-tlsUseCryptoHardware**

The *-tlsUseCryptoHardware* option is used to define whether crypto hardware, for example an *openCRYPT™* box (see the “*openCrypt*” manual) is to be used for calculating cryptographic algorithms. If the option specifies the use of crypt-hardware but the hardware/software requirements for using crypt-hardware are not met, FTP server startup is aborted.

<b>-tlsUseCryptoHardware</b>
<b><u>NO</u>   SERVER   CHILD   ALL</b>

### **NO**

No crypto hardware is used.  
NO is the default.

### **SERVER**

The crypto hardware is only used for the FTP server process.

### **CHILD**

The crypto hardware is only used for the child processes.

### **ALL**

The crypto hardware is used for both the FTP server process and for the child processes.

## **-tlsOpenSSLibName**

The *-tlsOpenSSLibName* 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.

<b>-tlsOpenSSLibName</b>
<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 85](#)), except that here the SIZE command is not deactivated completely, but the execution of the command will be aborted when the resource consumption becomes unreasonably large.

<b>-sizeCmdTimeLimit</b>
<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.

<b>-cmdBufSize</b>
<number-of-bytes>

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

## 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.

<b>STOP-FTP-DEMON</b>
<b>PORT-NUMBER=*STD-PORT/*ANY/&lt;integer 0..32767&gt;</b>

### **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
<pre> <i>SYSENT.TCP-IP-AP.nnn.FTPD:</i>  /.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 ccccccc\ -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) </pre>	<pre> <i>SYSENT.TCP-IP-AP.nnn.FTPD2:</i>  /.FTPS<b>2</b> LOGON ... /SYSFILE SYSLST=\$TSOS.SYSOUT.TCP-IP- AP.nnn.FTPD<b>2</b> ... -B 0 \ -F \$TSOS.SYSENT.TCP-IP-AP.nnn.FTPDC<b>2</b>\ -E \$TSOS.SYSDAT.TCP-IP-AP.nnn.SI<b>2</b>\ -K \$FTAC \ -J ccccccc\ -X 0\ -O 03600\ -A FTPS<b>2</b>\ -N 0015\ - D 0\ - T 0\ - S 0\ -C *MOD(\$TSOS.SYSLNK.TCP-IP-AP.nnn, FTPDC,RUN=ADV,PROG=ANY) \ <b>-P nnn</b> (nnn is the port number) </pre>
<pre> <i>SYSENT.TCP-IP-AP.nnn.FTPDC:</i>  /.FTPSR LOGON ... -N\ -E \$TSOS.SYSDAT.TCP-IP-AP.nnn.SI\ -D 0\ -T 0 </pre>	<pre> <i>SYSENT.TCP-IP-AP.nnn.FTPDC2:</i>  /.FTPS<b>2</b> LOGON ... -N\ -E \$TSOS.SYSDAT.TCP-IP-AP.nnn.SI<b>2</b>\ -D 0\ -T 0 </pre>
<pre> <b>Start of server task:</b> /START-FTP-DEMON </pre>	<pre> <b>Start of second server task:</b> /ENTER-JOB SYSENT.TCP-IP-AP.nnn.FTPD2 </pre>
<pre> <b>Connection setup:</b>  /FTP open &lt;host&gt; </pre>	<pre> <b>Connection setup:</b>  /FTP open &lt;host&gt; <b>nnn</b> (nnn is a port number) </pre>

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 console command */INTR <tsn>,shutdown*. *<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 69](#)).

- 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 38/page 57](#)), 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/OSD.
- 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 82](#)).
- 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 110](#)). 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.*nmn*.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 38/page 57](#))
- Options *-acctActive* and *-acctFile* (see [page 81](#))

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

```
/INTR <tsn>, acctActive:   Enable/disable FTP accounting (see page 124)  
/INTR <tsn>, acctFile:    Specify new accounting file (see page 124)
```

### 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 clear-down 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 118](#)).

**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 `/INTR` command `RDPROT` (see the [section “RDPROT - Save logging file of FTP servers” on page 125](#)).

## 4.8 Displaying the current settings of the FTP server

You can use the BS2000/OSD command `SHOW-FTP-TELNET-STATUS` to obtain information on the current settings of BS2000/OSD 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 `/INTR` (or the SDF command `/INFORM-PROGRAM`). Commands issued with `/INTR` 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 109](#)) can also be entered via the console.

### debug - Enable / disable trace to user level

<code>/INTR</code>
<code>&lt;tsn&gt; , <b>debug</b> &lt;debug-value&gt;</code>

`<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

```
/INTR
```

```
<tsn> , trace <trace-value>
```

<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

```
/INTR
```

```
<tsn> , shutdown
```

<tsn>

TSN of the server task to be shut down.

## acctActive - Enable / disable FTP accounting

```
/INTR
```

```
<tsn> , acctActive {ON | OFF}
```

<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

```
/INTR
```

```
<tsn> , acctFile <accounting-file-name>
```

<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.

```
/INTR
```

```
<tsn> , RDPROT
```

```
<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 (|l|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/OSD is the existence of sockets with IPv6 capability. In BS2000/OSD 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 37](#) 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 61](#)) or
- The `-systemExit 1` (see [page 79](#)) 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

(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(YAPXPARG) = 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 YAPSEPAFTPLPG 1        /* FTP login */
#define YAPSEPAFTPCMD 2        /* FTP command */
#define YAPSEPAFTPCMDE 3       /* FTP command end */
#define YAPSEPAFTPBYP 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_md1 {

    /* 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 SPMLNK.TCP-IP-AP.*nnn* (on SX 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.



```

    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;     /* Length 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 47](#)).

## 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 146](#))
- Configuration of TELNET via SDF command (see [page 148](#)) or option file (see [page 158](#))
- Startup and shutdown of TELNET servers (see [page 189](#))
- Saving the logging file of TELNET servers (see [page 192](#))
- Displaying the current settings of the TELNET server (see [page 192](#))
- Specification of commands using the console interface (see [page 195](#))
- Using IPv6 addresses in TELNET (see [page 197](#))
- Exits for TELNET clients and the TELNET server (see [page 198](#))

## 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 158](#)).
- With the aid of the installation command parameters SET-FTP-TELNET-PARAMETERS (see the [section “Configuring TELNET using an option file” on page 158](#)).

### 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 167](#))

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 147](#)). Provisions for authentication are negotiated by SSL to free TELNET of this load.
- AUTHENTICATION option (*-B*, see [page 187](#))

You use this option to negotiate the provisions for authentication. In BS2000/OSD 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 147](#)).
- ENCRYPTION option (*-H*, [page 188](#))

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	<a href="#">181</a>
-Z CipherSuite	Specify cipher suite preference list	<a href="#">174</a>
-Z RSAcertificateFile	Specify file which contains the RSA-based X.509 server certificate in PEM format	<a href="#">168</a>
-Z RSAkeyFile	Specify file which contains the private RSA server key in PEM format	<a href="#">169</a>
-Z DSACertificateFile	Specify file which contains the DSA-based X.509 server certificate in PEM format	<a href="#">170</a>
-Z DSAkeyFile	Specify file which contains the private DSA server key in PEM format	<a href="#">171</a>
-Z CertificateChainFile	Specify file in which all the certificates required for verification of the server certificate can be stored	<a href="#">180</a>
-Z CACertificateFile	Specify file which contains the certificates required for authentication of the TELNET client in PEM format	<a href="#">172</a>
-Z AcceptableClientCAFile	Specify file from which the names of the CAs that the server accepts as signatories of client certificates can be obtained	<a href="#">182</a>
-Z CARevocationFile	Specify file which contains the CRLs of the CAs	<a href="#">173</a>
-Z VerifyClient	Define whether the TELNET client must provide a certificate for server access	<a href="#">184</a>
-Z VerifyDepth	Define verification depth	<a href="#">183</a>
-Z RandFile	Specify file from which the data for initializing the PRNG is read when the server is started	<a href="#">179</a>
-Z UseCryptoHardware	Define whether crypto hardware is to be used for calculating crypto algorithms	<a href="#">186</a>
-Z OpenSSLLibName	Define the LMS file from which the OpenSSL library should be dynamically loaded	<a href="#">185</a>

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

**i** For the full command syntax and the description of the installation operands, see [page 38](#).

### SET-FTP-TELNET-PARAMETERS

(...)

, TELNET-SERVER-PROC= **\*NO** / **\*CREATE**(...)

**\*CREATE**(...)

**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-CONNECTIONS**= **\*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-REVOCAATION-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>

, **USE-CRYPTO-HARDWARE**= **\*STD** / **\*NO** / **\*YES**

**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 procedure 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 160](#).

**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 parameters TIME=NTL and PRIORITY=(120,EXPRESS) may be started in this job class.

**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 160](#).

**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 163](#).

**TRACE=\*STD**

Corresponds to the entry 0 or the value in the installation file.

**TRACE=<integer 0..9>**

Socket trace level.

**MAXIMUM-CONNECTIONS=**

Maximum number of connections that the server is to operate. See also the *-N* option on [page 161](#).

**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 160](#).

**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 164](#).

**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 164](#).

**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 167](#).

**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 187](#).

**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 181](#).

**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 174](#).

**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 168](#).

**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 169](#).

**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 170](#).

**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 171](#).

**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 172](#).

**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 182](#).

**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 180](#).

**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 173](#).

**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 179](#).

**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 185](#).

**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 184](#).

**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 183](#).

**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).

**USE-CRYPTO-HARDWARE=**

See the *-Z UseCryptoHardware* option on [page 186](#).

**USE-CRYPTO-HARDWARE=\*STD**

Default: \*NO

**USE-CRYPTO-HARDWARE=\*NO**

No crypto hardware is used.

**USE-CRYPTO-HARDWARE=\*YES**

Crypto hardware is used.

**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 188](#).

**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 185](#).

**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 161](#).

**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 162](#).

**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 38/page 148](#)), 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.

<b>-A</b>
<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.

<b>-D</b>
<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 199](#).

<b>-E</b>
<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.

<b>-N</b>
<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.

<b>-P</b>
<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.

<b>-S</b>
<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.

<b>-T</b>
<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*.

<b>-V</b>

## **-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.

<code>-X</code>
<code>&lt;ebcdic-code-table&gt;:&lt;iso-code-table&gt;</code>

`<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/OSD it is supported by the server option *-Z tls-required*.

- “Telnet Authentication Option” (RFC 2941) for negotiating an authentication method

In BS2000/OSD 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/OSD 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 166](#)), AUTHENTICATION option (see [page 187](#)) and ENCRYPTION option (see [page 188](#)) are described in detail in the following sections.

### 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 187](#)) 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 188](#)) and either the START-TLS option (`-tls-required`, see [page 167](#)) or the AUTHENTICATION option (`-B`, see [page 187](#)) 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.

<b>-Z tls-required</b>
[={ <b>yes</b>   <b>no</b> }]

### **yes**

START-TLS support is enabled.

### **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 169](#)).

<b>-Z RSACertificateFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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 168](#)), 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.

<b>-Z RSAKeyFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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 171](#)).

<b>-Z DSACertificateFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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 170](#)), 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.

<b>-Z DSAKeyFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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.

<b>-Z CACertificateFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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).)

<b>-Z CARevocationFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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.

<b>-Z CipherSuite</b>
=<specification>

<specification>

Specification of a cipher suite preference list (for details, see the description of the specification structure below).

ALL: !EXP: !ADH is the default.

### Description of the specification structure

The specification consists of one or more cipher mnemonics which are separated by a colon (:).

A cipher mnemonic can have 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 suite 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 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 using 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.

#### DEFAULT

The following cipher list is the default:

```
ALL: !ADH:RC4+RSA:+SSLv2:@STRENGTH
```

If it is used, this cipher mnemonic must be at the start of the specification.

#### 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. As 3DES is interpreted with a length of 168 bits, it belongs to this suite class.

#### MEDIUM

Cipher suites with a key length of 128 bits.

#### 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.

**kEDH**

Cipher suites with ephemeral Diffie-Hellman key negotiation.

**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, SSLv3, SSLv2**

TLSv1, SSLv3 and SSLv2 cipher suites. The TLSv1 suites and the SSLv3 suites are identical.

**AES**

Cipher suites with AES encryption (key length of 128 and 256 bits).

**DH**

Cipher suites with Diffie-Hellman key exchange, including anonymous exchange.

**ADH**

Cipher suites with anonymous Diffie-Hellman key exchange.

**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.

The available cipher suites are listed in the table below.

Name	Version	Key exchange	Authenti- cation	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	
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

Available cipher suites

<b>Name</b>	<b>Version</b>	<b>Key exchange</b>	<b>Authenti-cation</b>	<b>Encryption</b>	<b>Digest</b>	<b>Export</b>
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

## **-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.

<b>-Z RandFile</b>
=<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.

<b>-Z CertificateChainFile</b>
={<file-name 1..54>   * <b>NONE</b> }

<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 2 and 3 of the SSL protocol and also Version 1 of the TLS protocol. Some of these protocols can be activated selectively using the *-Z Protocol* option.

<b>-Z Protocol</b>
={+   -} {SSLv2   SSLv3   TLSv1   ALL } ...

**+**

The protocol specified after this sign is permissible.

**-**

The protocol specified after this sign is not permissible.

### **SSLv2**

SSL protocol Version 2



Version 2 of the SSL protocol displays some security-related deficiencies and should therefore not be used if possible.

### **SSLv3**

SSL protocol Version 3

### **TLSv1**

TLS protocol Version 1

### **ALL**

All protocols are to be enabled.

ALL-SSLv2 is the default.

### *Example*

The specifications `-tlsProtocol SSLv3+TLSv1` **and** `-tlsProtocol ALL-SSLv2` have the same effect.

## -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.

<b>-Z AcceptableClientCAFile</b>
={<file-name 1..54>   <b>*NONE</b> }

<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 184](#)) or *-Z VerifyClient=OPTIONAL*.
- Specifying the depth as 0 is meaningless. In this case only self-signed certificates would be permissible.

<b>-Z VerifyDepth</b>
=<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
-----------------

={ <u>NONE</u>   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 OpenSSLibName**

The *-Z OpenSSLibName* 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.

<b>-Z OpenSSLibName</b>
=<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.

