FUJITSU

interNet Services V3.4A

interNet Services V3.4A
Benutzerhandbuch

Ausgabe Dezember 2010

Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

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

Copyright und Handelsmarken

Inhalt

1 Einleitung

Das Produkt interNet Services ergänzt die TCP/IP-Funktionalität von openNet Server um folgende Standards:

- DNS Resolver und Server
- NTP Client und Server
- FTP Client und Server
- TELNET Client und Server
- OpenSSH
- Mail-Sender in BS2000/OSD
- Mail-Reader in BS2000/OSD
- Mail-Server in POSIX

1.1 Zielgruppe des Handbuchs

Das vorliegende Benutzerhandbuch wendet sich an BS2000/OSD-Nutzer und Administratoren, die TCP/IP-spezifische Anwendungen und/oder Dienste nutzen wollen. Kenntnisse des Betriebssystems BS2000/OSD sowie der TCP/IP-Grundbegriffe werden vorausgesetzt.
Neben dem hier vorliegenden Benutzerhandbuch existiert zu interNet Services zusätzlich ein Administratorhandbuch, das die Informationen für System- und Netzadministratoren enthält.

1.2 Wegweiser durch das Handbuch

Das vorliegende Handbuch ist folgendermaßen strukturiert:

- Kapitel 2: Überblick
  Dieses Kapitel stellt Grundlagen und Funktionalität der einzelnen Komponenten von interNet Services vor.
- Kapitel 3: SSL
  Dieses Kapitel beschreibt die Aspekte der Kommunikationssicherheit im Internet, die Grundlagen der Kryptographie und gibt einen Überblick über SSL. Außerdem erläutert das Kapitel, was beim Beantragen und Erstellen von Zertifikaten zu beachten ist.
- Kapitel 4: FTP
  In diesem Kapitel wird nutzerorientiert die Client- und Server-Funktionalität beschrieben, die interNet Services in BS2000/OSD zur Verfügung stellt. Dieses Kapitel enthält auch die detaillierte Beschreibung der Client-Kommandos (ergänzt durch Beispiele) sowie der Parametrisierung des FTP-Clients mithilfe einer Option-Datei.
- Kapitel 5: FTAC-Schnittstelle
  Dieses beschreibt die Nutzung der FTAC-Schnittstelle für FTP und beschreibt die FTAC-Kommandoschnittstelle.
- Kapitel 6: TELNET
  Die nutzerorientierte Beschreibung der Client- und Server-Funktionalität von TELNET ist Thema dieses Kapitels. Außerdem enthält dieses Kapitel die detaillierte Beschreibung der Client-Kommandos (ergänzt durch Beispiele) sowie der Parametrisierung des TELNET-Clients mithilfe einer Option-Datei.
- Kapitel 7: OpenSSH
  Dieses Kapitel beschreibt die Struktur und die Client-Anwendungen von OpenSSH. Außerdem werden die Basis-Utilities von OpenSSH erläutert.
- Kapitel 8: Mail-Reader
  Dieses Kapitel beschreibt die Funktionalität des Mail-Readers.
- Kapitel 9: Mail-Sender
  Dieses Kapitel beschreibt die Steuerung des Mail-Senders über SDF-Kommandos und über die Assembler-Makroschnittstelle bzw. über die C-Schnittstelle (Include-Dateien). Außerdem enthält dieses Kapitel eine detaillierte Beschreibung der Konfiguration des Mail-Senders über das Mail-Sender Frontend (Benutzer-Option-Datei).

1.3 Lizenzrechtliche Bestimmungen

Im Folgenden sind die lizenzrechtlichen Bestimmungen zum OpenSSL-Paket und zum TLS-FTP-Patch von Peter 'Luna' Runestig abgedruckt.

Die deutsche Fassung des Lizenztextes dient dem Leser nur als Hilfestellung zum leichteren Verständnis. Die deutsche Übersetzung ist nicht rechtsverbindlich. In Zweifelsfällen ist ausschließlich der englische Orginaltext maßgebend.

Deutsche Fassung des Lizenztextes (Übersetzung)

OpenSSL-Lizenz
==========

Copyright (c) 1998-2000 The OpenSSL Project. Alle Rechte vorbehalten.
Der Weitervertrieb und die Verwendung in Quell- und binären Formularen ist - mit oder ohne Veränderungen - grundsätzlich zulässig, sofern die folgenden Bedingungen eingehalten werden:
1. Weitervertriebene Quellcodes müssen den oben aufgeführten Copyright-Hinweis, die hier genannten Bedingungen und die nachstehende Ausschlussklausel enthalten.
2. Im Fall des Weitervertriebs in binärer Form müssen der oben genannte Copyright-Hinweis, die hier aufgeführten Bedingungen und die nachstehende Ausschlussklausel und/oder andere in der Bereitstellung enthaltene Materialien genannt werden.
3. Alle Werbematerialien, in denen Funktionen der Software erwähnt oder verwendet werden, müssen den folgenden Hinweis enthalten:
"Dieses Produkt enthält Software, die vom OpenSSL Project für die Verwendung mit dem OpenSSL Toolkit entwickelt wurde (http://www.openssl.org/)."
4. Die Bezeichnungen "OpenSSL Toolkit" und "OpenSSL Project" dürfen ohne vorherige schriftliche Genehmigung nicht zur Produktkennzeichnung oder zu sonstigen Werbezwecken verwendet werden. Schriftliche Genehmigungen erhalten Sie unter: openssl-core@openssl.org.
5. Auch für von dieser Software abgeleitete Produkte darf der Name "OpenSSL" weder als Produktbezeichnung noch als Bestandteil der Produktbezeichnung ohne vorherige schriftliche Genehmigung des OpenSSL Projects verwendet werden.
6. Der Weitervertrieb darf nur unter folgendem Hinweis erfolgen:
"Diese Produkt enthält Software, die vom OpenSSL Project für die Verwendung mit dem OpenSSL Toolkit entwickelt wurde (http://www.openssl.org/)."

DAS OPENSSL PROJECT STELLT DIESE SOFTWARE "OHNE MÄNGELGEWÄHR" BEREIT. DIESER GEWÄHRLEISTUNGSAUSSCHLUSS BEZIEHT SICH AUF VERTRAGLICHE ODER GESETZLICHE GARANTIEN, EINSCHLIESSLICH VON, ABER NICHT BESCHRÄNKT AUF, GESETZLICHE GARANTIEN BEZÜGLICH HANDELSÜBLICHER QUALITÄT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. IN KEINEM FALLE KÖNNEN OPENSSL PROJECT ODER SEINE MITARBEITER FÜR JEGLICHE DIREKTEN, INDIREKTEN, ZUFÄLLIGEN, SPEZIELLEN, EXEMPLARISCHEN SCHÄDEN

ODER FOLGESCHÄDEN (EINSCHLIESSLICH VON, JEDOCH NICHT BESCHRÄNKT AUF, BESCHAFFUNG VON ERSATZGÜTERN ODER -DIENSTLEISTUNGEN, NUTZUNGSAUSFÄLLEN, DATENVERLUSTEN ODER ENTGANGENEN GEWINNEN ODER BETRIEBSUNTERBRECHUNGEN) GLEICH WELCHEN URSPRUNGS HAFTBAR GEMACHT WERDEN. JEGLICHE HAFTUNGSANSPRÜCHE AUF VERTRAGSBASIS, IM HINBLICK AUF DELIKTSHAFTUNG ODER GEFÄHRDUNGSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT UND SONSTIGES), DIE AUS DER VERWENDUNG DIESER SOFTWARE ENTSTEHEN, WERDEN AUCH DANN AUSGESCHLOSSEN, WENN AUF DIE MÖGLICHKEIT DIESER SCHÄDEN HINGEWIESEN WURDE.

Das Produkt enthält kryptographische Software, die von Eric Young (eay@cryptsoft.com) entwickelt wurde. Das Produkt enthält Software, die von Tim Hudson (tjh@cryptsoft.com) entwickelt wurde.

SSLeay-Original-Lizenz
==========
Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com). Alle Rechte vorbehalten.
Bei dem vorliegenden Paket handelt es sich um eine SSL-Implementierung, die von Eric Young (eay@cryptsoft.com) entwickelt wurde. Die Implementierung wurde so erstellt, dass sie mit dem SSL von Netscape kompatibel ist.
Die Bibliothek steht zur kostenlosen gewerblichen und nichtgewerblichen Nutzung zur Verfügung, sofern die nachstehenden Bedingungen erfüllt werden.
Die nachstehenden Bedingungen gelten außer für den SSL-Code für alle in der Bereitstellung enthaltenen Codes, beispielsweise RC4, RSA, lhash, DES usw.
Für die in der Bereitstellung enthaltene SSL-Dokumentation gelten dieselben Copyrights, wobei als Eigentümer in diesem Fall Tim Hudson (tjh@cryptsoft.com) zu nennen ist.
Das Copyright verbleibt bei Eric Young, weshalb die Copyright-Hinweise innerhalb des Codes nicht entfernt werden dürfen.
Wenn das Paket innerhalb eines Produkts verwendet wird, ist Eric Young als Urheber der verwendeten Teile der Bibliothek zu erwähnen.
Dies kann in Form einer Textmeldung beim Programmstart oder in der dem Produktpaket beiligenden Dokumentation (online oder in Druckform) erfolgen.
Der Weitervertrieb und die Verwendung in Quell- und binären Formularen ist - mit oder ohne Veränderungen - grundsätzlich zulässig, sofern die folgenden Bedingungen eingehalten werden:
1. Weitervertriebene Quellcodes müssen den Copyright-Hinweis, die hier genannten Bedingungen und die nachstehende Ausschlussklausel enthalten.
2. Im Fall des Weitervertriebs in binärer Form müssen der oben genannte Copyright-Hinweis, die hier aufgeführten Bedingungen und die nachstehende Ausschlussklausel und/oder andere in der Bereitstellung enthaltene Materialien genannt werden.
3. Alle Werbematerialien, in denen Funktionen der Software erwähnt oder verwendet werden, müssen den folgenden Hinweis enthalten:
"Das Produkt enthält kryptographische Software, die von Eric Young (eay@cryptsoft.com) entwickelt wurde."

Das Wort "kryptographisch" muss nicht erwähnt werden, wenn die verwendeten Routinen aus der Bibliothek nicht mit kryptographischem Bezug verwendet werden.
4. Wenn Sie Windows-spezifische Codes (oder Ableitungen davon) aus dem Apps-Verzeichnis (Anwendungscode) verwenden, ist der folgende Hinweis erforderlich:
"Das Produkt enthält Software, die von Tim Hudson (tjh@cryptsoft.com) entwickelt wurde."
DIESE SOFTWARE WIRD VON ERIC YOUNG "OHNE MÄNGELGEWÄHR" BEREITGESTELLT. DIESER GEWÄHRLEISTUNGSAUSSCHLUSS BEZIEHT SICH AUF VERTRAGLICHE ODER GESETZLICHE GARANTIEN, EINSCHLIESSLICH VON, ABER NICHT BESCHRÄNKT AUF, GESETZLICHE GARANTIEN BEZÜGLICH HANDELSÜBLICHER QUALITÄT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. IN KEINEM FALLE KÖNNEN DER AUTOR ODER MITARBEITER FÜR JEGLICHE DIREKTEN, INDIREKTEN, ZUFÄLLIGEN, SPEZIELLEN, EXEMPLARISCHEN SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH VON, JEDOCH NICHT BESCHRÄNKT AUF, BESCHAFFUNG VON ERSATZGÜTERN ODER -DIENSTLEISTUNGEN, NUTZUNGSAUSFÄLLEN, DATENVERLUSTEN ODER ENTGANGENEN GEWINNEN ODER BETRIEBSUNTERBRECHUNGEN) GLEICH WELCHEN URSPRUNGS HAFTBAR GEMACHT WERDEN. JEGLICHE HAFTUNGSANSPRÜCHE AUF VERTRAGSBASIS, IM HINBLICK AUF DELIKTSHAFTUNG ODER GEFÄHRDUNGSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT UND SONSTIGES), DIE AUS DER VERWENDUNG DIESER SOFTWARE ENTSTEHEN, WERDEN AUCH DANN AUSGESCHLOSSEN, WENN AUF DIE MÖGLICHKEIT DIESER SCHÄDEN HINGEWIESEN WURDE.

Die Lizenz und die Bedingungen für den Weitervertrieb von allen öffentlich erhältlichen Versionen oder Ableitungen dieses Codes können nicht verändert werden, d.h., der Code kann nicht einfach kopiert und in eine andere Weitervertriebslizenz integriert werden [einschließlich der GNU Public Licence.]

Englischer Lizenztext (Originaltext)

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

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

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

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

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

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

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

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

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

1.4 Änderungen gegenüber der Vorgängerversion

Das folgende Änderungsprotokoll gibt einen Überblick über die Neuerungen in interNet Services V3.4A, die das vorliegende Handbuch betreffen.

Über Neuerungen, die das Handbuch „interNet Services Administratorhandbuch“ betreffen, informiert Sie das zugehörige Änderungsprotokoll.

Readme-Dateien

Folgende Readme-Dateien wurden in das Handbuch integriert:

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

FTAC-Schnittstelle

Das Kapitel 5 mit der Beschreibung der FTAC-Schnittstelle für FTP wurde überarbeitet und bezieht sich auf openFT V11.0.

Das Kapitel „Meldungen“ mit den FTAC-Meldungen ist entfallen.

1.5 Typographische Gestaltungsmittel

In diesem Handbuch werden folgende Mittel zur Darstellung von funktional wichtigen Textteilen verwendet:

kursive Schrift
: für Dateinamen, Programmnamen, Namen von Auftragsfenstern, Parameterbezeichnungen, Menütitel und Menüeinträge sowie Kommandos und Variablen im Fließtext.

<spitze Klammern>
: kennzeichnen Variable, wenn Sie dafür Werte einsetzen müssen.

[eckige Klammern]
: kennzeichnen optionale Angaben.

{geschweifte Klammern} ...
: kennzeichnen eine Liste von Alternativen, die durch „|“ voneinander getrennt sind.

`dicktengleiche Schrift`
: kennzeichnet Eingaben für das System, Systemausgaben und Dateinamen in Beispielen.

kommando
: In der Syntaxbeschreibung für Kommandos werden diejenigen Bestandteile (Bezeichnungen von Kommandos und Parametern) fett dargestellt, die unverändert eingegeben werden müssen.

► kennzeichnet Arbeitsschritte, die der Anwender durchführen muss.


für Hinweistexte


ACHTUNG!

für Warnhinweise

Verweise

Verweise innerhalb des Handbuchs geben die betreffende Seite im Handbuch und je nach Bedarf auch den Abschnitt bzw. das Kapitel an. Verweise auf Themen, die in einem anderen Handbuch beschrieben sind, enthalten den Kurztitel des Handbuchs. Die vollständigen Titel finden Sie im Literaturverzeichnis.

1.6 Readme-Dateien

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. den produktspezifischen Readme-Dateien. Sie finden die Readme-Dateien auf Ihrem BS2000/OSD-Rechner unter den folgenden Dateinamen:

SYSRME.TCP-IP-SV.*nnn*.D
SYSRME.TCP-IP-AP.*nnn*.D
SYSRME.MAIL.*nnn*.D

nnn steht für die Versionsangabe, z.B. 034 für die Version 3.4.

Die Benutzerkennung, unter der sich die Readme-Dateien befinden, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Dateien können Sie mit dem Kommando /SHOW-FILE oder mit einem Editor ansehen und auf einem Standarddrucker mit folgendem Kommando ausdrucken:

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

2 Internet Services in BS2000/OSD (Überblick)

BS2000/OSD stellt mit dem openNet Server eine Reihe von Internet Services bereit, die die Transportprotokolle TCP (Transmission Control Protocol) und UDP (User Datagram Protocol) nutzen. Die Liefereinheit openNet Server ist Voraussetzung für den Einsatz der interNet Services. Von der Open Group wurde im Product Standard X98PS für Internet Server die Menge der relevanten Internet Services festgelegt. Diese Services sind für BS2000/OSD in der Liefereinheit interNet Services integriert.

Die folgende Tabelle gibt einen Überblick über die von den einzelnen Produkten angebotenen Services.

Produkt	**Service-Gruppe**	**Service**	
openNet Server	TCP/IP Communications Services	TCP UDP IPv4 IPv6 -	Transmission Control Protocol User Datagram Protocol Internet Protocol Version 4 Internet Protocol Version 6 Requirements for Internet Hosts
interNet Services APACHE	File Transfer Service	FTP	File Transfer Protocol
	Terminal Service	TELNET	
	Name Service	DNS DDNS	Domain Name Service Dynamic DNS
	Time Service	NTP	Network Time Protocol
	Security Service OpenSSL	SSL TLS	Secure Socket Layer Transport Layer Security
	Security Service OpenSSH	SSH	Secure Shell
	Mail Services	SMTP POP IMAP	Simple Mail Transfer Protocol Post Office Protocol Internet Message Access Protocol
	Hypertext Services	HTTP	HyperText Transfer Protocol
SNMP-Basic-Agent BS2000	Network Management	SNMP	Simple Network Management Protocol

Die unterstützten Protokolle arbeiten in der Regel nach dem Client-/Server-Modell. Der Server stellt Dienste bereit, die von einem oder mehreren Clients angefordert und genutzt werden. Server und Client können auf denselben oder auf unterschiedlichen Systemen ablaufen.

In den folgenden Abschnitten werden die in interNet Services enthaltenen Services näher beschrieben.

2.1 FTP

Der Austausch von Daten stellt eine zentrale Anforderung bei der Vernetzung mehrerer Rechner dar. Die Vielfalt der auf dem Markt befindlichen Rechnertypen macht den Einsatz eines herstellerunabhängigen Standards notwendig. FTP (File Transfer Protocol) bietet die Möglichkeit, Daten unabhängig von Bauweise und Betriebssystem der Rechner auszutauschen. FTP setzt direkt auf TCP auf und kann Dateien aller Art (z.B. Text-, Bild-, Ton-, Video- oder Programmdateien) übertragen.

Der Benutzer kommuniziert über seine Benutzeroberfläche mit dem FTP-Client. Dieser baut eine Verbindung zum FTP-Server über dessen Port 21 auf (Kontrollverbindung). Über diese Verbindung sendet der Client Kommandos an den Server, der seinerseits Quittungsmeldungen für die Kommandos zurücksendet. Zum Austausch von Nutzdaten baut der FTP-Server eine zweite Verbindung, ausgehend von Port 20, zum FTP-Client auf (Datenverbindung).

BS2000/OSD stellt sowohl die Server- als auch die Client-Funktionalität des FTP zur Verfügung. Zusätzlich zum Standardprotokoll werden folgende Funktionen angeboten:

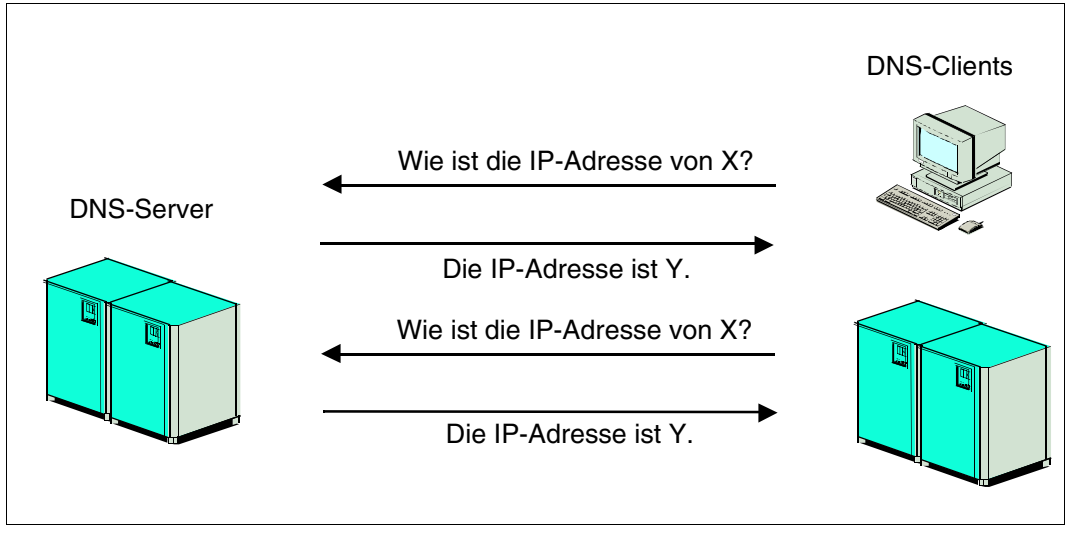
- die Unterstützung von BS2000-Dateiformaten (SAM, PAM)
- die Möglichkeit, Codetabellen für die Umsetzung EBCDIC nach ASCII und umgekehrt einzustellen
- Sicherheitsfunktionen durch den Anschluss an das optionale Sicherheitsprodukt openFT-AC mit den Möglichkeiten der Entkopplung von FTP-Zugangsberechtigung und Login-Berechtigung, Festlegung von benutzerspezifischen Zugriffsrechten und einer Protokollierung von Zugangsprüfungen.
- Zusätzliche Sicherheitsfunktionen durch Nutzung von TLS/SSL-Verschlüsselung für die Kontroll- und/oder Datenverbindung, um Vertraulichkeit, Authentizität und Integrität der zwischen Server und Client übertragenen Daten zu gewährleisten.

2.2 TELNET

TELNET ermöglicht eine Terminalsitzung auf einen am Netz angeschlossenen Rechner. Die von TELNET gebotene Funktionalität ähnelt der des von Unix-Systemen bekannten Kommandos rlogin. Das TELNET-Protokoll enthält jedoch speziell auf Großrechnersysteme zugeschnittene Protokollelemente. BS2000/OSD unterstützt sowohl die Client- als auch die Server-Funktionalität von TELNET.

2.3 DNS

DNS (Domain Name Service) ist ein globales Netzwerk von DNS-Servern, die die Abbildung von Namen auf IP-Adressen durchführen. Ohne DNS wäre weder der Betrieb des Internets noch der Betrieb von Intranets möglich.



DNS-Namen bilden eine Baumstruktur. Sie sind hierarchisch aufgebaut und in verschiedene Domänenebenen gegliedert. Es gibt eine Wurzel, die Rootdomain, die der Anker für alle Suchvorgänge innerhalb des DNS-Namensraumes ist. Neben Namen sind im DNS auch Adressen und andere Informationen gespeichert.

Der Domain Name Service ist eine verteilte, replizierte Datenbank mit DNS-Servern und DNS-Clients (Resolvern). Dabei werden die Daten von verschiedenen DNS-Servern verwaltet, die jeweils für eine oder mehrere DNS-Domänen verantwortlich sind. Dabei sind zur Erhöhung der Ausfallsicherheit redundante DNS-Server möglich. Die Resolver hingegen haben keine lokale Datenbasis. Die Clients wenden sich bei jeder DNS-Anfrage an einen oder mehrere DNS-Server, um die benötigte Information zu erhalten. Diese DNS-Anfragen können optional signiert erfolgen.

Kryptographische Signaturen sichern bei Abfragen nur die Kommunikation zwischen Servern. Für die Kommunikation zwischen Client und Server werden kryptographische Signaturen nicht unterstützt.

BS2000/OSD stellt seinen Nutzern sowohl die Server- als auch die Resolver-Funktionalität des DNS zur Verfügung.

Sowohl Server- als auch Resolver-Funktionalität sind eine Portierung des BIND-Codings, das als Standard-Implementierung des DNS gilt. Damit haben Anwender des BS2000/OSD Zugriff auf die Funktionen und Services des DNS. Zudem ist durch die hohe Verfügbarkeit des BS2000/OSD eine hohe Verfügbarkeit der darauf ablaufenden DNS-Server gewährleistet.

2.4 NTP

Das Network Time Protocol (NTP) erlaubt zum einen die Verteilung einer Referenz-Uhrzeit (Universal Coordinated Time, UTC) innerhalb eines Netzwerks und sorgt zum anderen für die Koordination der Uhren innerhalb beliebig großer Netzwerke.

Timeserver stehen in hierarchischer Beziehung zueinander, ein sekundärer Timeserver erhält seine Zeit über das Netzwerk von einem primären Timeserver. Zur Einspeisung der UTC-Zeit ins Netzwerk empfiehlt sich die Ausrüstung eines Servers mit einer funkgesteuerten Hardware-Uhr, die ein von einer Atomuhr generiertes Zeitsignal empfängt. Ein BS2000-Server kann sowohl die Funktion eines präzisen Timeservers als auch Clients ausführen.

2.5 OpenSSH

OpenSSH ist die freie, d.h. nicht lizenzkostenpflichtige Version von SSH (**S**ecure **Sh**ell).

SSH ist ein kryptographisches Protokoll für die Durchführung der folgenden Aufgaben:

- Login auf einen fernen Rechner
- interaktive/nicht interaktive Kommandoausführung auf einem fernen Rechner
- File Transfer zwischen verschiedenen Rechnern eines Netzes
- Tunneln von ungesicherten TCP-Protokollen über eine verschlüsselte SSH-Verbindung

SSH bezeichnet nicht nur das Protokoll selbst, sondern auch die konkreten Implementierungen.

OpenSSH ist die sichere Alternative zu den r-Utilities *rlogin*, *rcp*, *rsh* und den Programmen *telnet* und *ftp*, sofern diese Programme nicht durch TLS/SSL abgesichert sind. Im Gegensatz zu den genannten Programmen verschlüsselt OpenSSH den gesamten Netzverkehr (inklusive Passwörtern) und verhindert so Eavesdropping, Connection Hijacking und andere Attacken auf Netzebene. Darüber hinaus unterstützt OpenSSH eine Vielzahl von Tunnelling-Varianten sowie eine große Bandbreite an Authentifizierungsmechanismen.

2.6 Mail-Server

Beim Versenden und Empfangen von elektronischer Post (Mail) über das Internet erfüllen Mail-Server, auch Mail Transfer Agenten (MTA) genannt, die Funktionen von Post-Ämtern. Mail-Server erledigen den Transfer der Mails über das Netz und besorgen deren Zustellung in Postfächer (Mailboxen).

Der elektronische Postdienst im Internet basiert auf dem Simple Mail Transfer Protocol (SMTP), das in RFC 821 bzw. RFC 2821 definiert ist. Mail-Server, die den elektronischen Postdienst auf der Basis des SMTP-Protokolls abwickeln, werden auch als SMTP-Server bezeichnet. Während ursprünglich nur reine Textnachrichten übermittelt werden konnten, kann heute über den MIME-Mechanismus (Multipurpose Internet Mail Extensions, RFC 2045 bis 2049) ein breites Spektrum von Formaten, z.B. Bilder, übermittelt werden.

Der SMTP-Server erhält Nachrichten entweder von einem anderen SMTP-Server oder von einem Mail User Agenten (MUA), dem Mail-Sender. Dabei kann der SMTP-Server als Mail-Relay oder als Mail-Endsystem fungieren. Für die Wahl einer geeigneten Route zu einem Endsystem nutzt der SMTP-Server den Domain Name Service (DNS).

Die interNet Services in BS2000/OSD verwenden als SMTP-Server das in das BS2000/OSD portierte Produkt Postfix in der Version 2.7.1 (zum Zeitpunkt der Drucklegung des Handbuchs), das IPv6- und TLS-Funktionalität unterstützt. Dieser von Wietse Venema erstellte Open Source SMTP-Server zeichnet sich insbesondere durch hohe Performance, einfache Administrierbarkeit und ein hohes Maß an Sicherheit aus.

2.7 Mail-Sender und Mail-Reader

Mail-Sender und Mail-Reader sind Mail User Agenten (MUA), mit denen Sie im BS2000/OSD Mails versenden bzw. empfangen können:

- Mit dem Mail-Sender in interNet Services können Sie Mails an den lokalen Mail-Server in POSIX oder an ferne Mail-Server senden. Dabei stellt Ihnen der Mail-Sender gängige Funktionen zum Versenden von Mails, wie z.B. die Angabe von to-, cc- und bcc-Empfängern oder das Anfügen von Dateien, zur Verfügung. Darüber hinaus können Sie aus BS2000/OSD-Prozeduren heraus z. B. automatisiert Listen als Mail versenden.

  Die Mails können Sie gemäß S/MIME signieren und/oder verschlüsseln und über eine TLS/SSL-gesicherte SMTP-Verbindung zum Mail-Server senden.

- Für die automatisierte Verarbeitung von empfangenen Mails gibt es in interNet Services den Mail-Reader. Mit dem Mail-Reader können Sie im BS2000/OSD über die Zugriffs-Services POP3 und IMAP auf Mailboxen zugreifen, die im POSIX oder auf einem fernen Server liegen, und von dort Mails abholen und weiterverarbeiten.

  Der Mail-Reader im BS2000/OSD bietet Ihnen zwei Möglichkeiten zur Weiterverarbeitung:

  – Mithilfe von Prozeduren können Sie auf die Nachrichtenköpfe (Header), auf die Nachrichten (Messagebody) und auf die Anhänge (Attachments) einer Mail zugreifen.

  – An der C++-Programmschnittstelle wird die gesamte Mail als Instanz einer C++-Klasse zur Weiterbearbeitung an das Anwenderprogramm übergeben.

    Zur softwaremäßigen Darstellung einer Mail wird das Open-Source Produkt „Mimelib“ verwendet, das Klassen zur Verfügung stellt, die einen bequemen Zugriff auf die Mail gestatten.

3 SSL

SSL (Secure Sockets Layer) ist das derzeit am weitesten entwickelte Sicherheitsprotokoll im Internet. Ursprünglich von der Firma Netscape Communications für den sicheren Datentransfer via HTTP-Protokoll entwickelt, kann SSL inzwischen jedes Protokoll absichern, das im TCP/IP-Protokoll-Stack oberhalb der Transportschicht (TCP) angesiedelt ist.

Dieses Kapitel behandelt die folgenden Themen:

- Kommunikationssicherheit im Internet
- Kryptographische Grundlagen
- Überblick über SSL
- Beantragen und Erzeugen von X.509-Zertifikaten
- Überblick über die TLS/SSL-Unterstützung in FTP und TELNET

3.1 Kommunikationssicherheit im Internet

Durch das explosionsartige Wachstum des Internet und sein Vordringen in nahezu alle Bereiche des täglichen Lebens rücken auch die damit verbundenen Sicherheitsaspekte immer mehr in den Vordergrund:

- Kommunikationssicherheit, d.h. Datenauthentizität, Datenintegrität, Datenvertraulichkeit etc.
- Ausfallsicherheit, d.h. weitgehende Verfügbarkeit der beteiligten Systeme
- Schutz vor Viren, Würmern, Trojanischen Pferden, Backdoors u.a.

Gegenstand von SSL und damit des vorliegenden Kapitels sind die Aspekte der Kommunikationssicherheit.

3.1.1 Gefahren für die Kommunikationssicherheit

Bei den Bedrohungen für die Kommunikationssicherheit im Internet lassen sich aktive und passive Angriffe unterscheiden.

Aktive Angriffe auf die Kommunikationssicherheit

Zu den aktiven Angriffen auf die Kommunikationssicherheit zählen:

- Sender-Adresse der Nachricht fälschen (Address Spoofing)
- Inhalt der Nachrichten fälschen (Tampering)
- Nachrichten abfangen und erneut einspielen (Capture-Replay)
- Reihenfolge der gesendeten Nachrichten ändern

Passive Angriffe auf die Kommunikationssicherheit

Zu den passiven Angriffen auf die Kommunikationssicherheit zählen:

- Nachrichteninhalte lesen
- Verkehrsfluss der Nachrichten analysieren:
  - Wer sind die Kommunikationspartner?
  - Nachrichtenaufkommen im zeitlichen Verlauf
  - Länge und Häufigkeit der Nachrichten

3.1.2 Kommunikationssicherheit durch Kryptographie

Den Bedrohungen für die Kommunikationssicherheit begegnet SSL mithilfe kryptographischer Verfahren (siehe Abschnitt „Grundlagen der Kryptographie“ auf Seite 34).

Ziele der kryptographischen Verfahren sind im Einzelnen:

- Authentizität des Datenursprungs
  Datenursprungsauthentizität weist die angegebene Datenquelle als tatsächlichen Absender der Daten aus. Dies ist erforderlich für die Abwehr aktiver Angriffe, bei denen sich der Angreifer zwischen die beiden Kommunikationspartner stellt („Man in the middle“) und sich als der jeweils andere Partner ausgibt.
- Datenvertraulichkeit
  Datenvertraulichkeit verhindert, dass nicht autorisierte Personen den Datenverkehr lesen können.
- Datenintegrität
  Datenintegrität garantiert, dass die übertragenen Daten nicht verändert werden.
- Anti-Replay
  Anti-Replay verhindert, dass Daten von einem Eindringling abgefangen und anschließend wieder eingespielt werden.
- Vertraulichkeit des Verkehrsflusses
  Vertraulichkeit des Verkehrsflusses verhindert die nicht autorisierte Analyse des Nachrichtenverkehrs.
- Beweisbarkeit (Non-Repudiation)
  Non-Repudiation stellt sicher, dass der Kommunikationspartner nicht abstreiten kann, die transferierten Daten abgeschickt zu haben.

Die vier zuerst genannten Ziele lassen sich mit SSL realisieren. Dabei bietet SSL hohe Flexibilität bei der Auswahl der eingesetzten kryptographischen Verfahren und entlastet den Anwender gleichzeitig von der Notwendigkeit kryptographischer Detailkenntnisse.

3.2 Grundlagen der Kryptographie

Kryptographie realisiert die Ziele der Kommunikationssicherheit wie Datenvertraulichkeit, Datenintegrität etc. mithilfe der folgenden kryptographischen Methoden:

- Verschlüsselungsverfahren
- Kryptographische Hash-Funktionen und Message Authentication Codes (MAC)
- Digitale Signaturen

Die meisten kryptographischen Verfahren benötigen starke Zufallszahlen. Damit nun nicht jede einzelne kryptographische Anwendung Verfahren für die Generierung geeigneter Zufallszahlen implementieren muss, steht in BS2000/OSD zentral der Zufallsgenerator PRNGD (**P**seudo **R**andom **N**umber **G**enerator **D**emon) zur Verfügung (siehe Handbuch „interNet Services Administratorhandbuch“).

3.2.1 Verschlüsselungsverfahren

Es gibt zwei Klassen von Verschlüsselungsverfahren, die aufgrund ihrer spezifischen Vor- und Nachteile auf unterschiedliche Anwendungsbereiche zugeschnitten sind:

- Symmetrische Verschlüsselungsverfahren

  Symmetrische Verschlüsselungsverfahren werden für die Verschlüsselung der Nutzdaten verwendet (Vertraulichkeit).

- Asymmetrische Verschlüsselungsverfahren

  Asymmetrische Verschlüsselungsverfahren werden verwendet

  – in Schlüsselaustausch-Protokollen,

  – zur Erstellung von digitalen Signaturen (Non-Repudiation).