## **-Z UseCryptoHardware**

The *-Z UseCryptoHardware* option is used to define whether crypto hardware, for example an *openCRYPT*<sup>TM</sup> box (see the “*openCRYPT*” manual) is to be used for calculating cryptographic algorithms.

<b>-Z UseCryptoHardware</b>
= <b>{YES   <u>NO</u>}</b>

### **YES**

Crypto hardware is used.

### **NO**

No crypto hardware is used.  
NO is the default.

### 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/OSD 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 166](#)).

The options *-Z tls required* (START-TLS option, see [page 167](#)) 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>-B</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 Option -H - 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 167](#)) or *-B on* (see [page 187](#)). If it is, the following error message is issued:

```
error: SSL and encryption option
```

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 185](#)).

**-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:

<i>/START-TCP-IP-DEMON</i>	Enter Job for TCP-IP-AP
<i>/START-TELNET-DEMON</i>	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.

You shut down the TELNET server with the *STOP-TELNET-DEMON* command.

<b>STOP-TELNET-DEMON</b>
<b>PORT-NUMBER=*STD-PORT/*ANY/&lt;integer 0..32767&gt;</b>

#### **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.

### 5.4.3 Shutdown

Servers can still be shut down using the console command */INTR <tsn>,shutdown*. *<tsn>* is the TSN of the relevant server task.

### 5.4.4 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 158](#)).

- 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.

## 5.4.5 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 `/INTR` command `RDPROT` (see the [section “RDPROT - Save logging file of TELNET servers” on page 197](#)).

## 5.6 Displaying the current settings of TELNET servers

You can use the BS2000/OSD command `SHOW-FTP-TELNET-STATUS` to obtain information on the current settings of BS2000/OSD 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 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 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 `/INTR`. Commands issued with `/INTR` 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 189](#)) can also be entered via the console.

### debug - Enable / disable trace to user level

<code>/INTR</code>
<code>&lt;tsn&gt; , <b>debug</b> &lt;debug-value&gt;</code>

`<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

<b>/INTR</b>
<tsn> , <b>trace</b> <trace-value>

<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

<b>/INTR</b>
<tsn> , <b>shutdown</b>

<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.

```
/INTR
```

```
<tsn> , RDPROT
```

<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 *YAPTEXTIT*. 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 *YAPTEXTIT* 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:*`

- 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".

```
YAPTEXTIT (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 47](#)).

---

## 6 Generating random numbers

This chapter describes:

- Generating random numbers in BS2000/OSD with PRNGD
- Generating random numbers in POSIX with the *prngd* server daemon

## 6.1 Generating random numbers in BS2000/OSD 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/OSD provides a central random number generator. The BS2000/OSD 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/OSD 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/OSD PRNGD are described below.

### 6.1.1 Entropy sources of the BS2000/OSD PRNGD

The entropy sources used by the BS2000/OSD 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/OSD 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/OSD PRNGD

You configure the BS2000/OSD 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.

<b>poolSize</b>
<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.

<b>minimalEntropy</b>
<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.

<b>entropyThreshold</b>
<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 it 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.

<b>seedFile</b>
<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.

<b>file</b>
<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.

<b>fileInterval</b>
<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.

<b>cmd</b>
<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.

<b>cmdInterval</b>
<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.

<b>bcamInterval</b>
<time-interval>

<time-interval>

Time interval in seconds.

Default: 49

### 6.1.3 GPRBYTE program interface of the BS2000/OSD PRNGD

The GPRBYTE program interface of the BS2000/OSD 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 = <u>*NON_BLOCKING</u> ]



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 = { <u>G</u>   <name> }] [,MACID = { <u>PRB</u>   <name>}] [,EQUATES = { <u>YES</u>   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_md1 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

GPR0100 INTERNAL ERROR IN PRNGD-SUBSYSTEM-INITIALIZATION; ERROR CODE (&00)

### Meaning

Some operating resources aren't available. This resources are according to the error code:

- 0000 - No memory space for subsystem object available.
- 0001 - No memory space for PRNGD error trace available.
- 0002 - No memory space for PRNGD event trace available.
- 0003 - No memory space for task lock object available.
- 0004 - Task lock can't be created.
- 0005 - Server bourse can't be created.
- 0006 - No memory space for file object available.
- 0007 - No memory space for PRNG object available.
- 0008 - Error at reading configuration file.
- 0009 - Error at initialization of PRNG.
- 0010 - Error at accessing seed file.
- 0011 - Error at connection of holder task with SDF.
- 0012 - Error at notification of DSSM.

### Response

<None>

GPR0105 ERROR IN DSSM-PRNGD-INTERFACE; ERROR CODE (&00)

### Meaning

The cause for this message is according to the error code:

- 0001 - A notification of DSSM failed.
- 0002 - Some internal error occurred during restart of PRNGD subsystem.

### Response

<None>

GPR0110 THE SUBSYSTEM PRNGD IS READY

### Meaning

The subsystem PRNGD is as far initialized that it can fulfill requests for (pseudo) random numbers.

### Response

<None>

GPR0150 THE FOLLOWING SYSTEM DUMP OF TASK GPR1 SERVES AS DIAGNOSTIC AID

**Meaning**

If the reason for the occurred problem isn't already clear from the previous message, then take the dump please and send it to your support.

**Response**

<None>

## 6.2 Random number generation in POSIX with the prngd server daemon

The server program *prngd* used in OSD/POSIX provides random numbers (entropy) via a standardized interface for programs which run in the POSIX subsystem. *prngd* uses the PRNGD subsystem in BS2000/OSD if this has been started. However, if necessary *prngd* can calculate entropies independently. As the *prngd* server daemon and the BS2000/OSD subsystem PRNGD come from a common code base, they also share many concepts (non-blocking random number calculation, similarly structured configuration file, writing back of the so-called “seed file” etc.).



The other components of interNet Services which execute in the POSIX subsystem and require random numbers now call the BS2000 PRNG directly, and consequently the POSIX *prngd* might possibly no longer be supplied in a future interNet Services version.

People who write their own POSIX programs should therefore also address the BS2000 PRNGD directly via the GPRBYTE interface or, in POSIX-BC V7.0/V8.0 (corresponds to status A41) or higher, use the device file */dev/urandom* which also obtains random numbers from the BS2000 PRNGD.

The server program */opt/TCP-IP-SV/prngd/sbin/prngd* is automatically started when POSIX is started and then runs as a background process (daemon) under the SYSROOT ID if it is activated with AUTOSTART='yes' in */etc/default/TCP-IP-SV.prngd*. At startup it reads in its configuration from the file */etc/prngd.conf*.

Using the EGD protocol, the *prngd* server daemon is addressed via the Unix socket */var/run/egd-pool*. Further details on this can be found at <http://prngd.sourceforge.net/>. You can test the EGD interface and extract sample random numbers using the tool */opt/TCP-IP-SV/prngd/sbin/prngd-ctl*.

### Installing the prngd server daemon

Please read the Release Notice shipped with the product interNet Services in addition to the information in this section.

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 33](#).

For the installation and operation of the TCP-IP-SV component, the PLAM library *SINLIB.TCP-IP-SV.nnn.PRNGD* must be shareable. During installation of PRNGD, the installation path can be defined in the POSIX file system. The *prngd* server daemon is installed by default in the */opt/TCP-IP-SV/prngd* directory.

After installation is completed, the daemon is started in accordance with the reply to the installation query AutoStartPRNGD and is available to other POSIX processes for generating random numbers.

### Configuring the prngd server daemon

As the *prngd* server daemon normally gets its entropy from the BS2000/OSD subsystem PRNGD, no special configuration is required. Configuration file */etc/prngd.conf* describes the BS2000/OSD commands which are used as a fallback for entropy collection if the BS2000/OSD subsystem PRNGD is not available.

The configuration comprises three specifications per line for starting the POSIX program:

1. The first specification defines the command to be executed. This command is generally enclosed in double quotes.
2. The second specification defines the absolute path to the program (for example */sbin/sh*).
3. The third specification provides an estimate for the quality of the entropy delivered by the command (see 1.).

*Example of the configuration file /etc/prngd.conf*

```
"sh /opt/TCP-IP-SV/prngd/etc/prngd/bs2cmd.STATUS.LIST"      /sbin/sh      0.2
"sh /opt/TCP-IP-SV/prngd/etc/prngd/bs2cmd.STATUS.PROG"    /sbin/sh      0.2
#"df -lg"                                                  /usr/sbin/df  0.01
#"ps -aAcdefjl"                                           /sbin/ps      0.2
```

(See also the [section “Generating random numbers in BS2000/OSD with PRNGD” on page 206](#)).

Normally the BS2000/OSD subsystem PRNGD is available and the specifications in the */etc/prngd.conf* file are not used.

### Terminating the prngd server daemon

When the POSIX subsystem is terminated, so too is the *prngd* server daemon. Its status is saved to the seed file */etc/prngd-seed*, from where the seed value is read at the next startup. However, you can also stop the daemon manually by calling the following stop scripts:

```
# /etc/init.d/TCP-IP-SV.prngd stop
```

Do not forget to start it again manually later:

```
# /etc/init.d/TCP-IP-SV.prngd start
```

Otherwise components which do not yet have not yet been converted to direct access to the BS2000 PRNGD cannot obtain any random numbers. Thee TCP-IP-SV and MAIL components have already been converted to direct used of the BS2000 PRNGD.

---

## 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/OSD. 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 standards 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/OSD: 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

figure 1 illustrates the domain structure of the DNS name space.

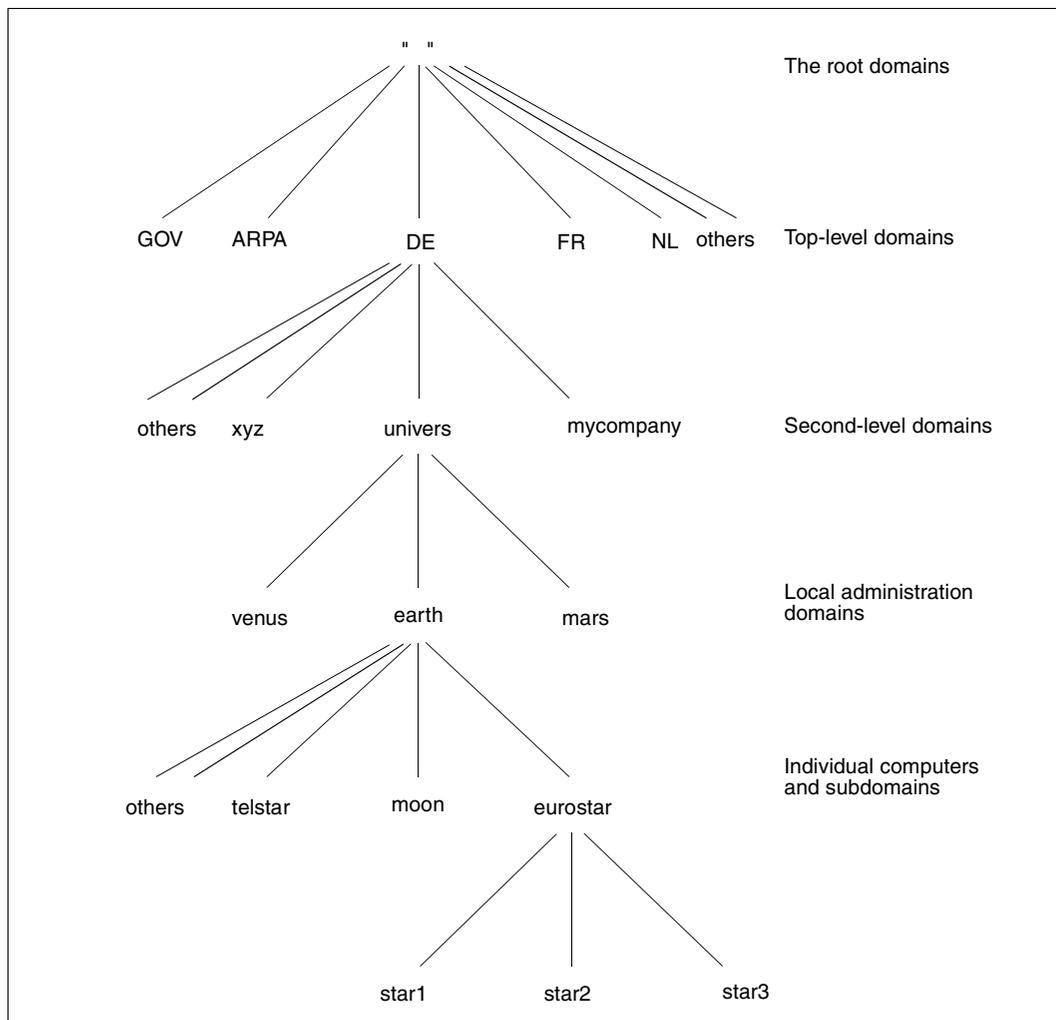


Figure 1: Domain structure of the DNS name space

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 files 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 228](#).

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](#)).

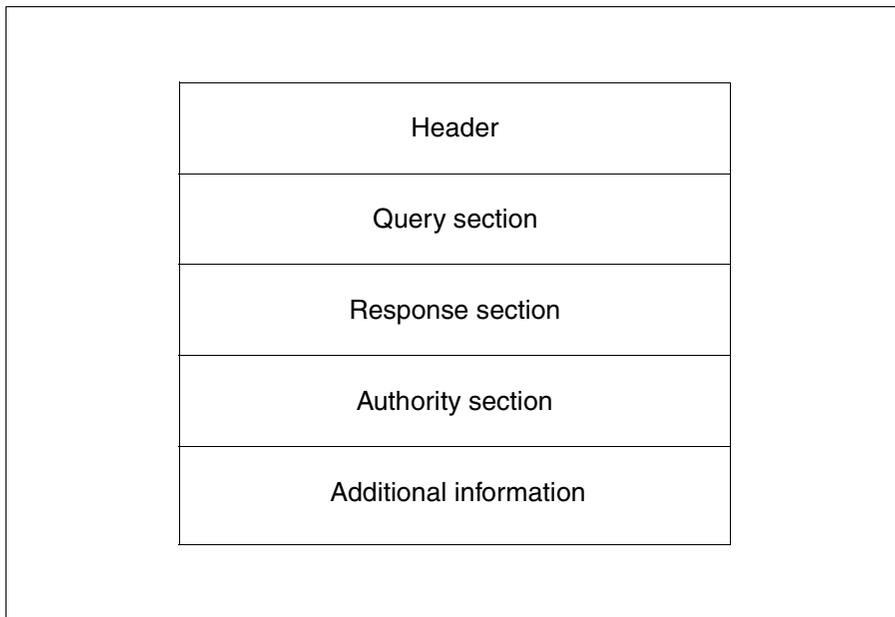


Figure 2: Format of a DNS message

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:

<i>gethostbyname()</i>	Supplies the associated IPv4 address for a host name.
<i>gethostbyaddr()</i>	Supplies the associated host name for an IPv4 address.
<i>getaddrinfo()</i>	Supplies information on host names, host addresses and service independently of the protocol.
<i>getnameinfo()</i>	Supplies the name of the communications partner independently of the protocol.
<i>getipnodebyaddr()</i>	Supplies information on host names independently of the protocol.
<i>getipnodebyname()</i>	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 237](#)).

## 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/OSD file system. This configuration file has the name \$TSOS.SYSDAT.SOCKETS.*nnn*.SOC6.RESOLV (up to and including openNet Server V3.0) or SYSDAT.LWRES.D.*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 235](#).

- 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/OSD or POSIX) and how the application is linked:

- Sockets application in BS2000/OSD:

The resolver functions initially use the internal Sockets resolver or, in Sockets 2.2 and higher, the LWRES.D 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 237](#)).

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/OSD. 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 33](#).

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 235](#), `<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 <code>/etc/init.d/TCP-IP-SV.dns</code> (to terminate when POSIX shuts down)

<sup>1)</sup> Only available if the DNS development modules are installed (see [page 237](#)).

Directory	File/Link	Description
/etc/rc2.d/	S70TCP-IP-SV.dns	Symbolic link to <i>/etc/init.d/TCP-IP-SV.dns</i> (to start when POSIX starts)
/usr/bin/	dig	Symbolic link to <i>&lt;instdir&gt;/bin/dig</i> , diagnostic tool for DNS
	host	Symbolic link to <i>&lt;instdir&gt;/bin/host</i> , host names DNS lookup tool
	nslookup	Symbolic link to <i>&lt;instdir&gt;/bin/nslookup</i> , lookup tool for DNS
/usr/lib/	libbind.a	Symbolic link to <i>&lt;instdir&gt;/lib/libbind.a</i> <sup>1)</sup>
<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 <i>&lt;instdir&gt;/sbin/dnsd</i>
<instdir>/etc/	install.TCP-IP-SV-DEV. <i>nnn</i> .DNS	Installation script for the DNS development modules
	remove.TCP-IP-SV-DEV. <i>nnn</i> .DNS	Uninstallation script for the DNS development modules
	rc.d/dnsd	Start/stop script for dnsd
<instdir>/lib/	libbind.a	Resolver library <sup>1)</sup>
<instdir>/readme/	*	man pages for the DNS tools and for the resolver library <sup>1)</sup> in HTML and text format
<instdir>/sbin/	dnsd	DNS resolver daemon

<sup>1)</sup> Only available if the DNS development modules are installed (see [page 237](#)).

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/OSD 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 */var/adm/messages* message file. 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 235](#)).

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 244](#), *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 244](#)).

- 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 *domain<sub>x</sub>* ( $x = 2 - 6$ ) 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 *domain<sub>x</sub>* (*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/messages* 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 240](#)).

### 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.dns* 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 246](#)).

## 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/messages* file. These entries have the following basic format:

```
Jan 22 15:44:58 LOG_NOTICE syslog[799]: dnssdemon: <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 *dnssdemon*) 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 244](#)) in the configuration file */etc/resolv.conf*. The diagnostic messages of the DNS resolver daemon are then logged in the */var/adm/messages* 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 33](#).

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 249](#), <instdir> stands for the installation directory */opt/TCP-IP-SV/dns-named*.

Directory	File	Description
<i>/etc/</i>	<i>named.conf</i>	DNS configuration file of the name server
<i>/etc/default</i>	<i>TCP-IP-SV.named</i>	Parameter file for the start/stop procedure
	<i>TCP-IP-SV.named.default</i>	Default for <i>TCP-IP-SV.named</i>
<i>/etc/init.d/</i>	<i>TCP-IP-SV.named</i>	Start/stop procedure

<sup>1)</sup> Only available if the NAMED development modules are installed (see [page 252](#)).

Directory	File	Description
/etc/rc0.d/	K64TCP-IP-SV.named	Symbolic link to <i>/etc/init.d/TCP-IP-SV.named</i> to terminate when POSIX shuts down
/etc/rc2.d/	S72TCP-IP-SV.named	Symbolic link to <i>/etc/init.d/TCP-IP-SV.named</i> to start when POSIX starts
/usr/bin/	nsupdate	Symbolic link to <i>&lt;instdir&gt;/bin/nsupdate</i>
/usr/lib/	liblwres.a libiscfg.a libisccc.a libisc.a libdns.a	Additional DNS libraries not required for running NAMED <sup>1)</sup>
/usr/sbin/	in.named	Symbolic link to <i>&lt;instdir&gt;/sbin/named</i>
	named-checkconf	Symbolic link to <i>&lt;instdir&gt;/sbin/named-checkconf</i>
	named-checkzone	Symbolic link to <i>&lt;instdir&gt;/sbin/named-checkzone</i>
	rndc	Symbolic link to <i>&lt;instdir&gt;/sbin/rndc</i>
	rndc-confgen	Symbolic link to <i>&lt;instdir&gt;/sbin/rndc-confgen</i>
	dnssec-keygen	Symbolic link to <i>&lt;instdir&gt;/sbin/dnssec-keygen</i>
	dnssec-signzone	Symbolic link to <i>&lt;instdir&gt;/sbin/dnssec-signzone</i>
<instdir>/etc/	install.TCP-IP-SV-DEV. <i>nnn</i> .NAMED	Installation script for the development modules
	remove.TCP-IP-SV-DEV. <i>nnn</i> .NAMED	Deinstallation script for the development modules
<instdir>/include/	*	Include files <sup>1)</sup>
<instdir>/readme/	bind9arm.pdf	Bind9 Administrator Reference Manual
	*	man pages in HTML and text format

<sup>1)</sup> Only available if the NAMED development modules are installed (see [page 252](#)).

Directory	File	Description
<instdir>/sbin/	named	DNS name server daemon
	named-checkconf	Syntax check of <i>named.conf</i> file
	named-checkzone	Syntax check of a zone data file
	rndc	Remote control program for name server daemon
	rndc-confgen	Generation program for <i>rndc</i> 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

<sup>1)</sup> Only available if the NAMED development modules are installed (see [page 252](#)).

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/OSD 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.032.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 `/var/adm/messages` message file. 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 249](#)).

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 258](#)).

##### 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/messages* 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/messages*. 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/messages*:

- 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 253](#).

#### 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:abcd:abcd:abcd:abc1
myhost.mydom.de.         604800  IN      AAAA    3ffe:1:1000:2000:abcd: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:abcd: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 Time Protocol 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/OSD

BS2000/OSD 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/OSD 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/OSD 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:

$$\text{Requirement} = (900 / \text{sync\_interval}) * 8 \text{ KB}$$

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/OSD. 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 manual "BS2000/OSD-BC - Introductory Guide to Systems Support", chapter "System time administration".

The time server with the highest priority is preferred by BS2000/OSD; all other time servers are then ignored. It is also not possible to operate NTP and DCE at the same time. Since this is not checked by BS2000/OSD, the system administrator must make suitable organizational arrangements.

By default the */var/adm/ntp.log* and */var/adm/messages* 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	<a href="#">page 277</a>
ntpq	Querying NTP status	<a href="#">page 296</a>
ntpdate	Setting date and time	<a href="#">page 293</a>
ntptrace	Tracing NTP servers	<a href="#">page 306</a>
ntpdcc	Querying NTP status (specifically)	
sntp	Client for simplified (SNTP) protocol	
ntp-keygen	Generating public and private keys	<a href="#">page 295</a>

## 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 33](#).

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 <sup>1</sup> :	Install packages on POSIX
IMON support ?	Y
name of product	TCP-IP-SV
package of product	NTP

...

<sup>1</sup> *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/OSD 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*.

Directory	File	Description
<i>&lt;instdir&gt;/bin</i>	<i>ntp-keygen</i>	NTP program
	<i>ntp-wait</i>	Script for synchronization with <i>ntpd</i>
	<i>ntpq</i>	NTP program
	<i>ntptrace</i>	Script for displaying the NTP server chain
<i>&lt;instdir&gt;/etc/rc.d</i>	<i>ntpd</i>	Start/stop script
<i>&lt;instdir&gt;/readme</i>	<i>&lt;diverse&gt;</i>	Original NTP documentation in various file formats
<i>&lt;instdir&gt;/sbin</i>	<i>ntpd</i>	NTP daemon
	<i>ntpdate</i>	NTP program
	<i>ntpdcc</i>	NTP program
	<i>sntp</i>	NTP program
	<i>start-ntp-daemon</i>	Auxiliary routine for starting <i>ntpd</i> , cannot be called directly
	<i>ntp.conf.default</i>	Default NTP configuration file
	<i>ntp.drift</i>	Drift file
<i>/etc/default</i>	<i>TCP-IP-SV.ntp</i>	Parameter file for the start/stop script
	<i>TCP-IP-SV.ntp.default</i>	Default parameter file for the start/stop script
<i>/etc/init.d</i>	<i>TCP-IP-SV.ntp</i>	Start/stop script for <i>ntpd</i>
<i>/etc/rc0.d</i>	<i>K17TCP-IP-SV.ntp</i>	Symbolic link to the start/stop script
<i>/etc/rc2.d</i>	<i>S97TCP-IP-SV.ntp</i>	Symbolic link to the start/stop script
<i>/var/run</i>	<i>ntp.pid</i>	File with the pid of <i>ntpd</i>
<i>/var/adm</i>	<i>ntpd_startup.log</i>	Start messages of <i>ntpd</i>
	<i>ntp.log</i>	Logfile of <i>ntpd</i>

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 279](#)). 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 */var/adm/messages* message file. 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/OSD, 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 293](#).



*ntpd* and *ntpdate* both use port number 123. Only one of the applications can therefore run at any given time.

### Addressing clocks

In OSD/POSIX only the system clock of the local computer can be configured as a 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. Further information is provided in the chapter "System time administration" in the manual "BS2000/OSD-BC - Introductory Guide to Systems Support".

In principle, however, NTP also allows other clocks such as radio receivers for DCF77 signals to be configured.

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/OSD.

#### 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  $2^{\text{minpoll}}$  seconds.

Default: 6 (64 sec)

Values permitted in BS2000/OSD:  $\geq 6$  (64 sec)

*maxpoll*

Integer that specifies the maximum polling interval for NTP messages as  $2^{\text{minpoll}}$  seconds.

Default: 10 (1024 sec)

Values permitted in BS2000/OSD: 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, *numer-address* 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 279](#)). 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/OSD.

*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 3. 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  $2^{\text{minpoll}}$  seconds.

Default: 6 (64 sec)

Values permitted in BS2000/OSD:  $\geq 6$  (64 sec)

*maxpoll*

Integer that specifies the maximum polling interval for NTP messages as  $2^{\text{minpoll}}$  seconds.

Default: 10 (1024 sec)

Values permitted in BS2000/OSD: 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  $2^{\text{minpoll}}$  seconds.  
Default: 6 (64 sec)  
Values permitted in BS2000/OSD:  $\geq 6$  (64 sec)

*maxpoll*

Integer that specifies the maximum polling interval for NTP messages as  $2^{\text{minpoll}}$  seconds.  
Default: 10 (1024 sec)  
Values permitted in BS2000/OSD: 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 *ntpd* 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 *ntpd*. 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 *ntpd* when they work with symmetrical keys.

---

**keys** keyfile

---

*keyfile*

Complete path of the key file.

**keymdir statement**

The *keymdir* statement specifies the default path name for autokey keys, parameters and certificates.

---

`keymdir 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.

<code>ntpd</code>
<code>[-bdgq] [-c &lt;conffile&gt;] [-D &lt;level&gt;] [-f &lt;driftfile&gt;] [-k &lt;keyfile&gt;] [-l &lt;logfile&gt;] [-p &lt;pidfile&gt;] [-r &lt;broadcastdelay&gt;]</code>

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/messages* 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 288](#)). 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.

**-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.

**-s**

This option instructs `ntpdate` to log the actions via the `syslog` function (output in `/var/adm/messages`), instead of using the standard output. This option is useful when `ntpdate` is executed by `cron`.

**-p <number\_of\_timestamps>**

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.

## &lt;server&gt;

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 seconds).

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 299](#)) 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/messages* 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/OSD. 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 (**Secure Shell**) 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 the free version of SSH (i.e. there is no license fee) which supports the SSH protocol versions 1.3, 1.5 and 2.0. The basis for porting OpenSSH to POSIX is provided by OpenSSH V3.8.

## 9.1 Concept of OpenSSH

OpenSSH is the secure alternative to the r utilities *rlogin*, *rcp* and *rsh*, 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 316](#))
- 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/OSD 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 3DES, Blowfish, AES and Arcfour:

- 3DES is a tried and tested encryption algorithm for strong encryption.
- Blowfish is a high-speed block encryptor for users who require a particularly fast encryption method.
- AES is also 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.
- Arcfour is a high-speed stream encryptor.

#### **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 Blowfish or 3DES.

#### **Strong authentication**

Authentication of the OpenSSH server to the OpenSSH client is based on the asymmetrical encryption algorithms RSA and DSA. Several methods are available for authenticating the OpenSSH client to the OpenSSH server (see [page 320](#)).

## Interoperability

OpenSSH Version 1 supports SSH 1.3 and SSH 1.5 and thus permits communication with most Unix and Windows operating systems and most of the other SSH implementations. OpenSSH Version 2 also supports SSH 2. Interoperability between the clients and servers of the two OpenSSH versions is guaranteed.



It is recommendable to work with SSH protocol Version 2 as SSH protocol Version 1 manifests security gaps and, for example, does not protect reliably against “man-in-the-middle” attacks. (You specify the protocol version in the */etc/ssh/sshd\_config* configuration file using the parameter *protocol 2*.)