Beiden Klassen ist gemeinsam, dass die Sicherheit auf der Geheimhaltung des bzw. der Schlüssel beruht, während das Verfahren selbst allgemein bekannt ist.

Symmetrische Verschlüsselung (Symmetric Key Encryption)

Bei der symmetrischen Verschlüsselung verwenden die kryptographischen Algorithmen für das Verschlüsseln der Daten beim Sender und das Entschlüsseln beim Empfänger denselben Schlüssel.

Wenn der Schlüssel vor seiner Anwendung zwischen Sender und Empfänger über dasselbe Medium ausgetauscht werden soll, über das auch die verschlüsselten Nutzdaten transportiert werden, muss der Gefahr der Schlüsselkompromittierung begegnet werden. Hierfür bietet sich z.B. die Verwendung asymmetrischer Verschlüsselungsverfahren wie RSA oder DH an. Das DH-Verfahren kann jedoch im Gegensatz zu RSA nicht die Authentizität der am Schlüsselaustausch beteiligten Partner garantieren. Dies muss über einen zusätzlichen Authentifizierungsmechanismus realisiert werden, z.B. via DSS (Digital Signature Standard).

Da jedes Paar von Kommunikationspartnern einen eigenen Schlüssel benötigt, verursacht die Schlüsselverwaltung einen erheblichen Aufwand, da die Anzahl der benötigten Schlüssel proportional zum Quadrat der Anzahl der Gruppenmitglieder ist.

Die Geschwindigkeit der symmetrischen Verfahren ist im Vergleich zu den asymmetrischen Verfahren hoch.

Die Sicherheit der symmetrischen Verschlüsselung korreliert mit der Schlüssellänge. Für eine sichere Verschlüsselung sollte die Schlüssellänge mindestens 80 Bits betragen.

Die bekanntesten Vertreter symmetrischer Verschlüsselung sind:

- DES (Digital Encryption Standard)
  DES ist das am besten untersuchte symmetrische Verfahren.
- 3-DES („Triple DES“)
  3-DES besteht in einer dreifach nacheinander angewandten DES-Verschlüsselung.
- AES (Advanced Encryption Standard)
  In einem Wettbewerb wurden Kandidaten für AES, dem DES-Nachfolge-Standard, gesucht. Als Sieger ging ein Verfahren mit dem Namen Rijndael aus dem Wettbewerb hervor.

Asymmetrische Verschlüsselung (Public Key Encryption)

Bei der asymmetrischen Verschlüsselung besitzt jeder Kommunikationspartner zwei verschiedene Schlüssel, zwischen denen ein mathematischer Zusammenhang besteht:

- Public Key

  Der Public Key (öffentlicher Schlüssel) ist allen Kommunikationspartnern bekannt und wird zum Verschlüsseln der Nachricht verwendet.

- Private Key

  Der Private Key (privater Schlüssel) ist nur dem Besitzer bekannt und wird zum Entschlüsseln der Nachricht verwendet.

Der Nachrichtenaustausch zwischen zwei Kommunikationspartnern A und B unter Verwendung asymmetrischer Verschlüsselung verläuft nach dem folgenden Schema:

1. Bevor A eine Nachricht an B sendet, muss A den Public Key von B kennen.
2. A verschlüsselt seine Nachricht mithilfe des Public Key von B.
3. A sendet die verschlüsselte Nachricht an B. (Die verschlüsselte Nachricht kann nun ausschließlich mithilfe des Private Key von B entschlüsselt werden.)
4. B entschlüsselt die Nachricht mithilfe seines Private Key.

Da einer der beiden Schlüssel öffentlich bekannt sein darf, ist nur ein Schlüsselpaar pro Empfänger erforderlich. Deshalb werden insgesamt wesentlich weniger Schlüssel benötigt als bei symmetrischen Verfahren.

Asymmetrische Verfahren sind im Vergleich zu den symmetrischen Verfahren wesentlich langsamer.

Bei asymmetrischen Verschlüsselungsverfahren kann nur der Eigentümer des privaten Schlüssels Operationen mit diesem durchführen. Auf dieser Basis lassen sich Signaturverfahren erstellen („elektronische Unterschrift“).

Die Sicherheit der asymmetrischen Verschlüsselung korreliert mit der Schlüssellänge. Für eine sichere Verschlüsselung sollte die Schlüssellänge bei RSA und DH mindestens 1024 Bits betragen.

Die bekanntesten Verfahren asymmetrischer Verschlüsselung sind:

- RSA
  RSA steht für die Erfinder Rivest, Shamir und Adleman.
- DH
  DH steht für die Erfinder Whitfield Diffie und Martin Hellman. DH kann nicht für digitale Signaturen verwendet werden. Hierfür steht z.B. DSS (Digital Signature Standard) zur Verfügung. DSS ist auch unter der Bezeichnung DSA (Digital Signature Algorithm) bekannt.
- ECC (Elliptic Curve Cryptography)
  Diese Verfahrensklasse ist noch relativ jung. Da die Anforderungen an die Leistungsfähigkeit der Hardware relativ gering sind, eignen sich diese Verfahren insbesondere auch für Smart-Cards.

3.2.2 Kryptographische Hash-Funktionen und MACs

Eine Hash-Funktion ist eine mathematische Funktion, die eine Zeichenkette beliebiger Länge auf eine Zeichenkette fester Länge abbildet. Somit lässt sich mit Hash-Funktionen zu einem umfangreichen Klartext ein charakteristisches Kennzeichen erstellen. Dieses Kennzeichen wird Check-Summe, Message Digest oder einfach Digest genannt.

Ein für kryptographische Zwecke geeigneter Hash-Algorithmus muss einer Reihe von Anforderungen genügen:

- Für identischen Input muss der Hash-Algorithmus denselben Output liefern.
- Minimale Änderungen des Input müssen in einem deutlich veränderten Message Digest resultieren.
- Aus dem Message Digest darf sich unter keinen Umständen der Input rekonstruieren lassen.
- Es sollte praktisch unmöglich sein, zwei unterschiedliche Klartexte zu finden, für die der Hash-Algorithmus denselben Message Digest liefert.

Hash-Funktionen mit den genannten Eigenschaften heißen kryptographische Hash-Funktionen. Kryptographische Hash-Funktionen eignen sich gut zur Sicherung der Datenintegrität.

Zwei sehr häufig verwendete Hash-Algorithmen sind MD5 und SHA-1. Die Digest-Länge beträgt bei MD5 128 Bits, bei SHA-1 160 Bits.

Message Authentication Code (MAC)

Message Authentication Codes (MACs) sind kryptographische Hash-Funktionen, die für die Erzeugung des Message Digest zusätzlich einen geheimen Schlüssel verwenden. MACs sichern Integrität und Authentizität des Datenverkehrs zwischen zwei Kommunikationspartnern, die sich einen geheimen Schlüssel teilen.

Der gebräuchlichste MAC ist HMAC. HMAC kann mit jedem kryptographischen Hash-Algorithmus verwendet werden und ist zurzeit der einzige in SSL und OpenSSL unterstützte MAC.

3.2.3 Digitale Signaturen

Neben der Sicherstellung der Datenintegrität werden kryptographische Hash-Algorithmen für die Erstellung digitaler Signaturen verwendet. Hierzu wird von einem Klartext zunächst der Hash-Wert (Message Digest) berechnet und dieser dann mit einem privaten Schlüssel verschlüsselt. Digitale Signaturen eignen sich insbesondere zur Sicherstellung der Beweisbarkeit (Non-Repudiation).

Aus Sicherheitsgründen sollte die Mindestlänge des verwendeten Public Key 1024 Bits betragen.

3.2.4 Reduzierung der CPU-Auslastung durch openCRYPT™-Anschluss

Durch die hohe Rechenintensität insbesondere der asymmetrischen Verschlüsselungsalgorithmen kann die CPU der Anlage, auf der der SSL-Client bzw. der SSL-Server abläuft, erheblich belastet werden. Eine Lösung dieses Problems besteht darin, die Berechnung rechenzeitintensiver Verschlüsselungsalgorithmen auf eine separate Hardware auszulagern. In BS2000/OSD stehen hierfür die openCRYPT™-Produkte zur Verfügung.

Folgende Merkmale sind für die openCRYPT™-Produkte hervorzuheben:

- Hohe Verarbeitungsleistung durch externen Co-Prozessor für S-Server-Anlagen (openCRYPT™-BOX bzw. dedizierte CPU für SX-Server-Anlagen (openCRYPT™-SOFT)
- BS2000-Anwendungsschnittstelle (openCRYPT™-SERV)
- Sichere Verschlüsselung durch bewährte und anerkannte Algorithmen und lange Schlüssel.

Voraussetzung für die Weiterleitung von Verschlüsselungsaufträgen ist auf der BS2000/OSD-Seite das Subsystem openCRYPT™-SERV mit der standardisierten Schnittstelle PKCS#11. In diesem Subsystem sind die PKCS#11-Funktionen und -Mechanismen für die Programmiersprachen ANSI C und /390-Assembler implementiert. openCRYPT™-SERV bietet die Arbeitsumgebung für den Anwender, der Daten aus einer Anwendung verschlüsseln möchte.

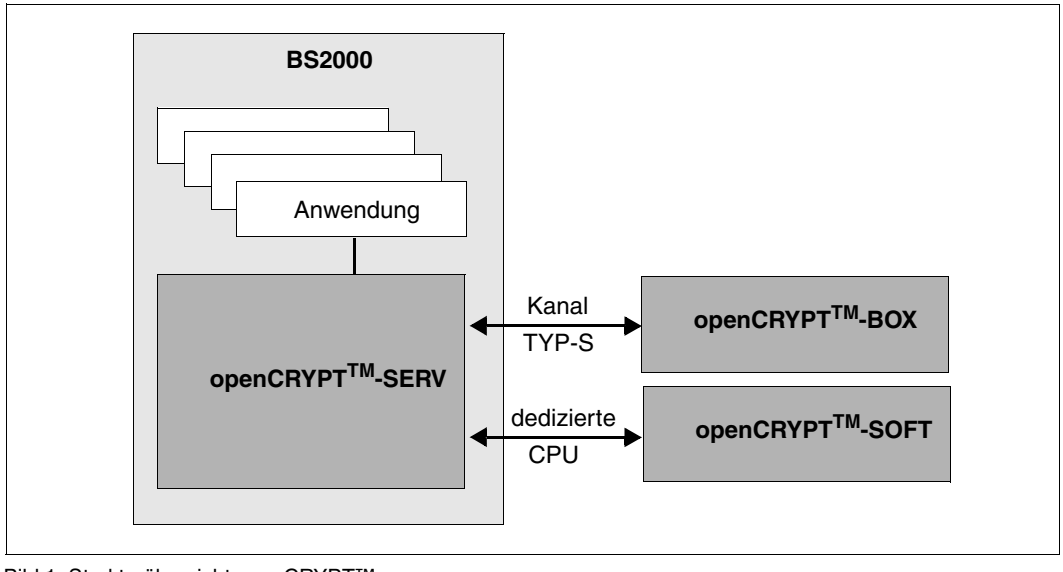


Bild 1: Strukturübersicht openCRYPT™

Das Subsystem openCRYPT™-SERV empfängt einen Verschlüsselungsauftrag von einer Anwendung im BS2000/OSD. Es leitet den Verschlüsselungsauftrag via Kanal an openCRYPT™-BOX bzw. zu einer dedizierten CPU an openCRYPT™-SOFT weiter. Dort wird der Auftrag bearbeitet.

Das Ergebnis – die verschlüsselten Daten – fließt zurück an das Subsystem openCRYPT™-SERV. Dieses leitet die Daten weiter an die BS2000/OSD-Anwendung, die den Auftrag erteilt hat. Die folgende Abbildung veranschaulicht diesen Vorgang für einen Kanalanschluss.

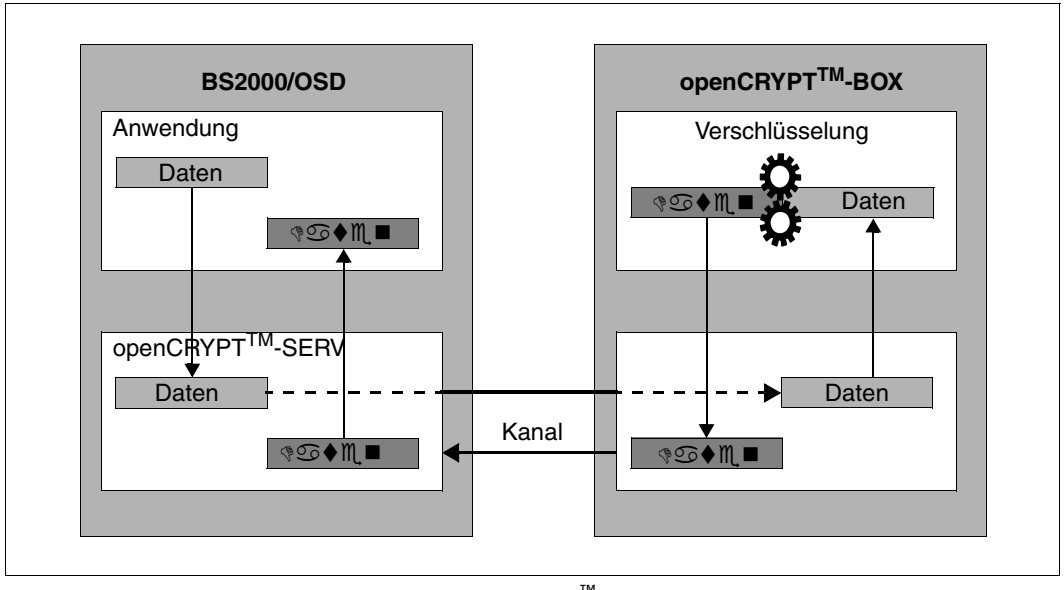


Bild 2: Ablaufdiagramm der Verschlüsselung via openCRYPT™-BOX

Nähere Informationen zu openCRYPT™ finden Sie im Handbuch „openCRYPT“.

3.3 Überblick über SSL

SSL (Secure Socket Layer) ermöglicht die gegenseitige Authentifizierung zweier kommunizierender Anwendungen und garantiert darüber hinaus Vertraulichkeit, Integrität und Authentizität der ausgetauschten Anwendungsdaten. So können Client-/Server-Systeme via SSL kommunizieren, ohne Gefahr zu laufen, dass die ausgetauschten Daten abgehört oder verfälscht werden. Für die Protokolle bzw. Anwendungen, die die Dienste von SSL nutzen, ist der Einsatz von SSL transparent.

Authentifizierung sowie Datenintegrität und Datenvertraulichkeit realisiert SSL mithilfe zweier untergeordneter Protokolle:

- SSL Record Protocol
- SSL Handshake Protocol

Das SSL Record Protocol definiert das für den Transfer der Daten verwendete Format. Das SSL Handshake Protocol ermöglicht es SSL-Client und SSL-Server, sich gegenseitig zu authentifizieren und Verschlüsselungsalgorithmen samt kryptographischem Schlüssel auszutauschen, bevor ein Protokoll der Anwendungsschicht die ersten Daten transferiert.

In interNet Services basieren die Implementierungen auf dem OpenSSL-Toolkit. Zum Zeitpunkt der Drucklegung des Handbuchs wurde die Version 1.0.0a des OpenSSL-Toolkits unterstützt. Die unterstützten Protokollversionen sind SSLv2, SSLv3 und TLSv1.

3.3.1 SSL im TCP/IP-Protokoll-Stack

Das SSL-Protokoll liegt im TCP/IP-Protokoll-Stack oberhalb des TCP-Protokolls und unterhalb der Anwendungsschicht (Application Layer):

- Das SSL Record Protocol setzt direkt auf dem TCP-Protokoll auf.
- Das SSL Handshake Protocol operiert auf dem SSL Record Protokoll.

3.3.2 SSL und TLS

Das Transport Layer Security-Protokoll V1.0 ist eine Weiterentwicklung des SSL V3.0-Protokolls, die durch die Internet Engeneering Task Force (IETF) im RFC 2246 standardisiert wurde. Obwohl zwischen TLS V1.0 und SSL V3.0 keine großen Unterschiede bestehen, ist Interoperabilität zwischen den beiden Protokollen nicht ohne Weiteres möglich.

Während SSL zunächst für das HTTP-Protokoll entwickelt wurde, lassen sich SSL bzw. TLS auch zur Absicherung von anderen Protokollen der Anwendungsschicht wie FTP, SMTP oder TELNET einsetzen.

i Das SSL-Protokoll in der Version 2 weist einige sicherheitstechnische Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

3.3.3 Cipher Suites

Bei TLS/SSL können nicht alle denkbaren Kombinationen der einzelnen kryptographischen Verfahren angewendet werden. Vielmehr sind in den TLS/SSL-Standards eine Reihe von verwendbaren Kombinationen aus Authentifizierungsverfahren (RSA, DSS), Schlüsselaustauschverfahren (RSA, DH), Symmetrischem Verschlüsselungsverfahren (DES, 3DES, AES, RC4 etc) und Message Digest festgelegt. Diese Kombinationen werden jeweils als Cipher Suite bezeichnet.

3.3.4 X.509-Zertifikate, Certificate Authorities (CA) und CRLs

Ein X.509-Zertifikat enthält alle für die Identifikation des Servers oder des Clients benötigten Informationen sowie den Public Key des Zertifikateigners. Zertifikate werden in eigenen Dateien abgelegt. Beim Aushandeln der Verbindung identifiziert SSL mithilfe der Zertifikatsdateien den Server, in manchen Anwendungen auch den Client.

Certificate Authority

Zertifikate werden von einer zentralen Stelle, der so genannten Certificate Authority (CA), durch Signieren mit dem privaten Schlüssel der CA ausgestellt, nachdem die Identität der im Zertifikat genannten Organisation und einer vertretungsberechtigten Person überprüft wurde. Die Signatur ist im Zertifikat enthalten und wird zum Zeitpunkt des Verbindungsaufbaus offengelegt, so dass der Client die Vertrauenswürdigkeit des Zertifikats verifizieren kann. Umgekehrt kann auch der Server ein Zertifikat vom Client fordern. Dies kommt allerdings in der Praxis nur selten vor.

Zertifikate, die von einer CA signiert sind, können durch Veröffentlichung in einer so genannten Certificate Revocation List (CRL) für ungültig erklärt werden.

X.509-Zertifikate

Im Zusammenhang mit SSL werden X.509-Zertifikate verwendet. X.509-Zertifikate arbeiten mit einer hierarchischen Vertrauensstruktur, an deren oberster Stelle die Certificate Authorities für die nachgewiesene Identität der Zertifikatsinhaber rechtlich bindend bürgen (Liability). Je nach Vertrauensstufe genügt den CAs als Nachweis der Identität bei Zertifikatsbeantragung eine gültige Mail-Adresse, ein gültiger Host-Name oder weitergehende Vertrauensnachweise (siehe Abschnitt „X.509-Zertifikate beantragen und erstellen“ auf Seite 44). Beispiele für zentrale CAs sind VeriSign, Thawte oder TC TrustCenter GmbH Hamburg.

Den Inhalt eines Zertifikats können Sie sich am Browser ansehen oder mithilfe der Prozedur SHOW.CERT ausgeben lassen (siehe Seite 53). (Die Prozedur SHOW.CERT ruft das OpenSSL-Kommando-Programm auf.)

Die folgende Abbildung zeigt ein Beispiel für ein X.509-Zertifikat.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=XY, ST=Snake Desert, L=Snake Town, O=Snake Oil, Ltd, OU=Certif
icate Authority, CN=Snake Oil CA/emailAddress=ca@snakeoil.dom
        Validity
            Not Before: Jul  4 18:48:05 2003 GMT
            Not After : Jul  4 18:48:05 2013 GMT
        Subject: C=XY, ST=Snake Desert, L=Snake Town, O=Snake Oil, Ltd, OU=Certi
ficate Authority, CN=Snake Oil CA/emailAddress=ca@snakeoil.dom
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:e0:c6:5e:47:eb:c9:ba:73:89:71:81:ba:58:6d:
                    ad:28:7a:36:ef:3e:e1:fe:0b:a5:49:27:07:24:44:
                    93:b3:4f:97:c6:f7:58:0d:fb:39:3d:57:7d:b3:c7:
                    d4:93:7d:70:84:cc:31:2b:89:f2:db:6a:13:9e:7d:
                    5e:d0:6c:d4:51:64:40:04:b6:16:76:3e:be:39:38:
                    37:24:e1:39:cb:c3:12:11:3d:0e:96:31:0d:6e:2d:
                    9d:6e:ed:90:12:a2:78:3c:54:03:1f:b8:f5:88:fe:
                    d0:75:dd:47:37:e1:d7:5e:be:07:44:0a:f5:88:7b:
                    c2:8c:8b:d4:b3:c3:6c:36:37
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                email:ca@snakeoil.dom
            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: md5WithRSAEncryption
        42:68:65:00:d3:b7:de:57:47:7b:dd:01:71:da:92:ce:ae:5c:
        b1:b7:0d:88:6c:1f:51:1d:bd:41:3a:91:65:f5:7b:d4:cb:04:
```

```
3b:16:f8:8c:de:39:de:cd:ff:42:7a:c6:32:ce:7f:26:b8:7f:
01:55:aa:7e:52:1e:8c:20:4e:9a:29:88:b3:26:b5:11:13:82:
c8:1e:27:da:2c:32:03:4e:75:1d:57:d2:f8:c3:ab:b2:6e:6b:
3f:b4:f2:3f:6c:ef:77:d8:ed:8b:0f:3e:51:9a:a8:c9:11:a9:
f4:a8:93:40:03:97:e2:72:ae:86:98:79:52:60:d0:50:81:7b:
ca:df
```

Beispiel für ein X.509-Zertifikat

Certificate Revocation List (CRL)

Zertifikatsverarbeitende Systeme müssen in der Lage sein, zu erkennen, wenn ein Zertifikat widerrufen wurde. Bei X.509-Zertifikaten erfolgt diese Statusprüfung mithife so genannter Certificate Revocation Lists (CRL). In einer CRL sind alle von einer bestimmten Certificate Authority (CA) erteilten Zertifikate aufgeführt, die nicht mehr gültig sind. Somit können Zertifikate, die von einer Certificate Authority herausgegeben wurden, durch Veröffentlichung einer CRL für ungültig erklärt werden.

3.3.5 X.509-Zertifikate beantragen und erstellen

In der Regel erwirbt man ein X.509-Zertifikat von einer kommerziellen Certificate Authority (CA) wie VeriSign (http://www.verisign.com), Thawte, TC TrustCenter GmbH Hamburg uva. Die von den CAs vergebenen Zertifikate sind i.A. nach Vertrauensstufen klassifiziert (z.B. „Class 3“).

Ein Differenzierungsmerkmal der einzelnen Vertrauensstufen ist der Aufwand, der für die Identifizierung des Antragstellers betrieben wird:

- Bei niedrigen Vertrauensstufen genügt die Zustellbarkeit an die angegebene Mail-Adresse.
- Bei höheren Vertrauensstufen muss der Antragsteller z.B. einen beglaubigten Handelsregisterauszug der beantragenden Firma mitliefern. Außerdem muss ein Zeichnungsberechtigter oder PKI-Verantwortlicher der Firma sich per Post Ident-Verfahren o.Ä. identifizieren.

  Höhere Vertrauensstufen bedeuten i.A. auch höhere Gewährleistungssummen im Schadensfall, beispielsweise wenn die CA einer unberechtigten Instanz ein Zertifikat ausstellt. Genauere Informationen sind auf den Websites der CAs zu finden.

Rechtzeitig vor dem Gültigkeitsende muss ein neues Zertifikat erworben und installiert werden. Wenn der private Schlüssel kompromittiert wurde oder die Angaben im Zertifikat nicht mehr gültig sind, muss das Zertifikat widerrufen werden („Revocation“).

Sollen die Zertifikate nur für Inhouse-Anwendungen verwendet werden, dann kann es auch sinnvoll sein, eine eigene CA einzurichten. Vor einem solchen Schritt empfiehlt sich aber eine gründliche Einarbeitung in das Thema PKI (Public Key Infrastructure) z.B. anhand geeigneter Literatur.

Neben den Identifizierungsunterlagen muss der Antragsteller auch einen sogenannten Certificate Signing Request (CSR) einreichen. Ein solcher CSR lässt sich z.B. mit dem OpenSSL-Kommandozeilen-Tool erstellen (siehe Abschnitt „Prozedur MAKE.CERT - Test-Zertifikate und CSRs erzeugen“ auf Seite 47).

3.3.6 SSL-Handshake

Die SSL-Kommunikation zwischen Client und Server beginnt immer mit dem so genannten Handshake. Der Handshake ermöglicht die Authentifizierung des Servers sowie die Vereinbarung der zu verwendenden Verschlüsselungsverfahren und Schlüssel.

Bei jedem Handshake muss sich der Server gegenüber dem Client mithilfe von Public Key-Verschlüsselung authentifizieren. Grundsätzlich kann auch der Server verlangen, dass sich der Client ihm gegenüber authentifiziert (ebenfalls via Public Key-Verschlüsselung). Dies ist jedoch eher die Ausnahme.

Vereinfacht dargestellt, läuft der SSL-Handshake in folgenden Schritten ab:

1. Der Client sendet eine Liste mit den von ihm unterstützten Verschlüsselungsverfahren (Cipher Suites) an den Server.
2. Der Server wählt ein Verfahren aus dieser Liste aus und und sendet es zusammen mit seinem Zertifikat, das den Public Key des Servers enthält, zurück an den Client.
3. Der Client prüft, ob das Server-Zertifikat bereits von einer CA signiert wurde, deren Zertifikat beim Client vorliegt, und der er somit implizit vertraut. Ferner prüft der Client, ob das Zertifikat für den Server ausgestellt wurde, zu dem er die Verbindung herstellen möchte. Im Einzelnen prüft der Client dabei,
   - ob der CN (Common Name) des Subject mit dem FQDN (Fully Qualified Domain Name) des Servers übereinstimmt, oder
   - ob der DNS-Teil des X.509v3 Subject Alternative Name mit dem FQDN des Servers übereinstimmt.
4. Der Client verschlüsselt mithilfe des Public Key des Servers ein PreMaster Secret und sendet dieses an den Server. Client und Server berechnen dann aus dem PreMaster Secret und den in den vorangegangenen Schritten ausgetauschten Zufallszahlen die Schlüssel, die für die verschiedenen Verschlüsselungs- und MAC-Algorithmen benötigt werden.
5. Bei entsprechender Konfigurierung fordert der Server im Rahmen des Handshakes ein Zertifikat vom Client an. Der Server prüft, ob dieses Zertifikat von einer CA signiert wurde, die er als vertrauenswürdig einstuft. Weitergehende Überprüfungen des Client-Zertifikats werden im Rahmen der interNet Services nicht durchgeführt.

3.4 Aufrufprozeduren für das OpenSSL-Toolkit

Für bestimmte Aufgaben im Zusammenhang mit der Erstellung von X.509-Zertifikaten steht das OpenSSL-Kommandowerkzeug (OpenSSL-Toolkit) zur Verfügung, so u.a. für

- Erzeugen von X.509 Certificate Signing Requests (CSR) und Testzertifikaten
- Anzeigen von Zertifikaten in lesbarer Form
- Wahl einer geeigneten Cipher Suite-Listen-Spezifikation

Für die Implementierung der SSL-Funktionen wurde das OpenSSL-Toolkit verwendet.

Da die Funktionalität des OpenSSL-Kommandowerkzeugs weit über den für FTP und TELNET benötigten Rahmen hinaus geht, gibt es zur Bedienungsvereinfachung einige SDF-P-Prozeduren, die in der LMS-Bibliothek SYSSPR.TCP-IP-AP.052 enthalten sind.

Diese Prozeduren liegen in folgender Form vor:

- als kompilierte Prozeduren. Auf diese Weise können die Prozeduren auch von Anwendern genutzt werden, denen nur SDF-P-BASYS, aber nicht das kostenpflichtige Subsystem SDF-P zur Verfügung steht (SYSJ-Elemente).
- als J-Elemente. Damit steht auch der Quelltext der Prozeduren zur Verfügung.

Nachfolgend finden Sie eine Beschreibung der folgenden Prozeduren:

- MAKE.CERT - Testzertifikate und CSRs (RSA oder DSA, 1024 Bit) erzeugen
- SHOW.CERT - X.509-Zertifikate im Klartext anzeigen
- SHOW.CIPHERLIST - Geeignete Cipher Suite-Listen-Spezifikation auswählen

3.4.1 Prozedur MAKE.CERT - Test-Zertifikate und CSRs erzeugen

Mithilfe der Prozedur MAKE.CERT können Sie Test-Zertifikate und CSRs (RSA oder DSA, 1024 Bit) erzeugen. MAKE.CERT besitzt eine Reihe von Aufruf-Parametern.

3.4.1.1 Beschreibung der Parameter

Parameter zur Spezifizierung der Snakeoil Certificate Authority (CA)

Die Parameter CA-SERIALFILE, CA-CERTFILE und CA-KEYFILE spezifizieren die Dateien für die so genannte Snakeoil-CA. Mithilfe dieser CA können Sie aus den generierten CSRs Test-Zertifikate erstellen, mit denen Sie die TLS-Funktionalität testen können, bevor Sie ein „richtiges“ Zertifikat kaufen. Die von dieser Snakeoil-CA erstellten Test-Zertifikate dürfen nicht für den Produktivbetrieb eingesetzt werden, da sie nicht vertrauenswürdig sind. (Der private Schlüssel dieser CA ist nicht geheim, so dass jeder, der Zugriff auf diesen Schlüssel hat, ein beliebiges, von der Snakeoil-CA signiertes Zertifikat ausstellen kann.)

CA-SERIALFILE

Dieser Parameter spezifiziert die Datei, in der die fortlaufende Seriennummer des jeweils generierten Test-Zertifkats gespeichert wird.

Wenn diese Datei beim Prozeduraufruf noch nicht existiert, wird sie angelegt und mit einer Zeile mit dem Inhalt „00“ initialisiert. Da diese Nummer in das jeweilige Test-Zertifikat eingetragen wird und einige Anwendungen diese Serien-Nummer zur Unterscheidung der von einer bestimmten CA ausgestellten Zertifikate verwenden, sollte diese Datei nach der erstmaligen Erzeugung nicht gelöscht werden. Andernfalls könnten u.U. Test-Zertifikate mit der gleichen Serien-Nummer erzeugt werden. Dies kann bei den genannten Anwendungen zu Problemen führen.

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.SNAKEOIL.SRL vorbelegt.

CA-CERTFILE

Dieser Parameter spezifiziert die Datei, in der das Root-Zertifikat der Snakeoil-CA gespeichert wird.

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.SNAKEOIL.CERT vorbelegt.

CA-KEYFILE

Dieser Parameter spezifiziert die Datei, in der der private Schlüssel der Snakeoil-CA gespeichert wird.

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.SNAKEOIL.KEY vorbelegt.

Parameter zur Spezifizierung der Generierungsdaten für den DSA-Schlüssel

DSA-PARAMFILE

Dieser Parameter spezifiziert die Datei, in der die für die Generierung eines DSA-Schlüssels benötigten Parameter gespeichert sind.

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.DSAPARAM vorbelegt.

Parameter zur Spezifizierung von Test-Zertifikat, privatem Schlüssel und CSR

Die Parameter CERTFILE, KEYFILE und CSRFILE spezifizieren die Dateien, in denen das Test-Zertifikat, der zugehörige private Schlüssel und der Certificate Signing Request (CSR) gespeichert werden.

CERTFILE

Dieser Parameter spezifiziert die Datei, in der das generierte Test-Zertifikat gespeichert wird. Der Name dieser Datei wird z.B. beim FTP-Installationskommando beim Operanden RSA-CERTIFICATE-FILE angegeben (siehe Handbuch „interNet Services Administratorhandbuch“).

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.NEW.CERT vorbelegt.

KEYFILE

Dieser Parameter spezifiziert die Datei, in der der zum Test-Zertifikat und CSR gehörende private Schlüssel gespeichert wird. Der Name dieser Datei wird z.B. beim FTP-Installationskommando beim Operanden RSA-KEY-FILE angegeben. Der Inhalt dieser Datei ist geheim zu halten, insbesondere, wenn später mit dem zugehörigen CSR ein Zertifikat für den Produktiv-Betrieb beantragt werden soll (siehe Handbuch „interNetServices Administratorhandbuch“).

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.NEW.KEY vorbelegt.

CSRFILE

Dieser Parameter spezifiziert die Datei, in der der CSR gespeichert wird. Wollen Sie ein Zertifikat für den Produktiv-Betrieb erwerben, dann schicken Sie diese Datei an eine kommerzielle CA. Nach dem Erhalt des Zertifikats von der CA führen Sie z.B. die FTP-Installation noch einmal durch, spezifizieren dann aber beim Operanden RSA-CERTIFICATE-FILE den Dateinamen des von der CA erhaltenen Zertifikats.

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.NEW.CSR vorbelegt.

Parameter zur Festlegung von Schlüsseltyp und Passphrase-Verschlüsselung

KEY-TYPE

Dieser Parameter spezifiziert, ob ein RSA- oder ein DSA-Schlüssel generiert werden soll (RSA / DSA).

Der Parameter ist mit RSA vorbelegt.

KEY-ENCRYPTION

Dieser Parameter spezifiziert, ob der generierte RSA- oder DSA-Schlüssel mit einer Passphrase verschlüsselt werden soll (YES / NO):

– Eine Verschlüsselung des privaten RSA- oder DSA-Schlüssel ist bei der Verwendung für einen Server wenig sinnvoll, da der Server dann beim Start nach der Passphrase fragt und damit nicht mehr automatisch startbar ist.

– Wenn der generierte Schlüssel für ein Client-Zertifikat verwendet werden soll, kann die Verschlüsselung des privaten Schlüssels sinnvoll sein.

Der Parameter ist mit NO vorbelegt.

3.4.1.2 Prozedurablauf

Nach dem Aufruf verfährt die Prozedur wie folgt:

1. Das RSA- bzw. DSA-Schlüsselpaar wird mit 1024 Bit Schlüssellänge generiert.
2. Der X.509-CSR wird generiert. Hierzu werden interaktiv einige Angaben vom Aufrufer erfragt.
3. Aus dem CSR wird mithilfe der Snakeoil-CA ein Test-Zertifikat generiert.

   Hierzu werden noch weitere Angaben vom Aufrufer erfragt:

   – Gültigkeitsdauer des Test-Zertifikats

   – Version des Zertifikats (X.509v1 oder X.509v3)

   – Bei der Angabe von „3“ (X.509v3) wird der DNS-Name im subjectAltName erfragt. Der DNS-Name ist in der Regel identisch mit „Common Name“ (CN) unter 2).
4. Das generierte Zertifikat wird im Klartext angezeigt.

Beispiel

Nachfolgend ist der Mitschnitt eines Prozeduraufrufs abgedruckt. Die Benutzereingaben sind dabei durch **Fettdruck** hervorgehoben.