## 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). 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/SMAPLus/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 three pairs of host key files (private/public keys) are searched for:

- *ssh\_host\_key* and *ssh\_host\_key.pub* (RSA1)
  - *ssh\_host\_rsa\_key* and *ssh\_host\_rsa\_key.pub* (RSA)
  - *ssh\_host\_dsa\_key* and *ssh\_host\_dsa\_key.pub* (DSA)
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 PRNGD socket and 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 three 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 three times for this purpose (for RSA1, RSA, DSA) 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 (*cs*h or *tc*sh) is installed, an analogous procedure is followed using the *ic*sh 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.nnm.OPENS*SH), 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 `MAXIMUM CPU LIMIT` for `SYROOT`. Using these values the script starts an `/ENTER` job with the job name `SSHLOGIN`.

The `sshd` daemon logs important messages in the `/var/adm/messages` 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 317](#)),
- 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 -words [-46Ddeigt] [-b <bits>] [-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. The `sshd` implementation described here supports versions 1 and 2 of the OpenSSH protocol in parallel.

### 9.3.3 Internal procedure when setting up a connection between `sshd` and `ssh`

The procedure for setting up a connection between the OpenSSH Server `sshd` and the OpenSSH client `ssh` (see the interNet Services User Guide) differs depending on whether SSH protocol Version 1 or Version 2 is used.

#### Connection setup between `sshd` and `ssh` in SSH protocol Version 1

To enable `sshd` to receive connection requests from `ssh`, the following requirements must be met for `sshd`:

- A host key (RSA, generally 1024 bits) exists on the computer on which `sshd` is running (server computer). The host key is the public key in a pair of public / private keys and identifies the server. This requirement is already satisfied by installation in POSIX.
- In addition, when it starts, `sshd` generates a server key S (RSA, generally 768 bits) which is updated every hour and is never stored.
- SSH protocol Version 1 must be activated in the configuration file `/etc/ssh/sshd_config` (default: "Protocol 2,1").

The following activities are performed for every `ssh` connection request:

1. `sshd` sends its two public RSA keys (host and server keys) to `ssh`.
2. `ssh` checks whether it knows `sshd` by comparing the RSA host key with the public keys in its own local database:
  - If the RSA host key is contained in this database, the system continues with 3.).
  - If the RSA host key is not contained in this database, `ssh` asks the user if he/she wishes to continue with connection setup.

Depending on the user's response, `ssh` proceeds as follows:

- If the user responds with “no”, `ssh` aborts connection setup.
- If the user responds with “yes”, `ssh` “learns” the RSA host key by entering it in the database of keys which it knows.  
Afterward `ssh` continues with 3.).

3. `ssh` generates a 256-bit random number, encrypts this with both the host key and the public key of `sshd` and sends the result to `sshd`. Both sides use the encrypted random number as the session key with which all communication over the connection is encrypted with immediate effect.
4. Both sides use this random number as the session key for encrypting all communication in the current session. The rest of the session is encrypted using a conventional encryption algorithm. Currently Blowfish or 3DES (default) is used for this purpose.
5. `ssh` and `sshd` now conduct an authentication dialog in which `ssh` proves its identity and authorization to `sshd` (see the [section “Authentication between OpenSSH client `ssh` and server `sshd`” on page 320](#)).

`sshd` authenticates itself implicitly to `ssh` as it can only ascertain the session key generated and encrypted by `ssh` if it knows its private RSA/DSA keys.

## Connection setup between sshd and ssh in SSH protocol Version 2

To enable *sshd* to receive connection requests from *ssh*, the following requirements must be met for *sshd*:

- A host key (RSA or DSA, generally 1024 bits) exists on the computer on which *sshd* is running (server computer). The host key is the public key in a pair of public / private keys and identifies the server. This requirement is already satisfied by installation in POSIX.
- Forward security is implemented by a Diffie-Hellman Key Agreement which results in a mutual session key.
- SSH protocol Version 2 must be activated in the configuration file */etc/ssh/sshd\_config*.

The following activities are performed for every *ssh* connection request:

1. *sshd* sends its host key and a list of the encryption algorithms it supports to *ssh*. The encryption methods that can be used at present are 128-bit AES, Blowfish, 3DES, CAST128, Arcfour, 192-bit AES and 256-bit AES.
2. *ssh* checks whether it knows *sshd* by comparing the RSA host key with the public keys in its own local database:
  - If the RSA host key is contained in this database, the system continues with 3.).
  - If the RSA host key is not contained in this database, the client proceeds as follows:
    - DNS lookup for host keys only for RSA/DSA keys (not RSA1). If the configuration is suitable (*VerifyHostKeyDNS Yes*), *ssh* ascertains the RSA host key with the aid of a DNS request and a user query. If the query is answered with “no”, *ssh* aborts the connection. If the answer is “yes”, the client “learns” the RSA host key by entering it in the database of keys which it knows. Continue with 3.).
    - Otherwise the connection is not set up.
3. *ssh* generates a 256-bit random number, encrypts this with both the host key and the public key of *sshd*, and sends the result to *sshd* together with details of the encryption method selected. Both sides use the encrypted random number as the session key with which all communication over the connection is encrypted with immediate effect.

Session integrity is guaranteed by a Message Authentication Code (MAC).

4. Both sides use this random number as the session key for encrypting all communication in the current session. The rest of the session is encrypted using a symmetrical encryption method. Currently 128-bit AES, Blowfish, 3DES, CAST128, Arcfour, 192-bit AES or 256-bit AES is used for this purpose. *ssh* selects the encryption algorithm from the list which it received from *sshd* together with the host key (see step 2).

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 320](#).

`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`

Different methods are used for authenticating the OpenSSH client to the OpenSSH server depending on whether SSH protocol Version 1 or Version 2 is used.

#### Client authentication in the SSH protocol Version 1

SSH protocol Version 1 supports the following methods for client authentication, it being possible to specifically admit or exclude some methods depending on the configuration:

1. rhosts authentication
2. rhosts RSA authentication
3. RSA-based authentication
4. Password authentication

The methods available in the actual configuration are used one after the other until a method has successfully provided authentication or until all methods have failed. The authentication methods are explained below.

#### 1. rhosts authentication

rhosts authentication corresponds to authentication with the `r` utilities `rlogin` and `rsh`. Unlike the `rlogin` server, `Sshd` uses DNS host names rather than BCAM host names. The DNS host names are determined by reverse lookup from the IP address (see the [section “BS2000/OSD-specific restrictions” on page 327](#)).



#### CAUTION!

rhosts authentication is extremely insecure and should on all accounts be disabled by the administrator. In the default configuration of the OpenSSH server, rhosts authentication has consequently already been disabled.

Successful rhosts authentication merely requires a corresponding entry for the computer from which the user logs on in one of the following files:

- */etc/hosts.equiv* or */etc/ssh/hosts.equiv*

Syntactically both files have the same format and are searched in the specified order.

If an entry for the computer from which the user logs on exists in one of these files on the remote computer, and if the user names on both sides also match, the user may log on immediately.

- *\$HOME/.rhosts* or *\$HOME/.shosts*

Syntactically both files have the same format and are searched in the specified order.

If one of the two files exists in the *\$HOME* directory of the user on the remote computer and if this file contains an entry with the name of the client server and the user name on this computer, the user may log on immediately.

## 2. rhosts authentication in conjunction with RSA-based authentication

With this method, rhosts authentication is complemented by an RSA-based authentication of the client computer. The known keys of the client computer are stored in the files */etc/ssh/known\_hosts* and *\$HOME/.ssh/known\_hosts* of the OpenSSH server *sshd*.

rhosts authentication in conjunction with RSA-based authentication prevents IP and DNS spoofing.

## 3. RSA-based authentication

In the case of RSA-based authentication each user generates a pair of keys comprising the public and private keys for authorization purposes. The OpenSSH server knows the user's public key if this key is stored in the server files */etc/ssh/known\_hosts* and *\$HOME/.ssh/known\_hosts*. Only the users themselves know their own private keys.

*ssh* handles the RSA authentication protocol automatically. The user generates the key pair by calling the OpenSSH basic tool *ssh-keygen*. RSA authentication can be performed most conveniently using the authentication agent *ssh-agent*. *ssh-keygen* and *ssh-agent* are described in the interNet Services User Guide.

#### 4. Password authentication

If the other authentication methods fail, *ssh* requests a user password and sends this to the remote computer in encrypted form so that it can be checked.

If authentication is not successful with any of the above-mentioned methods, the OpenSSH client's connection request is rejected.



SSH protocol Version 1 supports no mechanisms which ensure great integrity of the connection.

#### Client authentication in SSH protocol Version 2

SSH protocol Version 2 supports similar authentication methods to SSH protocol Version 1. When using the default value for the *PreferredAuthentication* option in the server configuration file *sshd\_config* (see [page 316](#)), the OpenSSH client executes the following authentication methods one after the other:

1. rhosts 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 is similar to RSA authentication and permits the use of RSA and DSA algorithms. The OpenSSH client signs the session ID with its private key (*\$HOME/.ssh/id\_dsa* or *\$HOME/.ssh/id\_rsa*) and sends the result to the OpenSSH server. The server checks whether the corresponding public key is contained in its *\$HOME/.ssh/authorized\_keys* file. If both keys exist and the signature is correct, the server accepts the connection.



SSH protocol Version 2 supports additional mechanisms which ensure confidentiality and integrity of the connection:

- Confidentiality is guaranteed by encrypting the data traffic using 3DES, Blowfish, CAST128 or Arcfour.
- Integrity is guaranteed by encrypting the data traffic using hmac-md5 or hmac-sha1.

## Server authentication

The OpenSSH client *ssh* automatically maintains and checks a database which contains the host IDs of all computers with which *ssh* has ever had contact. Host keys are stored on a user-specific basis in the `$HOME/.ssh/known_hosts` file. *ssh* also checks the `/etc/ssh/ssh_known_hosts` file for known computers. Each new computer is automatically stored in the user-specific file after it has been confirmed by the user.

As soon as the host ID of a computer changes, *ssh* reports this and blocks the existing passwords. This prevents a Trojan Horse from being able to call the users' passwords. This also permits "man-in-the-middle" attacks to be repelled which could otherwise bypass the encryption.

The *StrictHostKeyChecking* option in the configuration file *ssh\_config* (see [page 316](#)) is used to suppress logons to computers whose host key is unknown or has been changed. 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 316](#)), 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 321](#)) (SSH protocol Version 1).
- which are permitted for public key authentication (pubkey authentication, see [page 322](#)) (SSH protocol Version 2).

Using the `AuthorizedKeyFile` option in the `sshd_config` file (see [page 316](#)) 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_known_hosts`, `$HOME/.ssh/known_hosts`*

These files contain a list of all host public keys of all known remote computers. The global file `/etc/ssh/ssh_known_hosts` should be provided by the administrator (optional). The user-specific files `$HOME/.ssh/known_hosts` are maintained automatically. Each time users log on to an unknown computer, their key is entered in a user-specific file.

A detailed description of the syntax and options of the `/etc/ssh/ssh_known_hosts` and `$HOME/.ssh/known_hosts` files is provided in the OpenSSH man pages.

*/etc/ssh/ssh\_host\_key, /etc/ssh/ssh\_host\_dsa\_key, /etc/ssh/ssh\_host\_rsa\_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\_key.pub, /etc/ssh/ssh\_host\_dsa\_key.pub, /etc/ssh/ssh\_host\_rsa\_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\_key, /etc/ssh/ssh\_host\_dsa\_key, /etc/ssh/ssh\_host\_rsa\_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.

*\$HOME/.ssh/authorized\_keys*

This file contains a list of the public keys (RSA or DSA) with which you can log into a user's account. The file must be readable for a user with root authorization and should not be accessible for other users. The format of the file is described in the OpenSSH man pages. The user will transfer the contents of his/her files *identity.pub*, *id\_dsa.pub* and/or *id\_rsa.pub* to this file, as described for *ssh-keygen* (see the interNet Services User Guide).

*/etc/ssh/ssh\_known\_hosts, \$HOME/.ssh/known\_hosts*

These files are queried to check the host's public key in the event of

- combined *rhosts*/RSA authentication (see [page 321](#)) with SSH protocol Version 2.
- public key authentication (see [page 322](#)) with SSH protocol Version 2.

The key is accepted only if it is listed in one of the aforementioned files. The client uses the same files to ensure that it is connected to the intended remote computer.

Only users with root authorization and the owner should have write permission for these files. */etc/ssh/ssh\_known\_hosts* should be readable for all users. *\$HOME/.ssh/known\_hosts* can, but need not, be readable for all users.

*/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/OSD-specific restrictions

When working with OpenSSH in a BS2000/OSD 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/OSD side which is configured in the *\$TSOS.SYSDAT.SOCKETS.nnn.SOC6.RESOLV* or *\$TSOS.SYSDAT.LWRES.D.nnn.RESOLV.CONF* file, but the DNS resolver library of TCP-IP-SV:DNS(BS2000) (see [chapter "DNS" on page 221](#)). 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.LWRES.D.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 computers 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/OSD and OSD/POSIX. Thus in BS2000/OSD 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).

### Restrictions when using the BS2000/OSD dialog interface

You can only start the *ssh* client program in a remote login session, i.e. you must be logged onto POSIX by means of *rlogin*, *ssh*, or *slogin*.

In \$DIALOG or TELNET sessions you can only use *ssh* if `stdin`, `stdout` and `stderr` have been redirected to a file or a POSIX pipe, otherwise *ssh* will abort and issue the following message:

```
select: Invalid fileno
```

---

## 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/](http://www.postfix.org/) (Postfix)
- [www.washington.edu/imap/](http://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 821 and RFC 2821. 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/OSD is the product Postfix Version 2.7.1. This version applies for the creation time of this manual. Rebasings 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 manageability, 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 361](#).

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 345](#)).

To permit e-mails to be sent from BS2000/OSD, 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/OSD.

## 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.

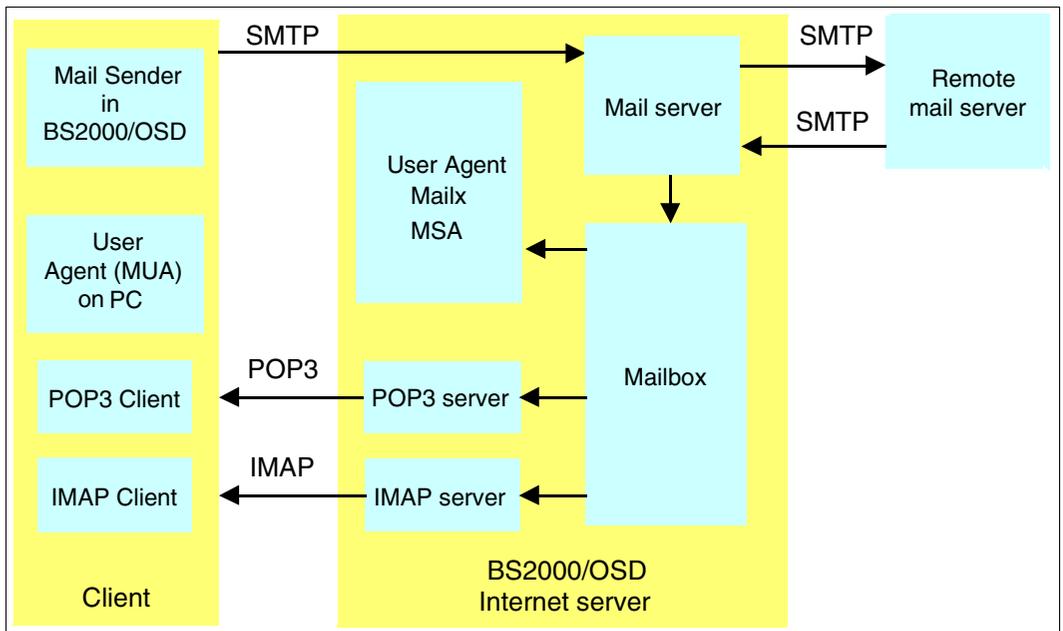


Figure 3: Client/server mail communication via SMTP, POP3 and IMAP

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, Eudora and 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/OSD 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 822 or 2822. 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 SMTP 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 238](#)).

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 `$TSOS.SYSDAT.LWRES.D.nnn.RESOLV.CONF` or `$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  $\leq$  Default POSIX User Number
  - POSIX Group Number  $\leq$  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/OSD account: SYSACC
  - Address space limit: 32 MBUnder 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/OSD account: SYSACC
  - Address space limit: 32 MBUnder 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.LWRES.D.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/messages* 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 possibly remaining 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/OSD 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/libexec/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             nmh     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 <i>inetd</i> )
/etc/inet/services	Text file	Assignment of service names to port numbers
/etc/inet/inetd.conf	Text file	Configuration file for <i>inetd</i>
/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	<i>nmn</i>	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 238](#)).

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 certificate 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.

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 357](#)) or *postalias* (see [page 359](#)) 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 349](#)).

#### Two different types of index files

The BS2000/OSD 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/OSD 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<sub>1</sub>, Value<sub>2</sub>, Value<sub>3</sub>, ..., Value<sub>n</sub>

#### 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<sub>1</sub>, Value<sub>2</sub>, Value<sub>3</sub>, ..., Value<sub>n</sub>

One or more receiver addresses.

The following can be specified as *Value<sub>i</sub>*, *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<sub>1</sub>, Value<sub>2</sub>, ..., Value<sub>n</sub>.

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 original 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.

<b>postconf</b>
[ -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.

<b>postsuper</b>
<code>[-p] [-s] [-d &lt;queue-id&gt;] [-h &lt;queue-id&gt;] [-H &lt;queue-id&gt;] [&lt;directory&gt; ...]</code>

**-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

## &lt;directory&gt; ...

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 348](#)). 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/OSD ported version of Postfix supports the *hash* and *btree* formats for index files.

<b>postmap</b>
[ <b>-q</b> <key>] [ <b>-d</b> <key>] [ <b>-i</b> ] [ <b>hash: btree:</b> ]<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/OSD 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 349](#)).

*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/OSD ported version of *postalias* supports the *hash* and *btree* formats for index files.

<b>postalias</b>
[ -q <key> ] [-d <key>] [-i <key>] [ <b>hash: btree:</b> ]<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 351](#).

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 335](#)).

## 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 335](#)).

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/OSD

## 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 370](#)).

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 367](#) and [“logLevel” on page 368](#)).

## 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.nmm.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.

<b>defaultOptionFileName</b>
<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 367](#)).

<b>backendConfigurationFileName</b>
<filename 1..54>

<filename 1..54>

Name of the configuration file.

Default: \$.SYSDAT.MAIL.*nnn*.SERVICE.OPT

## senderSuffix

When the *useSenderSuffix* option (see [page 366](#)) 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 concatenation of BS2000 user ID and suffix.

<b>senderSuffix</b>
<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.

<b>useSenderSuffix</b>
<b><u>NO</u>   YES</b>

### NO

No automatic generation of the sender address will be done. This is the default.

### YES

When additionally the *senderSuffix* option (see [page 365](#)) 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 363](#)). This can be overwritten using the commands `START-MAIL-SERVICE` and `MODIFY-MAIL-SERVICE-PARAMETERS`.

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.

<b>logFile</b>
<filename 1..54>

<filename 1..54>

Name of the logging file.

Default: \$.SYSDAT.MAIL.*nmn*.LOG

## logLevel

The *logLevel* option specifies the priority of the diagnostic messages to be logged.

<b>logLevel</b>
<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.

<b>logMailContent</b>
<b><u>NO</u> / YES</b>

**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.

<b>mailServer</b>
<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.

<b>mailServerPort</b>
<port-number>

<port-number>

Port number of the mail server.

Default: 25

## mailLogLevel

The *mailLogLevel* option specifies which data is to be logged.

<b>mailLogLevel</b>
<b>0   1   2</b>

### **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 371](#)).

<b>mailLogFile</b>
<filename 1..54>

<filename 1..54>

Name of the e-mail logging file.

Default: \$.SYSDAT.MAIL.*nnn*.MAILLOG

## retry Limit

This option specifies how often the mail delivery to a mail server shall be repeated when the server indicates with its return code a temporary error condition.

<b>retryLimit</b>
<limit>

<limit>

Number of retries, when the mail server indicates a temporary error condition. The time between two retries increases with every retry in an exponentially way, until a given limit is reached.

Default is 10.

## 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.

<b>tempFilePrefix</b>
<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.

<b>tlsSecureConnection</b>
<b><u>NONE</u>   OPTIONAL   REQUIRE</b>

### **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 2 and 3 of the SSL protocol and version 1 of the TLS protocol. You can use the *tlsProtocol* option to select which of these protocols are to be enabled.

<b>tlsProtocol</b>
[+   -] {SSLv2   SSLv3   TLSv1   ALL } ...

+

The protocol specified following this is permissible.

-

The protocol specified following this is not permissible.



If neither “+” nor “-” is specified, the effect is the same as when “+” is specified.

*Example*

`tlsProtocol SSLv3 TLSv1` **and** `-tlsProtocol ALL -SSLv2` **have the same effect.**

### SSLv2

Version 2 of the SSL protocol

### SSLv3

Version 3 of the SSL protocol

### TLSv1

Version 1 of the TLS protocol

### ALL

All protocols are to be enabled.

Default: ALL -SSLv2

## 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.

<b>tlsCipherSuite</b>
<specification>

<specification>

Specification in a list of preferred encryption methods (for more information, see the description of the structure of a specification below).

Default: ALL: !EXP: !ADH

### Description of the structure of a specification

The specification consists of one or more cipher mnemonics separated by a colon (:).

A cipher mnemonic can assume the following forms:

- A cipher mnemonic can consist of a single encryption suite (e.g. DES-CBC-SHA).
- A cipher mnemonic can represent:
  - A list of encryption suites that contain a particular algorithm
  - Encryption suites of a particular type

For example, SHA1 represents all encryption suites that use the SHA1 digest algorithm, and SSLv3 represents all SSL version 3 algorithms.

- Lists of encryption suites can be combined to form a single cipher mnemonic using the plus sign (+). This is then interpreted as a logical AND operation. SHA1+DES thus represents all encryption suites that contain the SHA1 and DES algorithms.
- You can precede each cipher mnemonic with an exclamation mark (!), a minus sign (-) or a plus sign (+):
  - If you precede it with an exclamation mark (!), the encryption suites are permanently removed from the list of preferred methods and do not reappear in it even if they are explicitly specified.
  - If you precede it with a minus sign (-), the encryption suites are removed from the list of preferred methods, but some or all of them can be added again using subsequent options.
  - If you precede it with a plus sign (+), the encryption suites are moved to the end of the list of preferred methods. No encryption suites are added to the list.

- If none of these three characters precedes the cipher mnemonic, it is interpreted as a list of encryption suites to be appended to the current list of preferred methods. If this includes an encryption suite that is already contained in the current list of preferred methods, it is ignored and not moved to the end of the list.
- The cipher mnemonic “@STRENGTH” can be inserted at any point to sort the current list of preferred methods by the strength of the encryption algorithms (i.e. by the length of the encryption key, above all).

#### *Permissible cipher mnemonics*

The permissible cipher mnemonics are described below.

#### ALL

All encryption suites with the exception of the eNULL ciphers. These must be activated explicitly.

#### HIGH

Encryption suites with key lengths greater than 128 bits. Since 3DES is evaluated with a length of 168 bits (instead of 112 bits, as evaluated by some cryptologists), it belongs to this suite class.

#### MEDIUM

Encryption suites with a key length of 128 bits.

#### LOW

Encryption suites with a key length of 64 or 56 bits. Export encryption suites are an exception.

#### 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 (i.e. algorithms without encryption). Since these do not offer encryption and are therefore a security risk, they are disabled by default and have to be explicitly specified, if required.

#### aNULL

Encryption suites without authentication. These are currently the anonymous Diffie-Hellman algorithms. These algorithms are vulnerable to “man in the middle” attacks, so it is not advisable to use them.

#### kRSA, RSA

Encryption suites with RSA key exchange.

**kEDH**

Encryption suites with a short-lived (ephemeral) Diffie-Hellman key agreement.

**aRSA**

Encryption suites with RSA authentication (i.e. the certificates contain RSA keys).

**aDSS, DSS**

Encryption suites with DSS authentication (i.e. the certificates contain DSS keys).

**TLSv1, SSLv3, SSLv2**

TLS v1, SSL v3 or SSL v2 encryption suites. The TLSv1 and SSLv3 suites are identical.

**DH**

Encryption suites with Diffie-Hellman key exchange, including anonymous exchange.

**ADH**

Encryption suites with anonymous Diffie-Hellman key exchange.

**AES**

Encryption suites with AES encryption (128- and 256-bit key length).

**3DES**

Encryption suites with Triple DES encryption.

**DES**

Encryption suites with DES encryption (not Triple DES).

**RC4**

Encryption suites with RC4 encryption.

**RC2**

Encryption suites with RC2 encryption.

**MD5**

Encryption suites with MD5 hash function.

**SHA1, SHA**

Encryption suites with SHA1 hash function.

The table below shows the available encryption suites.

Name	Version	Key exchange	Authenti- cation	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	
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

Available encryption suites

Name	Version	Key exchange	Authentication	Encryption	Digest	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 encryption suites

## 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).

<b>tlsCertificateFile</b>
<filename 1..54>   <b>*NONE</b>

<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.

<b>tlsKeyFile</b>
<filename 1..54>   <b>*NONE</b>

<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.

<b>tlsCACertificateFile</b>
<filename 1..54>   <b>*NONE</b>

<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).

<b>tlsCARevocationFile</b>
<filename 1..54>   <b>*NONE</b>

<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.

<b>tlsVerifyServer</b>
<b><u>YES</u></b>   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 382](#)).
- It makes no sense to specify a depth of 0. In this case, only self-signed certificates would be permissible.

<b>tlsVerifyDepth</b>
<depth>

<depth>

Number of maximum permissible certificates between the server certificate and the certificate known to the service task.

Default: 1

## tlsUseCryptoHardware

The *tlsUseCryptoHardware* option allows you to specify whether crypto hardware such as an *openCRYPT*<sup>™</sup> box (see the *openCRYPT* manual) is to be used to calculate cryptographic algorithms.

<b>tlsUseCryptoHardware</b>
<b>NO   YES</b>

### **NO**

Crypto hardware is not used (default).

### **YES**

Crypto hardware is used.

## 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
<p><b>ORDER-LIMIT = <u>255</u> / &lt;integer 1..32000&gt;</b></p> <p><b>,CONFIGURATION-FILE = *<u>STD</u> / &lt;filename 1..54 without-gen&gt;</b></p>

#### 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 367](#).

**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 363](#)).

**CONFIGURATION-FILE = <filename 1..54 without-gen>**

Specifies the file that contains the configuration options.

**Return codes**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Meaning</b>
	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 367](#).

### **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 372](#)). 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 367](#)). 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 368](#)).

### **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 368](#)).

**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 368](#)).

**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 368](#)).

**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 368](#)).

### 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

<b>SHOW-MAIL-SERVICE-PARAMETER</b>

### 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**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Meaning</b>
	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

<b>STOP-MAIL-SERVICE</b>

### 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

YML0100 INTERNAL ERROR IN MAILCLNT-SUBSYSTEM-INITIALIZATION;  
ERROR CODE '(&&00)'

### Meaning

Some operating resources aren't available. This resources are according to the error code:

0000 - No memory space for subsystem object available.

0001 - No memory space for MAILCLNT error trace available.

0002 - No memory space for MAILCLNT event trace available.

0004 - Task lock can't be created.

0005 - Server bourse can't be created.

0006 - No memory space for file object available.

0007 - Error at reading default configuration file.

0008 - Error at reading configuration file.

0011 - Error at connection of holder task with SDF.

0012 - Error at notification of DSSM.

### Response

<None>

YML0105 ERROR IN DSSM-MAILCLNT-INTERFACE;  
ERROR CODE '(&&00)'

### Meaning

The cause for this message is according to the error code:

0001 - A notification of DSSM failed.

0002 - Some internal error occurred during restart of MAILCLNT subsystem.

### Response

<None>

YML0110 THE FOLLOWING SYSTEM DUMP OF TASK YML1 SERVES AS DIAGNOSTIC AID

### Meaning

If the reason for the occurred problem isn't already clear from the previous message, then take the dump please and send it to your support.