```
/CALL-PROCEDURE *LIB($.SYSSPR.TCP-IP-AP.052,MAKE.CERT)
SSL Certificate Generation Utility
Copyright (c) 2003 Fujitsu Technology Solutions, All Rights Reserved

Generating test certificate signed by Snake Oil CA (TEST)
WARNING: Do not use this certificate for real-life/production systems.
         However, you can use the generated Certificate Signing
         Request (CSR) for requesting a real Server Certificate
         from a commercial Certificate Authority (CA).
-----------------------------------------------------------------------

STEP 1: Generating RSA private key (2048 bit)
%  BLS0523 ELEMENT 'OPENSSL', VERSION 'V05.2A00', TYPE 'L' FROM LIBRARY ':09FL:$
TSOS.SYSLNK.TCP-IP-AP.052' IN PROCESS
%  BLS0524 LLM 'OPENSSL', VERSION 'V05.2A00' OF '2010-11-22 20:53:09' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2010. ALL RIGHTS RESERVED
Generating RSA private key, 2048 bit long modulus
.
.
.
.
(Hier werden Zeilen ausgegeben, die den Fortschritt der Schlüsselgenerierung anzeigen)
.
.
.
+
+
+

.
.
.
.
e is 65537 (0x10001)
-----------------------------------------------------------------------

STEP 2: Generating X.509 certificate signing request
%  BLS0523 ELEMENT 'OPENSSL', VERSION 'V05.2A00', TYPE 'L' FROM LIBRARY ':09FL:$
TSOS.SYSLNK.TCP-IP-AP.052' IN PROCESS
%  BLS0524 LLM 'OPENSSL', VERSION 'V05.2A00' OF '2010-11-22 20:53:09' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2010. ALL RIGHTS RESERVED
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

```
1. Country Name             (2 letter code) [DE]:
*DE
2. State or Province Name   (full name)     [Bavaria]:
*Bavaria
3. Locality Name            (eg, city)      [Munich]:
*Munich
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)              : 731
%Certificate Version (1 or 3)               : 3
%9. subjectAltName:dNSName   (eg, FQDN)     : www.ts.fujitsu.com
%  BLSO523 ELEMENT 'OPENSSL', VERSION 'V05.2A00', TYPE 'L' FROM LIBRARY ':O9FL:$
TSOS.SYSLNK.TCP-IP-AP.052' IN PROCESS
%  BLSO524 LLM 'OPENSSL', VERSION 'V05.2A00' OF '2010-11-22 20:53:09' LOADED
%  BLSO551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2010. ALL RIGHTS RESERVED
Signature ok
subject=/C=DE/ST=Bavaria/L=Munich/O=Fujitsu Technology Solutions GmbH/OU=Interne
t Services/CN=ftp.ts.fujitsu.com/emailAddress=info@ts.fujitsu.com
Getting CA Private Key
------------------------------------------------------------------------

STEP 4: Show generated X.509 certificate
%  BLSO523 ELEMENT 'OPENSSL', VERSION 'V05.2A00', TYPE 'L' FROM LIBRARY ':O9FL:$
TSOS.SYSLNK.TCP-IP-AP.052' IN PROCESS
%  BLSO524 LLM 'OPENSSL', VERSION 'V05.2A00' OF '2010-11-22 20:53:09' LOADED
%  BLSO551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2010. ALL RIGHTS RESERVED
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=XY, ST=Snake Desert, L=Snake Town, O=Snake Oil, Ltd, OU=Certif
icate Authority, CN=Snake Oil CA/emailAddress=ca@snakeoil.dom
        Validity
            Not Before: Nov 24 13:28:02 2010 GMT
            Not After : Nov 24 13:28:02 2012 GMT
        Subject: C=DE, ST=Bavaria, L=Munich, O=Fujitsu Technology Solutions GmbH
, OU=Internet Services, CN=ftp.ts.fujitsu.com/emailAddress=info@ts.fujitsu.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:e3:90:89:bb:0a:40:51:b5:b1:b9:cf:2a:9b:4d:
                    6a:f7:63:c5:25:ce:f3:43:ee:80:38:40:66:fc:a0:
                    9e:89:41:c0:6c:1d:74:3c:f3:a9:2f:09:6d:ee:79:
                    5f:b3:36:bc:67:87:4e:0d:60:99:2e:a1:fc:b3:9f:
```

```
                    0c:8c:9d:33:c6:15:03:4a:cd:dd:82:54:85:48:54:
                    d2:f2:1c:82:da:1b:de:a0:5e:1d:bd:ac:07:04:07:
                    f7:df:de:97:18:29:25:df:2c:e8:25:29:25:b1:71:
                    21:2e:fb:ec:72:38:ca:33:d2:e7:6e:21:75:21:5e:
                    41:75:95:f6:88:3f:f1:30:04:e0:65:67:ee:60:c1:
                    0f:1b:7d:54:0b:e9:9d:c9:63:47:37:4d:b7:f8:df:
                    06:84:a3:83:af:09:25:9d:68:6c:53:1f:4d:9b:d7:
                    2b:eb:0c:34:1e:27:34:09:e6:11:20:90:1b:b7:ae:
                    94:6f:ab:94:b2:31:0b:9a:5e:02:a6:5a:b2:ec:88:
                    54:e8:14:46:d2:80:ad:18:b7:a4:37:d8:f6:1e:35:
                    2a:78:77:53:04:68:32:07:49:81:07:35:f1:b4:59:
                    7d:ab:01:d9:05:d9:ef:3f:a5:69:b5:5e:e1:3e:bf:
                    65:9d:b8:16:91:f1:c7:fa:9d:a0:78:e9:bc:39:b5:
                    55:a7
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:ftp.ts.fujitsu.com, email:info@ts.fujitsu.com
            Netscape Comment:
                interNET SERVICES generated test server certificate
    Signature Algorithm: sha1WithRSAEncryption
        67:9b:0a:cc:b8:90:43:6a:0e:cc:d7:99:c6:28:4a:39:0e:bc:
        c8:8a:a2:89:be:47:fb:78:ea:d9:04:87:e7:de:87:d7:e4:6b:
        f3:1d:52:20:79:20:ba:c3:9d:bd:3e:bf:20:53:9c:15:02:27:
        01:e7:4b:ac:46:92:3d:a1:75:48:d7:1f:85:85:c8:4a:f6:2c:
        fd:d1:00:3b:ae:49:07:fb:98:4c:fd:6d:1c:13:31:91:89:26:
        a0:40:f4:62:ad:3c:e8:f2:52:41:4f:99:ab:65:5e:51:12:d9:
        11:f9:28:37:02:80:5c:62:24:bd:e8:42:6b:48:7e:9a:92:8a:
        d9:43
-----------------------------------------------------------------------

RESULT: Server certification files

o  SYSDAT.TCP-IP-AP.052.NEW.KEY
   The PEM encoded RSA private key file. KEEP THIS FILE PRIVATE.

o  SYSDAT.TCP-IP-AP.052.NEW.CERT
   The PEM encoded X.509 certificate file.
   WARNING: Do not use this certficate for real-life/production
            systems.

o  SYSDAT.TCP-IP-AP.052.NEW.CSR
   The PEM encoded X.509 certificate signing request file which you
   can send to an official Certificate Authority (CA) in order to
   request a real server certificate (signed by this CA instead of
   our demonstration only Snake Oil CA) which later can replace the
   SYSDAT.TCP-IP-AP.052.NEW.CERT file.
```

3.4.2 Prozedur SHOW.CERT

Mithilfe der Prozedur SHOW.CERT können Sie die üblicherweise im PEM-Format gespeicherten X.509-Zertifikate im Klartext anzeigen. SHOW.CERT besitzt nur den folgenden Parameter:

CERTFILE

Dieser Parameter spezifiziert die Datei, die das anzuzeigende Zertifikat enthält.

Der Parameter ist mit SYSDAT.TCP-IP-AP.052.NEW.CERT vorbelegt.

Beispiel

```
/CALL-PROCEDURE *LIB($.SYSSPR.TCP-IP-AP.052,SHOW.CERT),-
                (CERTFILE=SYSDAT.TCP-IP-AP.052.SNAKEOIL.CERT)
```

3.4.3 Prozedur SHOW.CIPHERLIST

Die Prozedur SHOW.CIPHERLIST vereinfacht die Wahl der passenden

- *tlsCipherSuite*-Option (siehe Seite 97) in den Option-Dateien von FTP-Client und FTP-Server.
- *-Z CipherSuite*-Option (Seite 300) in den Option-Dateien von TELNET-Client und TELNET-Server.

SHOW.CIPHERLIST besitzt keinen Parameter.

Prozedurablauf

Nach dem Start fragt SHOW.CIPHERLIST nach der Spezifikation für die Verschlüsselungs-Suite. Nachdem Sie diese Spezifikation eingegeben haben, gibt SHOW.CIPHERLIST eine Liste von untereinander durch Doppelpunkte (:) getrennten Verschlüsselungs-Suites aus. Ein mit dieser Option gestarteter FTP-Client würde im SSL-Handshake als akzeptable Verschlüsselungs-Suiten diese Liste (in dieser Reihenfolge) zum Server schicken. Die Reihenfolge ist hier relevant, da die meisten Server aus dieser Liste die erste Suite auswählen, die sich in der Menge der von ihnen akzeptierten Verschlüsselungs-Suiten befindet.

Nach der Ausgabe fragt SHOW.CIPHERLIST erneut nach einer Verschlüsselungs-Suite-Spezifikation. Sobald Sie den gewünschten Option-String gefunden haben, können Sie die Prozedur durch Eingabe von `quit` beenden.

Beispiel

```
/CALL-PROCEDURE *LIB($.SYSSPR.TCP-IP-AP.052,SHOW.CIPHERLIST)
SSL Cipher List Show Utility
Copyright (c) 2007 Fujitsu Technology Solutions, All Rights Reserved

Show SSL Cipher List corresponding to cipher selection string.
--------------------------------------------------------------------------------

%  BLS0523 ELEMENT 'OPENSSL', VERSION 'V05.2A00', TYPE 'L' FROM LIBRARY ':09FL:$
TSOS.SYSLNK.TCP-IP-AP.052' IN PROCESS
%  BLS0524 LLM 'OPENSSL', VERSION 'V05.2A00' OF '2010-11-22 20:53:09' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2010. ALL RIGHTS RESERVED
OpenSSL>
%Cipher selection string: ALL:!EXP:!ADH
ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SH
A:DHE-RSA-CAMELLIA256-SHA:DHE-DSS-CAMELLIA256-SHA:AECDH-AES256-SHA:ECDH-RSA-AES2
56-SHA:ECDH-ECDSA-AES256-SHA:AES256-SHA:CAMELLIA256-SHA:PSK-AES256-CBC-SHA:ECDHE
-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3
-SHA:AECDH-DES-CBC3-SHA:ECDH-RSA-DES-CBC3-SHA:ECDH-ECDSA-DES-CBC3-SHA:DES-CBC3-S
HA:DES-CBC3-MD5:PSK-3DES-EDE-CBC-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA
:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DHE-RSA-SEED-SHA:DHE-DSS-SEED-SHA:DHE-RSA
-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-SHA:AECDH-AES128-SHA:ECDH-RSA-AES128-SHA:EC
DH-ECDSA-AES128-SHA:AES128-SHA:SEED-SHA:CAMELLIA128-SHA:RC2-CBC-MD5:PSK-AES128-C
BC-SHA:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:AECDH-RC4-SHA:ECDH-RSA-RC4-SHA:ECDH
-ECDSA-RC4-SHA:RC4-SHA:RC4-MD5:RC4-MD5:PSK-RC4-SHA:EDH-RSA-DES-CBC-SHA:EDH-DSS-D
ES-CBC-SHA:DES-CBC-SHA:DES-CBC-MD5
OpenSSL>
%Cipher selection string: ALL:!EXP:!ADH:!LOW
ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SH
A:DHE-RSA-CAMELLIA256-SHA:DHE-DSS-CAMELLIA256-SHA:AECDH-AES256-SHA:ECDH-RSA-AES2
56-SHA:ECDH-ECDSA-AES256-SHA:AES256-SHA:CAMELLIA256-SHA:PSK-AES256-CBC-SHA:ECDHE
-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3
-SHA:AECDH-DES-CBC3-SHA:ECDH-RSA-DES-CBC3-SHA:ECDH-ECDSA-DES-CBC3-SHA:DES-CBC3-S
HA:DES-CBC3-MD5:PSK-3DES-EDE-CBC-SHA:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA
:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:DHE-RSA-SEED-SHA:DHE-DSS-SEED-SHA:DHE-RSA
-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-SHA:AECDH-AES128-SHA:ECDH-RSA-AES128-SHA:EC
DH-ECDSA-AES128-SHA:AES128-SHA:SEED-SHA:CAMELLIA128-SHA:RC2-CBC-MD5:PSK-AES128-C
BC-SHA:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:AECDH-RC4-SHA:ECDH-RSA-RC4-SHA:ECDH
-ECDSA-RC4-SHA:RC4-SHA:RC4-MD5:RC4-MD5:PSK-RC4-SHA
OpenSSL>
%Cipher selection string: ALL:!EXP:!ADH:!LOW:-HIGH:3DES:+RC2:+RC4
DHE-RSA-SEED-SHA:DHE-DSS-SEED-SHA:SEED-SHA:ECDHE-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DE
S-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:AECDH-DES-CBC3-SHA:ECDH-RSA
-DES-CBC3-SHA:ECDH-ECDSA-DES-CBC3-SHA:DES-CBC3-SHA:DES-CBC3-MD5:PSK-3DES-EDE-CBC
-SHA:RC2-CBC-MD5:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:AECDH-RC4-SHA:ECDH-RSA-RC
4-SHA:ECDH-ECDSA-RC4-SHA:RC4-SHA:RC4-MD5:RC4-MD5:PSK-RC4-SHA
```

```
OpenSSL>
%Cipher selection string: ALL:!EXP:!ADH:!LOW:-HIGH:3DES:+RC2:+RC4:+kEDH
SEED-SHA:ECDHE-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-SHA:AECDH-DES-CBC3-SHA:ECDH
-RSA-DES-CBC3-SHA:ECDH-ECDSA-DES-CBC3-SHA:DES-CBC3-SHA:DES-CBC3-MD5:PSK-3DES-EDE
-CBC-SHA:RC2-CBC-MD5:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:AECDH-RC4-SHA:ECDH-RS
A-RC4-SHA:ECDH-ECDSA-RC4-SHA:RC4-SHA:RC4-MD5:RC4-MD5:PSK-RC4-SHA:DHE-RSA-SEED-SH
A:DHE-DSS-SEED-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA
OpenSSL>
%Cipher selection string: quit
```

3.5 Nutzung von TLS/SSL

interNet Services bietet für FTP, TELNET, Mail-Reader, Mail-Sender und Mail-Server optional TLS/SSL-Unterstützung an.

3.5.1 TLS/SSL-Unterstützung in FTP

Die TLS/SSL-Unterstützung im FTP-Client und FTP-Server lässt sich auf verschiedene Weise steuern.

TLS/SSL-Unterstützung im FTP-Client einstellen

Für die Einstellung der TLS/SSL-Unterstützung im FTP-Client stehen Ihnen folgende Instrumente zu Verfügung:

- Option-Datei
  Die TLS/SSL-Unterstützung bietet ein breites Spektrum an Einstellmöglichkeiten. Diese Einstellungen können via Options in einer oder mehreren Option-Dateien hinterlegt werden.
- FTP-Client-Kommandos für die TLS/SSL-Unterstützung
  Dies sind Kommandos für die Aktivierung/Deaktivierung der TLS/SSL-Unterstützung, für das Lesen der Option-Datei u.a.
- erweiterte Status-Informationen
  Die Ausgabe des *status*-Kommandos enthält TLS-spezifische Informationen.

Die TLS/SSL-Unterstützung im FTP-Client ist ausführlich beschrieben im Abschnitt „TLS/SSL-Unterstützung im FTP-Client“ auf Seite 88.

TLS/SSL-Unterstützung im FTP-Server einstellen

Für die Einstellung der TLS/SSL-Unterstützung im FTP-Server stehen Ihnen folgende Instrumente zu Verfügung:

- Option-Datei
  Die TLS/SSL-Unterstützung bietet ein breites Spektrum an Einstellmöglichkeiten. Diese Einstellungen können via Options in einer oder mehreren Option-Dateien hinterlegt werden.
- Installationskommando SET-FTP-TELNET-PARAMETERS
  Die Einstellungen zur TLS/SSL-Absicherung können Sie auch mithilfe zusätzlicher Parameter des Installationskommandos vornehmen.

- FTP-Protokoll-Kommandos
  Der FTP-Server unterstützt die FTP-Protokoll-Kommandos AUTH, PBSZ und PROT, um die TLS/SSL-Absicherung der Kontroll- und Datenverbindungen aktivieren bzw. deaktivieren zu können.
- erweiterte Status-Informationen
  Die Ausgabe des STAT-Kommandos enthält TLS-spezifische Informationen.

Die TLS/SSL-Unterstützung im FTP-Server ist ausführlich beschrieben im Handbuch „interNet Services Administratorhandbuch“.

3.5.2 TLS/SSL-Unterstützung in TELNET

Die TLS/SSL-Unterstützung im TELNET-Client und TELNET-Server lässt sich auf verschiedene Weise steuern.

TLS/SSL-Unterstützung im TELNET-Client einstellen

Für die Einstellung der TLS/SSL-Unterstützung im TELNET-Client stehen Ihnen folgende Möglichkeiten zu Verfügung:

- Option-Datei
  Die TLS/SSL-Unterstützung bietet ein breites Spektrum an Einstellmöglichkeiten. Diese Einstellungen können via Options in einer oder mehreren Option-Dateien hinterlegt werden.
- TELNET-Client-Kommandos für die TLS/SSL-Unterstützung
  Dies sind Kommandos für die Aktivierung/Deaktivierung der TLS/SSL-Unterstützung, für das Lesen der Option-Datei u.a.

Die TLS/SSL-Unterstützung im TELNET-Client ist ausführlich beschrieben im Abschnitt „Sicherheit im TELNET-Client“ auf Seite 289.

TLS/SSL-Unterstützung im TELNET-Server einstellen

Für die Einstellung der TLS/SSL-Unterstützung im TELNET-Server stehen Ihnen folgende Möglichkeiten zu Verfügung:

- Option-Datei
  Die TLS/SSL-Unterstützung bietet ein breites Spektrum an Einstellmöglichkeiten. Diese Einstellungen können via Options in einer oder mehreren Option-Dateien hinterlegt werden.

- Installationskommando SET-FTP-TELNET-PARAMETERS
  Die Einstellungen zur TLS/SSL-Absicherung können Sie auch mithilfe zusätzlicher Parameter des Installationskommandos vornehmen.

Die TLS/SSL-Unterstützung im TELNET-Server ist ausführlich beschrieben im Handbuch „interNet Services Administratorhandbuch“.

3.5.3 TLS/SSL-Unterstützung im Mail-Reader

TLS/SSL kann für die Verbindung des Mail-Readers zum POP3- bzw. IMAP-Server verwendet werden, um die Übertragung des jeweiligen Passworts abzusichern.

Die Einstellung der TLS/SSL-Unterstützung erfolgt in der Konfigurationsdatei des Mail-Readers (siehe Abschnitt „POP3/IMAP-Server: Parameterbereich SERVER“ auf Seite 379).

3.5.4 TLS/SSL-Unterstützung im Mail-Sender

TLS/SSL kann für die Absicherung der Verbindung der Service-Task des MAILCLNT-Subsystems zum Mail-Server verwendet werden.

Die Einstellung der TLS/SSL-Unterstützung erfolgt in der Konfigurationsdatei für das Mail-Sender Backend. Die TLS/SSL-Unterstützung im Mail-Sender ist ausführlich beschrieben im Handbuch „interNet Services Administratorhandbuch“.

3.5.5 TLS/SSL-Unterstützung im Mail-Server

TLS/SSL kann für die Absicherung der Verbindungen des Mail-Servers zu den Mail-Clients oder zu anderen Mail-Servern verwendet werden.

Die Einstellung der TLS/SSL-Unterstützung erfolgt in der Konfigurationsdatei für den Mail-Server. Die TLS/SSL-Unterstützung im Mail-Server ist beschrieben im Handbuch „interNet Services Administratorhandbuch“.

4 FTP

interNet Services beinhaltet u.a. die TCP/IP-basierte Anwendung FTP. **F**ile **T**ransfer **P**rotocol dient zum Datentransfer zwischen Rechnern verschiedener Hersteller und ist unabhängig vom Betriebssystem dieser Rechner. Das FTP-Protokoll ist genormt, d.h. die Server-Kommandos und die Antwortmeldungen sind im RFC 959 und weiteren RFCs für neuere Funktionen festgelegt. RFC 959 schreibt nicht alle FTP-Funktionen zwingend vor, so dass verschiedene Implementierungen herstellerabhängig geringfügig voneinander abweichen können.

FTP funktioniert nach dem Client-/Server-Prinzip. Das Client-/Server-Prinzip setzt voraus, dass zwei sich ergänzende Prozesse existieren. Der Initiator einer Verbindung oder Anforderung wird als Client bezeichnet, während sein Partner, der die Anforderung empfängt und beantwortet, als Server bezeichnet wird. Der Server-Prozess, der im Normalfall ständig gestartet ist, wird als Dämon bezeichnet. Der Client-Prozess wird meist durch einen entsprechenden Aufruf, z.B. *ftp*, initiiert.

Die FTP-Funktionalität im BS2000/OSD teilt sich in die Server- und die Client-Funktionalität auf.

4.1 FTP-Server im BS2000/OSD

Der FTP-Server nimmt Aufträge von FTP-Clients am lokalen Netz entgegen und führt sie aus.

Voraussetzung für die Benutzung des FTP-Servers an einem BS2000/OSD-Partner-Rechner ist, dass vom Administrator der FTP-Server hochgefahren wurde.

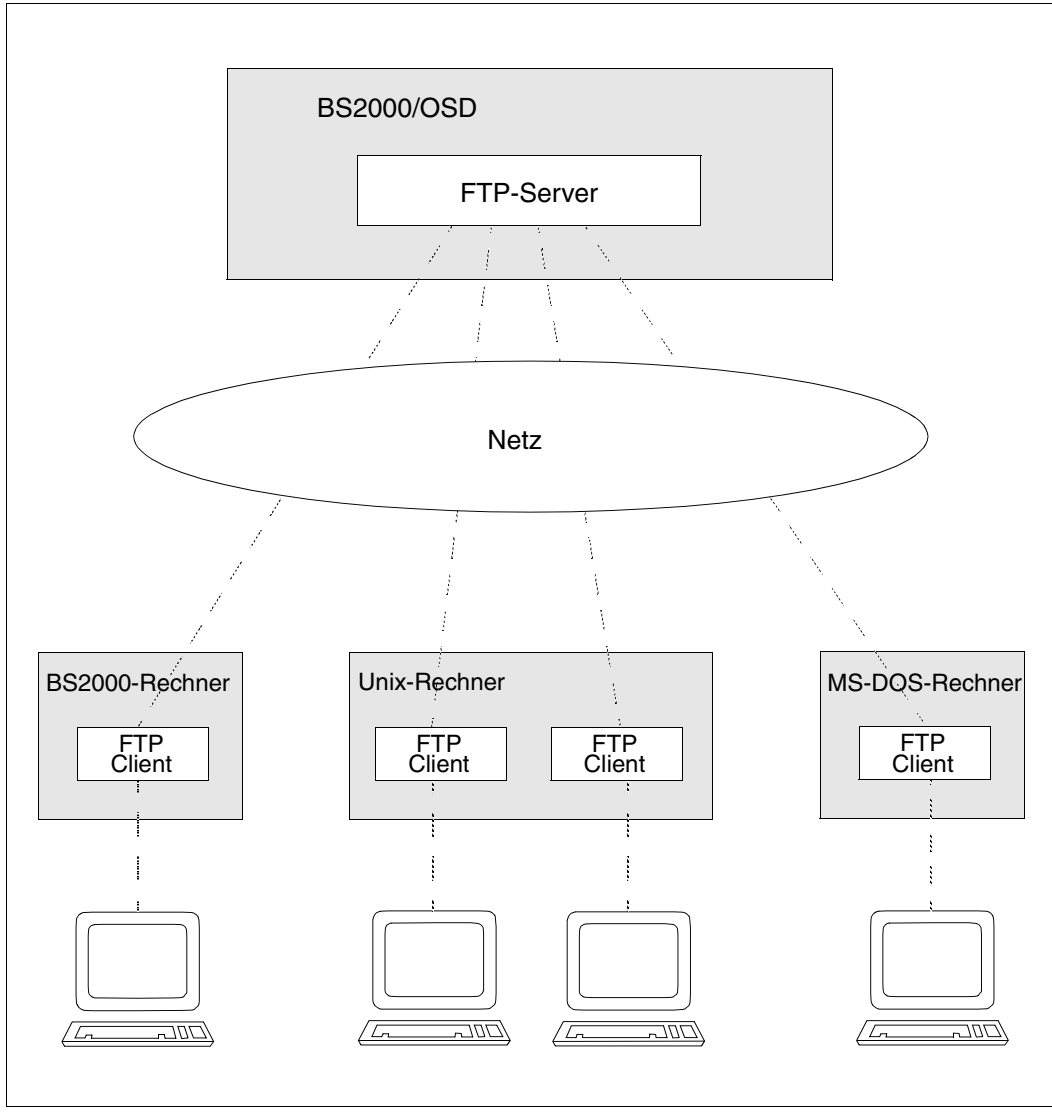


Bild 3: FTP-Server in BS2000/OSD

In der Folge wird der Rechner mit dem FTP-Server als ferner Rechner bezeichnet, damit die Server-Beschreibung mit der Client-Beschreibung konsistent ist (obwohl, vom Standpunkt des Servers betrachtet, der lokale Rechner sein eigener ist).

Arbeitsweise

Für jeden Verbindungswunsch eines Client (Kommando *open*) erzeugt der Server eine eigene BS2000-Task. Die dazu benötigten Daten (Benutzerkennung, Abrechnungsnummer, Passwort) werden vom Client angefordert. Danach wird ein Programm (FTPDC) gestartet, das den eigentlichen Dateitransfer und die vom Client angestoßenen lokalen Dateizugriffe durchführt. Die DMS-Zugriffsrechte des BS2000/OSD (*RDPASS*, *WRPASS*, *EXPASS*, *SHARE=YES|NO*, *ACCESS=READ|WRITE*) werden dabei vollständig beachtet; ebenso die Zugriffsrechte von POSIX.

i Wenn sich mehrere Clients zum exakt gleichen Zeitpunkt am Server anmelden, kann es zur Ablehnung des Verbindungswunsches kommen. In diesem Fall muss der Client den Verbindungsaufbau wiederholen.

Passwortschutz

Im BS2000 können einzelne DVS-Dateien durch Passworte geschützt sein. Der BS2000-FTP-Server erwartet Funktionsaufrufe mit Dateinamen in der Form

dateiname,C'passwort' oder *dateiname,X'passwort'*.

Dies gilt nicht für das POSIX-Dateisystem.

Beispiel

Von einem Unix-System aus (Dateiname *ZWATON*) soll die mit dem Lese-Passwort *OTTO* geschützte BS2000-Datei *ANTON* überschrieben werden:

```
put ZWATON ANTON,C'OTTO'
```

Dateien, die mit unterschiedlichen Passworten geschützt sind (z.B. *RDPASS=C'OTTO'* und *WRPASS=C'KARL'*), können nicht direkt angesprochen werden. Es muss zumindest eines der beiden Passworte mit der Funktion *site exec* des Kommandos *quote* vorher eingegeben werden:

```
quote site exec PASSWORD C'KARL'
```

quote site exec darf nur verwendet werden,

- wenn der FTP-Zugang nicht mit FTAC kontrolliert wird bzw.
- wenn es nicht mithilfe der Server-Option *-disableSiteExecCommand* deaktiviert wurde (siehe Handbuch „interNet Services Administratorhandbuch“).

PASSIVE-Modus

Der PASSIVE-Modus erlaubt es, von einem Client aus eine Datenübertragung zwischen zwei Servern zu initiieren. Der FTP-Client im BS2000/OSD unterstützt dieses Verfahren durch das Client-Kommando *proxy* (siehe Seite 172).

Im BS2000/OSD-FTP-Server wird diese Funktionalität durch die Server-Kommandos PASV bzw. EPSV unterstützt. Im BS2000-FTP-Client kann der PASSIVE-Modus durch das Client-Kommando *passive* (siene Seite 168) eingestellt werden.

Im PASSIVE-Modus wird der Server veranlasst, an einem Datenport einen Verbindungswunsch zu erwarten, anstatt selbst aktiv eine Verbindung aufzubauen. Der Verbindungsaufbau zwischen den beiden Server-Tasks erfolgt über die erste generierte Leitungsadresse. Falls über diese Leitung keine Verbindung zwischen den beiden Servern möglich ist, wird der Verbindungswunsch abgelehnt (Returncode 425, can't build data connection).

Der PASSIVE-Modus wird außerdem verwendet, wenn auf einen FTP-Server hinter einer Firewall zugegriffen werden soll, da vom Server ausgehende Verbindungen oft Konflikte mit den Firewall-Regeln verursachen. Aus diesem Grund nutzen viele Clients standardmäßig den PASSIVE-Modus.

Umfang der Implementierung

Die Funktionen eines FTP-Server sind in genormten Protokollelementen (Serverkommandos) realisiert.

Die folgende Tabelle enthält eine Übersicht über die Funktionen des FTP-Servers im BS2000/OSD.

Server-kommando	Server-Funktion	Im Server auf BS2000/OSD (DVS)	(POSIX)	Korrespondierendes (k) / absetzendes (a) Client-Kommando
acct	Abrechungsnummer bekanntgeben	+	+	
abor	Dateiübertragung abbrechen	+	+	
allo	Speicher zuweisen	(+)	(+)	
appe	Datei an eine bestehende Datei anhängen	+	+	append (a)
auth	TLS/SSL-Verbindung initiieren	+	+	open(a)
cdup	Wechsel in das nächsthöhere ADV (POSIX) bzw. Entfernen der Teilqualifizierungsstufe (DVS)	+ (1)	+	cdup (a)

FTP-Server-Funktionen

Server-kommando	Server-Funktion	Im Server auf BS2000/OSD (DVS)	(POSIX)	Korrespondierendes (k) / absetzendes (a) Client-Kommando
cmod	1:1-Übertragung einschalten	+	+	copymode(k)
cwd	Arbeits-Dateiverzeichnis wechseln (POSIX) bzw. Teilqualifizierung ändern (DVS)	+	+	cd (a)
dele	Eine Datei löschen	+	+	delete (a)
eprt	extended port command für IPv6	+	+	
epsv	extended passive command für IPv6	+	+	
feat	Neue Features des Servers anzeigen	+	+	
help	Hilfe anfordern	+	+	
list	Dateiverzeichnisse und Dateinamen auflisten	+	+	dir (a)
mdtm	Datum und Uhrzeit der bislang letzten Änderung einer Datei ausgeben	+	+	
mkd	Dateiverzeichnis einrichten	-	+	
mode	Übertragungsmodus definieren	+	+	mode (a,k)
nlst	Dateinamen auflisten	+	+	ls (a)
noop	nichts tun	+	+	
opts	Optionen angeben	(+)	(+)	
pass	Passwort bekanntgeben	+	+	
pasv	PASSIVE-Modus einstellen	+	+	
pbsz	Speicher für Datenverschlüsselung zuweisen	(+)	(+)	private (a)
port	Datenverbindungs-Port definieren	+	+	
prot	Verschlüsselung der Datenverbindung aktivieren/deaktivieren	+	+	private (a)
pwd (identisch mit xpwd)	Aktuelles Arbeits-Dateiverzeichnis ausgeben	+	+	pwd (a)
quit	Sitzung beenden	+	+	close (a)
rest	Datei-Position spezifizieren, an der ein Datentransfer aufsetzen soll	+	+	reget (a) reput (a)
retr	Eine Datei wiedererlangen	+	+	get (a)

FTP-Server-Funktionen

Server-kommando	Server-Funktion	Im Server auf BS2000/OSD (DVS)	(POSIX)	Korrespondierendes (k) / absetzendes (a) Client-Kommando
rmd (identisch mit xrmd)	Dateiverzeichnis löschen	-	+	
rnfr	Eine Datei umbenennen: alten Namen angeben	+	+	rename (a)
rnto	Eine Datei umbenennen: neuen Namen angeben	+	+	rename (a)
site	Kommando an fernes BS reichen bzw. Präfix für BS2000/OSD-spezifische Funktionen des FTP-Servers	+	+	
size	Größe einer Datei ausgeben	+	+	
stat	Server-Status-Informationen ausgeben	+	+	status (k)
stor	Eine Datei abspeichern	+	+	put (a)
stou	Eindeutiges Speichern von Dateien	+	+	put (a)
stru	Übertragungsstruktur definieren	+	+	struct (a+k)
syst	Systeminformation über den Server	+	+	system (a)
type	Übertragungs-Typ definieren	+	+	type (a+k)
user	Benutzer-Name definieren	+	+	user (a)
xcup (identisch mit cdup)	Wechsel in das nächsthöhere ADV (POSIX) bzw. Entfernen der Teilqualifizierungsstufe (DVS)	+ (1)	+	
xcwd (identisch mit cwd)	Arbeits-Dateiverzeichnis wechseln (POSIX) bzw. Teilqualifizierung ändern (DVS)	+	+	
xmkd	Dateiverzeichnis einrichten	-	+	mkdir (a)
xpwd	Aktuelles Arbeits-Dateiverzeichnis ausgeben	+	+	pwd (a)
xrmd	Dateiverzeichnis löschen	-	+	rmdir (a)

FTP-Server-Funktionen

Legende

ADV Arbeits-Dateiverzeichnis
\+ Die Funktion ist realisiert.
(+) Die Funktion ist als Leerfunktion realisiert.

- Die Funktion ist nicht realisiert
(a) Client-Kommando setzt Server-Kommando ab.
(k) Client-Kommando hat die Bedeutung des Server-Kommandos.
(a+k) Client-Kommando hat die Bedeutung des Server-Kommandos und setzt Server-Kommando ab.

(1) *cdup* und *xcup* erfüllen im BS2000/OSD-Server die Funktion „Ändern des Arbeits-Dateiverzeichnisses (ADV)“ durch Entfernen einer Teilqualifizierungsstufe.

BS2000/OSD-spezifische Funktionen des FTP-Servers

Die folgende Tabelle gibt einen Überblick über die BS2000/OSD-spezifischen Funktionen des FTP-Servers.

Server-kommando	Server-Funktion	Im Server auf BS2000/OSD (DVS)	(POSIX)	Korrespondierendes Client-Kommando
cmod	1:1-Übertragung einschalten	+	+	copymode (k)
exec	Kommando an fernes Betriebssystem reichen	+	+	
file	DVS-Dateiattribute wechseln	+	+	file (k)
ftyp	Dateibearbeitungstyp wechseln	+	+	ftyp (k)
help	Information zu FTP-Kommandos	+	+	help (k)
modc	Zeichenkette (String) für Wechsel POSIX / DVS ändern	+	+	modchar (k)
setc	Code-Tabellen wechseln	+	+	setcode (k)
sfil	Spezielle EOF-Marker für PAM-Dateien ein-/ausschalten; Auffüllen von leeren SAM-Records ein-/ausschalten	+	(+)	setfile (k)
exit	Parameter für Exit-Routinen definieren	+	+	rexit (a)

BS2000/OSD-spezifische Funktionen des FTP-Servers

Legende

+ Die Funktion ist realisiert.
(+) Die Funktion ist als Leerfunktion realisiert.
(k) Client-Kommando hat die Bedeutung des Server-Kommandos.

Es empfiehlt sich, die BS2000/OSD-spezifischen Funktionen des FTP-Servers immer mit dem Präfix *site* aufzurufen, obwohl aus Kompatibilitätsgründen vorerst noch bei den Kommandos *file*, *ftyp*, *modc* und *setc* auch die Verwendung ohne vorangesetztes *site* möglich ist.

Insbesondere beim Erstellen von Prozeduren wird aber dringend empfohlen, immer die aktuelle Form mit vorangesetztem *site* zu verwenden, um Problemen vorzubeugen, die eventuell bei Erweiterung des FTP-Standards auftreten können.

Folgt auf *site* keines der in der Tabelle genannten Kommandos, so wird *site* in der Bedeutung von *site exec* interpretiert. Die auf *site* folgende Zeichenkette wird dann als Kommando an das ferne Betriebssystem gereicht und wird wie in früheren FTP-Server-Versionen interpretiert.

Aufruf der FTP-Server-Funktionen durch den FTP-Client des Partnerrechners

Für die meisten Funktionen des FTP-Servers gibt es einen Kommando-Aufruf im Client des Partnerrechners. Für Funktionen ohne einen entsprechenden Kommando-Aufruf im Client kann mit dem Client-Kommando *quote* (siehe Abschnitt „quote - Aufruf von Server-Funktionen“ auf Seite 178) die Funktion direkt über den FTP-Server aufgerufen werden.

Für FTP-Clients, die keine spezielle Eingabemöglichkeit für eine Abrechnungsnummer anbieten (z.B. Web-Browser), kann beim BS2000/OSD-FTP-Server eine Abrechnungsnummer, durch ein Komma getrennt, an die Benutzerkennung angehängt werden.

Die Funktionen "Einrichten eines Dateiverzeichnisses" und "Löschen eines Dateiverzeichnisses" finden im DVS-Dateisystem des BS2000/OSD keine sinnvolle Entsprechung. Für das POSIX-Dateisystem des BS2000/OSD sind diese Funktionen im Rahmen der POSIX-Unterstützung realisiert. Über die FTP-Client-Kommandos *mkdir* und *rmdir* können Dateiverzeichnisse im POSIX-Dateisystem eingerichtet und gelöscht werden.

Grundsätzlich kann jede Server-Funktion des Partnerrechners mit dem Client-Kommando *quote* aufgerufen werden.

Im Folgenden werden die Server-Funktionen *file*, *ftyp*, *modc*, *setc*, *sfil*, *cmod* und *exit* näher behandelt, da diese Funktionen nur im BS2000/OSD-FTP-Server existieren und immer über das Client-Kommando *quote* angesprochen werden müssen.

- Die Funktion *site exit* legt die Parameter für die Exit-Routinen fest (siehe Handbuch „interNet Services Administratorhandbuch“). Das *exit*-Kommando können Sie auch über das Client-Kommando *rexit* zum Server schicken (siehe Seite 187).
  Einzugeben ist:

  quote site exit receive:<recv-parm>

  quote site exit send:<send-parm>

  quote site exit receive:<recv-parm>!send:<send-parm>

  z.B.: *quote site exit receive:-C5!send:-D7*
  *quote site exit send:*NONE*

- Die Funktion *site file* legt die Datei-Attribute einer zu übertragenden DVS-Datei am fernen Rechner (jener Rechner, auf dem der FTP-Server läuft) fest. Die Funktion *file* entspricht dem Client-Kommando *file,* das auf Seite 131 detailliert beschrieben ist.
  Einzugeben ist:
  quote site file <ferne-datei,file-operandenliste>
  z.B.: *quote site file datei1,fcbtype=sam*

- Die Funktion *site ftyp* legt fest, ob die SAM-Dateien am fernen Rechner als Text- oder Binär-Dateien bearbeitet werden sollen. Die Funktion *ftyp* entspricht dem Client-Kommando *ftyp*, das auf Seite 134 detailliert beschrieben ist.
  Einzugeben ist:
  quote site ftyp <Dateibearbeitungstyp>
  z.B.: *quote site ftyp binary*

- Die Funktion *site modc* legt das erste Zeichen der Zeichenfolge zum Wechsel zwischen dem DVS-Dateisystem und dem POSIX-Dateisystem fest. Die Funktion *modc* entspricht dem Client-Kommando *modchar*, das auf Seite 159 detailliert beschrieben ist.
  Einzugeben ist:
  quote site modc <Character>
  z. B.: *quote site modc $*

- Die Funktion *site setc*, ermöglicht die Einstellung der Code-Tabellen zur Code-Konvertierung. Die Funktion *setc* entspricht dem Client-Kommando *setcode,* das auf Seite 194 detailliert beschrieben ist. Einzugeben ist:
  quote site setc <EBCDIC-Tabelle> <ISO-Tabelle>
  z. B.: *quote site setc EDF049 ISO88599*

- Die Funktion *sfil* legt spezielle Verhaltensweisen beim Transfer von Dateien fest und entspricht dem Client-Kommando *setfile*, das auf Seite 195 detailliert beschrieben ist. Einzugeben ist:

  quote site sfil datend on\|off
  Ein-/Ausschalten des speziellen EOF-Markers (Default: eingeschaltet).

  quote site sfil pademptyrec on\|off
  Ein-/Ausschalten des Auffüllens von leeren SAM-Records (Default: ausgeschaltet).

  z.B.: *quote site sfil datend off*
  quote site sfil pademptyrec on

- Die Funktion *site cmod* (siehe Beschreibung des Client-Kommandos *copymode* auf Seite 124).

Kommandos des FTP-Servers, die die Restart-Fähigkeit des FTP-Client unterstützen

Folgende Kommandos des FTP-Servers unterstützen die Restart-Fähigkeit von FTP-Client und FTP-Server:

– mdtm
– size
– rest

mdtm- Datum und Uhrzeit der bislang letzten Änderung einer Datei ermitteln

Das *mdtm*-Kommando liefert Datum und Uhrzeit der bislang letzten Änderung einer Datei:

Meldung: `213 <JJJJMMTThhmmss>`

mdtm
<datei>

<datei>
Datei, für die das *mdtm*-Kommando Datum und Uhrzeit der letzten Änderung liefert.

Beispiel

```
quote site mdtm testdatei
213 20101015204331
```

size - Größe einer Datei ermitteln

Das *size*-Kommando gibt an, wieviele Bytes im Falle eines Transfers dieser Datei über das Netz übertragen würden. Dabei werden die aktuellen Einstellungen für *mode*, *type*, *struct* und *ftyp* berücksichtigt. Für *mode* <> *stream* wird das *size*-Kommando mit einem Fehler-Code abgewiesen.

Es kann vorkommen, dass bei einem BS2000/OSD-FTP-Server das *size*-Kommando aus technischen Gründen deaktiviert ist.

Meldung: 213 <Größe der Datei (in byte)>

size
<datei>

<datei>
Datei, für die das *size*-Kommando die Größe liefert.

Beispiel

```
quote site size testdatei
213 498665
```

rest - Datei-Position spezifizieren, an der ein Dateitransfer aufsetzen soll

Das *rest*-Kommando spezifiziert eine beliebige Byte-Position, an deren korrespondierenden Datei-Position (statt am Dateianfang) ein durch ein nachfolgendes *stor*- oder *recv*-Kommando ausgelöster Dateitransfer aufsetzen soll.

rest
<position>

<position>
Byte-Position, an dessen korrespondierender Datei-Position der nachfolgende Transfer einer Datei aufsetzen soll.

Kommando zur Anzeige der Server-Fähigkeiten

Der FTP-Server unterstützt das *FEAT*-Kommando (RFC 2389). Standardmäßig meldet *FEAT* die Unterstützung der Kommandos *SIZE*, *MDTM* und *REST STREAM*. Die Unterstützung von *SIZE* wird nicht angezeigt, wenn *SIZE* via Server-Option *-disableSizeCommand* deaktiviert wurde (siehe Handbuch „interNet Services Administratorhandbuch“).

Wenn die TLS-Unterstützung des FTP-Servers aktiviert ist, meldet *FEAT* außerdem die Unterstützung von *AUTH TLS*, *PBSZ* und *PROT*.

Das von RFC 2389 im Zusammenhang mit dem FEAT-Kommando geforderte *OPTS*-Kommando ist als Leerfunktion implementiert.

FEAT

FTAC-Schnittstelle

Die Vorteile des FTP-Protokolls liegen in der u.a. durch die Normung bedingten weiten Verbreitung von FTP-Client-Programmen, der Aspekt Sicherheit hingegen bewegt sich nicht auf dem BS2000/OSD-üblichen Niveau. D.h. jeder, der Ihre Berechtigungsdaten kennt, kann sich Daten von Ihrer Kennung holen, Daten auf Ihrer Kennung speichern, Daten löschen oder Dateimerkmale ändern.
interNet Services bieten deshalb für FTP den Zugriff auf die FTAC-Schnittstelle an. Zum File Transfer openFT wird seit langem der Zugangs- und Zugriffsschutz openFT-AC angeboten.

FTAC bietet zum Schutz des BS2000/OSD-Servers folgende Möglichkeiten:

– Entkopplung von FTP-Zugangsberechtigung und Login-Berechtigung
– Zugriffsrechte abhängig von Partnersystemen
– benutzerspezifische Zugriffsrechte
– flexible Abstufung der Zugriffsrechte
– Protokollierung jeder Berechtigungsprüfung
– einfache Anwendung

Ausführliche Information zur FTAC-Unterstützung für FTP finden Sie im Kapitel „FTAC-Schnittstelle“ auf Seite 225.

4.2 SNMP-Subagent für FTP

Für den FTP-Server gibt es einen eigenen Subagenten (FTP-Subagent), der über eine Management-Anwendung, den BCAM-Manager, bedient wird.

Im Handbuch „SNMP-Management für openNet Server und interNet Services“ finden Sie Informationen zu den Themen

- Handhabung des BCAM-Managers,
- Software-Voraussetzungen,
- Installation und Deinstallation,
- In- und Außerbetriebnahme des FTP-Subagenten.

Interaktion zwischen FTP-Subagent und FTP-Server

Der FTP-Server erreicht den FTP-Subagenten unter der festen Portnummer 3237. Unmittelbar nach dem Start meldet sich der FTP-Server beim FTP-Subagenten, sofern dieser gestartet ist, und liefert ihm folgende Informationen:

- Portnummer, unter der der FTP-Subagent den FTP-Server erreichen kann
- Server-Portnummer für die Kontrollverbindung zu den FTP-Clients

Sofern nicht bereits ein Server-Entry mit dieser Server-Portnummer existiert, legt der FTP-Server einen entsprechenden Server-Entry an.

Jeder FTP-Server schreibt beim Start seine beiden Portnummern in die Datei SYSDAT.TCP-IP-AP.052.SNMP. Falls der FTP-Subagent erst nachträglich gestartet wird, kann er sich in SYSDAT.TCP-IP-AP.052.SNMP über die momentan aktiven FTP-Server informieren und die entsprechenden Datenstrukturen anlegen.

Wenn der FTP-Server beendet wird, löscht er seinen Eintrag aus der Datei SYSDAT.TCP-IP-AP.052.SNMP.

4.3 1:1-Übertragung von BS2000/OSD-Plattendateien

Die 1:1-Übertragung realisiert den Transfer von BS2000/OSD-Plattendateien unter Beibehaltung von Dateieigenschaften. Voraussetzung ist eine homogene Kopplung, d.h. Quell- und Zielsystem müssen BS2000/OSD-Systeme sein.

Bei der 1:1-Übertragung bleiben u.a. folgende Eigenschaften des Dateiformats unverändert:

- Blockungsfaktor
- Blockstruktur (PAMKEY, DATA, NO)
- Satzformat
- Definitionen des ISAM-Schlüssels

Außerdem werden bei der 1:1-Übertragung Schlüsselinhalte mit übertragen.

Neben den inhalts- und strukturerhaltenden Dateiattributen können optional auch die Dateischutzattribute (USER-ACCESS, ACCESS, BASIC-ACL, AUDIT, RETENTION-PERIOD) außer Passwörtern und GUARDS-Regeln in die Zieldatei übernommen werden.

Mit der 1:1-Übertragung ist es ferner möglich, BS2000/OSD-Dateien auf Nicht-BS2000/OSD-Systemen zwischenzulagern, um sie anschließend auf BS2000/OSD-Systemen einzurichten. Dies entspricht bei openFT einer Übertragung im Transparent-Modus.

- Mit PAMKEY auf NK-Platten oder mit ungeradem Blockungsfaktor bei NK4-Platten ist keine Übertragung möglich. In diesen Fällen müssen Sie zuvor eine Konvertierung mit dem Dienstprogramm PAMCONV durchführen.
- Der Aufbau der übertragenen Dateien wird nicht offengelegt. Somit ist die Anwendung z.B. von per FTP-Exit implementierten Code-Umwandlungsroutinen nicht sinnvoll. Es empfiehlt sich deshalb, entsprechende Exit-Routinen während einer 1:1-Übertragung zu deaktivieren.
- Die 1:1-Übertragung von Banddateien wird nicht unterstützt.

1:1-Übertragung aktivieren

Die 1:1-Übertragung aktivieren/deaktivieren Sie wie folgt:

- Auf FTP-Client-Seite mit dem Kommando *copymode* (siehe Seite 124).
- Auf FTP-Server-Seite mit dem proprietären Kommando CMOD, das Sie am FTP-Client mit *quote site cmod ...* abschicken.

4.4 FTP-Client im BS2000/OSD

Jeder Anwender von FTP im BS2000/OSD eröffnet eine eigene FTP-Client-Task. Beim Kommando *open* stellt der Client die Verbindung zum gewünschten FTP-Server an einem fernen Rechner her.

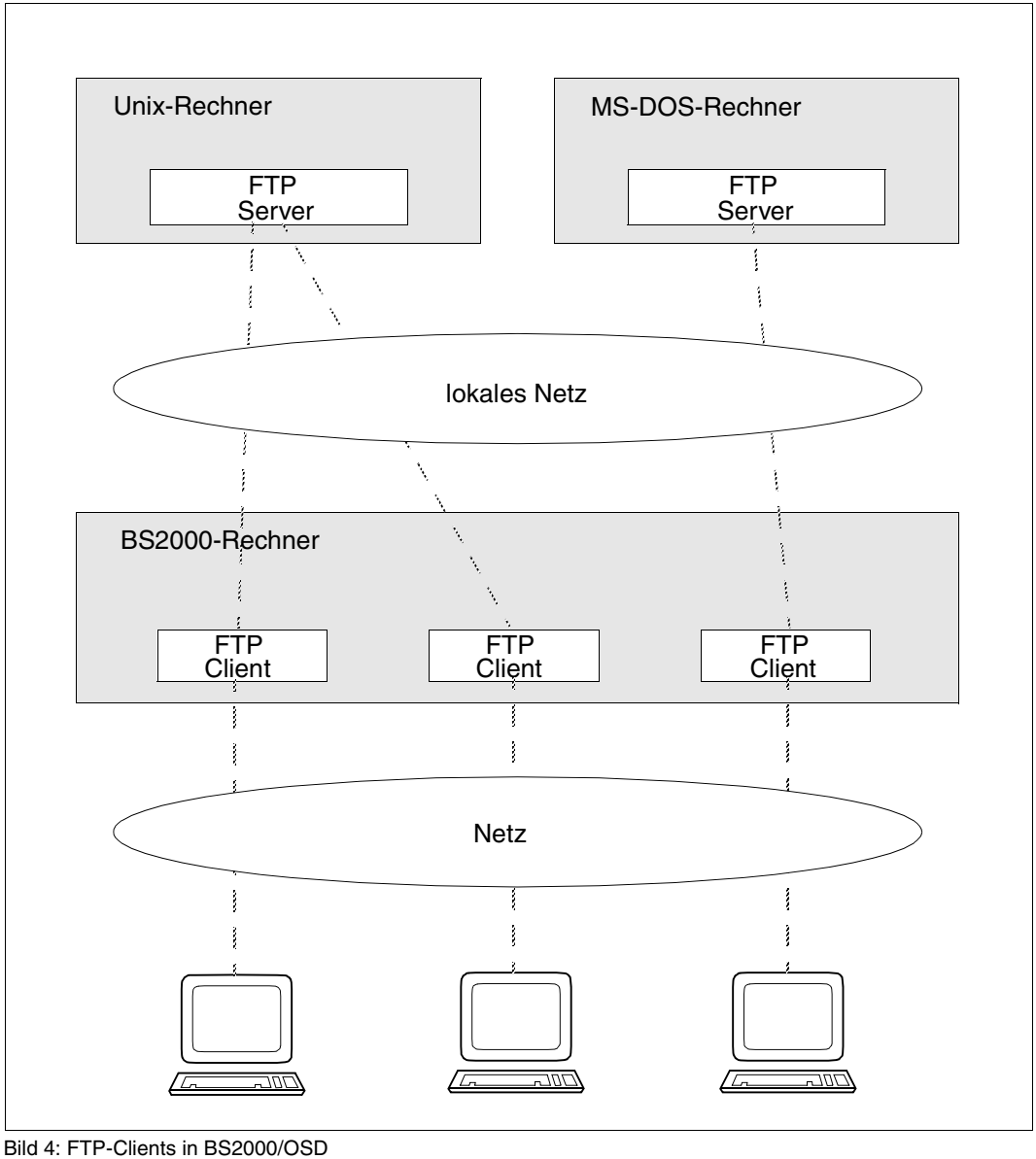


Bild 4: FTP-Clients in BS2000/OSD

Client starten und beenden

Den FTP-Client starten Sie wie folgt:

START–FTP bzw. FTP

Das Beenden des FTP-Client erfolgt mit den Kommandos *quit* oder *bye*.

Verbindung eröffnen und schließen

Vor Operationen, die (auch) einen fernen Rechner betreffen, muss eine Verbindung hergestellt werden. Dies geschieht mit dem Kommando *open*. Existiert am fernen Rechner ein Zugriffsschutzmechanismus, verlangt FTP nun die notwendigen Angaben. Die Zugangsberechtigung im BS2000/OSD ergibt sich z.B. aus den Werten für Benutzerkennung, Abrechnungsnummer und Passwort. Diese Werte sind systemabhängig, siehe folgende Tabelle.

Ist der Partner ein FTAC-geschütztes BS2000/OSD-System, dann muss die Zugangsberechtigung entsprechend der FTAC-Einstellung angegeben werden.

System	**Benutzerkennung**	**Abrechnungs-nummer**	**Passwort**
BS2000/OSD ohne FTAC-Schutz	1 - 8 alphanumerische Zeichen	1 - 8 alphanumerische Zeichen	1 - 8 Zeichen langer C-String bzw. 1 - 16 Zeichen langer X-String
BS2000/OSD mit FTAC-Schutz	$FTAC bzw. entsprechender, bei der Serverinstallation definierter String		8 - 32 Zeichen langer C-String bzw. 15 - 64 Zeichen langer X-String
Unix-System	1 - 32 Zeichen	Unix-Systeme kennen lokal keine Abrechnungsnummer	alphanumerische Zeichen (die Länge ist systemabhängig), es wird zwischen Groß- und Kleinschreibung unterschieden
Windows	1 - 36 Zeichen	Windows kennt lokal keine Abrechnungsnummer	8 - 32 Zeichen langer C-String bzw. 15 - 64 Zeichen langer X-String
MVS	1 - 7 alphanumerische Zeichen	max. 40 Zeichen, Großbuchstaben, Ziffern und Sonderzeichen $, @, #	1 - 8 alphanumerische Zeichen

Die Verbindung zum fernen Rechner kann mit dem Kommando *close* wieder aufgelöst werden.

Im Dialog meldet sich FTP mit einer Versionsnummer und der Eingabeaufforderung *ftp>*. Waren die Berechtigungsangaben nach dem *open* fehlerhaft, bleibt bei nicht gesetztem Auftragsschalter 1 die Verbindung zum fernen Rechner erhalten. Mit dem Kommando *user* können die richtigen Angaben nachträglich angegeben werden. Durch Setzen des Auftragsschalters 1 wird ein SPIN-OFF-Mechanismus eingeschaltet (siehe aber FTP-Client-Kommandos *jobvar* bzw. *svar*). Bei fehlerhaften Angaben an den Zugriffsschutzmechanismus wird der FTP-Client beendet. Nach dem Start von FTP im Dialog ist die Rückfragefunktion eingeschaltet, sie kann mit dem Kommando *prompt* ausgeschaltet werden.

Im Batch müssen nach *open* die Berechtigungsangaben (Kennung, Passwort, Abrechnungsnummer), falls das ferne System es verlangt, einzeln in Folgezeilen angegeben werden. Sind diese Angaben fehlerhaft, so beendet sich der FTP-Client. Das Lesen von Kommandos erfolgt im Batch mit RDATA. Kommandos müssen mit Kleinbuchstaben eingegeben werden. Nach dem Start von FTP im Batch ist die Rückfragefunktion ausgeschaltet.

Für den Batch-Betrieb muss der Auftragsschalter 1 gesetzt werden.

Kommando	Funktion	Seite
open	Eröffnen der Verbindung zu einem fernen Rechner	164
close	Schließen der Verbindung zum fernen Rechner	123
!	In den BS2000/OSD-Kommando-Modus wechseln	213
bye	FTP beenden	118
quit	gleiche Funktion wie bye	177

Kommandos zur Steuerung von FTP

Aktuelle Informationen zu Client- und Server-Einstellungen, zu Kommandos und zu Trace- und Debug-Einstellungen bieten folgende Kommandos:

Kommando	Funktion	Seite
help	Kurzinformation zu FTP-Kommandos ausgeben	140
?	gleiche Funktion wie help	212
status	Ausgabe von FTP-STATUS-Information	197
remotehelp	Information zu Funktionen des fernen FTP-Server	183
debug	DEBUG-Ausgaben ein-/ausschalten	126
trace	TRACE-Ausgaben ein-/ausschalten	205
verbose	Server-Antworten ein-/ausschalten	210
system	Information über den FTP-Server	203

Allgemeine Information über FTP

Dateiverzeichnisse

Nach dem erfolgreichen Verbindungsaufbau ist sowohl am lokalen Rechner als auch am fernen Rechner jeweils ein Arbeits-Dateiverzeichnis definiert. Über das aktuell eingestellte lokale Arbeits-Dateiverzeichnis können Sie sich mit *lpwd* informieren, mit *pwd* über das aktuelle ferne Arbeits-Dateiverzeichnis. Die im lokalen Arbeits-Dateiverzeichnis vorhandenen Dateien können mit dem Kommando *ldir* oder *lls* aufgelistet werden, die Dateien im fernen

Arbeits-Dateiverzeichnis mit dem Kommando *dir* oder *ls*. Werden in den Kommandos Dateien nicht durch ihren vollständigen Pfadnamen identifiziert, so befindet sich die Datei relativ zum aktuell eingestellten Arbeits-Dateiverzeichnis.

Ein lokaler Dateiname muss den Konventionen des lokalen und ein ferner Dateiname denen des fernen Rechners entsprechen. Dabei ist auf Groß- und Kleinschreibung zu achten.

Kommando	Funktion	Seite
ldir	Information über lokale Datei(en)	145
lpwd	Ausgeben des lokalen Arbeits-Dateiverzeichnisses	147
lls	Auflisten von Dateinamen am lokalen Rechner	146

Informationen über lokale Dateien und Dateiverzeichnisse

Kommando	Funktion	Seite
dir	Information über ferne Datei(en)	128
mdir	Information über ferne Dateinamen in mehreren Verzeichnissen am fernen Rechner	152
ls	Auflisten von Dateinamen am fernen Rechner	148
mls	Auflisten von Dateinamen in mehreren Verzeichnissen am fernen Rechner	157
pwd	Ausgabe des fernen Arbeits-Dateiverzeichnisses	176

Informationen über ferne Dateien und Dateiverzeichnisse

BS2000/OSD arbeitet wahlweise mit zwei verschiedenen Dateisystemem, dem proprietären DVS-Dateisystem und dem an Unix-Systeme angelehnten POSIX-Dateisystem. Im BS2000/OSD werden Dateiverzeichnisse durch das Ansprechen von Dateigruppen durch Teilqualifizierung nachgebildet. Der vollständige bzw. absolute Pfadname im BS2000/OSD entspricht dem vollqualifizierten Dateinamen (z.B. *:CATID:$USERID.DATEINAME*). Das aktuell am lokalen Rechner eingestellte Dateisystem kann über das Kommando *lpwd* ermittelt werden. Mit dem Setzen des Arbeits-Dateiverzeichnisses kann zwischen dem DVS-Dateisystem und dem POSIX-Dateisystem gewechselt werden.

Auf Dateien des DVS- bzw. des POSIX-Dateisystems kann nur zugegriffen werden, wenn man sich im aktuellen Arbeits-Dateiverzeichnis in dem entsprechenden Dateisystem befindet. Ein Zugriff auf POSIX-Dateien aus dem DVS-Dateisystem oder ein Zugriff auf DVS-Dateien aus dem POSIX-Dateisystem ist nicht möglich. Dieser Grundsatz ist bei allen FTP-Kommandos und -Funktionen zu beachten.

Einige Betriebssysteme (z.B. Unix-Betriebssystem, MS-DOS) kennen ein hierarchisch organisiertes System von Dateiverzeichnissen. Das POSIX-Dateisystem entspricht in seiner Struktur dem Unix-Dateisystem (UFS). Es verfügt über einen hierarchischen Aufbau.
Im DVS-Dateisystem werden Dateiverzeichnisse durch das Ansprechen von Dateigruppen durch Teilqualifizierung nachgebildet.

Beispiel

```
:5:$TCPTEST.AAA.DATEI1
:5:$TCPTEST.AAA.DATEI2
:5:$TCPTEST.AAA.BBB.DATEI3
:5:$TCPTEST.AAA.BBB.DATEI4
:5:$TCPTEST.AAA.CCC.DATEI5
:5:$TCPTEST.DDD.DATEI6
:5:$TCPTEST.DATEI7
```

:5:$TCPTEST ist das Hauptverzeichnis, das alle Dateien enthält. *AAA* und *DDD* sind Unterverzeichnisse des Hauptverzeichnisses; *BBB* und *CCC* sind Unterverzeichnisse von *AAA*. Der Teil der Dateiangabe, der aus den Verzeichnissen besteht, wird als Pfadname bezeichnet (z.B. ist *:5:$TCPTEST.AAA.BBB* der Pfadname für die Dateien *DATEI3* und *DATEI4*).

Wechsel zwischen DVS- und POSIX-Dateisystem

Der Wechsel vom DVS- in das POSIX-Dateisystem oder umgekehrt erfolgt durch Angabe der aktuell dafür eingestellten Zeichenfolge. Hierbei ist die Groß- und Kleinschreibung zu beachten.

Der Standardwert dieser Zeichenfolge lautet:

%POSIX / %posix zum Wechsel in das POSIX-Dateisystem
%BS2000 / %bs2000 zum Wechsel in das DVS-Dateisystem

Das erste Zeichen dieser Zeichenfolge (%-Zeichen) kann durch das Client-Kommando *modchar* für das lokale System und durch die Server-Funktion *site modc* für das ferne System geändert werden. Da es für die Server-Funktion *site modc* kein eigenes Client-Kommando gibt, muss *site modc* mit dem Client-Kommando *quote* aufgerufen werden.

Abhängig davon, ob man das lokale oder das ferne Dateisystem wechseln möchte, findet der Wechsel des Dateisystems mit den Client-Kommandos *lcd* oder *cd* statt.
Mit *lcd %POSIX* würde man z.B. am lokalen Rechner in das POSIX-Dateisystem wechseln.

Nach dem Wechsel in das gewünschte Dateisystem befindet man sich immer im HOME-Directory des entsprechenden Dateisystems. Der Name dieses HOME-Directorys wird im POSIX-Dateisystem vom Systemverwalter festgelegt, im DVS-Dateisystem entspricht es *:CATID:$USERID*.

Für das Dateimanagement lokaler und ferner Dateien und Dateiverzeichnisse stehen folgende Kommandos zur Verfügung:

Kommando	Funktion	Seite
lcd	Ändern des lokalen Arbeits-Dateiverzeichnisses	143

Veränderung von lokalen Dateien und Dateiverzeichnissen

Kommando	Funktion	Seite
mkdir	Einrichten eines fernen Dateiverzeichnisses	156
rmdir	Löschen eines fernen Dateiverzeichnisses	188
cd	Ändern des fernen Arbeits-Dateiverzeichnisses	120
delete	Löschen einer fernen Datei	127
mdelete	Löschen mehrerer ferner Dateien	150
rename	Umbenennen von fernen Dateien	184
cdup	Wechseln in das nächsthöhere Verzeichnis	122

Veränderung von fernen Dateien und Dateiverzeichnissen

Metazeichen in Dateinamen

Dateinamen lokaler und ferner Dateien sind Operanden vieler FTP-Kommandos. Bei einigen Kommandos muss der Dateiname genau eine Datei bezeichnen; in anderen Fällen können durch Angabe von Meta-Zeichen (z.B. * und ?) auch Dateigruppen bezeichnet werden.

Syntax und Semantik der Metazeichen sind nicht genormt und von der speziellen Implementierung abhängig. Da sich spezielle Implementierungen meist auf bereits vorhandene Betriebssystem-Aufrufe abstützen, sind die erlaubten Meta-Zeichen mit denen im Betriebssystem generell erlaubten identisch. In den meisten Fällen bestimmt der FTP-Server, ob und welche Metazeichen erlaubt sind; in einigen Fällen auch der FTP-Client. Die folgende Tabelle gibt eine Zusammenfassung der erlaubten Metazeichen in den BS2000/OSD-, Unix- und MS-DOS-Betriebssystemen.

Funktion	BS2000/OSD		Unix-System	MS-DOS
	DVS	POSIX		
Eine beliebige (auch leere) Zeichenfolge ersetzen	*	*	*	*
Genau ein Zeichen ersetzen	/	keine Entsprechung	?	?
Die angeführten Zeichenfolgen ersetzen	<s1,...>	keine Entsprechung	keine Entsprechung	keine Entsprechung
Eine Zeichenfolge, die in der lexikalischen Ordnung zwischen den Zeichenfolgen s1 und s2 liegt, ersetzen	<s1:s2>	keine Entsprechung	[s1-s2] (*)	keine Entsprechung
Die entsprechenden Dateien sollen ausgeschlossen werden	-s1 (nur am Anfang der Zeichenfolge)	keine Entsprechung	keine Entsprechung	keine Entsprechung

Metazeichen in Dateinamen
(*) nur einzelne Zeichen sind erlaubt

Backslash

Der FTP-Client im BS2000/OSD entfernt den einfachen Backslash, deshalb muss der Backslash, wenn er beispielsweise als Directory-Trennzeichen verwendet werden soll, doppelt gesetzt werden.

Dateipasswörter

Im BS2000/OSD können einzelne DVS-Dateien durch Dateipasswörter geschützt sein. Der BS2000/OSD-FTP-Server erwartet Funktionsaufrufe mit Dateinamen in der Form

```
<dateiname>,C'<passwort>' oder <dateiname>,X'<passwort>'.
```

Dies gilt nur für das DVS-Dateisystem.

Beispiel

Von einem Unix-System aus (Dateiname *ZWATON*) soll die mit dem Lese-Passwort *OTTO* geschützte BS2000/OSD-Datei *ANTON* überschrieben werden:

```
put ZWATON ANTON,C'OTTO'
```

Umsetzen von Dateien bei der Übertragung (nicht 1:1-Übertragung)

Bei der Übertragung von Dateien mit FTP von einem Betriebssystem auf ein anderes sind zwei Aspekte zu beachten:

- Unterschiedliche Betriebssysteme kennen und akzeptieren unterschiedliche Dateitypen. BS2000/OSD kennt beispielsweise SAM-, ISAM- und PAM-Dateien; Unix-Systeme und MS-DOS kennen nur unstrukturierte Dateien.
- Unterschiedliche Betriebssysteme verwenden verschiedene Codes zur Darstellung von Zeichen. BS2000/OSD codiert Zeichen beispielsweise in EBCDIC; Unix-Systeme und MS-DOS in ASCII.

Durch die Verwendung von XHCS bietet sich die Möglichkeit, zwischen verschiedenen ASCII-/EBCDIC-Codekonvertierungen zu wechseln. Dies wird durch die Verwendung des Kommandos *setcode* beim Client bzw. *quote site setc* (siehe Seite 178) beim Server ermöglicht.

Durch den Übertragungstyp wird die Umsetzung von Dateitypen und Codes beeinflusst. So gilt für einen BS2000/OSD-FTP-Client lokal:

- Beim Übertragungstyp ASCII (Standardwert) erfolgt eine Codeumsetzung von EBCDIC nach ASCII (senden) bzw. umgekehrt (empfangen). Es können SAM-, ISAM- (ohne Key) und PAM-Dateien (ohne PAMKEY) gelesen werden. Es wird standardmäßig eine SAM-Datei mit variabler Satzlänge erzeugt. Diese Voreinstellung kann aber durch Kommando modifiziert werden.
- Beim Übertragungstyp BINARY findet keine Codeumsetzung statt. Es können SAM-, ISAM- (ohne Key) und PAM-Dateien (ohne PAMKEY) gelesen werden. Es wird standardmäßig eine PAM-Datei erzeugt. Diese Voreinstellung kann aber durch Kommando modifiziert werden.
- Beim Übertragungstyp EBCDIC findet keine Codeumsetzung statt. Es können SAM-, ISAM- (ohne Key) und PAM-Dateien (ohne PAMKEY) gelesen werden. Es wird eine SAM-Datei mit variabler Satzlänge erzeugt.

Zusätzlich können Dateiattribute und Bearbeitungstyp über das Kommando *quote site file* bzw. *quote site ftyp* beeinflusst werden:

- *quote site file*:
  Die Dateiattribute einer zu erzeugenden Datei können voreingestellt werden. Hierbei kann festgelegt werden, ob eine PAM- oder SAM-Datei erzeugt werden soll. Im Weiteren werden alle Dateiattribute unterstützt, die von C (BS2000/OSD) bei STREAM I/O unterstützt werden.