### Response

<None>

YML0120 SUBSYSTEM ASTI IS NOT AVAILABLE

### Meaning

The ASTI subsystem must be available for command execution.

### Response

Inform your system administrator.

YML0130 CONFIGURATION FILE '(&&00)' UNREADABLE OR DOESN'T EXIST

**Meaning**

The specified configuration file is unreadable or does not exist.

**Response**

Specify the name of a readable file.

YML0131 Error accessing the file '(&00)

**Meaning**

The specified file cannot be accessed.

**Response**

Specify the valid name of an accessible file.

YML0132 STARTING MAIL SERVICE FAILED. ASTI RC: '(&&00)'

**Meaning**

The start of the ASTI service MAILCLNT failed with the displayed ASTI return code.

**Response**

<None>

YML0133 BCAM IS NO LONGER AVAILABLE. MAILCLNT SERVICE WILL BE TERMINATED

**Meaning**

The ASTI service MAILCLNT was terminated because BCAM is no longer available.

**Response**

Start the MAILCLNT service after BCAM reavailability again.

YML0134 THE MAIL SERVICE IS ALREADY RUNNING

**Meaning**

The ASTI service MAILCLNT is already running.

**Response**

<None>

YML0136 IMON COULD NOT FIND LOAD LIBRARY

**Meaning**

IMON could not find the load library with the program for the ASTI service.

**Response**

Check the product installation.

YML0138 MAIL SERVICE IS NOT RUNNING

**Meaning**

The mail service is not active and can therefore not be stopped.

**Response**

<None>

YML0140 MAIL SERVICE IS NOT RUNNING

**Meaning**

For using this command the mail service must be running.

**Response**

Start the mail service and repeat the command.

YML0142 USER OPTION FILE DOES NOT EXIST OR IS INACCESSIBLE

**Meaning**

The user option file does not exist or is not readable.

**Response**

Check the specified file name and the file attributes.

YML0144 MAIL ATTACHMENT FILE DOES NOT EXIST OR IS INACCESSIBLE

**Meaning**

At least one of the specified attachment files does not exist or is not readable.

**Response**

Check the specified file names and the file attributes.

YML0146 MAIL SEND ORDER TOO BIG

**Meaning**

The mail send order built from the command operands is too big.

**Response**

Downsize the built mail send order by e.g. splitting the mail into several with smaller recipient lists or with fewer attachments or by putting message or attachment texts into files instead specifying them directly as command operands.

YML0148 ORDER LIMIT FOR MAIL SERVICE REACHED

**Meaning**

The mail service reached a workload limit and therefore can't accept further orders at present.

**Response**

Repeat the command after some waiting time. If the problem continues to exist even after several repetitions or if it occurs frequently, please inform your system administrator.

YML0150 PARAMETER ERROR IN CALL TO MAILCLNT SUBSYSTEM

**Meaning**

The calling entity has made an invalid call to the MAILCLNT subsystem or an unexpected internal error occurred.

**Response**

Contact the system administrator so that diagnostic material can be produced if necessary and possible.

YML0160 MAIL SEND ORDER SUBMITTED WITH ORDER ID '(&&00)'

**Meaning**

The mail send order is submitted to ASTI with the specified order id.

**Response**

<None>

YML0170 MAIL SEND ORDER CARRIED OUT; MAIL SERVER RESULT: '(&&00)'

**Meaning**

The mail send order is submitted to an SMTP mail server.

**Response**

<None>

YML0171 MAIL SEND ORDER FAILED; MAIL SERVER RESULT: ' '(&&00)'

**Meaning**

The submission of the mail send order to an SMTP mail server failed.

**Response**

If the mail server result doesn't already enable you to recognize and correct the error, contact the system administrator.

YML0172 MAIL SERVICE NOT ACTIVE OR WAS NOT ABLE TO COMPLETE ORDER

**Meaning**

The action could not be performed, because the ASTI mail service is not started or ASTI could not report a successful order completion, because e.g. the service terminated abnormally. In some cases the order may have nevertheless completed successfully.

**Response**

Inform your system administrator.

YML0173 SUBSYSTEM ASTI REPORTED '(&00)'. IN SYSTEM MODE: /HELP-MSG (&00)

**Meaning**

For more detailed information about the DMS error code enter /HELP-MSG in system mode.

**Response**

None.

YML0174 NO SENDER ADDRESS SPECIFIED

**Meaning**

Neither per FROM operand nor per user option file a sender address has been specified.

**Response**

Specify a sender address.

YML0176 UNEXPECTED ASTI ERROR; ASTI RC: '(&00)'

**Meaning**

The call of the ASTI subsystem failed with an unexpected error code. The corresponding diagnostic information can be found within the Software Error Logging file under the rectype YML9000.

**Response**

Inform your system administrator.

YML0178 ORDER CANCELLED BECAUSE OF TIMEOUT

**Meaning**

The order has been cancelled because the specified wait time has been reached.

**Response**

Wait some time and repeat the command. If after some repetitions the error still occurs, inform your system administrator please.

YML0203 INVALID OR NON-EXISTENT S/MIME FILE

**Meaning**

A key, certificate or crl file specified in the command or in an option file is non-existent or contains invalid data.

**Response**

<None>

YML0210 THE ORDER COULD NOT BE FOUND

**Meaning**

The specified order could not be found.

**Response**

Check the order ID.

YML0211 NO ORDER FOUND

**Meaning**

No order found with specified selection conditions.

**Response**

Check the selection conditions.

YML0212 YOU ARE NOT ALLOWED TO ACCESS THIS ORDER

**Meaning**

The specified order is owned by another user.

**Response**

Check the order ID.

YML0214 COMMAND CAN'T BE EXECUTED BECAUSE OF RESOURCE SATURATION

**Meaning**

Some resources needed for command execution are not available at the moment.

**Response**

Wait some time and repeat the command.

YML0215 NO RESULT REQUESTED AT ORDER SUBMISSION

**Meaning**

No result can be delivered because at order submission no result return has been requested.

**Response**

<None>

YML0216 NO ACCESS TO ORDER, BECAUSE SUBMITTED BY FOREIGN TASK

**Meaning**

The order can't be accessed as long the order submitting task is still active.

**Response**

Change to the order submitting task or wait for the completion of this task.

YML0218 ORDER IS NOT A MAIL SEND ORDER

**Meaning**

The order is not a mail send order but an order used for internal purposes.

**Response**

<None>

YML0220 YOU ARE NOT ALLOWED TO DELETE THIS ORDER

**Meaning**

You are not allowed to delete this order, because it is submitted by an other user.

**Response**

<None>

YML0221 YOU ARE NOT AUTHORIZED TO DELETE THE ORDERS OF OTHER USERS

**Meaning**

You do not have the required privileges to delete the orders of other users.

**Response**

<None>

YML0222 ORDER IS STILL NOT COMPLETED

**Meaning**

The order is just in processing so still no result is available.

**Response**

Wait some time and repeat then the command.

---

## 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 (BS2000/OSD)**

**BCAM**

User Guide

**interNet Services (BS2000/OSD)**

User Guide

**openNet Server (BS2000/OSD)**

**IPv6 Introduction and Conversion Guide, Stage 1**

User Guide

**openNet Server (BS2000/OSD)**

**IPSec**

User Guide

**openCRYPT (BS2000/OSD)**

**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/OSD)  
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**  
**Introductory Guide to Systems Support**  
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](http://www.ietf.org)

---

# Index

- \$HOME/.rhosts 321
- \$HOME/.shosts 321
- \$HOME/.ssh/authorized\_keys 326
- \$HOME/.ssh/known\_hosts 324, 326
- /etc/hosts 223
- /etc/hosts.allow 326
- /etc/hosts.deny 326
- /etc/hosts/equiv 321
- /etc/imap 341
- /etc/imap/certs 341
- /etc/imap/MAKE.CERT.sh 341
- /etc/imap/private 341
- /etc/inet 341
- /etc/inet/inetd.conf 341
- /etc/inet/services 341
- /etc/init.d/MAIL.postfix 339
- /etc/named.conf
  - DNS configuration file 249
  - examples 253
- /etc/nologin 326
- /etc/ntp.conf 279
- /etc/postfix 338
- /etc/postfix/main.cf 339
- /etc/postfix/master.cf 338
- /etc/rc0.d/K17MAIL.postfix 339
- /etc/rc2.d/S97MAIL.postfix 339
- /etc/ssh/moduli 325
- /etc/ssh/shosts.equiv 321
- /etc/ssh/ssh\_host\_dsa\_key 325
- /etc/ssh/ssh\_host\_dsa\_key.pub 325
- /etc/ssh/ssh\_host\_key 325
- /etc/ssh/ssh\_host\_key.pub 325
- /etc/ssh/ssh\_host\_rsa\_key 325
- /etc/ssh/ssh\_host\_rsa\_key.pub 325
- /etc/ssh/ssh\_known\_hosts 324, 326
- /etc/ssh/sshd\_config 316
- /opt/MAIL 338, 341
- /opt/MAIL/imap 341
- /opt/MAIL/imap/readme 341
- /opt/MAIL/imap/sbin 341
- /opt/MAIL/imap/sbin/imapd 341
- /opt/MAIL/imap/sbin/ipop3d 341
- /opt/MAIL/imap/share 341
- /opt/MAIL/postfix 338
- /opt/MAIL/postfix/bin 338
- /opt/MAIL/postfix/bin/mailq 338
- /opt/MAIL/postfix/bin/newaliases 338
- /opt/MAIL/postfix/libexec/postfix 338
- /opt/MAIL/postfix/libexec/postfix/master 338
- /opt/MAIL/postfix/libexec/postfix/pickup 338
- /opt/MAIL/postfix/libexec/postfix/qmgr 338
- /opt/MAIL/postfix/libexec/postfix/smtp 338
- /opt/MAIL/postfix/libexec/postfix/smtpd 338
- /opt/MAIL/postfix/libexec/postfix/local 338
- /opt/MAIL/postfix/readme 338
- /opt/MAIL/postfix/sbin 338
- /opt/MAIL/postfix/sbin/postalias 338
- /opt/MAIL/postfix/sbin/postcat 338
- /opt/MAIL/postfix/sbin/postconf 338
- /opt/MAIL/postfix/sbin/postdrop 338
- /opt/MAIL/postfix/sbin/postfix 338
- /opt/MAIL/postfix/sbin/postmap 338
- /opt/MAIL/postfix/sbin/postqueue 338
- /opt/MAIL/postfix/sbin/postsuper 338
- /opt/MAIL/postfix/sbin/sendmail 338
- /opt/MAIL/postfix/share 338
- /usr/sbin/in.imapd 341
- /usr/sbin/in.ipop3d 341

/var/adm/messages  
  message file 35, 237, 251, 276  
/var/empty 325  
/var/mail 339, 341  
/var/mail/USER 339, 341  
/var/run/sshd.pid 325  
/var/spool/postfix 339

### A

-A 70, 160  
A (Address) 227  
AAAA (quad A) 227  
accounting in FTP 115  
-acctActive 81  
acctActive 124  
-acctFile 81  
acctFile 124  
activate debugging function  
  NTP 300  
active mode 127  
addressing 279  
  clock 278, 279, 283  
adjtime call  
  NTP 272  
administration  
  NAMED 259  
administration and operation (DNS and  
  security) 260  
administration domains  
  local 226  
agent forwarding (OpenSSH) 311  
alias 23  
alias file 349  
  example 351  
alias file, see also postmap source file  
alias format 359, 360  
aliases, function 332  
-allowTosLogin 82  
alphanum-name (data type) 24  
-appPrefix 70  
arpfile 255  
association  
  NTP 271

ASSOCIATIONS command  
  NTP 301  
authentication  
  between ssh and sshd 320  
  client authentication 320, 322  
  password authentication 322  
  public key authentication 322  
  rhosts authentication 320, 321, 322  
  RSA-based authentication 321  
  server authentication 323  
AUTHENTICATION option 165, 187

### B

-B 71, 187  
backendConfigurationFileName 365  
Balanced-Tree (B-Tree), see btree  
BCAM host file  
  DNS 223  
bcamInterval 212  
BROADCASTCLIENT statement  
  NTP 287  
BROADCASTDELAY statement  
  NTP 287  
BS2000/OSD-PRNGD  
  see PRNGD (BS2000/OSD)  
btree file 348

### C

-C 72  
caching 232  
cat-id (data type) 24  
-childEnterJob 73  
-childJobClass 74  
-childName 72  
client (DNS) 223  
client authentication 320, 322  
clock  
  addressing 278, 283  
CLOCKVAR/CV command  
  NTP 302  
cmd 211  
cmdInterval 212  
CNAME (Canonical Name) 228

- code tables
  - selecting 164
- command-line options
  - NTP 296
- command-rest (data type) 24
- composed-name (data type) 24
- concept
  - DNS 223
- configuration
  - BS2000/OSD-PRNGD 207
  - DNS resolver 238
  - FTP 53
  - NTP daemon 291
  - OpenSSH 316
  - TELNET 145
- configuration file
  - /etc/ntp.conf (NTP) 291
  - /etc/ssh/sshd\_config 316
  - create for NTP 279
  - mail sender 367
- configuration file (DNS)
  - syntax 247
- configuration parameters
  - NTP 279
- configuring
  - FTP via option file 69
  - TELNET via option file 158
- connection setup between sshd and ssh 317, 319
- control connection 126
  - secure 103
- controlkey statement (NTP) 288
- convSelector 78
- COOKED command
  - NTP 299
- creating
  - DNS configuration file 238
- c-string (data type) 24
- D**
- D 72, 160
- data connection 126
  - secure 104
- data type
  - alphanum-name 24
  - cat-id 24
  - command-rest 24
  - composed-name 24
  - c-string 24
  - date 24
  - device 24
  - filename 25
  - fixed 24
  - integer 26
  - name 26
  - partial-name 27
  - posix-filename 27
  - posix-pathname 27
  - product-version 28
  - structured-name 28
  - text 28
  - time 28
  - vsn 28
  - x-string 29
  - x-text 29
- data types in SDF 24
- date
  - set with NTP 293
- date (data type) 24
- debug 122, 195
- DEBUG command
  - NTP 300
- debug level
  - specifying 160
- debugLevel 72
- defaultOptionFileName 364
- delay
  - NTP 270
- device (data type) 24
- diagnosis 248
  - DNS resolver 248
  - NAMED 262, 263
  - NTP 305
- dig (DNS) 264
  - examples 265
- directories
  - IMAP/POP3 server 341