- *quote site ftyp*:
  Der Bearbeitungstyp beim Schreiben einer SAM-Datei kann voreingestellt werden. Der Bearbeitungstyp legt fest, ob die Dateien beim Schreiben als Textdateien (satz- bzw. zeilenweise organisiert) oder als Binärdateien (als Folge von Byte organisiert) bearbeitet werden sollen.

– PAMKEYs werden nicht mit übertragen. Es ist somit nicht möglich, ausführbare Dateien zwischen zwei BS2000/OSD-Rechnern direkt auszutauschen. Daher müssen ausführbare Dateien in PLAM-Bibliotheken verpackt übertragen werden.

– ISAM-Dateien verlieren bei der Übertragung die ISAM-Schlüssel. Auch ISAM-Dateien sollten daher vor der Übertragung unbedingt in PLAM-Bibliotheken verpackt werden.

Wie Sie BS2000/OSD-Plattendateien unter Beibehaltung ihrer Eigenschaften transferieren, ist im Abschnitt „1:1-Übertragung von BS2000/OSD-Plattendateien“ auf Seite 72 beschrieben.

Kommando	Funktion	Seite
user	Benutzerkennung am fernen Rechner angeben	208
ascii	Übertragungstyp ASCII einschalten	114
binary	Übertragungstyp BINARY einschalten	117
copymode	1:1-Übertragung von BS2000/OSD-Plattendateien einschalten	124
file	DVS-Dateiattribute wechseln	131
ftyp	DVS-Dateibearbeitungstyp ändern bzw. abfragen	134
modchar	Zeichenfolge zum Wechsel zwischen BS2000/OSD- und POSIX-Dateisystem ändern	159
setcode	Code-Tabellen wechseln	194
setfile	Verwendung von speziellen EOF-Markern und Auffüllen von Leersätzen mit Blank ein-/ausschalten	195
tenex	gleiche Funktion wie binary	204
type	Übertragungstyp ändern oder abfragen	206
mode	Übertragungsmodus ändern oder abfragen	161
struct	Übertragungsstruktur ändern oder abfragen	199
form	Übertragungsformat ändern oder abfragen	133
runique	eindeutiger Zieldateiname beim Überschreiben mit get	189
sunique	eindeutiger Zieldateiname beim Überschreiben mit put	200

Kommandos zur Steuerung der Dateiübertragung

Dateien übertragen

Die eigentliche Dateiübertragung kann unabhängig von der Übertragungsrichtung für eine oder mehrere Dateien angestoßen werden. Bei der Übertragung mehrerer Dateien im Dialog kann mit dem Kommando *prompt* gesteuert werden, ob die Übertragung jeder einzelnen Datei nachgefragt wird oder nicht.

Der von einem Client initiierte Abbruch einer Dateiübertragung wird vom FTP-Server über das Server-Kommando *abor* realisiert. Der BS2000/OSD-FTP-Server meldet in diesem Fall nach erfolgreichem Abbruch. Dieser Abbruch funktioniert nicht, wenn der FTP-Zugang über FTAC erfolgt.

```
226 Abort successful
426 Transfer aborted. Data connection closed.
```

Der FTP-Client im BS2000/OSD erkennt den Wunsch nach Abbruch der Dateiübertragung durch Drücken der K2-Taste. Nach mehrmaligem Drücken der K2-Taste ist eine Rückkehr zu FTP nicht mehr möglich.

Kommando	Funktion	Seite
append	Anhängen einer lokalen an eine ferne Datei	113
get	Holen einer Datei	136
recv	gleiche Funktion wie get	180
mget	Holen mehrerer ferner Dateien	154
reget	Holen einer Datei nach einem Transfer-Abbruch	181
put	Senden einer lokalen Datei	175
send	gleiche Funktion wie put	190
mput	Senden von mehreren lokalen Dateien	162
reput	Senden einer lokalen Datei nach einem Transfer-Abbruch	185

Dateien übertragen

Weitergabe von Kommandos

Sind im FTP-Server des fernen Rechners Funktionen implementiert, für die es keinen Kommando-Aufruf im eigenen FTP-Client gibt, kann die entsprechende Funktion mit dem Kommando *quote* direkt aufgerufen werden. Die notwendigen Parameter werden transparent übergeben; im FTP-Client erfolgt keine Überprüfung. Welche Funktionen im FTP-Server des fernen Rechners implementiert sind, kann mit dem Kommando *remotehelp* ermittelt werden.

Eine mögliche Server-Funktion, die mit *quote* angesprochen werden muss, ist *site exec*. Mit *site exec* können bei der BS2000/OSD-Implementierung Kommandos an das Betriebssystem des fernen Rechners gegeben werden. Bei Nutzung der FTAC-Funktionalität darf *site exec* nicht verwendet werden, es sei denn, es wurde mit der Server-Option *-disableSiteExecCommand* ein anderes Verhalten konfiguriert (siehe Handbuch „interNet Services Administratorhandbuch“).

Kommando	Funktion	Seite
bell	Klingelzeichen ein-/ausschalten	116
hash	Anzeige des Übertragungsfortschritts ein-/ausschalten	139
quote	Übergabe von Parametern an den fernen FTP-Server	178
glob	Expansion von Metazeichen ein-/ausschalten	138
prompt	Rückfrage ein-/ausschalten	170
sendport	Port-Kommando ein-/ausschalten	191
settime	Einstellen des Timeout-Werts für Server-Antworten	196
exit	Parameter für lokale Exit-Routine definieren	130
rexit	Parameter für ferne Exit-Routine definieren	187
jobvar	Verwendung einer Jobvariablen ein-/ausschalten	141
svar	Verwendung einer S-Variablen (SDF-P) ein-/ausschalten	201
passive	PASSIVE-Modus ein-/ausschalten	168

Weitere FTP-Funktionen

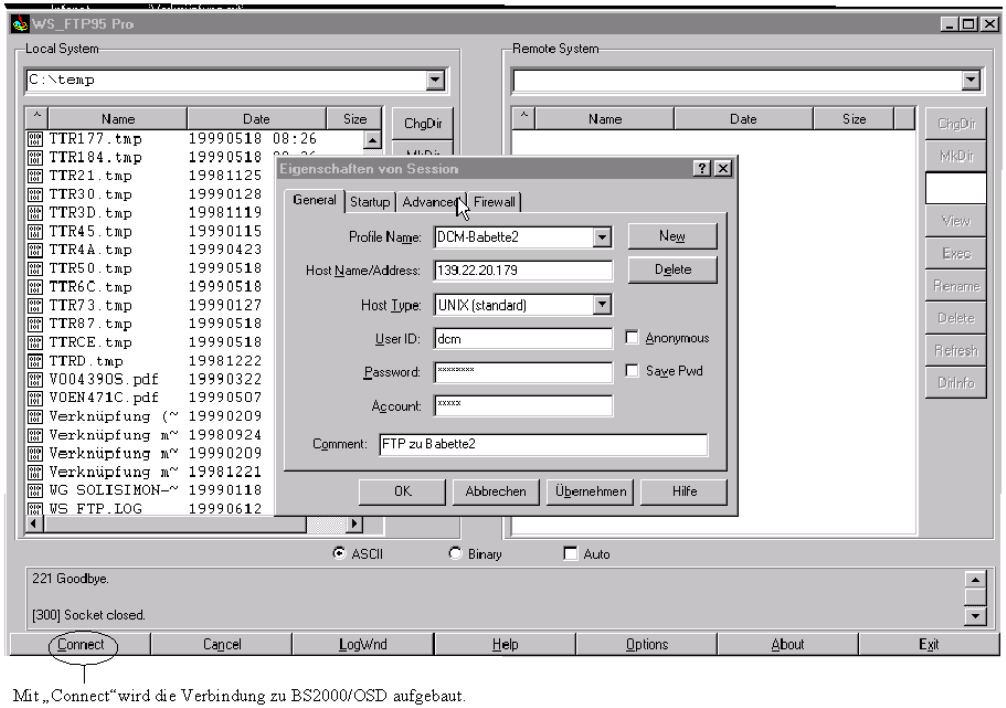
Restart-Fähigkeit des FTP-Client

Die Restart-Fähigkeit des FTP-Client ermöglicht die Wiederaufnahme abgebrochener FTP-Transfers an der Position der Zieldatei, an der der FTP-Transfer unterbrochen wurde. Die Restart-Fähigkeit des FTP-Client wird realisiert durch die FTP-Client-Kommandos *reget* (siehe Seite 181) und *reput* (siehe Seite 185).

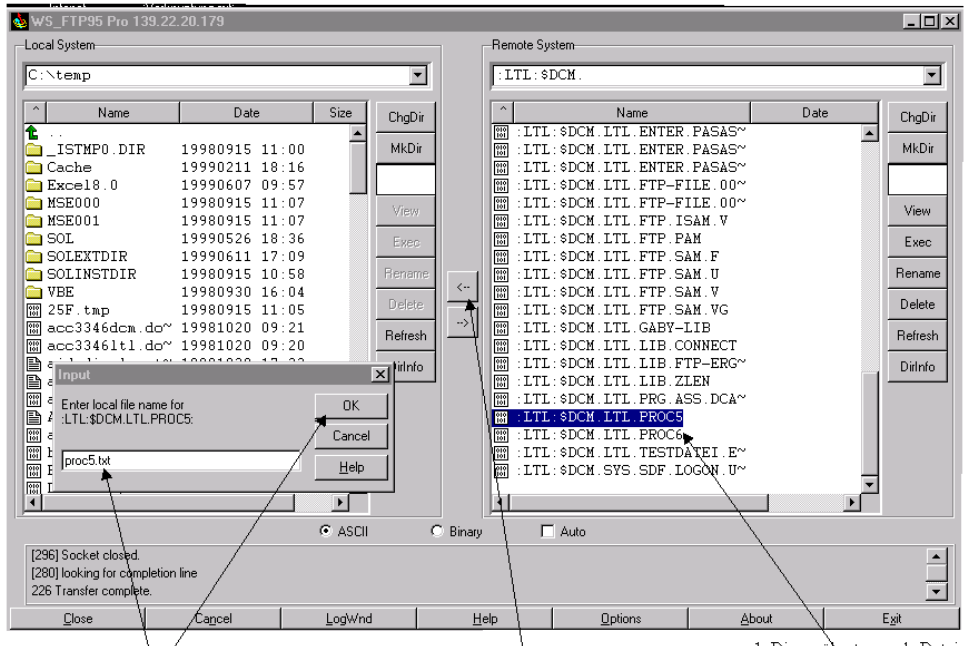
Batch-Fähigkeit des FTP-Client

Der FTP-Client kann in (Batch-)Prozeduren eingesetzt werden. Die Batch-Fähigkeit des FTP-Client ermöglicht es der Prozedur, festzustellen, bei welchem Kommando welcher Fehler aufgetreten ist. Hierzu werden das fehlerhafte Client-Kommando, ggf. das zugehörige fehlerhafte Server-Kommando und die zugehörigen Fehlermeldungen in einer zuvor spezifizierten Job-Variablen und/oder SDF-P-Variablen abgelegt, sodass die aufrufende Prozedur in geeigneter Weise reagieren kann. Die Batch-Fähigkeit wird unterstützt durch die FTP-Client-Kommandos *jobvar* (siehe Seite 141) und *svar* (siehe Seite 201).

Beispiel einer FTP-Session zwischen einem PC und BS2000/OSD



Mit „Connect“ wird die Verbindung zu BS2000/OSD aufgebaut.



3. Der den Windows-Regeln entsprechende Dateiname wird angegeben und mit „OK“ markiert. Die Datei wird in das gewünschte Verzeichnis übertragen.

2. Durch Markieren „<-“ und gleichzeitiges Drücken der „Shift“-Taste wird das Input“-Fenster geöffnet.

1. Die zu übertragende Datei wird markiert

4.5 FTP-Client in POSIX

Wenn der FTP-Client unter POSIX abläuft, gibt er die folgende Startmeldung aus:

```
POSIX-FTP <vers> <date> <time>
```

<date> und <time> spezifizieren hier Datum und Uhrzeit zum Zeitpunkt der Übersetzung des Main-Moduls. Zum Zeitpunkt des Starts wechselt der FTP-Client in das lokale POSIX-Dateiverzeichnis.

Dem Batch-Betrieb in BS2000/OSD entspricht in POSIX der FTP-Aufruf mit Schalter -n.

Beispiel

Es sollen Informationen über ferne Dateien in eine Datei dir.erg geschrieben werden:

```
→ ftp -n <remote host> >>dir.erg <<END
   user <kennung> <password> <account>
   dir
   close
   bye
   END
```

TLS/SSL-Unterstützung des FTP-Client in POSIX

Die TLS-Unterstützung des FTP-Client erfolgt analog zur TLS-Unterstützung des FTP-Client in BS2000/OSD (siehe Abschnitt „TLS/SSL-Unterstützung im FTP-Client“ auf Seite 88).

Die private Option-Datei hat den Dateinamen $HOME/.ftp.options. Die Dateinamen beziehen sich auf das POSIX-Dateisystem. Wenn Sie eine BS2000/OSD-Datei als CertificateFile verwenden wollen, müssen Sie deshalb dem Dateinamen den Präfix „/BS2/“ voranstellen.

4.6 TLS/SSL-Unterstützung im FTP-Client

interNet Services unterstützt die Absicherung der Kontroll- und – optional – der Datenverbindung mithilfe des TLS- bzw. SSL-Protokolls.

Die TLS/SSL-Unterstützung wird durch die folgenden Vorkehrungen realisiert:

- Option-Datei

  Die TLS/SSL-Unterstützung bietet ein breites Spektrum an Einstellmöglichkeiten. Diese Einstellungen können via Options in einer oder mehreren Option-Dateien hinterlegt werden (siehe Abschnitt „Parametereinstellung mithilfe von Option-Dateien“ auf Seite 89).

- FTP-Client-Kommandos für die TLS/SSL-Unterstützung

  Kommandos für die Aktivierung/Deaktivierung der TLS/SSL-Unterstützung, für das Lesen der Option-Datei u.a. (siehe Abschnitte „Kommandos zur TLS/SSL-Unterstützung“ auf Seite 110 und „Kommandoübersicht (FTP-Client)“ auf Seite 112).

- erweiterte Status-Informationen

  Die Ausgabe des *status*-Kommandos enthält TLS-spezifische Informationen (siehe Abschnitt „status - Ausgabe von FTP-Status-Informationen“ auf Seite 197).

Im Folgenden wird für „TLS/SSL“ abkürzend nur noch „TLS“ geschrieben.

4.7 Parametereinstellung mithilfe von Option-Dateien

Insbesondere die für die TLS-Nutzung erforderlichen Einstellungen werden in einer oder mehreren Option-Dateien hinterlegt. Die Option-Datei wird beim Start des FTP-Client eingelesen. Sie können jedoch auch zu einem späteren Zeitpunkt die TLS-Einstellungen den aktuellen Erfordernissen anpassen, indem Sie mit dem FTP-Client-Kommando *readopt* (siehe Seite 179) eine Option-Datei einlesen.

Ermittlung der relevanten Option-Datei durch den FTP-Client

Bei der Ermittlung der relevanten Option-Dateien verfährt der FTP-Client wie folgt:

1. Zunächst sucht der FTP-Client nach einer zentral abgelegten Option-Datei mit dem folgenden Standarddateinamen: $.SYSDAT.TCP-IP-AP.052.F-GOPT

   Bei der Installation mit IMON können Sie durch Umdefinieren der Logical-Id SYSDAT.F-GOPT auf eine andere Datei verweisen, die der FTP-Client dann als Option-Datei verwendet.

2. Unabhängig von der unter 1.) genannten zentralen Datei sucht der FTP-Client zusätzlich eine anwendereigene Option-Datei unter dem Namen SYSDAT.TCP-IP-AP.052.FTP.OPT. Sofern diese Datei existiert, liest der FTP-Client aus ihr die Options.

Falls eine Option in beiden Dateien mit unterschiedlichen Werten vorkommt, gilt der in der anwendereigenen Datei definierte Wert.

Falls Einstellungen der TLS-Absicherung via FTP-Client-Kommando vorgenommen werden, haben diese Vorrang vor den in den Option-Dateien spezifizierten Angaben.

Zu den Besonderheiten bei der TLS-Unterstützung des FTP-Client in POSIX siehe Seite 87.

Notation der Options in der Option-Datei

Die einzelnen Options müssen gemäß den folgenden Regeln in die Option-Datei eingetragen sein:

- Jede Option muss in einer eigenen Zeile stehen.
- Erstrecken sich die Argumente einer Option über mehrere Zeilen, dann muss jede fortzusetzende Zeile mit dem Fortsetzungszeichen „\“ abgeschlossen werden.
- Eine Zeile, die mit dem Zeichen „#“ in Spalte 1 beginnt, wird beim Einlesen ignoriert.
- Bei den Option-Namen wird Klein-/Groß-Schreibung nicht unterschieden.

Zeitpunkt, zu dem Options bzw. Option-Änderungen wirksam werden

Nach dem Start des FTP-Client können Sie mit dem Client-Kommando *readopt* (siehe Seite 179) die gewünschte Option-Datei einlesen. Diesen Vorgang können Sie beliebig oft wiederholen.

Die einzelnen Options bzw. Option-Änderungen werden zu unterschiedlichen Zeitpunkten wirksam:

- Bei folgenden Options werden Änderungen mit dem nächsten Aufbau einer Kontroll-Verbindung wirksam:

  -protect
  -private
  -tlsCipherSuite
  -tlsKeyFile
  -tlsCertificateFile
  -tlsProtocol
  -tlsCACertificateFile
  -tlsCARevocationFile
  -tlsVerifyServer
  -tlsVerifyDepth
  -tlsRandomSeed
  -tlsUseCryptoHardware

- Folgende Option wird während einer FTP-Sitzung nur einmal ausgewertet, und zwar zum Zeitpunkt des Ladens der OpenSSL-Bibliothek (Meldung: `[Loading OpenSSL library...]`):

  -tlsOpenSSLlibName

  Wenn Sie für diese Options nach der TLS-Initialisierung eine Verbindung mit anderen Werten dieser Options aufbauen wollen, gehen Sie wie folgt vor:

  1. FTP-Client beenden
  2. ggf. Standard-Option-Datei anpassen
  3. FTP-Client neu starten
  4. vor einer via *protect-*, *private-* oder *open*-Kommando angestoßenen TLS-Initialisierung:

     ggf. mit *readopt*-Kommando eine passende Option-Datei einlesen

  Die Schritte 2.) und 4.) sind alternativ auszuführen.

Beschreibung der Options

Nachfolgend sind die einzelnen Options beschrieben.

-transferType - Übertragungstyp einstellen

Mit der Option *-transferType* wird ein Übertragungstyp (vor)eingestellt (siehe auch Kommando „type - Übertragungstyp ändern oder abfragen“ auf Seite 206).

Im Gegensatz zum *type*-Kommando ist die Option *-transferType* nur lokal wirksam und wird erst aktiv, wenn eine neue Verbindung aufgebaut wird. Wenn hingegen die Option-Datei bei einer bestehenden FTP-Verbindung mit dem *readopt*-Kommando eingelesen wird, ist diese Verbindung von der Option *-transferTyp* nicht betroffen.

-transferType
ascii | **binary** | **ebcdic** | **tenex**

ascii
: Stellt den Übertragungstyp ASCII ein. Dieser Übertragungstyp sollte immer für Textdateien gewählt werden.
ascii ist Voreinstellung.

binary
: Stellt den Übertragungstyp BINARY ein. Dieser Übertragungstyp sollte immer bei Binärdateien gewählt werden.

ebcdic
: Stellt den Übertragungstyp EBCDIC ein. Dieser Übertragungstyp sollte gewählt werden, wenn beide Partner mit EBCDIC arbeiten, d.h. keine Codeumsetzung gewünscht ist und im BS2000/OSD eine SAM-Datei erzeugt werden soll.

tenex
: Entspricht dem Übertragungstyp BINARY.

-initialCommand - FTP-Client-Kommando spezifizieren

Mit der Option *-initialCommand* wird ein FTP-Client-Kommando spezifiziert, das direkt nach dem Start des FTP-Client ausgeführt wird. Es können nur solche Kommandos sinnvoll verwendet werden, die für eine erfolgreiche Ausführung keine Verbindung zu einem FTP-Server voraussetzen. Die Option *-initialCommand* kann wiederholt angegeben werden, um mehrere FTP-Client-Kommandos zu spezifizieren.

Das spezifizierte Kommando wird erst nach dem Einlesen beider Option-Dateien (zentrale und lokale) ausgeführt. Enthalten die Option-Dateien mehrere *-initialCommand*-Options, dann wird die Ausführungsreihenfolge der Kommandos durch die Reihenfolge festgelegt, in der die zugehörigen Options in den Option-Dateien abgelegt sind.

-initialCommand
<ftp-client-cmd>

<ftp-client-cmd>
FTP-Client-Kommando, das direkt nach dem Start des FTP-Client ausgeführt werden soll.

-protect - TLS-Absicherung für Kontrollverbindungen

Mit der Option *-protect* wird festgelegt, dass alle nachfolgend aufgebauten Kontrollverbindungen mit TLS abgesichert werden. Wenn keine Option *-protect* spezifiziert ist, wird die TLS-Absicherung nicht durchgeführt.

Auf eine bereits bestehende Kontrollverbindung hat die Option *-protect* keinen Einfluss.

-protect

-private - TLS-Absicherung für Kontroll- und Datenverbindungen

Mit der Option *-private* wird festgelegt, dass alle nachfolgend aufgebauten Kontroll- und Datenverbindungen mit TLS abgesichert werden. Auf eine bereits bestehende Kontrollverbindung hat die Option *-private* keinen Einfluss. Falls aber die bestehende Kontrollverbindung mit TLS abgesichert ist, werden alle nachfolgend aufgebauten Datenverbindungen ebenfalls mit TLS abgesichert.

-private

-tlsRandomSeed - Pseudo-Zufallszahlengenerator initialisieren

Mit der Option *-tlsRandomSeed* wird spezifiziert, wie der von TLS verwendete Pseudo-Zufallszahlengenerator initialisiert wird. Eine gute Initialisierung mit möglichst zufälligen, nicht vorhersagbaren Werten ist für die TLS-Absicherung von entscheidender Bedeutung. Falls auf dem Rechner, auf dem der FTP-Client abläuft, das BS2000/OSD-Subsystem PRNGD (**P**seudo **R**andom **N**umber **G**enerator **D**emon) aktiv ist, wird PRNGD für die Initialisierung verwendet, so dass die Einstellung der Option *-tlsRandomSeed* praktisch ohne Bedeutung ist. Das Subsystem PRNGD ist beschrieben im Handbuch „interNet Services Administratorhandbuch“.

-tlsRandomSeed
PROGRAM | **USER**

PROGRAM
Es werden programm-interne Funktionen verwendet. Diese nutzen vor allem die Schwankungen der Echtzeituhr in Relation zum Taktgeber der Rechner-CPU, um Zufallszahlen für die Initialisierung zu generieren.

USER
Ein Teil der Initialisierung erfolgt analog der Initialisierung bei der Angabe PROGRAM. Zusätzlich wird der Anwender wiederholt aufgefordert, Zeichen in möglichst zufälliger Auswahl via Tastatur einzugeben und/oder die ENTER-Taste zu betätigen. Für die Initialisierung wird zum einen der Zeitstempel der Eingabe verwendet. Darüber hinaus werden die eingegebenen Zeichen ebenfalls für die Initialisierung verwendet. Da jedoch die Zufälligkeit der eingegebenen Zeichen nicht bekannt ist und diese Zeichen außerdem in vielen Fällen abgehört werden können, werden sie nicht berücksichtigt bei der Abschätzung, ob schon genügend Initialisierungsmaterial vorliegt. In diese Abschätzung geht nur die Anzahl der Betätigungen der ENTER-Taste ein.
USER ist Voreinstellung.

Wenn der FTP-Client im Batch-Modus betrieben wird, empfiehlt sich in der Regel die Einstellung PROGRAM, da im Batch-Modus kein Anwender für die Eingabe von zufälligen Zeichen zur Verfügung steht.

-tlsProtocol - TLS/SSL-Protokoll-Auswahl

OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-Protokoll in der Version 1. Mit der Option *-tlsProtocol* können einige dieser Protokolle selektiv aktiviert werden.

-tlsProtocol
[**+** | **-**] {**SSLv2** | **SSLv3** | **TLSv1** | **All** } ...

+
Das nachfolgend spezifizierte Protokoll ist zugelassen.

Wenn weder „+“ noch „-“ angegeben werden, hat dies dieselbe Wirkung wie die Angabe von „+“.

-
Das nachfolgend spezifizierte Protokoll ist nicht zugelassen.

SSLv2
SSL-Protokoll der Version 2

Das SSL-Protokoll in der Version 2 weist einige sicherheitstechnische Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

SSLv3
SSL-Protokoll der Version 3

TLSv1
TLS-Protokoll der Version 1

ALL
Alle Protokolle sollen aktiviert werden.

All -SSLv2 ist Voreinstellung,

Beispiel

Die Angaben `-tlsProtocol SSLv3 TLSv1` und `-tlsProtocol All -SSLv2` haben dieselbe Wirkung.

-tlsCipherSuite - Spezifikation einer Verschlüsselungsverfahren-Vorzugsliste

Mit der Option *-tlsCipherSuite* wird eine Verschlüsselungsverfahren-Vorzugsliste spezifiziert. Falls diese Option nicht angegeben wird, wird eine voreingestellte Vorzugsliste verwendet.

-tlsCipherSuite
<spezifikation>

<spezifikation>
Spezifikation einer Verschlüsselungsverfahren-Vorzugsliste (Näheres siehe nachfolgende Beschreibung des Aufbaus einer Spezifikation).
ALL: !EXP: !ADH ist Voreinstellung.

Beschreibung des Aufbaus einer Spezifikation

Die Spezifikation besteht aus einem oder mehreren Chiffre-Mnemonics, die durch einen Doppelpunkt (:) getrennt sind.

Ein Chiffre-Mnemonic kann folgende Formen annehmen:

- Ein Chiffre-Mnemonic kann aus einer einzelnen Verschlüsselungs-Suite wie z.B. DES-CBC-SHA bestehen.
- Ein Chiffre-Mnemonic kann repräsentieren:
  - Liste von Verschlüsselungs-Suites, die einen bestimmten Algorithmus enthalten
  - Verschlüsselungs-Suites eines bestimmten Typs

  Beispielsweise repräsentiert SHA1 alle Verschlüsselungs-Suiten, die den Digest-Algorithmus SHA1 benutzen und SSLv3 repräsentiert alle SSL-Version 3-Algorithmen.
- Listen von Verschlüsselungs-Suiten können mithilfe des „+“-Zeichens zu einem einzelnen Chiffre-Mnemonic kombiniert werden. Dies wird dann als logische UND-Operation interpretiert. So repräsentiert SHA1+DES alle Verschlüsselungs-Suiten, die die SHA1- und DES-Algorithmen enthalten.

- Jedem Chiffre-Mnemonic kann optional eines der Zeichen „!“, „-“ oder „+“ vorangestellt werden:
  - Bei Voranstellen von „!“ werden die betreffenden Verschlüsselungs-Suiten dauerhaft aus der Vorzugsliste gelöscht. Sie erscheinen auch dann nicht wieder in der Vorzugsliste, wenn sie explizit angegeben werden.
  - Bei Voranstellen von „-“ werden die betreffenden Verschlüsselungs-Suiten aus der Vorzugsliste gelöscht, aber einige von ihnen oder alle können durch nachfolgende Optionen wieder hinzugefügt werden.
  - Bei Voranstellen von „+“ werden die betreffenden Verschlüsselungs-Suiten an das Ende der Vorzugsliste verschoben. Hiermit werden keine Verschlüsselungs-Suiten zur Vorzugsliste hinzugefügt, sondern nur existierende verschoben.
  - Wenn keines der drei Zeichen „!“, „-“ oder „+“ vorangestellt ist, wird der Chiffre-Mnemonic als eine Liste von Verschlüsselungs-Suiten interpretiert, die an die aktuelle Vorzugsliste angehängt wird. Wenn dies eine Verschlüsselungs-Suite einschließt, die schon in der aktuellen Vorzugsliste enthalten ist, dann wird diese ignoriert, sie wird nicht an das Ende der Vorzugsliste verschoben.
- Der Chiffre-Mnemonic @STRENGTH kann an beliebiger Stelle eingefügt werden, um die aktuelle Vorzugsliste nach der Länge der Verschlüsselungsschlüssel zu sortieren.

Zulässige Chiffre-Mnemonics

Nachfolgend sind die zulässigen Chiffre-Mnemonics beschrieben.

ALL
: Alle Verschlüsselungs-Suiten mit Ausnahme der eNULL-Chiffren. Letztere müssen explizit aktiviert werden.

HIGH
: Verschlüsselungs-Suiten mit Schlüssellängen größer 128 Bit. Da 3DES mit 168 Bit Länge (anstatt 112 Bit wie von manchen Kryptologen) bewertet wird, zählt es zu dieser Suiten-Klasse.

MEDIUM
: Verschlüsselungs-Suiten mit Schlüssellänge 128 Bit.

LOW
: Verschlüsselungs-Suiten mit 64 oder 56 Bit Schlüssellänge, ausgenommen Export-Verschlüsselungs-Suiten.

EXP, EXPORT
: Export-Verschlüsselungs-Algorithmen einschließlich 40- und 56-Bit-Algorithmen.

EXPORT40
: 40-Bit-Export-Verschlüsselungs-Algorithmen.

EXPORT56
: 56-Bit-Export-Verschlüsselungs-Algorithmen.

eNULL, NULL
: „NULL“-Verschlüsselungs-Algorithmen, d.h. solche ohne Verschlüsselung. Da diese keine Verschlüsselung bieten und damit ein Sicherheitsrisiko sind, werden sie standardmäßig deaktiviert und müssen gegebenenfalls explizit angegeben werden.

aNULL
: Verschlüsselungs-Suiten ohne Authentifizierung. Dies sind im Augenblick die anonymen Diffie-Hellman-Algorithmen. Diese Algorithmen sind anfällig für „man in the middle“-Angriffe, so dass von ihrer Benutzung abgeraten wird.

kRSA, RSA
: Verschlüsselungs-Suiten mit RSA-Schlüsselaustausch.

kEDH
: Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüsselvereinbarung.

aRSA
: Verschlüsselungs-Suiten mit RSA-Authentifizierung, d.h. die Zertifikate enthalten RSA-Schlüssel.

aDSS, DSS
: Verschlüsselungs-Suiten mit DSS-Authentifizierung, d.h. die Zertifikate enthalten DSS-Schlüssel.

TLSv1, SSLv3, SSLv2
: TLSv1-, SSLv3- bzw. SSLv2-Verschlüsselungs-Suiten. Die TLSv1-Suiten und die SSLv3-Suiten sind identisch.

DH
: Verschlüsselungs-Suiten mit Diffie-Hellman-Schlüsselaustausch, einschließlich anonymem Austausch.

ADH
: Verschlüsselungs-Suiten mit anonymem Diffie-Hellman-Schlüsselaustausch.

AES
Verschlüsselungs-Suiten mit AES-Verschlüsselung (128 und 256 Bit Schlüssellänge)

3DES
Verschlüsselungs-Suiten mit Triple-DES-Verschlüsselung.

DES
Verschlüsselungs-Suiten mit DES-Verschlüsselung (kein Triple-DES).

RC4
Verschlüsselungs-Suiten mit RC4-Verschlüsselung.

RC2
Verschlüsselungs-Suiten mit RC2-Verschlüsselung.

MD5
Verschlüsselungs-Suiten mit MD5-Hash-Funktion.

SHA1, SHA
Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.

In der nachfolgenden Tabelle sind die verfügbaren Verschlüsselungs-Suiten zusammengefasst.

Name	Version	Schlüssel-Austausch	Authentifizierung	Veschlüsselung	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	

Verfügbare Verschlüsselungs-Suiten

Name	Version	Schlüssel-Austausch	Authentifi-zierung	Veschlüsse-lung	Digest	Export
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	keine	AES(256)	SHA1	
ADH-AES128-SHA	SSLv3	DH	keine	AES(128)	SHA1	
ADH-DES-CBC3-SHA	SSLv3	DH	keine	3DES(168)	SHA1	
ADH-DES-CBC-SHA	SSLv3	DH	keine	DES(56)	SHA1	
ADH-RC4-MD5	SSLv3	DH	keine	RC4(128)	MD5	
EXP-ADH-DES-CBC-SHA	SSLv3	DH(512)	keine	DES(40)	SHA1	export
EXP-ADH-RC4-MD5	SSLv3	DH(512)	keine	RC4(40)	MD5	export
NULL-SHA	SSLv3	RSA	RSA	keine	SHA1	
NULL-MD5	SSLv3	RSA	RSA	keine	MD5	

Verfügbare Verschlüsselungs-Suiten

-tlsCertificateFile - Datei mit X.509-Client-Zertifikat im PEM-Format

Mit der Option *-tlsCertificateFile* wird eine Datei spezifiziert, die das X.509-Client-Zertifikat zur Client-Authentifizierung im PEM-Format enthält. Diese Datei kann auch den privaten Schlüssel des Client enthalten. In der Regel werden aber Zertifikat und Schlüssel in getrennten Dateien abgelegt. In diesem Fall wird die Schlüsseldatei mithilfe der Option *-tlsKeyFile* (siehe Seite 103) spezifiziert.

-tlsCertificateFile
<dateiname 1 .. 54> | ***NONE**

<dateiname 1 .. 54>
Name der Datei, die das X.509-Client-Zertifikat im PEM-Format enthält.

***NONE**
Es wird kein Client-Zertifikat (und somit auch keine Client-Authentifizierung) verwendet. *NONE ist Voreinstellung.

-tlsKeyFile - Datei mit Client-Schlüssel im PEM-Format spezifizieren

Mit der Option *-tlsKeyFile* wird eine Datei spezifiziert, die den privaten Client-Schlüssel im PEM-Format enthält.

Wenn sowohl X.509-Client-Zertifikat als auch privater Client-Schlüssel innerhalb derselben Datei enthalten sind (siehe Option *-tlsCertificateFile* auf Seite 102), braucht die Option *-tlsKeyFile* nicht angegeben zu werden.

Wenn der Client-Schlüssel mit einer Passphrase geschützt ist, dann muss diese Passphrase nach dem Start des FTP-Client beim ersten Aufbau einer mit TLS gesicherten FTP-Verbindung eingegeben werden.

-tlsKeyFile
<dateiname 1 .. 54> | ***NONE**

<dateiname 1 .. 54>
Name der Datei, die den privaten Client-Schlüssel enthält.

***NONE**
Es wird keine separate Client-Schlüssel-Datei verwendet.

Voreinstellung ist der in der Option *-tlsCertificateFile* (siehe Seite 102) spezifizierte Dateiname.

-tlsCACertificateFile - Datei mit Server-Authentifizierung spezifizieren

Mit der Option *-tlsCACertificateFile* wird eine Datei spezifiziert, die die für die Authentifizierung des FTP-Servers erforderlichen CA-Zertifikate im PEM-Format enthält. Die einzelnen PEM-Zertifikate sind sequenziell in der Datei angeordnet.

Zum Hinzufügen oder Löschen von Zertifikaten können Sie die Datei mit einem beliebigen Text-Editor bearbeiten. Die einzelnen Zertifikate sind wie folgt in der Datei notiert:

```
-----BEGIN CERTIFICATE-----
< CA-Zertifikat in Base64-Codierung >
-----END CERTIFICATE-----
```

Text außerhalb dieser Sequenzen wird vom FTP-Client ignoriert und kann deshalb zum Kennzeichnen der Zertifikate verwendet werden, die wegen der ASN.1/Base64-Codierung in nicht lesbarer Form vorliegen.

-tlsCACertificateFile
<dateiname 1 .. 54> | ***NONE**

<dateiname 1 .. 54>
Name der Datei, die die für die Authentifizierung des FTP-Servers erforderlichen Zertifikate im PEM-Format enthält.

***NONE**
Es wird keine Datei mit CA-Zertifikaten angegeben.
*NONE ist Voreinstellung.

-tlsCARevocationFile - Datei mit CRL spezifizieren

Mit der Option *-tlsCARevocationFile* wird eine Datei spezifiziert, die die CRLs (Certificate Revocation List) der Zertifizierungsinstanzen (Certificate Authority, CA) enthält. (Zertifikate, die von von einer Zertifizierungsinstanz herausgegeben wurden, können durch Veröffentlichung einer so genannten Certificate Revocation List (CRL) für ungültig erklärt werden.)

-tlsCARevocationFile
<dateiname 1 .. 54> | ***NONE**

<dateiname 1 .. 54>
Name der Datei, die die CRLs der Zertifizierungsinstanzen enthält.

***NONE**
Es wird keine Datei mit CRLs angegeben.
*NONE ist Voreinstellung.

-tlsVerifyServer - FTP-Server-Zertifikat verifizieren (ja/nein)

Mit der Option *-tlsVerifyServer* wird festgelegt, ob das FTP-Server-Zertifikat verifiziert werden soll.

-tlsVerifyServer
YES | **NO**

YES
Das FTP-Server-Zertifikat soll verifiziert werden.
YES ist Voreinstellung.

NO
Das FTP-Server-Zertifikat soll nicht verifiziert werden.
Mit dieser Einstellung wird man anfällig für „man in the middle“-Angriffe.

-tlsVerifyDepth - Verifizierungstiefe festlegen

Mit der Option *-tlsVerifyDepth* wird die so genannte Verifizierungstiefe festgelegt, d.h. die Anzahl der maximal zulässigen Zertifikate zwischen dem FTP-Server-Zertifikat und dem Zertifikat, das dem FTP-Client bekannt ist.

Im Einzelnen ist zu beachten:

- Wird für die maximale Tiefe der Wert 1 spezifiziert (Default), dann muss das Server-Zertifikat direkt von einer dem FTP-Client bekannten Certificate Authority (CA) signiert worden sein, damit es akzeptiert wird.
- Wird die maximale Tiefe überschritten, dann wird die Verbindung abgebrochen, sofern nicht aufgrund von *-tlsVerifyServer NO* (siehe Seite 106) die Verifizierung des FTP-Server-Zertifikats ausgeschaltet ist.
- Die Spezifikation der Tiefe 0 ist nicht sinnvoll. In diesem Fall wären nur selbstsignierte Zertifikate zulässig.

-tlsVerifyDepth
<tiefe>

<tiefe>
Anzahl der maximal zulässigen Zertifikate zwischen dem FTP-Server-Zertifikat und dem Zertifikat, das dem FTP-Client bekannt ist (einschließlich des FTP-Server-Zertifikats).
Voreinstellung: 1

-tlsUseCryptoHardware - Krypto-Hardware verwenden (ja/nein)

Mit der Option *-tlsUseCryptoHardware* wird festgelegt, ob zur Berechnung kryptographischer Algorithmen Krypto-Hardware, z.B. eine openCRYPT^{TM}-BOX (siehe Handbuch „openCRYPT“), verwendet werden soll.

-tlsUseCryptoHardware
YES | **NO**

YES
Es wird Krypto-Hardware verwendet.

NO
Es wird keine Krypto-Hardware verwendet.
NO ist Voreinstellung.

-tlsOpenSSLlibName - LMS-Datei für OpenSSL-Bibliothek festlegen

Mit der Option *-tlsOpenSSLlibName* wird festgelegt, aus welcher LMS-Datei die OpenSSL-Bibliothek nachgeladen werden soll. Eine vom Standardnamen abweichende Angabe kann z.B. erforderlich sein, wenn die OpenSSL-Bibliothek auch von anderen Produkten verwendet wird.

Das Nachladen der OpenSSL-Bibliothek lässt sich durch Cache-Speichern mithilfe von DAB beschleunigen. Bei gemeinsamer Verwendung einer OpenSSL-Bibliothek durch mehrere Produkte wird die Größe des verwendeten DAB-Puffers verringert.

-tlsOpenSSLlibName
<openssl-libname>

<openssl-libname>
Name der LMS-Datei, aus der die OpenSSL-Bibliothek nachgeladen werden soll.
Voreinstellung: $.SYSLNK.TCP-IP-AP.052

4.8 Kommandos zur TLS/SSL-Unterstützung

Für die TLS-Unterstützung sind eine Reihe von FTP-Client-Kommandos von Bedeutung. In der nachfolgenden Tabelle sind diese Kommandos zusammen mit einer Kurzbeschreibung ihrer Funktionalität aufgelistet. Ausführlich beschrieben sind diese Kommandos im Abschnitt „Kommandoübersicht (FTP-Client)“ auf Seite 112.

Kommando	Funktionalität	Seite
open	Bei eingeschalteter TLS-Absicherung der Kontrollverbindung: TLS-abgesicherte Verbindung zu einem fernen Rechner eröffnen.	164
private	TLS-Absicherung der Datenverbindung ein-/ausschalten.	169
protect	TLS-Absicherung der Kontrollverbindung ein-/ausschalten.	171
readopt	Option-Datei einlesen	179
status	Anzeigen, ob – die Datenverbindungen mit TLS abgesichert werden, – eine Kontrollverbindung besteht, ob diese verschlüsselt ist und ggf. mit welchem Algorithmus verschlüsselt wird.	197

FTP-Client-Kommandos im Zusammenhang mit der TLS-Unterstützung

Die via FTP-Client-Kommando vorgenommenen Einstellungen zur TLS/SSL-Sicherung haben Vorrang vor den in den Options spezifizierten Angaben.

4.9 Bandverarbeitung mit FTP

Bei der Bandverarbeitung mit FTP sind die folgenden Formate, Einstellungen und Modi möglich:

- alle Dateiformate: SAM-V, SAM-F, SAM-U, PAM
- alle Einstellungen zu BLKCTRL, BLKSIZE, RECSIZE
- alle Übertragungsmodi ASCII, EBCDIC, BINARY, ftyp text, textbin, binary

Die Bandverarbeitung wird durch ein BS2000/OSD-FILE-Kommando gesteuert (siehe Handbuch „Benutzerkommandos (SDF-Format)“.

Bei einer Remote-Banddatei müssen Sie das FILE-Kommando mit vorangestelltem *quote site ...* absetzen.

Beispiele

1. Eine SAM-Datei soll auf der MBK SADW48 vom Typ T-C2 angelegt werden. Auf dem Datenträger befindet sich schon eine Datei (-> FSEQ=2, da die gewünschte Datei die zweite auf dem Band ist):

   ```
   FILE <DATEI>,FCBTYP=SAM,RECFORM=F,RECSIZE=300,FSEQ=2,VOL=SADW48,DEV=T-C2
   ```

2. Die Datei soll auf den zwei leeren MBKs SADW48,SADW49 angelegt werden:

   ```
   FILE <DATEI>,FCBTYP=SAM,RECFORM=FIX,RECSIZE=300,VOL=(SADW48,SADW49),DEV=T-C2
   ```

4.10 Kommandoübersicht (FTP-Client)

FTP-Kommandos können bis zur Eindeutigkeit abgekürzt werden. Operanden werden durch Leerzeichen getrennt.

Kommen in Kommandos Dateinamen als Operanden vor, so können sie in voller Länge angegeben werden (vollständiger Pfadname). Wird nur ein relativer Dateiname angegeben, so wird der Dateiname durch das im Augenblick aktuelle Arbeits-Dateiverzeichnis ergänzt.

Im folgenden Abschnitt sind die Kommandos in alphabetischer Reihenfolge beschrieben.

append - Anhängen einer lokalen an eine ferne Datei

Mit dem Kommando *append* wird eine Datei vom lokalen Rechner zum fernen Rechner übertragen und dort an eine eventuell bereits bestehende Datei angehängt. Bei DVS-Dateien mit dem Dateityp SAM kann durch das Kommando *ftyp* der Bearbeitungstyp beeinflusst werden.

append
<lokale-datei> [<ferne-datei>]

<lokale-datei>
: Name einer POSIX- oder DVS-Datei am lokalen Rechner, die zum fernen Rechner übertragen werden soll. Metazeichen sind nicht erlaubt.

<ferne-datei>
: Name einer Datei am fernen Rechner (Metazeichen sind nicht erlaubt). Existiert die Datei bereits, wird die lokale Datei daran angehängt. Existiert die Datei noch nicht, wird eine neue Datei angelegt.
Fehlt der Operand *ferne-datei*, wird der Name der lokalen Datei verwendet (in diesem Fall muss der Name der lokalen Datei auch den Dateinamenskonventionen des fernen Rechners entsprechen). Großbuchstaben im Namen der lokalen Datei werden in Kleinbuchstaben umgewandelt. Ist der ferne Rechner ein BS2000/OSD-Rechner, so setzt der Server die Kleinbuchstaben wieder in Großbuchstaben um (siehe auch FTP-Client-Kommando *setcase* auf Seite 193).

Beispiel

Die Datei *:5:$TCPTEST.MAN.FTP.C* wird vom lokalen Rechner auf den fernen BS2000/OSD-Rechner übertragen und dort an die Datei *:110:$TSOS.FTP.2* angehängt. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.

→ pwd
   257 ":110:$TSOS." is current directory.

→ append ftp.c ftp.2
   200 PORT command successful.
   ...
```

ascii - Übertragungstyp ASCII einschalten

Das Kommando *ascii* stellt den Übertragungstyp ASCII ein. Dieser Übertragungstyp sollte immer dann gewählt werden, wenn Textdateien zu übertragen sind. Nach dem Start von FTP ist der Übertragungstyp ASCII voreingestellt.

Der Übertragungstyp kann auch mit den Kommandos *binary*, *tenex* und *type* geändert werden. Mit den Kommandos *status* und *type* kann der aktuell eingestellte Übertragungstyp ermittelt werden. Der Übertragungstyp ist genauer auf Seite 81 beschrieben.

ascii

Beispiel

1. Der aktuelle Übertragungstyp wird abgefragt.

```
→ type
   Using binary mode to transfer files.
```

2. Der Übertragungstyp wird von BINARY auf ASCII gesetzt.

```
→ ascii
   200 Type set to A.
```

3.

```
→ status
Connected to anlaged, port 21.
No proxy connection.
Passive Mode: off
Mode: stream; Type: ascii; Form: non-print; Structure: file
Copymode: off; Ftyp: textbin
Verbose: on; Bell: off; Prompting: on; Globbing: on
Filesystem is: BS2000
The character to change the filesystem is: %
ISO-codetable is ISO88591, EBCDIC-codetable is EDF041
Time limit for server responses: 30 secs
Time limit for file size determination: 60 secs
Store unique: off; Receive unique: off
Use special EOF marker (C-DATEIENDE): on
Pad empty record with blank: off
Case sensitivity: OFF
Used job variable: *NONE
Used SDF-P variable: *NONE
Used receive selector: *NONE
Used send selector: *NONE
Hash mark printing: off; Use of PORT cmds: on
Protected control channel: off
Private data channel: off
Cipher: clear
```

bell - Klingelzeichen ein-/ausschalten

Das Kommando *bell* ist in BS2000/OSD wirkungslos und nur aus Kompatibilitätsgründen implementiert.

bell

binary - Übertragungstyp BINARY einschalten

Das Kommando *binary* stellt den Übertragungstyp BINARY ein. Dieser Übertragungstyp ist immer dann anzugeben, wenn Binärdateien übertragen werden sollen.
Nach dem Start von FTP ist der Übertragungstyp ASCII voreingestellt.

Der Übertragungstyp kann auch mit den Kommandos *binary*, *tenex* und *type* geändert werden. Der aktuell eingestellte Übertragungstyp kann mit den Kommandos *status* und *type* ermittelt werden. Der Übertragungstyp ist auf Seite 81 beschrieben.

binary

Beispiel

1. Die aktuelle Übertragungsart wird abgefragt.

```
→ type
   Using ascii mode to transfer files.
```

2. Die Übertragungsart wird von ASCII auf BINARY geändert.

```
→ binary
   200 Type set to I.
```

3. Es wird eine PAM-Datei übertragen. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ put pamela
   200 PORT command successful.
   ...
```

4. Die neue Übertragungsart wird abgefragt.

```
→ type
   Using binary mode to transfer files.
```

bye - FTP beenden

Das Kommando *bye* beendet das Programm FTP. Falls noch eine Verbindung zu einem fernen Rechner existiert, wird diese geschlossen (impliziter *close*).
Das Kommando *quit* kann synonym zu *bye* eingegeben werden.

bye

Beispiel

```
→ bye
   221 Goodbye.
```

ccc - TLS-Absicherung der Kontrollverbindung ausschalten

Das Kommando *ccc* schaltet die TLS-Absicherung der Kontrollverbindung wieder aus. Hauptanwendungsgebiet hierfür sind FTP-Verbindungen, die über Firewalls und NAT-Geräte geführt werden, da diese Geräte bestimmte FTP-Kommandos (PORT, PASV, EPRT, EPSV) lesen bzw. verändern können müssen, um ihre Aufgaben zu erfüllen.

ccc

Mit dem *ccc*-Kommando können Sie die Verschlüsselung der Kontrollverbindung nach der Übermittlung von User-Id, Passwort und gegebenenfalls Account wieder aufheben, um nachfolgend Dateien beispielsweise über Firewalls hinweg zu transferieren.

Nach dem Absetzen des *ccc*-Kommandos werden die Kommandos für das Einstellen der Verschlüsselung der Datenverbindungen (PBSZ, PROT) abgewiesen, d.h. diesbezüglich bleibt der Status zum Zeitpunkt des Absetzen des *ccc*-Kommandos erhalten. Wenn beispielweise die Verschlüsselung der Datenverbindungen aktiviert war, dann werden auch weiterhin die Datenverbindungen verschlüsselt.

i Sie sollten das *ccc*-Kommando nur verwenden, wenn es unbedingt notwendig ist, da danach die Namen der angesprochenen Dateien und Verzeichnisse, die für sich genommen schon eine geheime Information darstellen können, wieder für einen Angreifer sichtbar werden. Auch bestimmte Angriffe auf einen FTP-Server, die durch TLS-Verwendung verhindert werden, sind bei einer Kontrollverbindung im Klartext-Modus wieder möglich.

cd - Wechseln des fernen Arbeits-Dateiverzeichnisses

Das Kommando *cd* ändert das aktuelle Arbeits-Dateiverzeichnis am fernen Rechner. Besteht eine Verbindung zu einem BS2000/OSD-FTP-Server, so kann durch das Kommando *cd*, wenn am fernen Rechner ein POSIX-Dateisystem vorhanden ist, zwischen DVS- und POSIX-Dateisystem gewechselt werden (siehe Seite 78). Das aktuell eingestellte ferne Arbeits-Dateiverzeichnis kann mit dem Kommando *pwd* ermittelt werden. Beim Wechsel des Dateisystems wird in das entsprechende HOME-Dateiverzeichnis gewechselt.

cd
.|**..**| <fernes-dateiverzeichnis> | **%BS2000** | **$HOME** | **%POSIX**

.
Das lokale Arbeits-Dateiverzeichnis wird nicht geändert (nur aus Kompatibilität zu anderen Implementierungen vorhanden).

..
Im DVS-Dateisystem wird die letzte (am weitesten rechts stehende) Teilqualifizierung aus dem Namen des lokalen Arbeits-Dateiverzeichnisses entfernt.
Im POSIX-Dateisystem wird in das übergeordnete Dateiverzeichnis gewechselt.

<fernes-dateiverzeichnis>
Name des neuen Arbeits-Dateiverzeichnisses

%BS2000
Es wird in das DVS-HOME-Dateiverzeichnis gewechselt. Wenn Sie sich bereits im DVS-Dateisystem befinden, ist die Angabe von %BS2000 mit der Angabe $HOME identisch.

$HOME
Es wird in das DVS-HOME-Dateiverzeichnis gewechselt. Voraussetzung für die Angabe von $HOME ist, dass Sie sich bereits in einem DVS-Dateiverzeichnis befinden.

%POSIX
Es wird in das POSIX-HOME-Dateiverzeichnis gewechselt.

Beispiel

Der ferne Rechner ist ein BS2000/OSD-Rechner mit POSIX-Dateiverzeichnis.

1. Abfragen des fernen Arbeits-Dateiverzeichnisses.

```
→ pwd
   257 "/home/usr/tcptest" is current directory.
```

2. Wechseln des Arbeits-Dateiverzeichnisses durch Anhängen des Suffixes *man/sam/nach.bs2000*.

```
→ cd man/sam/nach.bs2000
   250 "/home/usr/tcptest/man/sam/nach.bs2000" is current directory now.
```

3. Abfragen des neuen Arbeits-Dateiverzeichnisses.

```
→ pwd
   257 "/home/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

4. Wechseln in das DVS-Dateiverzeichnis.

```
→ cd %BS2000
   250 :4:$TEST is current directory now.
```

cdup - Wechseln ins nächsthöhere Dateiverzeichnis

Das Kommando *cdup* initiiert am fernen Rechner den Wechsel ins nächsthöhere Dateiverzeichnis. Alternativ kann dies auch durch die Angabe *cd* .. erreicht werden.

cdup

close - Schließen der Verbindung zum fernen Rechner

Mit dem Kommando *close* wird die Verbindung zu einem fernen Rechner geschlossen.

close

Beispiel

1. Die Abfrage des Status zeigt an, dass eine Verbindung zum Rechner *anlaged* besteht.

```
→ status
Connected to anlaged, port 21.
No proxy connection.
Passive Mode: off
Mode: stream; Type: ascii; Form: non-print; Structure: file
Copymode: off; Ftyp: textbin
Verbose: on; Bell: off; Prompting: on; Globbing: on
Filesystem is: BS2000
The character to change the filesystem is: %
ISO-codetable is ISO88591, EBCDIC-codetable is EDF041
Time limit for server responses: 30 secs
Time limit for file size determination: 60 secs
Store unique: off; Receive unique: off
Use special EOF marker (C-DATEIENDE): on
Pad empty record with blank: off
Case sensitivity: off
Used job variable: *NONE
Used SDF-P variable: *NONE
Used receive selector: *NONE
Used send selector: *NONE
Hash mark printing: off; Use of PORT cmds: on
Protected control channel: off
Private data channel: off
Cipher: clear
```

2. Durch *close* wird diese Verbindung geschlossen.

```
→ close
221 Goodbye.
```

copymode - 1:1-Übertragung von BS2000/OSD-Plattendateien ein-/ausschalten

Mit dem Kommando *copymode* könnnen Sie die 1:1-Übertragung von BS2000/OSD-Plattendateien ein- und ausschalten. Näheres zur 1:1-Übertragung finden Sie im Abschnitt „1:1-Übertragung von BS2000/OSD-Plattendateien“ auf Seite 72.

copymode
[**on** | **same** | **off**]

on
Schaltet 1:1-Übertragung ein (ohne Erhaltung der Dateischutz-Attribute). Von der Quelldatei übernommen werden inhalts- und strukturerhaltende Attribute, wie z.B. FCB-Typ, Blocklänge, Satzlänge, Satzformat etc.
Wenn kein Operand spezifiziert ist, entspricht dies der Angabe `copymode on`.

same
Schaltet 1:1-Übertragung ein (mit Erhaltung der Dateischutz-Attribute). Von der Quelldatei übernommen werden inhalts- und strukturerhaltende Attribute, wie z.B. FCB-Typ, Blocklänge, Satzlänge, Satzformat etc.
Mit Ausnahme von Passwörtern und GUARDS-Regeln werden die Dateischutzattribute (USER-ACCESS, ACCESS, BASIC-ACL, AUDIT, RETENTION-PERIOD) in die Zieldatei übernommen.

off
Schaltet 1:1-Übertragung aus.
off ist Voreinstellung, wenn keine *copymode*-Anweisung angegeben wird.

Wenn Sie *copymode* beim FTP-Server einstellen wollen, geben Sie *quote site cmod* an. Die Parameter entsprechen denen von *copymode*. Bei *quote site cmod* muss jedoch immer einer der Parameter on, same oder off angegeben werden.

Beispiel

1. Übertragung zwischen zwei BS2000/OSD-Rechnern mit Erhaltung der Dateischutz-Attribute:
   - `copymode same` am FTP-Client eingeben
   - `quote site cmod same` an den FTP-Server senden
   - Transfer der gewünschten Datei starten

2. FTP-Client befindet sich auf einem Unix-Rechner.
   Übertragung einer BS2000/OSD-Datei auf einen Unix-Rechner mit anschließender Übertragung der Datei auf einen BS2000/OSD-Rechner.

   - `quote site cmod` ... an den BS2000/OSD-Rechner senden
   - BS2000/OSD-Datei auf den Unix-Rechner übertragen
   - `quote site cmod` ... an den BS2000/OSD-Zielrechner senden
   - BS2000/OSD-Datei an den BS2000/OSD-Zielrechner senden

     Es muss `type binary` eingestellt sein, damit der Zielrechner die erhaltenen Dateien nicht verändert.

debug - DEBUG-Ausgaben ein-/ausschalten

Die DEBUG-Ausgaben dienen vor allem Netzadministratoren und Kundendienst-Mitarbeitern zur Diagnose von Problemen im Netz. Der Anwender benötigt in der Regel keine DEBUG-Ausgaben.

debug
<debug-wert>

<debug-wert>
Zulässig sind Werte zwischen 0 und 9.

0 Keine DEBUG-Ausgaben

1 Ausgabe aller Nachrichten vom FTP-Client an den FTP-Server und vom FTP-Server an den FTP-Client.

Zusätzlich zu 1:

2 Ausgabe von Informationen über Dateizugriffe.

Wird kein Operand angegeben, wird die DEBUG-Ausgabe umgeschaltet, d.h. war die DEBUG-Ausgabe eingeschaltet, so wird sie ausgeschaltet; war sie ausgeschaltet, so wird sie eingeschaltet. Angaben größer 2 werden wie 2 behandelt.

delete - Löschen einer fernen Datei

Mit dem Kommando *delete* wird eine Datei am fernen Rechner gelöscht. Das Kommando *mdelete* dient dem teilqualifizierten Löschen von Dateien am fernen Rechner.

delete
<ferne-datei>

<ferne-datei>
Name der zu löschenden Datei am fernen Rechner.

Beispiel

1. Das ferne Arbeits-Dateiverzeichnis wird angezeigt.

```
→ pwd
   257 ":110:$TSOS." is current directory.
```

2. Die Datei *anton* wird gelöscht.

```
→ delete anton
   200 DELE command okay.
```

dir - Informationen über ferne Datei(en)

Das Kommando *dir* liefert Informationen über Dateien am fernen Rechner. Informationen über ferne Dateien können auch mit den Kommandos *mdir*, *ls* und *mls* ermittelt werden.

dir
[<ferne-datei> [<lokale-datei>]]

<ferne-datei>
Name einer Datei am fernen Rechner. Fehlt dieser Operand, wird eine Liste aller Dateien im aktuellen Arbeits-Dateiverzeichnis des fernen Rechners geliefert.

<lokale-datei>
Name einer lokalen Datei, in die die Ausgabe des Kommandos geschrieben wird. Ist dieser Operand nicht angegeben, erfolgt die Ausgabe an die Datenstation.

Wenn die Verbindung zum Partner über ein FTAC-Profil aufgebaut wurde, wird die Datei-Information nur ab dem relativ eingestellten Pfadnamen ausgegeben.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Das lokale und das ferne Arbeits-Dateiverzeichnis werden abgefragt.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.

→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Es wird Information über Dateien des fernen Arbeits-Dateiverzeichnisses abgefragt. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ dir
   200 PORT command successful.
   ...
   -rw-rw-r-- 1 tcptest 229   Sep 7 12:55 anton.1
   -rw-rw-r-- 1 tcptest 229   Sep 7 12:55 anton.2
   -rw-rw-r-- 1 tcptest 229   Sep 7 12:55 anton.3
   -rwx------ 1 tcptest 10505 Sep 5 18:35 berta
   -rw-rw-r-- 1 tcptest 229   Sep 7 12:55 berta.1
   -rw-rw-r-- 1 tcptest 229   Sep 7 12:55 berta.2
   -rwx------ 1 tcptest 10505 Sep 5 18:19 caesar
   -rw-rw-r-- 1 tcptest 229   Sep 7 12:55 zwaton
   ...
   461 bytes received in 2.54 seconds (1.58E-01 Kbytes/s)
```

3. Information über die fernen Dateien *anton.** wird in der lokalen Datei *anton.prot* abgelegt.

```
→ dir anton.* anton.prot
   200 PORT command okay.
   ...
```

exit - Parameter für lokale Exit-Routine definieren

Exit-Routinen für den FTP-Client stellen Sie mit dem Kommando *exit* ein.

Nähere Informationen zu FTP-Exit-Routinen finden Sie im Handbuch „interNet Services Administratorhandbuch“.

exit
[receive:<receive-selector>] [send:<send-selector>]

<receive-selector>
: Mit <receive-selector> können Sie unter mehreren beim Empfangen von Daten zur Verfügung stehenden Aktionen die gewünschte Aktion auswählen. Zu diesem Zweck versorgen Sie die Exit-Routine mit einem String, auf den <receive-selector> verweist. Dabei können Sie frei entscheiden, welche Strings mit welchen Bedeutungen verwendet werden. Lediglich dem String "*NONE" ist eine feste Bedeutung zugeordnet: Bei Angabe von "*NONE" wird die Exit-Routine deaktiviert.

  Wenn Sie <receive-selector> nicht angeben, wird keine Exit-Routine angesprungen.

<send-selector>
: Mit <send-selector> können Sie unter mehreren beim Senden von Daten zur Verfügung stehenden Aktionen die gewünschte Aktion auswählen. Zu diesem Zweck versorgen Sie die Exit-Routine mit einem String, auf den <send-selector> verweist. Dabei können Sie frei entscheiden, welche Strings mit welchen Bedeutungen verwendet werden. Lediglich dem String "*NONE" ist eine feste Bedeutung zugeordnet: Bei Angabe von "*NONE" wird die Exit-Routine deaktiviert.

  Wenn Sie <send-selector> nicht angeben, wird keine Exit-Routine angesprungen.

file - Attribute einer Datei am lokalen Rechner festlegen

Das Kommando *file* legt - bei einer Nicht-1:1-Übertragung - die Datei-Attribute einer zu übertragenden Datei am lokalen Rechner fest. *file* wird auf das BS2000/OSD-Kommando *FILE* abgebildet. Dieses Kommando kann auch am FTP-Server, mit dem Kommando *quote site* vorangestellt, abgesetzt werden (siehe Seite 65).

file
<lokale-datei> <,file-operandenliste>

<lokale-datei>
Name einer Datei am lokalen Rechner, deren Datei-Attribute festgelegt werden sollen. Es muss ein im *FILE*-Kommando erlaubter vollqualifizierter Dateiname oder das Zeichen "*" verwendet werden.

<file-operandenliste>
Die möglichen Operanden sind den entsprechenden BS2000/OSD-Handbüchern zu entnehmen.
Bei der Angabe eines Dateinamens für *lokale-datei* gilt das *file*-Kommando nur für das nächste *get*-, *mget*-, *reget*- oder *recv*-Kommando, das sich auf den Dateinamen bezieht.
Bei der Angabe eines "*" Zeichens für *lokale-datei* gilt das *file*-Kommando nur für das nächste *get*-, *recv-, reget*- oder *mget*-Kommando. Die Dateiattribute gelten dann für die im *get*-, *reget*-, *recv-* oder *mget*-Kommando angegebenen lokalen Dateinamen.

Erfolgt die Eingabe mehrerer *file*-Kommandos, so gelten nur die Angaben des letzten *file*-Kommandos.

Wird kein *file*-Kommando angegeben und existiert die Datei noch nicht, so gilt wie bisher:

- Beim Übertragungstyp *ascii* (siehe *type*- oder *ascii*-Kommando) werden SAM-Dateien mit variabler Satzlänge erzeugt.
- Beim Übertragungstyp *binary* (siehe *type*- oder *binary*-Kommando) werden PAM-Dateien erzeugt.
- Existiert die Datei bereits, so werden die Attribute aus dem Katalogeintrag übernommen und die Datei wird überschrieben.
- Der Operand *LINK* darf in der *file-operandenliste* nicht angegeben werden, da automatisch ein Linkname ergänzt wird.

Einschränkungen

Es werden nur jene Dateiattribute unterstützt, die auch vom C-RTS V2.0 bei STREAM I/O unterstützt werden (siehe Handbuch „C-Compiler“).

Bei ISAM-Dateien werden für die Schlüsselposition und die Schlüssellänge nur die Einstellungen *KEYPOS=5* und *KEYLEN=8* unterstützt.

Bei der Übertragung von ISAM-Dateien mit dem Kommando *put* werden die Satzschlüssel nicht mitübertragen. Der Operand *recform* wird nicht ausgewertet. ISAM-Dateien können somit nur gelesen werden.

Wie Sie aber BS2000/OSD-Plattendateien unter Beibehaltung ihrer Eigenschaften transferieren, ist im Abschnitt „1:1-Übertragung von BS2000/OSD-Plattendateien“ auf Seite 72 beschrieben.

Beispiel

1. Es wird der Übertragungstyp *binary* eingestellt.

```
→ type binary
   200 Type set to I.
```

2. Es wird ein *file*-Kommando für die *datei1* abgesetzt.

```
→ file datei1,fcbtype=sam
```

3. Die Datei *datei1* wird als SAM-Datei angelegt.

```
→ get datei2 datei1
```

form - Übertragungsformat ändern oder abfragen

Das Kommando *form* stellt das Übertragungsformat ein bzw. gibt das aktuelle Übertragungsformat aus. Das aktuell eingestellte Übertragungsformat kann alternativ auch mit dem Kommando *status* ermittelt werden.

form
[file]

file

Stellt das Übertragungsformat FILE ein.
Ist kein Operand angegeben, wird das aktuelle Übertragungsformat am Bildschirm ausgegeben.

Die FTP-Norm stellt optional verschiedene Übertragungsformate zur Verfügung. In BS2000/OSD ist nur das Übertragungsformat FILE implementiert. Das Kommando *form* ist aus Kompatibilität zu anderen FTP-Implementierungen vorhanden.

ftyp - Bearbeitungstyp für Dateien am lokalen Rechner festlegen

Das Kommando *ftyp* legt fest, ob SAM-Dateien am lokalen Rechner als Text- oder Binär-Dateien bearbeitet werden sollen.

Das Kommando *ftyp* ist nur für SAM-Dateien des DVS-Dateisystems wirksam. PAM-Dateien werden wie bisher als Binärdateien behandelt und ISAM-Dateien als Textdateien. POSIX-Dateien sind von dieser Einschränkung nicht betroffen.

Das Kommando kann auch am FTP-Server, mit dem Kommando *quote site* vorangestellt, abgesetzt werden (siehe Seite 65).

ftyp
text | **binary** | **textbin**

Die Voreinstellung hat sich gegenüber älteren FTP-Versionen geändert, da mit der Einstellung ftyp=text kein Restart des Transfers möglich ist (siehe hierzu auch Abschnitt „Fragen und Antworten (FAQ)“ auf Seite 220).

text

Die SAM-Dateien am lokalen Rechner sollen als Textdateien bearbeitet werden.
Bei dieser Einstellung werden beim Schreiben in eine Datei Tabulatorzeichen (X'05' in EBCDIC) in eine entsprechende Anzahl Leerzeichen umgesetzt (Tabulatorpositionen 1, 9, 17 ...) und Neue-Zeile-Zeichen (X'15' in EBCDIC) führen zu einem Zeilenwechsel (Satzwechsel).
Beim Lesen werden an jeden gelesenen Satz Neue-Zeile-Zeichen (X'15') angefügt (siehe Handbuch „C-Compiler“).

binary

Die SAM-Dateien am lokalen Rechner sollen als Binärdateien bearbeitet werden.
Bei dieser Einstellung werden beim Schreiben in eine Datei die Daten identisch in die Datei übernommen (keine Umsetzung von Tabulatorzeichen X'05' und Neue-Zeile-Zeichen X'15') und beim Lesen ohne das Anfügen von Neue-Zeile-Zeichen (X'15') gelesen.
Bei dieser Einstellung können SAM-Dateien mit fester, variabler und undefinierter Satzlänge gelesen und geschrieben werden.
Bei fester Satzlänge wird der letzte Satz mit binären Nullen aufgefüllt (falls notwendig). Will man dies umgehen, so ist mit variabler Satzlänge zu arbeiten.

textbin

Die SAM-Dateien am lokalen Rechner sollen als binäre Textdateien bearbeitet werden. Bei dieser Einstellung werden beim Schreiben in eine Datei Tabulatorzeichen (X'05' in EBCDIC) nicht in Leerzeichen umgesetzt. Neue-Zeile-Zeichen (X'15' in EBCDIC) führen jedoch zu einem Zeilenwechsel (Satzwechsel).
Beim Lesen werden an jeden gelesenen Satz Neue-Zeile-Zeichen (X'15') angefügt.
Diese Einstellung ist nach dem Start von FTP voreingestellt.

Beispiel

1. Der Dateibearbeitungstyp wird auf *binary* gesetzt.

```
→ ftyp binary
   ftyp set to binary.
```

2. Die Datei *datei1* wird ohne Umsetzung der Tabulator- und Neue-Zeile-Zeichen angelegt.

```
→ get datei2 datei1
```

get - Holen einer Datei

Mit dem Kommando *get* wird eine Datei vom fernen Rechner zum lokalen Rechner übertragen. Auch mit den Kommandos *mget* und *recv* können Dateien vom fernen zum lokalen Rechner übertragen werden. Für die Restart-Unterstützung gibt es eine spezielle Ausprägung des *get*-Kommandos, das *reget*-Kommando (siehe Seite 181).

get
<ferne-datei> [<lokale-datei>]

<ferne-datei>
Name einer Datei am fernen Rechner, die zum lokalen Rechner übertragen werden soll.