- disableSiteExecCommand 84
  - disableSizeCommand 85
  - dispersion (NTP) 270
  - displaying current settings
    - of FTP and TELNET servers 119, 192
  - DNS
    - /etc/hosts 223
    - administration tools rndc, rndc-confgen 264
    - BCAM host file 223
    - concept 223
    - configuration file named.conf 249
    - diagnostic tools dig/, host/, nslookup 264
    - diagnostic tools, dig (examples) 265
    - domain level 225
    - domain structure 224
    - explanation 221
    - functionality 223
    - message format 228
    - modify configuration file 247
    - name server 231
    - NAMED 231
    - NAMED, stop 260
    - restart resolver (restartdns) 246
    - root domains 225
    - shutdown daemon (stopdns) 36
    - startup calls 246
    - storing information 227
    - tools 264
    - Transaction SIGNatures 257
    - zone 226
  - DNS (resource records) 227
  - DNS client 223
  - DNS development modules
    - installing/uninstalling 237
  - DNS name server, see NAMED
  - DNS name space 224
  - DNS resolver 223, 229
    - configuring 238
    - diagnosis and maintenance 248
    - DOMAIN entry 240
    - installing 235
    - modify configuration 247
    - NAMESEVER entry 239
    - OPTIONS entry 244
    - SEARCH entry 242
    - terminate 247
    - uninstalling 237
  - DNS resolver start 246
  - DNS resolver stop 247
  - DNS SECurity (DNSSEC) 233, 257
  - DNS server
    - see also DNS-NAMED 225
  - DNS-NAMED 231
  - DNSSEC 233, 257
  - DNSSEC (DNS SECurity) 233
  - domain
    - in-addr.arpa 226
  - DOMAIN entry
    - DNS resolver 240
  - domain level (DNS) 225
  - domain structure (DNS) 224
  - DRIFTFILE statement
    - NTP 288
  - DSSidLength 77
  - DUMMY module (TELNET exit) 198
  - dummy routine (FTP exit) 139
- ## E
- E 73, 161
  - ENCRYPTION option 165, 188
  - entropy sources (BS2000/OSD-PRNGD) 206
  - entropyThreshold 209
  - event
    - FTP exit 131
  - example
    - FTP exit 135
    - NTP configuration file 280
  - exit
    - TELNET 198
  - exit mechanisms
    - for FTP 130
    - for FTP server 139
  - exit routine
    - specifying (TELNET) 161
  - exit routines, TELNET 199
  - EXITTEL.C 198
  - explanation
    - DNS 221

- F**
- F 73
  - file 210
  - fileInterval 210
  - filename (data type) 25
  - fixed (data type) 24
  - format
    - DNS message 228
    - of a message (SMTP) 333
  - forwarder (server)
    - DNS 233
  - forwarding server 233
  - FTAClevel 71
  - FTACuserId 74
  - FTP
    - accounting 115
    - configuration and operation 53
    - configuring via option file 69
    - installing/uninstalling in POSIX 47
    - IPv6 addresses 126
    - TLS/SSL support 54
  - FTP exit 130
    - dummy routine 139
    - dummy routine, user-defined 140
    - events 131
    - example 135
    - security check 130
  - FTP exit routine
    - result 143
    - return code 143
    - user-defined 143
  - FTP server 129
    - displaying current settings 119, 192
    - exit mechanisms 139
    - logging file 119, 192
    - passive mode 127
    - proxy mechanism 128
    - shutdown 110, 112, 190
    - start 109, 189
    - TLS/SSL support 54
  - FTP server options
    - acctActive 81
    - acctFile 81
    - allowTosLogin 82
    - appPrefix (-A) 70
    - childEnterJob (-F) 73
    - childJobClass (J) 74
    - childName (-C) 72
    - convSelector (-U) 78
    - debugLevel (-D) 72
    - disableSiteExecCommand 84
    - disableSizeCommand 85
    - DSSidLength (-S) 77
    - FTAClevel (-B) 71
    - FTACuserId (-K) 74
    - initialChildCmds (-Z) 80
    - logonExtension (-L) 75
    - maxConn (-N) 75
    - OpenSSLlibName 107
    - portNumber (-P) 76
    - serverInfoFile (-E) 73
    - sizeCmdTimeLimit 107, 108
    - socketTraceLevel (-T) 77
    - systemExit (-X) 79
    - timeout (-O) 76
    - tlsAcceptableClientCAFile 99
    - tlsCAcertificateFile 98
    - tlsCArevocationFile 100
    - tlsCertificateChainFile 97
    - tlsCipherSuite 87
    - tlsDSACertificateFile 95
    - tlsRandFile 105
    - tlsRSACertificateFile 93
    - tlsRSAkeyFile 94, 96
    - tlsSecureControlConnection 103
    - tlsSecureDataConnection 104
    - tlsUseCryptoHardware 106
    - tlsVerifyClient 101, 184
    - tlsVerifyDepth 102
    - verbose (-V) 78
  - FTP subagent 129
  - FTP subevent
    - FTPBYE 131
    - FTPCMD 132
    - FTPCMDE 133
    - FTPLOG 131
  - FTPBYE (FTP subevent) 131
  - FTPCMD (FTP subevent) 132

FTPCMDE (FTP subevent) [133](#)

FTPLOG (FTP subevent) [131](#)

FUDGE statement

  NTP [283](#)

function [333](#)

  aliases [332](#)

functionality

  DNS [223](#)

  IMAP [333](#)

  POP3 [333](#)

  Sendmail [332](#)

## G

generating random numbers [205](#)

  in BS2000/OSD with PRNGD [206](#)

  in POSIX with prngd server daemon [219](#)

GPRBYTE [213](#)

## H

-H [188](#)

hash file [348](#)

header (message header) [333](#)

heterogeneous networks [128](#)

hierarchy

  time server [273](#)

host (DNS) [264](#)

HOST command

  NTP [298](#)

host file (DNS) [223](#)

HOSTNAMES command

  NTP [299](#)

## I

IMAP [333](#)

  TLS/SSL protection [345](#)

IMAP server

  directories [341](#)

  files [341](#)

  installing [340](#)

  port number [334](#)

  starting up [344](#)

  uninstalling [342](#)

implementation

  NTP [272](#)

in-addr.arpa

  domain [226](#)

index file [348](#)

  alias format [359, 360](#)

  generating [357, 359, 360](#)

  Postfix format [357](#)

  processing [357, 359](#)

index files

  types [348](#)

initial operation

  interNet Services [35](#)

-initialChildCmds [80](#)

input file, see source file [349](#)

installation

  DNS development modules [237](#)

  DNS resolver [235](#)

  LDAP [312](#)

  mail sender [363](#)

  NAMED [249](#)

  NAMED development modules [252](#)

  NTP [274](#)

  OpenSSH [312](#)

  TCP-IP-SV [33, 219, 235, 249, 274, 312](#)

installing

  FTP/TELNET in POSIX [47](#)

  IMAP server [340](#)

  mail server [334](#)

  OpenSSH [312](#)

  POP3 server [340](#)

  SMTP server [334](#)

integer (data type) [26](#)

interNet Services

  initial operation [35](#)

  stop calls [247, 260](#)

  uninstall [35](#)

  uninstallation [237, 251, 276](#)

  uninstalling [276](#)

INTR interface [122, 195](#)

  acctActive [124](#)

  acctFile [124](#)

  debug [122, 195](#)

  RDPROT [125, 197](#)

  shutdown [123, 196](#)

  trace [123, 196](#)

- IPv6 addresses
  - in FTP 126
  - in TELNET 197
- iterative query
  - DNS 232
- J**
- J 74
- K**
- K 74
- keyid-Kommando (NTP) 303
- keys statement (NTP) 289
- keysdir statement (NTP) 290
- L**
- L 75
- LDAP 329
  - installation 312
- licensing regulations 15
- local administration domains 226
- logFile 367
- logging
  - DNS resolver 248
  - NAMED 262
  - NTP 305
- logging file of FTP and TELNET servers 119, 192
- logging function
  - DNS resolver 248
- login process (sshd) 323
- logLevel 368
- logonExtension 75
- loopback 256
- M**
- Mail Delivery Agent (MDA) 332
- mail queue 354
  - displaying messages 356
  - processing 354, 355
- mail sender 363
  - configuration file 367
  - installation 363
  - MODIFY-MAIL-SERVICE-PARAMETER 386
  - option file 363
  - service commands 384
  - SHOW-MAIL-SERVICE-PARAMETER 389
  - START-MAIL-SERVICE 384
  - STOP-MAIL-SERVICE 391
- mail sender options
  - defaultOptionFileName 364
  - logFile 367
  - logLevel 368
  - mailLogLevel 371, 372
  - mailServer 370
  - mailServerPort 370
  - retryLimit 372
  - tempFilePrefix 373
  - tlsCACertificateFile 381
  - tlsCARevocationFile 382
  - tlsCertificateFile 380
  - tlsCipherSuite 375
  - tlsKeyFile 380
  - tlsProtocol 374
  - tlsSecureConnection 373
  - tlsUseCryptoHardware 383
  - tlsVerifyDepth 383
  - tlsVerifyServer 382
- mail senders in BS2000/OSD 363
- Mail server, starting up 343
- mail server 332
  - installing 334
  - uninstalling 334
- mail server options
  - backendConfigurationFileName 365
  - senderSuffix 365
  - useSenderSuffix 366
- mail server, see also SMTP server
- Mail Transfer Agent (MTA) 329
- Mail Transfer Agent, see also SMTP server
- Mail User Agent (MUA) 329
- mailbox 332
- mailLogLevel 371, 372
- mailServer 370
- mailServerPort 370
- maintenance
  - DNS resolver 248
  - NAMED 262
  - NTP 305

- master server [232](#)
- maxConn [75](#)
- MDA [332](#)
- message file
  - /var/adm/messages [35](#), [237](#), [251](#), [276](#)
- message header [333](#)
- message text [333](#)
- message, format (SMTP) [333](#)
- migration from Sendmail to Postfix [361](#)
- MIME mechanism [331](#), [333](#)
- minimalEntropy [208](#)
- modify
  - configuration of the DNS resolvers [247](#)
  - DNS configuration file [247](#)
  - zone data files of NAMED [260](#)
- MODIFY-MAIL-SERVICE-PARAMETER [386](#)
- MTA (Mail Transfer Agent) [329](#)
- MUA, see Mail User Agent
- Multipurpose Internet Mail Extensions, see MIME
- MX (Mail Exchanger) [227](#)

## N

- N [75](#), [161](#)
- name (data type) [26](#)
- name prefix, specifying [160](#)
- name server (DNS) [231](#)
- name space (DNS) [224](#)
- NAMED [249](#), [262](#)
  - administering [259](#)
  - configuration file [253](#)
  - diagnosis [263](#)
  - diagnosis and maintenance [262](#)
  - diagnostic options [263](#)
  - executing without root authorization [258](#)
  - installing [249](#)
  - logging [262](#)
  - modify zone data files [260](#)
  - restart call [259](#)
  - security [257](#)
  - shutting down [260](#)
  - start [259](#)
  - startup call [259](#)
  - uninstalling [251](#)
  - zone data files [254](#)

- NAMED development modules
  - installing/uninstalling [252](#)
- NAMED see DNS NAMED
- named.conf [253](#)
- NAMESERVER entry
  - DNS resolver [239](#)
- network security with OpenSSH [309](#)
- Network Time Protocol, see NTP
- networks, heterogeneous [128](#)
- newaliases [360](#)
- NS (Name Server) [227](#)
- nslookup (DNS) [264](#)
- NTP
  - /etc/ntp.conf [291](#)
  - activate debugging function [300](#)
  - addressing [278](#), [279](#), [283](#)
  - adjtime call [272](#)
  - association [271](#)
  - ASSOCIATIONS command [301](#)
  - BROADCASTCLIENT statement [287](#)
  - BROADCASTDELAY statement [287](#)
  - CLOCKVAR/CV command [302](#)
  - configuration parameters [279](#)
  - controlkey statement [288](#)
  - COOKED command [299](#)
  - DEBUG command [300](#)
  - delay [270](#)
  - diagnosis [305](#)
  - dispersion [270](#)
  - DRIFTFILE statement [288](#)
  - FUDGE statement [283](#)
  - HOST command [298](#)
  - HOSTNAMES command [299](#)
  - implementation [272](#)
  - installation [274](#)
  - keyid-Kommando [303](#)
  - keys statement [289](#)
  - keymdir statement [290](#)
  - logging [305](#)
  - maintenance [305](#)
  - ntp [277](#)
  - ntpdate [277](#), [293](#), [295](#)
  - ntp-keygen [295](#)
  - NTPTRACE command [306](#)

- NTPVERSION command 300
- offset 270
- passwd command 304
- PEER statement 284
- PEERS command 302
- primary time server 269
- program 273
- query ntpq status 296
- query status 297
- RAW command 299
- requestkey statement 289
- restart 277
- RESTRICT statement 281
- sample configuration file 280
- secondary time server 269
- SERVER statement 279
- set date/time 293
- shutdown 277
- shutdown daemon 36
- shutting down 277
- starting 277
- startup 277
- startup options 291
- stratum 269
- time server 269
- TIMEOUT command 298
- trace functionality 306
- trustedkey statement 289
- NTP daemon
  - options 291
  - startup options 291
- NTP time synchronization 277
- NTPD
  - set date 293
  - set time 293
- ntpd
  - NTP 277
  - options 291
  - startup options 291
- ntpdate
  - NTP 277, 295
  - NTP program 293
- ntp-keygen
  - NTP program 295
- ntpq
  - query NTP status 296
- NTPTRACE command
  - NTP 306
- NTPVERSION command
  - NTP 300
- O**
- O 76
- offset
  - NTP 270
- OpenSSH 307
  - BS2000/OSD-specific restrictions 327
  - component parts 308
  - concept 308
  - configuring 316
  - features 310
  - network security 309
  - server daemon sshd 316
  - uninstalling 315
- OpenSSH server daemon
  - files 324
- OpenSSH server daemon see also sshd
- operation
  - FTP 53
  - Postfix server 348
  - TELNET 145
- option file
  - FTP 69
  - mail sender 363
  - SYSSSI 363
  - TELNET 158
- options
  - parameter lines 69
- OPTIONS entry, DNS resolver 244
- options see also FTP server options, TELNET server options, TLS/SSL options 69
- P**
- P 76, 162
- parameter line options 69
- partial-filename (data type) 27
- passive mode 127
- passwd command (NTP) 304