<lokale-datei>
Name einer POSIX- oder DVS-Datei am lokalen Rechner. Existiert die Datei bereits, wird sie entweder überschrieben oder - mit entsprechendem Suffix versehen - neu angelegt. Dieses Verhalten wird mit *runique* gesteuert. Falls die Datei nicht existiert, wird sie neu angelegt.
Fehlt der Operand *lokale-datei*, wird der Name der fernen Datei verwendet (in diesem Fall muss der Name der fernen Datei auch den Dateinamenskonventionen des lokalen Rechners entsprechen).

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des lokalen und des fernen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.VON.UNIX.

→ pwd
   257 /home/usr/man is current directory.
```

2. Der Inhalt der Datei *anton* (im fernen Arbeits-Dateiverzeichnis) wird in eine DVS-Datei mit gleichem Namen (im lokalen Arbeits-Dateiverzeichnis) übertragen. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ get anton
   200 PORT command okay.
   ...
```

3. Wechseln in das POSIX-HOME-Dateiverzeichnis.

```
→ lcd %POSIX
   Local directory now /home/usr
```

4. Die ferne Datei anton wird zum lokalen Rechner übertragen und im POSIX-Dateisystem mit dem Namen */home/usr/berta* gespeichert.

```
→ get anton berta
   200 PORT command okay.
   ...
```

glob - Expansion von Metazeichen ein-/ausschalten

Das Kommando *glob* schaltet die Expansion von Metazeichen in lokalen Dateinamen ein bzw. aus. Nach dem Start von FTP ist die Metazeichen-Expansion eingeschaltet.

In den Kommandos *mget*, *mput*, *mdir*, *mls* und *mdelete* können in den lokalen Dateinamen Metazeichen angegeben werden. Ist die Metazeichen-Expansion eingeschaltet, werden Metazeichen als solche erkannt und den Regeln entsprechend ausgewertet. Ist die Metazeichen-Expansion ausgeschaltet, werden Metazeichen nicht erkannt, sondern als Teil des Dateinamens interpretiert.

glob

Beispiel

```
mget a*
```

Wenn die Metazeichen-Expansion ausgeschaltet ist - in der Status-Anzeige durch *off* ersichtlich - fasst FTP die Angabe `a*` als vollqualifizierten Dateinamen auf und bringt die Fehlermeldung:

```
550 a* not a plain file or does not exist
```

hash - Anzeige des Übertragungsfortschritts ein-/ausschalten

Das Kommando *hash* schaltet die Anzeige des Fortschritts einer Dateiübertragung ein bzw. aus. Ist die Anzeige eingeschaltet, wird nach jeder Übertragung eines Blockes das Zeichen # am Bildschirm ausgegeben.

Nach dem Start von FTP ist *hash* ausgeschaltet. Bei Einschalten von *hash* wird die verwendete Blocklänge ausgegeben. Wenn FTP im Batch aufgerufen wird, also der Auftragsschalter 1 gesetzt ist, ist *hash* wirkungslos.

Die aktuelle Einstellung von *hash* kann mit dem Kommando *status* ermittelt werden.

hash

Beispiel

1. Abfragen des lokalen und des fernen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.

→ pwd
   257 "/usr/tcptest/man/sam/from.bs2000" is current directory.
```

2. Einschalten. Beim Kommando *append* wird implizit die FTP-Server-Funktion PORT aufgerufen.

```
→ hash
   Hash mark printing on (8192 bytes/hash mark).

→ append sam.to.sinix.anton.upd anton
   200 PORT command okay.
```

3. Dateiübertragung mit Ausgabe des Zeichens # nach jedem übertragenen Block von 8192 Byte.

```
   ##################################################
   226 Transfer complete.
   109089 bytes sent in 12.83 seconds (8.30 Kbytes/s)
```

help - Information zu FTP-Kommandos

Das Kommando *help* gibt eine Liste aller FTP-Kommandos aus. Zu allen FTP-Kommandos sind auch Kurzinformationen erhältlich.

Das Kommando *?* gibt ebenfalls eine Liste aller FTP-Kommandos aus. Eine Liste aller Funktionen des FTP-Servers am fernen Rechner ist mit dem Kommando *remotehelp* erhältlich.

help
[< kommando >]

<kommando>
FTP-Kommando, zu dem Informationen gewünscht sind.
Ist kein FTP-Kommando angegeben, wird eine Liste aller FTP-Kommandos ausgegeben.

Beispiel

1. Anfordern einer Liste der FTP-Kommandos.

```
→ help
   Commands may be abbreviated.  Commands are:

!               exit            mdelete         quit            setfile
?               modchar         mdir            quote           settime
append          file            mget            recv            status
ascii           form            mkdir           readopt         struct
bell            ftyp            mls             reget           sunique
binary          get             mode            remotehelp      svar
bye             glob            mput            rename          system
ccc             hash            open            reput           tenex
cd              help            passive         rexit           trace
cdup            jobvar          private         rmdir           type
copymode        lcd             prompt          runique         user
close           ldir            protect         send            verbose
delete          lls             proxy           sendport
debug           lpwd            put             setcase
dir             ls              pwd             setcode
```

2. Anfordern einer Kurzinformation zu mget.

```
→ help mget
   mget get multiple files
```

jobvar - Fehlerinformationen in einer Jobvariablen hinterlegen

Mit Kommando *jobvar* kann dem FTP-Client mitgeteilt werden, ob er Fehlerinformationen in einer Jobvariablen hinterlegen soll.

Eine Alternative zur Hinterlegung von Fehlerinformationen bietet das Kommando *svar* (siehe Seite 201).

Tritt bei der Bearbeitung des *jobvar*-Kommandos ein Fehler auf und ist der Schalter 1 gesetzt (Batch-Berieb), dann wird der FTP-Client mit TERMJ beendet. Der Batch-Job bzw. die Prozedur wird somit erst nach der nächsten STEP-Anweisung fortgesetzt.

Wenn die Versorgung einer Job-Variablen aktiviert ist, verändert sich das Verhalten von Kommandos, die mehrere Einzelaktionen ausführen (Kommandos *mdir*, *mls*, *mget*, *mput*, *mdelete*). Bei diesen Kommandos wird die Kommando-Verarbeitung nach der ersten fehlerhaften Aktion abgebrochen.

Dagegen wird der FTP-Client im Batch-Betrieb nicht mit TERMJ beendet, wenn die Berechtigungsdaten beim *open*-Kommando fehlerhaft sind.

jobvar
< jv-name> | ***NONE**

<jv-name>
: Name der Job-Variablen, die der FTP-Client nach Absetzen des *jobvar*-Kommandos mit den Kommando-Return-Informationen versorgen soll. Falls die Job-Variable <jv-name> nicht bereits existiert, wird sie neu erstellt.

***NONE**
: Die Angabe von *NONE veranlasst, dass nicht länger eine Job-Variable mit Kommando-Return-Informationen des FTP-Clients versorgt wird.

Layout der Job-Variablen

Offset / Länge	Feld	Feldbeschreibung
0 / 3	Status-Indicator	$S: Kommando erfolgreich ausgeführt $E: Kommando mit Fehler $T: FTP-Client normal beendet $A: FTP-Client wegen Fehler abnormal beendet
3 / 1	Filler	stets „0“
4 / 4	TSN	TSN der FTP-Client-Task
8 / 4	catid	mit Leerzeichen versorgt
12 / 4	Session number	System-Laufnummer
16 / 16	User command name	Name des FTP-Client-Kommandos
32 / 96	Command Parameter	FTP-Kommando-Parameter
128 / 4	FTP protocol command	vom Client an den Server gesendetes Kommando
132 /124	FTP message	Lokale Meldung bzw. Server-Antwort

lcd - Wechseln des lokalen Arbeits-Dateiverzeichnisses

Das Kommando *lcd* ändert das aktuelle lokale Arbeits-Dateiverzeichnis oder das Dateisystem am lokalen Rechner. Beim Wechsel des Dateisystems wird in das entsprechende HOME-Dateiverzeichnis gewechselt. Das aktuell eingestellte lokale Arbeits-Dateiverzeichnis kann mit dem Kommando *lpwd* ermittelt werden. Das ferne Arbeits-Dateiverzeichnis wird mit dem Kommando *cd* eingestellt.

lcd
. | **..** | **%POSIX** | **%BS2000** | **$HOME** | <pfad>

.
Das lokale Arbeits-Dateiverzeichnis wird nicht geändert (nur aus Kompatibilität zu anderen Implementierungen vorhanden).

..
Im DVS-Dateisystem wird die letzte (am weitesten rechts stehende) Teilqualifizierung aus dem Namen des lokalen Arbeits-Dateiverzeichnisses entfernt.
Im POSIX-Dateisystem wird in das übergeordnete Dateiverzeichnis gewechselt.

%POSIX
Es wird in das lokale POSIX-HOME-Dateiverzeichnis gewechselt.

%BS2000
Es wird in das lokale DVS-HOME-Dateiverzeichnis gewechselt. Wenn Sie sich bereits im DVS-Dateisystem befinden, ist die Angabe von %BS2000 mit der Angabe $HOME identisch.

$HOME
Es wird in das lokale DVS-HOME-Dateiverzeichnis gewechselt. Voraussetzung für die Angabe von $HOME ist, dass Sie sich bereits in einem DVS-Dateiverzeichnis befinden.

<pfad>
Wird im DVS-Dateisystem ein vollständiger Pfadname (beginnend mit einer Katalogangabe und/oder einer Benutzerkennung) angegeben, wird das bestehende Arbeits-Dateiverzeichnis ersetzt. Fehlen Katalogangabe und Benutzerkennung, wird <alternname>.<pfad> eingestellt.

Im DVS-Dateisystem werden die Pfadnamen nicht auf ihre Richtigkeit überprüft.
Im POSIX-Dateisystem kann <pfad> ein absoluter oder relativer POSIX-Pfadname sein.

Beispiel

1. Abfragen des lokalen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.
```

2. Ändern des Arbeits-Dateiverzeichnisses durch Hinzufügen der Teilqualifizierung MAN.

```
→ lcd man
   Local directory now :5:$TCPTEST.MAN.
```

3. Wechseln in das POSIX-HOME-Dateiverzeichnis.

```
→ lcd %POSIX
   Local directory now /home/usr
```

4. Ändern des Arbeits-Dateiverzeichnisses auf */home/usr/test*.

```
→ lcd test
   Local directory now /home/usr/test
```

5. Wechseln in das DVS-HOME-Dateiverzeichnis.

```
→ lcd %BS2000
   Local directory now :5:$TCPTEST.
```

6. Ändern des Arbeits-Dateiverzeichnisses durch Angabe einer Teilqualifizierung.

```
→ lcd XXX
   Local directory now :5:$TCPTEST.XXX.
```

7. Ändern des Arbeits-Dateiverzeichnisses durch Entfernen einer Teilqualifizierung.

```
→ lcd ..
   Local directory is :5:$TCPTEST.
```

ldir - Information über lokale Dateien

Das Kommando *ldir* liefert Informationen über Dateien am lokalen Rechner. Informationen über lokale Dateien können auch mit dem Kommando *lls* ermittelt werden.

ldir
[<lokale-datei>] [,<fstatus-operandenliste>]

<lokale-datei>
ldir wird im DVS-Dateisystem auf das BS2000/OSD-Kommando *FSTATUS* abgebildet. <*lokale-datei*> muss ein im *FSTATUS*-Kommando erlaubter voll- oder teilqualifizierter Dateiname sein.
Im POSIX-Dateisystem wird *ldir* auf das Kommando *ls -l* abgebildet. <*lokale-datei*> muss ein erlaubter voll- oder teilqualifizierter Dateiname sein.
Ist keine lokale Datei angegeben, werden Informationen über alle Dateien des aktuellen Arbeits-Dateiverzeichnisses ausgegeben.

<fstatus-operandenliste>
fstatus-operandenliste sind beliebige Operanden des *FSTATUS*-Kommandos. Das Kommando *FSTATUS* ist im Handbuch „Benutzerkommandos (ISP-Format)“ beschrieben.

Beispiel

1. Abfragen des lokalen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.
```

2. Informieren über die DVS-Dateien mit dem Dateiattribut SAM im Arbeits-Dateiverzeichnis.

```
→ ldir ,fcbtype=sam
   0000003 :5:$TCPTEST.MAN.ANTON.PROT
   0000003 :5:$TCPTEST.MAN.P.TRC
   0000003 :5:$TCPTEST.MAN.P.USR
   0000003 :5:$TCPTEST.MAN.SAM.NACH.MSDOS.ANTON.UPD
   0000003 :5:$TCPTEST.MAN.SAM.NACH.MSDOS.BERTA.1
   0000003 :5:$TCPTEST.MAN.SAM.NACH.SINIX.ANTON
   0000006 :5:$TCPTEST.MAN.SAM.VON.SINIX.BERTA
   0000003 :5:$TCPTEST.MAN.SAM.VON.SINIX.BERTA.1
   0000003 :5:$TCPTEST.MAN.SAM.VON.SINIX.BERTA.2
   0000006 :5:$TCPTEST.MAN.SAM.VON.SINIX.CAESAR
   :5: PUBLIC: 30 FILES. RES= 99, FREE= 51, REL= 0 PAGES
```

lls - Auflisten von Dateinamen am lokalen Rechner

Das Kommando *lls* listet Dateinamen am lokalen Rechner auf.

lls
[<lokale-datei>] [,<fstatus-operandenliste>]

<lokale-datei>
lls wird im DVS-Dateisystem auf das BS2000/OSD-Kommando *FSTATUS* abgebildet. <*lokale-datei*> muss ein im *FSTATUS*-Kommando erlaubter voll- oder teilqualifizierter Dateiname sein.
Im POSIX-Dateisystem wird *lls* auf das Kommando *ls* abgebildet. <*lokale-datei*> muss ein erlaubter voll- oder teilqualifizierter Dateiname sein.
Ist keine lokale Datei angegeben, so werden Informationen über alle Dateien des aktuellen Arbeits-Dateiverzeichnisses ausgegeben.

<fstatus-operandenliste>
fstatus-operandenliste sind beliebige Operanden des *FSTATUS*-Kommandos. Das Kommando *FSTATUS* ist im Handbuch „Benutzerkommandos (ISP-Format)“ beschrieben.

Beispiel

1. Abfragen des lokalen Arbeits-Dateiverzeichnisses (DVS-Dateisystem).

```
→ lpwd
   Local directory is :5:$TCPTEST.
```

2. Anfordern einer Liste von DVS-Dateien im lokalen Dateiverzeichnis.

```
→ lls sam.nach.*
   SAM.NACH.MSDOS.ANTON.UPD
   SAM.NACH.MSDOS.BERTA.1
   SAM.NACH.MSDOS.BERTA.2
   SAM.NACH.SINIX.ANTON
   SAM.NACH.SINIX.BERTA.1
   SAM.NACH.SINIX.BERTA.2
   SAM.NACH.SINIX.CAESAR
```

lpwd - Ausgeben des lokalen Arbeits-Dateiverzeichnisses

Das Kommando *lpwd* gibt den Namen des aktuell eingestellten lokalen Arbeits-Dateiverzeichnisses aus. Das lokale Arbeits-Dateiverzeichnis wird mit dem Kommando *lcd* eingestellt.

lpwd

Beispiel

1. Abfragen des lokalen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.
```

2. Ändern des lokalen Arbeits-Dateiverzeichnisses durch Hinzufügen der Teilqualifizierung *MAN.SAM*.

```
→ lcd man.sam
   Local directory now :5:$TCPTEST.MAN.SAM.

→ lpwd
   Local directory is :5:$TCPTEST.MAN.SAM.
```

3. Ändern des Arbeitsdateiverzeichnisses im POSIX-Dateisystem.

```
→ lcd %POSIX
   local directory now /home/usr/tcp.

→ lpwd
   local directory is /home/usr/tcp.
```

ls - Auflisten von Dateinamen am fernen Rechner

Das Kommando *ls* listet Dateinamen am fernen Rechner auf. Informationen über ferne Dateien können auch mit den Kommandos *dir*, *mdir* und *mls* ermittelt werden.

ls
[<ferne-datei> [<lokale-datei>]]

<ferne-datei>
Name einer Datei am fernen Rechner. Fehlt dieser Operand, wird eine Liste aller Dateien im aktuellen Arbeits-Dateiverzeichnis des fernen Rechners geliefert.

<lokale-datei>
Name einer lokalen Datei, in die die Ausgabe des Kommandos geschrieben wird. Ist dieser Operand nicht angegeben, erfolgt die Ausgabe an die Datenstation.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des Arbeits-Dateiverzeichnisses.

```
→ pwd
   257 "/usr/tcptest" is current directory.
```

2. Auflisten der Dateien im Arbeits-Dateiverzeichnis am Bildschirm. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ ls
   200 PORT command okay.
   ...
   FIL.files
   FIL.tools
   FTP
   TELNET
   TU
   man
   pool
   backup
   ...
```

3. Auflisten der mit *DAT.** bezeichneten fernen Dateien in die lokale Datei *dat.prot*.

```
→ ls DAT.* dat.prot
   200 PORT command okay.
   ...
```

mdelete - Löschen mehrerer ferner Dateien

Mit dem Kommando *mdelete* werden Dateien am fernen Rechner gelöscht. Die Funktion Rückfrage wird mit dem Kommando *prompt* ein- und ausgeschaltet. Mit dem Kommando *delete* können ebenfalls Dateien am fernen Rechner gelöscht werden.

mdelete
<ferne-datei> [<ferne-datei>] [<ferne-datei>]

<ferne-datei>
Name der zu löschenden Datei im fernen Rechner. Es können mehrere Dateinamen angegeben werden.
Wenn die Funktion Rückfrage eingeschaltet ist, fragt FTP vor dem Löschen jeder Datei, ob die Datei wirklich gelöscht werden soll oder nicht.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des fernen Arbeits-Dateiverzeichnisses.

```
→ pwd
   257 "/usr/tcptest/man/sam/von.msdos" is current directory.
```

2. Löschen der Dateien *anton*.* und *berta*.*. Die Rückfragefunktion (Prompting) ist eingeschaltet.

```
→ mdelete anton.* berta.*
   mdelete anton.2 (y/n/q)?

→ y
   200 DELE command okay.
   mdelete anton.3 (y/n/q)?

→ n
   mdelete anton.upd (y/n/q)?

→ y
   200 DELE command okay.
   mdelete berta.1 (y/n/q)?

→ y
   200 DELE command okay.
   mdelete berta.2 (y/n/q)?

→ n
```

3. Ausschalten der Rückfragefunktion.

```
→ prompt
   Interactive mode off.
```

4. Löschen der übrigen Dateien im fernen Arbeits-Dateiverzeichnis. Die Rückfragefunktion (Prompting) ist ausgeschaltet.

```
→ mdelete *
   200 DELE command okay.
```

mdir - Information über ferne Dateien ausgeben

Das Kommando *mdir* liefert Informationen über Dateien am fernen Rechner. Informationen über ferne Dateien können auch mit den Kommandos *dir*, *ls* und *mls* ermittelt werden.

mdir
[<ferne-datei>] [<ferne-datei>] [<ferne-datei>] ... - | <lokale-datei>

<ferne-datei>
Name einer Datei am fernen Rechner. Es können mehrere Dateinamen in einem einzigen Aufruf angegeben werden.

<lokale-datei>
Name einer lokalen Datei, in die die Ausgabe des Kommandos geschrieben wird.

-
Die Informationen werden auf den Bildschirm ausgegeben.

Werden keine Operanden angegeben, so werden diese im Dialog abgefragt.

Wenn die Verbindung zum Partner über ein FTAC-Profil aufgebaut wurde, wird die Datei-Information nur ab dem relativ eingestellten Pfadnamen ausgegeben.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des lokalen und des fernen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.

→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Ausgeben von Informationen über die fernen Dateien *berta*.* und *anton*.* in die lokale Datei *SEVERAL.PROT*. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ mdir berta.* anton.* several.prot
   200 PORT command okay.
   150 ASCII data connection for berta.* (89.16.100.0,1187).
   226 Transfer complete.
   100 bytes received in 0.71 seconds (1.36E-01 Kbytes/s)
   200 PORT command okay.
   150 ASCII data connection for anton.* (89.16.100.0,1188).
   226 Transfer complete.
   150 bytes received in 1.64 seconds (8.90E-02 Kbytes/s)
```

mget - Holen mehrerer ferner Dateien

Mit dem Kommando *mget* werden Dateien vom fernen Rechner zum lokalen Rechner übertragen. Mit den Kommandos *get*, *reget* und *recv* können einzelne Dateien vom fernen zum lokalen Rechner übertragen werden.

mget
<ferne-datei> [<ferne-datei>] [<ferne-datei>] ...

<ferne-datei>
Name einer Datei am fernen Rechner, die zum lokalen Rechner übertragen werden soll. Es können mehrere Dateien angegeben werden.
Die Dateien heißen am lokalen Rechner genauso wie am fernen Rechner. Die Dateinamen müssen daher sowohl den Regeln des lokalen Rechners als auch des fernen Rechners gehorchen.
Wenn die Funktion Rückfrage (siehe *prompt*) eingeschaltet ist, fragt FTP vor dem Übertragen jeder Datei, ob die Datei wirklich übertragen werden soll oder nicht.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfrage des lokalen und des fernen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.SAM.VON.SINIX.

→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Alle Dateien des fernen Arbeits-Dateiverzeichnisses werden übertragen. Implizit wird die FTP-Server-Funktion PORT aufgerufen. Die Rückfragefunktion ist in diesem Beispiel ausgeschaltet (siehe *prompt*).

```
→ mget *
   200 PORT command okay.
   150 ASCII data connection for anton (89.16.100.0,1192).
   226 Transfer complete.
   242 bytes received in 0.06 seconds (3.69 Kbytes/s)
   200 PORT command okay.
   150 ASCII data connection for anton.1 (89.16.100.0,1193).
   226 Transfer complete.
   242 bytes received in 0.06 seconds (3.75 Kbytes/s).
   .
   further transfers
   .
   200 PORT command okay.
   150 ASCII data connection for caesar (89.16.100.0,1199).
   226 Transfer complete.
   10845 bytes received in 0.78 seconds (13.56 Kbytes/s)
```

mkdir - Einrichten eines fernen Dateiverzeichnisses

Mit dem Kommando *mkdir* wird ein neues Dateiverzeichnis am fernen Rechner eingerichtet. Das Kommando ist nicht für das DVS-Dateisystem gültig.

mkdir
<fernes-dateiverzeichnis>

<fernes-dateiverzeichnis>
Vollständiger oder relativer Name des Dateiverzeichnisses am fernen Rechner

mls - Auflisten von Dateinamen in mehreren Verzeichnissen am fernen Rechner

Das Kommando *mls* liefert Informationen über Dateien am fernen Rechner. Informationen über ferne Dateien können auch mit den Kommandos *dir*, *mdir* und *ls* ermittelt werden.

mls
[<ferne-datei>] [<ferne-datei>] [<ferne-datei>] ... - | <lokale-datei>

<ferne-datei>
: Name einer Datei am fernen Rechner. Es können mehrere Dateien in einem einzigen Aufruf angegeben werden.

<lokale-datei>
: Name einer lokalen Datei, in die die Ausgabe des Kommandos geschrieben wird.

-
: Die Informationen werden auf den Bildschirm ausgegeben.

Werden keine Operanden angegeben, so werden diese im Dialog abgefragt.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Das lokale und das ferne Arbeits-Dateiverzeichnis wird abgefragt.

```
→ lpwd
   Local directory is :5:$TCPTEST.MAN.

→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Auflisten der mit *berta*, *berta*.* und *anton*.* bezeichneten Dateien in der lokalen Datei *listen.prot*. Implizit wird die FTP-Server-Funktion PORT aufgerufen. Die Rückfragefunktion ist ausgeschaltet.

```
→ mls berta berta.* anton.* listen.prot
   200 PORT command okay.
   150 ASCII data connection for berta (89.16.100.0,1184).
   226 Transfer complete.
   6 bytes received in 0.05 seconds (9.93E-02 Kbytes/s)
   200 PORT command okay.
   150 ASCII data connection for berta (89.16.100.0,1185).
   226 Transfer complete.
   16 bytes received in 0.21 seconds (7.37E-02 Kbytes/s)
   200 PORT command okay.
   150 ASCII data connection for anton.* (89.16.100.0,1186).
   226 Transfer complete.
   24 bytes received in 0.14 seconds (1.58E-01 Kbytes/s)
```

modchar - Zeichenkette wechseln

Zum Wechsel zwischen dem lokalen DVS-Dateisystem und dem lokalen POSIX-Dateisystem muss bei dem Kommando *lcd* eine bestimmte Zeichenfolge angegeben werden (siehe Seite 143). Sollten bei der Verwendung der Standardzeichenfolgen %BS2000 und %POSIX Probleme auftreten (z.B. weil ein POSIX-Dateiverzeichnis gleichen Namens existiert), dann kann das erste Zeichen dieser Zeichenfolge mit *modchar* geändert werden.

Die aktuell eingestellte Zeichenfolge kann mit dem Kommando *status* ermittelt werden.

Die entsprechende Einstellung können Sie an einem BS2000/OSD-FTP-Server mit dem Kommando *quote site modc* vornehmen (siehe Seite 65).

modchar
<character>

<character>
Zeichen gegen das das eingestellte erste Zeichen der Zeichenfolgen zum Wechsel zwischen dem DVS-Dateisystem und der POSIX-Dateisystem ersetzt werden soll.

Beispiel

1. Das lokale Arbeits-Dateiverzeichnis wird abgefragt.

```
→ lpwd
   Local directory is :4:$TEST.
```

2. Wechsel vom DVS-HOME-Dateiverzeichnis in das POSIX-HOME-Dateiverzeichnis.

```
→ lcd %POSIX
   Local directory now /home/usr.

→ lpwd
   Local directory is /home/usr.
```

3. Das eingestellte erste Zeichen der Zeichenfolgen zum Wechsel zwischen dem DVS- und dem POSIX-Dateisystem wird auf # geändert.

```
→ modchar #
   The prefix has been set to #.
```

4. Wechsel vom POSIX-HOME-Dateiverzeichnis in das DVS-HOME-Dateiverzeichnis.

```
→ lcd #BS2000
   Local directory now :4:$TEST.
   lpwd
   Local directory is :4:$TEST.
```

mode - Übertragungsmodus ändern oder abfragen

Das Kommando *mode* stellt den Übertragungsmodus ein bzw. gibt den aktuell eingestellten Übertragungsmodus aus. Ist kein Operand angegeben, wird der aktuelle Übertragungsmodus am Bildschirm ausgegeben. Der aktuell eingestellte Übertragungsmodus kann auch mit dem Kommando *status* ermittelt werden.

Die FTP-Norm stellt optional verschiedene Übertragungsmodi zur Verfügung. Im BS2000 sind davon *stream* und *block* implementiert.

mode
[**stream** | **block**]

stream
Stellt den Übertragungsmodus STREAM ein.

block
Dieser Operand erlaubt es, Dateien ohne Änderung der vorhandenen Satzstruktur zu übertragen. Voraussetzung für die Angabe von *block* sind

- Übertragungstyp EBCDIC und
- die Dateieigenschaften RECFORM=V und FCBTYPE=SAM.

mput - Senden von mehreren lokalen Dateien

Mit dem Kommando *mput* werden Dateien vom lokalen Rechner zum fernen Rechner übertragen. Dateien können auch mit den Kommandos *put*, *reput*, *append* und *send* zu einem fernen Rechner übertragen werden.

mput
<lokale-datei> [<lokale-datei>] [<lokale-datei>] ...

<lokale-datei>

Name einer Datei am lokalen Rechner, die zum fernen Rechner übertragen werden soll. Es können mehrere Dateien mit einem einzigen Aufruf angegeben werden.
Die Dateien heißen im fernen Rechner genauso wie im lokalen Rechner. Die Dateinamen müssen daher sowohl den Regeln des lokalen Rechners als auch denen des fernen Rechners gehorchen. Per Default werden Großbuchstaben im Namen der lokalen Datei in Kleinbuchstaben umgewandelt. Diese Voreinstellung können Sie jedoch mithilfe des Client-Kommandos *setcase* (siehe Seite 193) deaktivieren.
Wenn die Funktion Rückfrage (siehe Kommando *prompt*) eingeschaltet ist, fragt FTP vor dem Übertragen jeder Datei, ob die Datei wirklich übertragen werden soll oder nicht. Wenn eine Datei nicht übertragen werden konnte, werden auch die nachfolgenden Dateien nicht übertragen.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des lokalen und des fernen Arbeits-Dateiverzeichnisses.

```
→ lpwd
Local directory is :5:$TCPTEST.MAN.SAM.NACH.SINIX.

→ pwd
257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Senden der mit *anton*, *anton*.* und *berta*.* bezeichneten Dateien im lokalen Arbeits-Dateiverzeichnis. Implizit wird die FTP-Server-Funktion PORT aufgerufen. Die Rückfragefunktion ist in diesem Beispiel ausgeschaltet. Siehe Kommando *prompt* auf Seite 170.

```
→ mput anton anton.* berta.*
   200 PORT command okay.
   150 ASCII data connection for anton (89.16.100.0,1201).
   226 Transfer complete.
   242 bytes sent in 0.03 seconds (6.05 Kbytes/s)
   200 PORT command okay.
   150 Opening data connection for anton.upd (89.16.100.0,1202).
   .
   further transfers
   .
   226 Transfer complete.
   242 bytes sent in 0.04 seconds (5.76 Kbytes/s)
   200 PORT command okay.
   150 ASCII data connection for berta.2 (89.16.100.0,1207).
   226 Transfer complete.
   242 bytes sent in 0.04 seconds (5.90 Kbytes/s)
```

open - Eröffnen der Verbindung zu einem fernen Rechner

Mit dem Kommando *open* wird eine Verbindung zu einem fernen Rechner eröffnet. Es muss entweder der Name oder die Internet-Adresse dieses Rechners bekannt sein. Der Rechner muss entweder zum lokalen Netz gehören oder über einen Gateway-Rechner erreichbar sein. Namen und Adressen der mit *open* erreichbaren Rechner können vom Verwalter des lokalen Netzes erfragt werden.

Falls mithilfe des *protect*- oder *private*-Kommandos die TLS-Absicherung der Kontrollverbindung aktiviert wurde, resultiert dies in einem modifizierten Verhalten des *open*-Kommandos (siehe Seite 166).

open
<ipadr> | <remotehost> | localhost | loopback [<port>]

<ipadr>
Internet-Adresse (IPv4- oder IPv6-Adresse) des fernen Rechners, zu dem die Verbindung aufgebaut werden soll:

- Eine IPv4-Adresse muss in der üblichen „decimal-dotted“-Notation angegeben werden.
- Eine IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

<remotehost>
Symbolischer Name des fernen Rechners, zu dem die Verbindung aufgebaut werden soll

localhost
Symbolischer Name, der für den eigenen Rechner reserviert ist (meist nur für Testzwecke sinnvoll). Für *localhost* ist eine eigene Internet-Adresse generiert, die nicht mit der Adresse des eigenen Rechners im lokalen Netz identisch ist. Diese Adresse ist nur dem lokalen Rechner bekannt; andere Rechner können sie nicht verwenden.

loopback
loopback steht für die Internet-Adresse, mit der der lokale Rechner tatsächlich im LAN erreichbar ist.

<port>
Portnummer des FTP-Servers. Der FTP-Server hat standardmäßig die Portnummer 21. Die Angabe einer Portnummer ist beispielsweise dann nötig, wenn auf einem Rechner mehrere FTP-Server betrieben werden.

Falls am fernen Rechner ein Mechanismus zur Zugangskontrolle implementiert ist (wie z.B. in BS2000/OSD und in Unix-Systemen), ermittelt FTP die notwendigen Berechtigungsdaten (Benutzerkennung, Abrechnungsnummer, Passwort) im Dialog. Es werden nur die tatsächlich benötigten Informationen abgefragt (Unix-Systeme benötigen z.B. keine Abrechnungsnummer; daher entfällt die entsprechende Abfrage).

Werden die Berechtigungsdaten fehlerhaft eingegeben, bleibt im Dialogbetrieb die Verbindung zum fernen Rechner bestehen. Mit dem Kommando *user* besteht dann die Möglichkeit, die Berechtigungsdaten nachträglich einzugeben.

Im Batch-Betrieb bzw. im Dialogbetrieb, wenn der Auftragsschalter 1 gesetzt ist, müssen die notwendigen Berechtigungsinformationen in eigenen Zeilen (in der richtigen Reihenfolge) dem Kommando *open* folgen. Sind die Berechtigungsdaten fehlerhaft, wird die Verbindung wieder geschlossen. FTP wird mit TERMJ beendet; d.h. der Batch-Job bzw. die Prozedur wird erst nach der nächsten STEP-Anweisung fortgesetzt. Siehe hierzu auch die Client-Kommandos *jobvar* auf Seite 141 bzw. *svar* auf Seite 201.

Die Berechtigungsdaten müssen den Konventionen des Partnerrechners entsprechend angegeben werden. Beachten Sie in diesem Zusammenhang bitte die Tabelle auf Seite 75.

Ist kein Kennwort nötig, wird *'*NONE'* (Großbuchstaben) angegeben.

Beispiel

Der ferne Rechner ist ein Linux-Rechner.

1. Aufruf des FTP-Client.

```
/START-FTP
%  BLS0523 ELEMENT 'FTP', VERSION '05.2A00' FROM LIBRARY
                          ':DC16:$TSOS.SYSLNK.TCP-IP-AP.052'INPROCESS
%  BLS0524 LLM 'FTP', VERSION '05.2A00' OF '2010-10-29 19:19:43' LOADED
%  BLS0551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2010. ALL RIGHTS RESERVED
BS2000-FTP Vers V05.2A00 Oct 24 2010 16:50:19
ftp>
```

2. Herstellung der Verbindung zum Rechner anlaged.

```
→ open anlaged
   Connected to anlaged, port21.
   220 anlaged (vsFTPd 2.0.5)
```

3. Eingabeaufforderung für eine Kennung (Name).

```
   Name (anlaged:TCPTEST):
→ KENNUNG
   331 Password required for tcptest.
```

4. Eingabeaufforderung für ein Passwort.

```
   Password (anlaged:tcptest):
→ PASSWORT
   230 Login successful.
```

Verhalten des *open*-Kommandos bei bestehender TLS-Absicherung der Kontrollverbindung

Bei einer via *protect*- oder *private*-Kommando bzw. durch die entsprechenden Options aktivierten TLS/SSL-Absicherung der Kontrollverbindung verhält sich das *open*-Kommando wie folgt:

1. Falls das TLS-Subsystem noch nicht initialisiert ist, wird zunächst die Meldung `[Initialising TLS]` ausgegeben und Sie müssen – falls das BS2000/OSD-Subsystem PRNGD nicht aktiv ist – selbst für die Initialisierung des Pseudozufallszahlen-Generators sorgen (siehe Abschnitt „protect - TLS-Absicherung der Kontrollverbindung ein-/ausschalten“ auf Seite 171).
2. Falls in der Option-Datei ein privater Schlüssel (für Client-Zertifizierung) spezifiziert wurde, wird nach dem Verbindungsaufbau zum Server zunächst die Passphrase für den privaten Schlüssel abgefragt, falls dieser verschlüsselt abgespeichert wurde. Es wird aus Sicherheitsgründen dringend empfohlen, den privaten Schlüssel verschlüsselt abzuspeichern.