- password authentication [322](#)
  - PEER statement
    - NTP [284](#)
  - PEERS command
    - NTP [302](#)
  - PLAM library
    - SINLIB.MAIL.nnn.POSTFIX [334, 340](#)
  - poolSize [208](#)
  - POP3 server
    - directories [341](#)
    - files [341](#)
    - function [333](#)
    - installing [340](#)
    - port number [334](#)
    - starting up [344](#)
    - TLS/SSL protection [345](#)
    - uninstalling [342](#)
  - port forwarding (OpenSSH) [311](#)
  - port number
    - IMAP server [334](#)
    - POP3 server [334](#)
    - SMTP server [334](#)
    - specifying [162](#)
  - portNumber [76](#)
  - POSIX
    - terminate subsystem [36, 247, 260](#)
  - posix-filename (data type) [27](#)
  - posix-pathname (data type) [27](#)
  - postalias [359](#)
    - source file [349](#)
  - postcat [356](#)
  - postconf [353](#)
  - Postfix
    - format for source files [349](#)
    - migration from Sendmail [361](#)
    - TLS/SSL protection [347](#)
  - postfix [352](#)
  - Postfix server
    - displaying and modifying configuration
      - parameters [353](#)
    - lookup tables [348](#)
    - operation [348](#)
    - post-installation script [336](#)
    - starting and stopping [352](#)
      - starting up [343](#)
      - uninstalling [334, 339](#)
  - Postfix server, see also SMTP server
  - post-installation script (OpenSSH) [313](#)
  - post-installation script (Postfix server) [336](#)
  - postmap [357](#)
    - source file [349](#)
    - source file, see also alias file
  - postqueue (mailq) [354](#)
  - postsuper [355](#)
  - primary time server
    - NTP [269](#)
  - PRNGD (BS2000/OSD)
    - bcamInterval [212](#)
    - cmd [211](#)
    - cmdInterval [212](#)
    - configuration [207](#)
    - entropy source [206](#)
    - entropyThreshold [209](#)
    - file [210](#)
    - fileInterval [210](#)
    - messages [217](#)
    - minimalEntropy [208](#)
    - poolSize [208](#)
    - program interface GPRBYTE [213](#)
    - seedFile [209](#)
  - product-version (data type) [28](#)
  - program
    - NTP [273](#)
  - program interface GPRBYTE [213](#)
  - proxy mechanism [128](#)
  - PTR (Domain Name Pointer) [227](#)
  - public key authentication [322](#)
  - PuTTY [308](#)
- ## Q
- query
    - NTP status [296, 297](#)
    - recursive (DNS) [231](#)
  - query, iterative
    - DNS [232](#)

**R**

RAW command  
  NTP 299  
RDPROT 125, 197  
recursive query  
  DNS 231  
regulations, licensing 15  
requestkey statement (NTP) 289  
resolver  
  DNS 223, 229  
resource records (DNS) 227  
restart  
  NTP 277  
restart call  
  DNS resolver (restartedns) 246  
  NAMED 259  
  TCP-IP-SV 35  
RESTRICT statement  
  NTP 281  
result  
  FTP exit routine 143  
retryLimit 372  
return code  
  FTP exit routine 143  
RFCs 221  
rhosts authentication 320, 321, 322  
rndc-confgen (DNS) 264  
rnds (DNS) 264  
root domains 225  
RSA-based authentication 321

**S**

-S 77, 162  
SEARCH entry  
  DNS resolver 242  
secondary time server  
  NTP 269  
second-level domains 226  
secure  
  control connection 103  
  data connection 104  
Secure Shell, see OpenSSH  
security (NAMED) 257

security check  
  FTP exit 130  
seedFile 209  
senderSuffix 365  
Sendmail  
  functionality 332  
  migration to Postfix 361  
server  
  forwarder (DNS) 233  
  forwarding 233  
  master 232  
  primary, see master server 232  
  secondary, see slave server 232  
  slave 232  
server (DNS) 223  
server authentication 323  
SERVER statement  
  NTP 279  
-serverInfoFile 73  
set time  
  NTPD 293  
SET-FTP-TELNET-PARAMETERS  
  TELNET 148  
SHOW-MAIL-SERVICE-PARAMETER 389  
shutdown 123, 196  
  DNS daemon (stopdns) 36  
  FTP server 110, 112, 190  
  interNet Services 36  
  NTP 277  
  NTP daemon 36  
  TELNET server 112, 189, 190  
shutdown call  
  TCP-IP-SV 36  
shutting down  
  named 260  
  NTP 277  
Simple Mail Transfer Protocol (SMTP) 331  
SINLIB.MAIL.nnn.POSTFIX  
  PLAM library 334, 340  
-sizeCmdTimeLimit 107, 108  
slave server 232  
SMTP  
  Simple Mail Transfer Protocol 331  
  TLS/SSL protection 347

- SMTP server
    - installing [334](#)
    - port number [334](#)
    - post-installation script [336](#)
    - uninstalling [334](#), [339](#)
  - SMTP server, see also Postfix server
  - SNMP subagent for FTP [129](#)
  - SOA (Start Of Authority) [227](#)
  - socket trace level
    - specifying [163](#)
  - socketTraceLevel [77](#)
  - software requirements [19](#)
  - source file
    - for postalias [349](#)
    - for postmap [349](#)
    - postalias format [349](#)
    - Postfix format [349](#)
  - ssh
    - connection setup with sshd [317](#), [319](#)
  - SSH, see OpenSSH
  - sshd
    - connection setup with ssh [317](#), [319](#)
    - files [324](#)
    - login process [323](#)
    - starting [317](#)
    - stopping [317](#)
  - sshd (OpenSSH server daemon) [316](#)
  - sshd, see also OpenSSH server daemon [316](#)
  - standard port number
    - IMAP server [334](#)
    - POP3 server [334](#)
    - SMTP server [334](#)
  - start
    - DNS resolver [246](#)
    - FTP server [109](#), [189](#)
    - NAMED [259](#)
    - TELNET server [109](#), [189](#)
  - start call
    - TCP-IP-SV [35](#)
  - starting
    - NTP [277](#)
    - Postfix server [352](#)
  - START-MAIL-SERVICE [384](#)
  - START-TLS option [165](#)
  - startup
    - IMAP server [344](#)
    - NTP [277](#)
    - POP3 server [344](#)
    - Postfix server [343](#)
    - SMTP server [343](#)
  - startup call
    - DNS [246](#)
    - NAMED [259](#)
  - startup options
    - NTP [291](#)
  - statements
    - DNS configuration file [239](#)
  - stop
    - DNS resolver [247](#)
    - NAMED [260](#)
  - stop calls
    - interNet Services [247](#), [260](#)
  - STOP-MAIL-SERVICE [391](#)
  - stopping
    - Postfix server [352](#)
  - storing information (DNS) [227](#)
  - stratum
    - NTP [269](#)
  - structured-name (data type) [28](#)
  - subagent for FTP [129](#)
  - subevent, see FTP subevent
  - subsystem
    - terminate POSIX [36](#), [247](#), [260](#)
  - suffixes for data types [30](#)
  - syntax, DNS configuration file [247](#)
  - SYSSSI [363](#)
  - system clock [269](#)
  - systemExit [79](#)
- ## T
- T [77](#), [163](#)
  - TCP forwarding (OpenSSH) [311](#)
  - TCP-IP-SV [19](#)
    - installing [33](#), [219](#), [235](#), [249](#), [274](#), [312](#)
    - restart calls [35](#)
    - shutdown call [36](#)
    - software requirements [19](#)
    - start calls [35](#)

- TELNET
  - configuration and operation 145
  - configuring via option file 158
  - configuring via SET-FTP-TELNET-PARAMETERS 148
  - exit routine 199
  - installing/uninstalling in POSIX 47
  - IPv6 addresses 197
  - TLS/SSL support 146
- Telnet authentication option 165
- Telnet data encryption option 165
- TELNET exit 198
  - DUMMY module 198
  - EXITTEL.C 198
  - user-defined 202
  - YAPTEXT 198
- TELNET server
  - displaying current settings 119, 192
  - logging file 119, 192
  - shutdown 112, 189, 190
  - start 109, 189
  - TLS/SSL support 146
- TELNET server options
  - A 160
  - AUTHENTICATION option 165
  - B 187
  - D 160
  - E 161
  - ENCRYPTION option 165
  - H 188
  - N 161
  - P 162
  - S 162
  - START-TLS option 165
  - T 163
  - Telnet authentication option 165
  - Telnet encryption option 165
  - tlsCertificateChainFile 180
  - V 163
  - X 164
  - Z AcceptableClientCAFile 182
  - Z CACertificateFile 172
  - Z CARevocationFile 173
  - Z CipherSuite 174
  - Z DSACertificateFile 170
  - Z DSAKeyFile 171
  - Z OpenSSLLibname 185
  - Z Protocol 181
  - Z RandFile 179
  - Z RSACertificateFile 168
  - Z RSAKeyFile 169
  - Z tls-required 167
  - Z UseCryptoHardware 186
  - Z VerifyDepth 183
- tempFilePrefix 373
- terminal name
  - defining 162
- terminate
  - DNS resolver 247
  - NAMED 260
  - POSIX subsystem 36, 247, 260
- text (data type) 28
- text of a message 333
- time
  - set with NTP 293
- time (data type) 28
- time server
  - hierarchy 273
  - NTP 269
  - primary (NTP) 269
  - secondary (NTP) 269
- time synchronization
  - hierarchy 273
  - NTP 269
- timeout 76
- TIMEOUT command
  - NTP 298
- TLS/SSL options
  - START-TLS option 165
  - tlsAcceptableClientCAFile 99
  - tlsCAcertificateFile 98
  - tlsCArevocationFile 100
  - tlsCertificateChainFile 97
  - tlsCipherSuite 87
  - tlsDSACertificateFile 95
  - tlsOpenSSLlibName 107
  - tlsProtocol 86
  - tlsRandFile 105

- tlsRSACertificateFile 93
  - tlsRSAkeyFile 94, 96
  - tlsSecureControlConnection 103
  - tlsSecureDataConnection 104
  - tlsUseCryptoHardware 106
  - tlsVerifyClient 101, 184
  - tlsVerifyDepth 102
  - Z CACertificateFile 172
  - Z CARevocationFile 173
  - Z CertificateChainFile 180
  - Z CipherSuite 174
  - Z DSACertificateFile 170
  - Z DSAKeyFile 171
  - Z OpennSSLibname 185
  - Z Protocol 181
  - Z RandFile 179
  - Z RSACertificateFile 168
  - Z RSAKeyFile 169
  - Z tls-required 167
  - Z UseCryptoHardware 186
  - Z VerifyDepth 183
- TLS/SSL protection
- IMAP/POP3 345
  - Postfix 347
- TLS/SSL support
- on the FTP server 54
  - on the TELNET server 146
- tlsAcceptableClientCAFile 99
  - tlsCACertificateFile 381
  - tlsCAcertificateFile 98
  - tlsCARevocationFile 382
  - tlsCArevocationFile 100
  - tlsCertificateChainFile 97
  - tlsCertificateFile 380
  - tlsCipherSuite 87
  - tlsCipherSuite 375
  - tlsDSACertificateFile 95
  - tlsKeyFile 380
  - tlsOpenSSLibname 107
  - tlsProtocol 86
  - tlsProtocol 374
  - tlsRandFile 105
  - tlsRSACertificateFile 93
  - tlsRSAkeyFile 94, 96
  - tlsSecureConnection 373
  - tlsSecureControlConnection 103
  - tlsSecureDataConnection 104
  - tlsUseCryptoHardware 106
  - tlsUseCryptoHardware 383
  - tlsVerifyClient 101, 184
  - tlsVerifyDepth 102
  - tlsVerifyDepth 383
  - tlsVerifyServer 382
  - tools (DNS) 264
  - top-level domains 225
  - trace 123, 196
  - trace functionality
    - NTP 306
  - Transaction SIGNatures 233
  - Transaction SIGNatures (TSIG) 257
  - Transaction SIGNatures, DNS 257
  - trustedkey statement (NTP) 289
  - TSIG 257
  - TSIG (Transaction SIGNatures) 233
  - TSN 122, 123, 195, 196
- ## U
- U 78
  - uninstalling
    - DNS development modules 237
    - DNS resolver 237
    - FTP/TELNET in POSIX 47
    - IMAP server 342
    - interNet Services 35, 237, 251, 276
    - mail server 334
    - NAMED 251
    - NAMED development modules 252
    - OpenSSH 315
    - POP3 server 342
    - Postfix server 334, 339
    - SMTP server 334
  - Universal Coordinated Time (UTC) 269
  - user-defined
    - FTP exit routine 140, 143
    - TELNET exit 202
  - useSenderSuffix 366

**V**

-V [78](#), [163](#)

Verbose

enabling/disabling [163](#)

-verbose [78](#)

vsn (data type) [28](#)

**X**

-X [79](#), [164](#)

x-string (data type) [29](#)

x-text (data type) [29](#)

**Y**

YAPTEXTIT [198](#)

**Z**

-Z [80](#)

-Z AcceptableClientCAFile [182](#)

-Z CACertificateFile [172](#)

-Z CARevocationFile [173](#)

-Z CertificateChainFile [180](#)

-Z CipherSuite [174](#)

-Z DSACertificateFile [170](#)

-Z OpenSSLLibname [185](#)

-Z Protocol [181](#)

-Z RandFile [179](#)

-Z RSACertificateFile [168](#)

-Z RSAKeyFile [169](#), [171](#)

-Z tls-required [167](#)

-Z UseCryptoHardware [186](#)

-Z VerifyDepth [183](#)

zone (DNS) [226](#)

zone data files [254](#)