   Der private Schlüssel wird nur beim ersten mit TLS abgesicherten *open*-Kommando geladen, so dass bei nachfolgenden *open*-Kommandos innerhalb derselben FTP-Sitzung keine erneute Eingabe der Passphrase erforderlich ist.
3. Es wird die Meldung `[Starting SSL/TLS negotiation ...]` ausgegeben. Treten bei dieser TLS-Verhandlung keine Fehler auf, dann werden die Daten des FTP-Server-Zertifikats angezeigt:
   - Bezeichnung des Zertifikat-Inhabers
   - Bezeichnung des Zertifikat-Ausstellers
   - Gültigkeitsdauer
4. Wenn der im *open*-Kommando spezifizierte Host-Name des FTP-Servers nicht identisch mit dem im Zertifikat eingetragenen Namen ist, dann wird ein entsprechender Warnhinweis ausgegeben und Sie werden gefragt, ob Sie fortfahren möchten. Für das Fortfahren sollten Sie sich nur dann entscheiden, wenn Sie sich sicher sind, dass der durch das Zertifikat gekennzeichnete Server auch tatsächlich der gewünschte Server ist. Andernfalls können Sie Opfer eines so genannten „man in the middle“-Angriffs werden: Beim „man in the middle“-Angriff stellt sich der Angreifer zwischen Server und Client und gibt sich dem Client gegenüber als der gewünschte Server aus.

Wenn Hostname und Zertifikatseigentümer übereinstimmen oder wenn Sie eingewilligt haben, trotz Namensdifferenzen fortzufahren, wird die gewohnte Aufforderung angezeigt, den Benutzernamen für den FTP-Server einzugeben: Das weitere Vorgehen entspricht dann dem Vorgehen beim nicht durch TLS gesicherten *open*-Kommando.

passive - PASSIVE-Modus ein-/ausschalten

Das Kommando *passive* schaltet den PASSIVE-Modus ein bzw. aus. Der PASSIVE-Modus ist mittlerweile Standardeinstellung in vielen Implementierungen und dient dazu, aus einem lokalen Netz über eine Firewall ins Internet zu gelangen. Solche Firewalls unterbinden häufig den aktiven Verbindungsaufbau in das durch die Firewall geschützten Netz hinein.

Näheres zum Passive-Modus finden Sie im Handbuch „interNetServices Administratorhandbuch“.

passive
[**on** | **off**]

on
Der PASSIVE-Modus wird eingeschaltet.
Wenn kein Operand spezifiziert ist, entspricht dies der Angabe `passive on`.

off
Der PASSIVE-Modus wird ausgeschaltet.
off ist Voreinstellung, wenn keine passive-Anweisung angegeben ist.

Im PASSIVE-Modus werden alle nachfolgenden Dateitransfer-Kommandos mit aktivem Client und passivem Child abgewickelt, d.h. mittels PASV- bzw. EPSV-Kommandos. Dies gilt bis zur negativen Quittierung eines PASV- bzw. EPSV-Kommandos durch den Server, z.B. wenn die Verbindung zum Server beendet wurde.

Für das EPSV-Kommando auf Child-Seite gilt: Wenn Sie den Parameter ALL beim EPSV-Kommando angeben, werden alle darauffolgenden PORT- bzw. EPRT-Kommandos mit „522 PORT command not successful“ bzw. „522 EPRT command not successful“ abgewiesen.

Mit dem *status*-Kommando können Sie überprüfen, ob der PASSIVE-Modus eingestellt ist:

```
ftp> status
Connected to PGAB0021, port 21.
No proxy connection.
Passive Mode: on
...
```

Jedesmal wenn Sie das *passive*-Kommando eingeben, wird außerdem die neue Einstellung ausgegeben.

private - TLS-Absicherung der Datenverbindung ein-/ausschalten

Mit dem Kommando *private* kann der Schalter für die TLS-Absicherung der Datenverbindungen verändert werden.

Die aktuelle Einstellung dieser Option können Sie mit dem Kommando *status* ermitteln.

Datenverbindungen werden nur dann TLS-gesichert, wenn auch die Kontrollverbindung TLS-gesichert ist. Das Kommando *private* wird somit abgewiesen, wenn es bei bestehender, aber nicht TLS-abgesicherter Kontrollverbindung abgesetzt wird.

private
[**on** | **off**]

on
Alle nachfolgend aufgebauten Datenverbindungen werden mit TLS abgesichert.

off
Nachfolgend aufgebaute Datenverbindungen werden nicht mit TLS abgesichert.

Wenn kein Argument angegeben wird, wechselt der Schalter je nach Ausgangswert von *on* auf *off* bzw. von *off* auf *on*.

Wenn keine durch TLS abgesicherte Kontrollverbindung besteht, wird bei *private on* implizit *protect on* ausgeführt.

prompt - Rückfrage ein-/ausschalten

Bei den Kommandos *mdir*, *mget*, *mls*, *mput* und *mdelete* können in einem einzigen Aufruf mehrere Dateioperationen durchgeführt werden. Ist die Rückfrage eingeschaltet, wird der Anwender vor jeder Dateioperation gefragt, ob diese auch wirklich durchgeführt werden soll. Das Kommando *prompt* schaltet die Rückfrage ein bzw. aus. Ist die Rückfrage eingeschaltet, wird sie durch *prompt* ausgeschaltet und umgekehrt. Nach dem Start von FTP ist die Rückfrage eingeschaltet. Die aktuelle Einstellung dieser Option kann mit dem Kommando *status* ermittelt werden.

prompt

Bei eingeschalteter Rückfrage-Option sind auf die Rückfrage folgende Antworten möglich:

- j die Aktion wird durchgeführt
- y die Aktion wird durchgeführt
- n die Aktion wird nicht durchgeführt. Das Kommando wird nicht abgebrochen, sondern mit der nächsten Teilaktion fortgesetzt.
- q die Aktion wird nicht durchgeführt. Das gesamte Kommando wird abgebrochen.

i Eingaben ungleich *n* oder *q* werden wie *j* bzw. *y* gewertet.

Beispiel

Siehe Kommando *mdelete* auf Seite 150.

protect - TLS-Absicherung der Kontrollverbindung ein-/ausschalten

Mit dem Kommando *protect* kann der Schalter für die TLS-Absicherung der Kontrollverbindungen verändert werden.

Beim ersten Einschalten der TLS-Absicherung werden Sie nach der Meldung `[Initializing TLS ...]` eventuell aufgefordert, einige Zeichen in möglichst zufälliger Auswahl über Tastatur einzugeben oder nur die ENTER-Taste zu betätigen (siehe Option „-tlsRandomSeed - Pseudo-Zufallszahlengenerator initialisieren“ auf Seite 95). Auf diese Weise wird der Pseudo-Zufallszahlen-Generator initialisiert. Gegebenenfalls werden Sie erneut aufgefordert, Zeichen in möglichst zufälliger Auswahl über Tastatur einzugeben oder nur die ENTER-Taste zu betätigen, bis insgesamt genügend Material für eine Initialisierung zur Verfügung steht.

Die aktuelle Einstellung dieser Option können Sie mit dem Kommando *status* ermitteln.

Das Kommando *protect* hat keine Auswirkung auf eine aktuell bestehende Kontrollverbindung.

protect
[**on** | **off**]

on
Alle nachfolgend aufgebauten Kontrollverbindungen werden mit TLS abgesichert.

off
Nachfolgend aufgebaute Kontrollverbindungen werden nicht mit TLS abgesichert.

Wenn kein Argument angegeben wird, wechselt der Schalter je nach Ausgangswert von *on* auf *off* bzw. von *off* auf *on*.

proxy - Verbindung zu zwei fernen Rechnern steuern

Das Kommando *proxy* steuert gleichzeitig eine Verbindung zu zwei fernen Rechnern für die Übertragung von Dateien zwischen diesen beiden fernen Rechnern. Voraussetzung hierfür ist, dass der zweite ferne Rechner das Kommando PASV bzw. EPSV unterstützt.

proxy
<ftp-kommando>

<ftp-kommando>
Spezifiziert ein FTP-Client-Kommando:

- Um die Kontrollverbindung zum zweiten Rechner herzustellen, muss das erste <ftp-kommando> das Kommando *open* sein.
- Durch Eingabe von `proxy help` erhalten Sie die weiteren FTP-Kommandos angezeigt, die auf der sekundären Verbindung ausführbar sind.

Folgende Kommandos verhalten sich anders, wenn ihnen *proxy* vorangestellt ist:

- *get* und *mget* übertragen Dateien vom ersten Server auf den zweiten Server.
- *put*, *mput* und *append* übertragen Dateien vom zweiten Server auf den ersten Server.

Die folgende Abbildung skizziert den grundsätzlichen Ablauf der Übertragung einer Datei <datei> zwischen zwei fernen Servern A und B. C ist der Client.

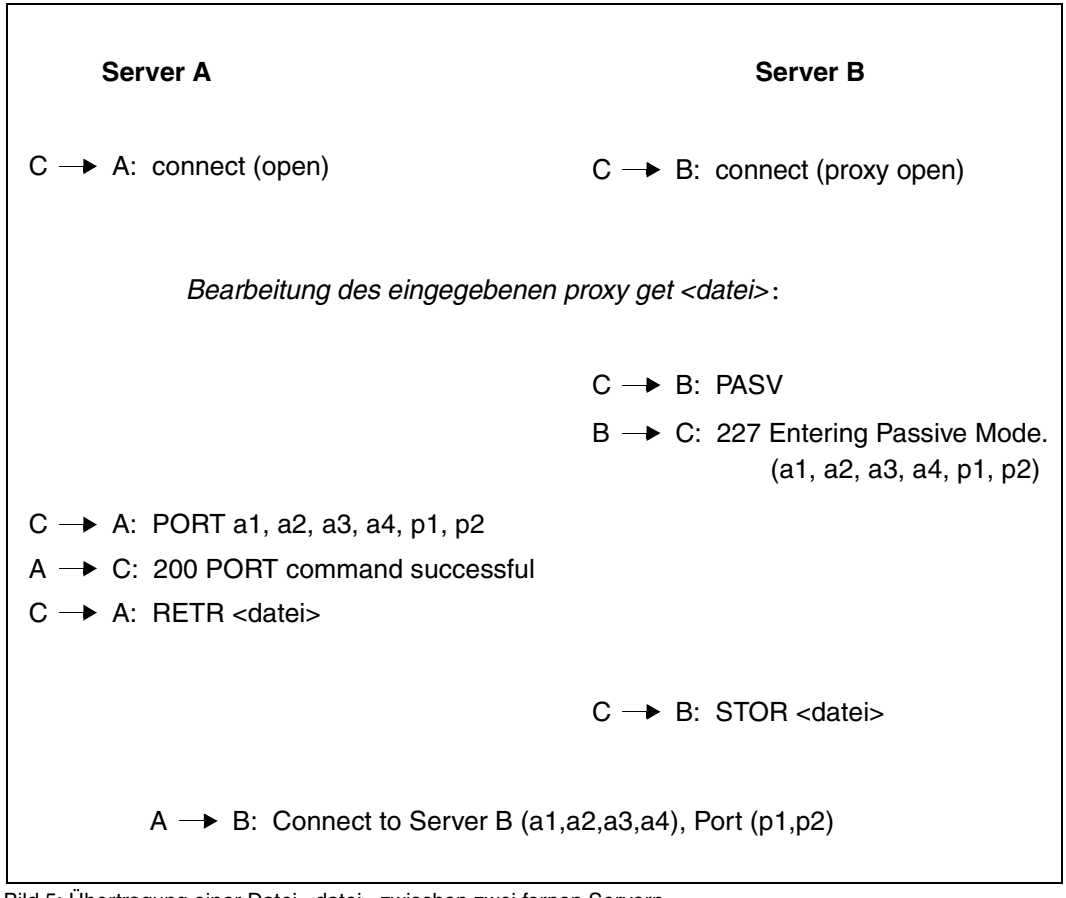


Bild 5: Übertragung einer Datei <datei> zwischen zwei fernen Servern

Beispiel

Alle Dateien der Kennung *userid1* auf dem remote *system1* sollen auf die Kennung *userid2* auf dem remote *system2* übertragen werden.

```
ftp> open system1        Öffne Kontrollverbindung zum 1. Server
userid1                  .
.                        .
.                        .
.                        .
ftp> proxy open system2  Öffne Kontrollverbindung zum 2. Server
userid2                  .
.                        .
.                        .
.                        .
ftp> proxy?              Welche proxy-Kommandos werden unterstützt?
ftp> proxy mget*         Übertrage Dateien system1 -> system2
ftp> proxy ls            Kontrolliere Ergebnis
ftp> proxy close         Schließe sekundäre Kontrollverbindung
```

Das *status*-Kommando liefert in diesem Fall folgende Ausgabe:

```
ftp> status
Connected to PGAB0021, port21.
Connected for proxy commands to system2.
Passive Mode: off
...
```

put - Senden einer lokalen Datei

Mit dem Kommando *put* wird eine Datei vom lokalen Rechner zum fernen Rechner übertragen. Dateien können auch mit den Kommandos *mput*, *append* und *send* zum fernen Rechner übertragen werden. Für die Restart-Unterstützung gibt es eine spezielle Ausprägung des *put*-Kommandos, das *reput*-Kommando (siehe Seite 185).

put
<lokale-datei> [<ferne-datei>]

<lokale-datei>
Name einer POSIX- oder DVS-Datei am lokalen Rechner, die zum fernen Rechner übertragen werden soll.

<ferne-datei>
Name einer Datei am fernen Rechner. Existiert die Datei bereits, wird sie entweder überschrieben oder - mit entsprechendem Suffix versehen - neu angelegt. Dieses Verhalten wird mit *sunique* gesteuert. Falls die Datei nicht existiert, wird sie neu angelegt. Fehlt der Operand *ferne-datei*, wird der Name der lokalen Datei verwendet (in diesem Fall muss der Name der lokalen Datei auch den Dateinamenskonventionen des fernen Rechners entsprechen). Per Default werden Großbuchstaben im Namen der lokalen Datei in Kleinbuchstaben umgewandelt. Diese Voreinstellung können Sie jedoch mithilfe des Client-Kommandos *setcase* (siehe Seite 193) deaktivieren.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des lokalen und des fernen Arbeits-Dateiverzeichnisses.

```
→ lpwd
   Local directory is :5:$TCPTEST.

→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Übertragen des Inhalts der lokalen Datei *MAN.SAM.NACH.SINIX.ANTON* in die ferne Datei *anton*. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ put man.sam.nach.sinix.anton anton
   200 PORT command okay.
   ...
```

pwd - Ausgabe des fernen Arbeits-Dateiverzeichnisses

Das Kommando *pwd* gibt den Namen des aktuell eingestellten fernen Arbeits-Dateiverzeichnisses aus. Die Einstellung des fernen Arbeits-Dateiverzeichnisses erfolgt mit dem Kommando *cd*.

pwd

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

Abfragen des fernen Arbeits-Dateiverzeichnisses.

```
→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

quit - FTP beenden

Das Kommando *quit* beendet das Programm FTP. Falls noch eine Verbindung zu einem fernen Rechner existiert, wird diese geschlossen (impliziter *close*). Das Kommando *bye* kann synonym zu *quit* eingegeben werden.

quit

Beispiel

```
→ quit
   221 Goodbye.
```

quote - Aufruf von Server-Funktionen

Mit dem Kommando *quote* können Funktionen des FTP-Servers am fernen Rechner aufgerufen werden, für die es kein Kommando des FTP-Client am eigenen Rechner gibt.

Ein Beispiel dafür ist die Funktion *site exec*, die der FTP-Server des BS2000/OSD anbietet, für die es aber kein Kommando im FTP gibt. Mit *site exec* können beliebige BS2000/OSD-Kommandos im Server zur Ausführung gebracht werden, sofern

- die FTAC-Funktionalität auf dem Server nicht eingesetzt ist bzw.
- die *site exec*-Funktion nicht via Server-Option *-disableSiteExecCommand* deaktiviert ist (siehe Handbuch „interNet Services Administratorhandbuch“).

Da die mit *quote* aufrufbaren Funktionen vom Server abhängen, muss an dieser Stelle auf eine weitere Beschreibung verzichtet werden. Funktionen des BS2000/OSD-Server sind in Abschnitt „FTP-Server im BS2000/OSD“ auf Seite 59 beschrieben.

Die im Server implementierten Funktionen können mit dem Kommando *remotehelp* abgefragt werden. Das Kommando *help* gibt eine Liste der im lokalen FTP-Client vorhandenen Kommandos aus.

quote
<argument> [<argument>] [<argument>]

Mit dem Kommando *quote site setc* können die Codekonvertierungs-Tabellen beim BS2000/OSD-Server geändert werden. Die Kommandosyntax entspricht der von *setcode* (siehe Seite 194).

Beispiel

Der ferne Rechner ist ein BS2000/OSD-Rechner.

1. Mit dem Kommando *quote CWD* wird das Dateiverzeichnis am fernen Server gewechselt.

```
→ quote CWD test
   200 ":110:$TSOS.TEST." is current directory now.
```

2. Mit dem Kommando *quote PWD* wird das aktuelle Dateisystem am fernen FTP-Server ausgegeben.

```
→ quote PWD
   257 ":110:$TSOS.TEST." is current directory.
```

readopt - Option-Datei einlesen

Mit dem Kommando *readopt* wird eine Option-Datei eingelesen. Das Kommando ergänzt/ersetzt die Optionen, die beim Start des FTP-Client aus globaler und lokaler Option-Datei gelesen wurden (sofern diese Dateien zum Start-Zeitpunkt existierten). Das Kommando *readopt* können Sie beliebig oft ausführen, um z.B. Option-Sätze zusammenzusetzen, die auf verschiedene Dateien verteilt sind, oder um zwischen unterschiedlichen Option-Sätzen zu wechseln.

Das Kommando *readopt* sollte möglichst nicht bei einer bestehenden FTP-Verbindung ausgeführt werden. Andernfalls kann es je nach Inhalt der Option-Datei zu Inkonsistenzen zwischen den Einstellungen auf dem lokalen und dem fernen Rechner kommen.

Beachten Sie bei einigen TLS-Optionen, dass eine Änderung nach der TLS-Initialisierung wirkungslos bleibt. Wenn Sie mit solchen veränderten TLS-Optionen arbeiten wollen, müssen Sie den FTP-Client beenden und anschließend neu starten.

readopt
<option-datei>

<option-datei 1 .. 54>
Name der Option-Datei

recv - Holen einer Datei

Mit dem Kommando *recv* wird eine Datei vom fernen Rechner zum lokalen Rechner übertragen.

recv
<ferne-datei> [<lokale-datei>]

Siehe Kommando *get* auf Seite 136.

reget - Holen einer Datei mit Restart-Unterstützung

Das Kommando *reget* ist eine spezielle Ausprägung des *get*-Kommandos (siehe Seite 136) und unterstützt die Restart-Fähigkeit des FTP-Clients.

Dabei ist darauf zu achten, dass für *mode*, *type*, *struct* und *ftyp* dieselben Einstellungen vorgenommen werden, wie sie beim abgebrochenen *get*-Aufruf vorlagen.

Analog zum *get*-Kommando überträgt das Kommando *reget* eine Datei vom fernen Rechner zum lokalen Rechner. Beim *reget*-Kommando wird jedoch vor dem Start des Transfers geprüft, ob die Zieldatei (lokale Datei) bereits existiert.

Sofern die Zieldatei bereits existiert, sind zwei Fälle zu unterscheiden:

- Die Ausführung des *reget*-Kommandos unterstellt, dass die Datei bereits teilweise übertragen wurde, und setzt den Transfer an der Stelle fort, die dem aktuellen Ende der Zieldatei entspricht, wenn die folgenden Bedingungen erfüllt sind:
  - Die Zieldatei ist kleiner und wurde später erstellt als die Quelldatei.
  - Der FTP-Server unterstützt die für die Prüfung und Positionierung notwendigen Server-Kommandos *mdtm*, *size* und *rest* (siehe Seite 68).
  - Als Übertragungsmodus ist "stream" eingestellt (siehe *mode*-Kommando auf Seite 161).
  - Die *ftyp*-Einstellung ist *textbin*.
  - Die *setfile*-Einstellung ist *datend: on*, *pademptyrec: off*
  - *runique* ist ausgeschaltet.
- Andernfalls entspricht die Ausführung des *reget*-Kommandos der Ausführung von *get*, d.h. die Quelldatei wird vollständig übertragen.

Bei der Kommandoausführung wird die lokale Datei nicht auf den korrekten Datei-Inhalt bzw. auf die korrekten Datei-Attribute überprüft.

reget
<ferne-datei> [<lokale-datei>]

<ferne-datei>
Name einer Datei am fernen Rechner, die zum lokalen Rechner übertragen werden soll.

<lokale-datei>
Name einer POSIX- oder DVS-Datei am lokalen Rechner. Falls die Datei nicht existiert, wird sie neu angelegt und das *reget*-Kommando verhält sich wie das *get*-Kommando. Fehlt der Operand *lokale-datei*, wird der Name der fernen Datei verwendet (in diesem Fall muss der Name der fernen Datei auch den Dateinamenskonventionen des lokalen Rechners entsprechen).

remotehelp - Information zu Funktionen des fernen FTP-Server

Das Kommando *remotehelp* gibt eine Liste aller am FTP-Server des fernen Rechners erkannten Funktionen aus. Die einzelnen Server-Funktionen können als Kommandos ausgegeben werden. Server-Funktionen, für die kein entsprechendes Kommando im FTP-Client implementiert ist, können mit dem Kommando *quote* aufgerufen werden. Eine Liste aller Kommandos des lokalen FTP-Client ist mit dem Kommando *help* erhältlich.

remotehelp
<server-fkt>

<server-fkt>
FTP-Server-Funktion, zu der Informationen ausgegeben werden soll.
Ist kein Operand angegeben, wird eine Liste aller erkannten Funktionen ausgegeben.

Beispiel

Der ferne Rechner ist ein BS2000/OSD-Rechner.

Auflistung der FTP-Server Funktionen. Die mit * versehenen Funktionen sind nicht implementiert.

```
→ remotehelp
   214-The following commands are recognized (* =>'s unimplemented).
    USER     EPRT     APPE     MRCP*    XCWD     MKD      XCUP     OPTS     MODC
    PASS     EPSV     MLFL*    ALLO     LIST     XMKD     STOU     AUTH     SETC
    ACCT     TYPE     MAIL*    RNFR     NLST     RMD      SYST     PBSZ
    REIN*    STRU     MSND*    RNTO     SITE     XRMD     REST     PROT
    QUIT     MODE     MSOM*    ABOR     STAT     PWD      MDTM     CCC
    PORT     RETR     MSAM*    DELE     HELP     XPWD     SIZE     FILE
    PASV     STOR     MRSQ*    CWD      NOOP     CDUP     FEAT     FTYP
   214 Direct comments to TSOS at BCAMVM06.
→ remotehelp site
   214-The following SITE commands are recognized (* =>'s unimplemented).
    CMOD     EXEC     FILE     FTYP     HELP     MODC     SETC     SFIL     EXIT
   214 Direct comments to TSOS at BCAMVM06.
```

rename - Umbenennen von fernen Dateien

Mit dem Kommando *rename* wird eine Datei am fernen Rechner umbenannt.

rename
<ferne-datei1> <ferne-datei2>

<ferne-datei1>
Alter Name der Datei.

<ferne-datei2>
Neuer Name der Datei.
Eine bereits bestehende Datei wird i.a. überschrieben, nicht aber im DVS-Dateiverzeichnis.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des fernen Arbeits-Dateiverzeichnisses.

```
→ pwd
   257 "/usr/tcptest/man/sam/to.bs2000" is current directory.
```

2. Umbenennen der fernen Datei *anton* in *zwaton*.

```
→ rename anton zwaton
   350 File exists, ready for destination name
   250 RNTO command successful.
```

reput - Senden einer lokalen Datei mit Restart-Unterstützung

Das Kommando *reput* ist eine spezielle Ausprägung des *put*-Kommandos (siehe Seite 175) und unterstützt die Restart-Fähigkeit des FTP-Clients.

Dabei ist darauf zu achten, dass für *mode*, *type*, *struct* und *ftyp* dieselben Einstellungen vorgenommen werden wie beim abgebrochenen *put*-Aufruf vorlagen.

Analog zum *put*-Kommando überträgt das Kommando *reput* eine Datei vom lokalen Rechner zum fernen Rechner. Beim *reput*-Kommando wird jedoch vor dem Start des Transfers geprüft, ob die Zieldatei (Datei auf dem Server) bereits existiert.

Sofern die Zieldatei bereits existiert, sind zwei Fälle zu unterscheiden:

- Die Ausführung des *reput*-Kommandos unterstellt, dass die Datei bereits teilweise übertragen wurde, und setzt den Transfer an der Stelle fort, die dem aktuellen Ende der Zieldatei entspricht, wenn die folgenden Bedingungen erfüllt sind:
  - Die Zieldatei ist kleiner und wurde später erstellt als die Quelldatei.
  - Der FTP-Server unterstützt die für die Prüfung und Positionierung notwendigen Server-Kommandos *mdtm*, *size* und *rest* (siehe Seite 68).
  - Der Übertragungsmodus ist "stream" (siehe *mode*-Kommando auf Seite 161).
  - Die *ftyp*-Einstellung ist "textbin".
  - Die *setfile*-Einstellung ist "datend: on, pademptyrec: off"
  - *sunique* ist ausgeschaltet.
- Andernfalls entspricht die Ausführung des *reput*-Kommandos der Ausführung von *put*, d.h. die Quelldatei wird vollständig übertragen.

Bei der Kommandoausführung wird die ferne Datei nicht auf den korrekten Datei-Inhalt bzw. auf die korrekten Datei-Attribute überprüft.

reput
<lokale-datei> [<ferne-datei>]

<lokale-datei>
Name einer POSIX- oder DVS-Datei am lokalen Rechner, die zum fernen Rechner übertragen werden soll.

<ferne-datei>
Name einer POSIX- oder DVS-Datei am fernen Rechner. Falls die Datei nicht existiert, wird sie neu angelegt, und das *reput*-Kommando verhält sich wie das *put*-Kommando. Fehlt der Operand *ferne-datei*, so wird der Name der lokalen Datei verwendet (in diesem Fall muss der Name der lokalen Datei auch den Dateinamenskonventionen des fernen Rechners entsprechen). Großbuchstaben im Namen der lokalen Datei werden in Kleinbuchstaben umgesetzt (siehe auch FTP-Kommando *setcase* auf Seite 193).

rexit - Parameter für ferne Exit-Routine definieren

Im FTP-Client haben Sie die Möglichkeit, benutzerdefinierte BS2000/OSD-FTP-Server-Exits mit dem Kommando *rexit* (remote exit) einzustellen.

Nähere Informationen zu FTP-Server-Exit-Routinen finden Sie im Handbuch „interNet Services Administratorhandbuch“.

rexit
[receive:<receive-selector>]![send:<send-selector>]

<receive-selector>
Mit <receive-selector> können Sie unter mehreren beim Empfang der Daten (aus Server-Sicht) zur Verfügung stehenden Aktionen die gewünschte Aktion auswählen. Zu diesem Zweck versorgen Sie die Exit-Routine mit einem String, auf den <receive-selector> verweist. Dabei können Sie frei entscheiden, welche Strings mit welchen Bedeutungen verwendet werden. Lediglich den Strings „*“ und „*NONE“ ist eine feste Bedeutung zugeordnet:

- Bei Angabe von „*“ für <receive-selector> wird die Exit-Routine wieder mit dem in der FTP-Server-Option *-U* eingestellten <receive-selector> aufgerufen. Wenn in dieser Server-Option kein <receive-selector> eingestellt ist, ist die Angabe von „*“ gleichbedeutend mit der Angabe von „*NONE“.
- Bei Angabe von „*NONE“ wird die Exit-Routine deaktiviert.

<send-selector>
Mit <send-selector> können Sie unter mehreren beim Senden der Daten (aus Server-Sicht) zur Verfügung stehenden Aktionen die gewünschte Aktion auswählen. Zu diesem Zweck versorgen Sie die Exit-Routine mit einem String, auf den <send-selector> verweist. Dabei können Sie frei entscheiden, welche Strings mit welchen Bedeutungen verwendet werden. Lediglich den Strings „*“ und „*NONE“ ist eine feste Bedeutung zugeordnet:

- Bei Angabe von „*“ für <send-selector> wird die Exit-Routine wieder mit dem in der FTP-Server-Option *-U* eingestellten <*send-selector*> aufgerufen. Wenn in dieser Server-Option kein <*send-selector*> eingestellt ist, ist die Angabe von „*“ gleichbedeutend mit der Angabe von „*NONE“.
- Bei Angabe von „*NONE“ wird die Exit-Routine deaktiviert.

rmdir - Löschen eines fernen Dateiverzeichnisses

Mit dem Kommando *rmdir* wird ein leeres Dateiverzeichnis am fernen Rechner gelöscht.

rmdir
<fernes-dateiverzeichnis>

<fernes-dateiverzeichnis>
Vollständiger oder relativer Name eines Dateiverzeichnisses am fernen Rechner.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Ausgeben des Inhalts des aktuellen Arbeits-Dateiverzeichnisses. Implizit wird die FTP-Server-Funktion PORT aufgerufen.

```
→ dir
   200 PORT command okay.
   ...
   -rwx------ 1 tcptest 10505 Sep 5 18:20 alaska
   -rw-rw-r-- 1 tcptest 229   Sep 7 13:56 anton
   -rw-rw-r-- 1 tcptest 229   Sep 7 13:56 anton.3
   -rw-rw-r-- 1 tcptest 229   Sep 7 13:56 anton.upd
   -rwx------ 1 tcptest 10505 Sep 5 18:20 georgia
   drwxrwxrwx 2 tcptest 176   Sep 7 12:55 nach.bs2000
   drwxrwxrwx 2 tcptest 176   Sep 7 13:51 von.bs2000
   drwxrwxrwx 2 tcptest 32    Sep 7 14:40 von.msdos
   ...
```

2. Löschen des (leeren) Dateiverzeichnisses *von.msdos*.

```
→ rmdir von.msdos
   250 RMD command successful.
```

runique - Eindeutiges Abspeichern lokaler Dateien

Das Kommando *runique* steuert die Behandlung von Dateien beim Abspeichern auf dem lokalen Rechner, d.h. wenn mit *get*, *mget* oder *recv* gearbeitet wurde. Die Angabe *runique* schaltet diese Behandlung ein bzw. aus. Die festgelegte Einstellung gilt bis zur nächsten Angabe von *runique*.

Nach dem Start von FTP ist *runique* standardmäßig ausgeschaltet.

runique ausgeschaltet: Existiert eine Datei gleichen Namens, wird sie, entsprechende Schreibberechtigung vorausgesetzt, überschrieben.

runique eingeschaltet: Existiert die zu übertragende Datei bereits, wird sie mit einem Suffix (.1 bis .99) versehen neu angelegt. Sollten die Suffixe .1 bis .99 für diesen Dateinamen bereits belegt sein, wird eine Fehlermeldung ausgegeben und die Übertragung abgebrochen.

runique

send - Senden einer lokalen Datei

Mit dem Kommando *send* werden Dateien vom lokalen Rechner zum fernen Rechner übertragen.

send
<lokale-datei> [<ferne-datei>]

Siehe Kommando *put* auf Seite 175.

sendport - Port-Kommando ein-/ausschalten

Das Kommando *sendport* legt fest, ob FTP am Beginn einer Datei-Übertragung die Funktion PORT des Servers des fernen Rechners aufrufen soll oder nicht. Ist die PORT-Funktion eingeschaltet, dann wird sie durch *sendport* ausgeschaltet und umgekehrt. Nach dem Start von FTP ist die PORT-Funktion eingeschaltet.

Normalerweise ist die Funktion PORT in jedem Server implementiert. Es gibt jedoch Server, die sich nicht normgerecht verhalten und PORT nicht implementiert haben. Diese reagieren auf ein PORT-Kommando mit der Fehlermeldung

```
550: Command not understood
```

oder

```
502: PORT command not implemented
```

In diesem Fall muss die PORT-Funktion ausgeschaltet werden.

sendport

set - Setzen und Anzeigen von Variablen

Mit dem Kommando *set* können Variablen gesetzt und angezeigt werden, die das Verhalten des FTP-Clients beeinflussen.

Im Augenblick wird nur die Variable *sizeCmdTimeLimit* mit diesem Kommando verändert, die das Verhalten des *reget*- und *reput*-Kommandos beeinflusst.

Bei einem *reget*- oder *reput*-Kommando wird u.a. die Größe der lokalen Datei bestimmt, um einen Wiederaufsetzpunkt für den Datentransfer zu finden. Insbesondere wenn der Übertragungstyp nicht binary ist, kann diese Größenbestimmung sehr zeitaufwendig sein, so dass sich durch die Verwendung von *reget*/*reput* im Vergleich zu *get*/*put* kaum ein Zeitvorteil ergibt.

Mit der Variablen *sizeCmdTimeLimit* kann die Dauer der Größenbestimmung begrenzt werden, d.h. beim Überschreiten des Werts wird die Größenbestimmung abgebrochen und in Folge davon statt eines *reget*-/*reput*-Kommandos das entsprechende *get*-/*put*-Kommando ausgeführt.

Der aktuell gültige Wert kann durch das *set*-Kommando ohne Parameter oder durch das *status*-Kommando abgefragt werden.

set
`sizeCmdTimeLimit` <Wert>

sizeCmdTimeLimit <Wert>
: Maximale Dauer der Größenbestimmung in Sekunden
Die Voreinstellung ist 60 Sekunden.

setcase - Groß-/Kleinschreibung der Dateinnamen im Zielsystem

Das Kommando *setcase* legt die Groß-/Kleinschreibung in den Dateinamen der mit *put* und *mput* transferierten Dateien im Zielsystem fest.

setcase
[**all** | **posix** | **off**]

all
Beim Transfer aus BS2000/OSD und POSIX ins Zielsystem bleiben Groß- und Kleinbuchstaben in den Dateinamen erhalten. Für den Transfer aus BS2000/OSD bedeutet dies Großschreibung der Dateinamen.
Wenn kein Operand spezifiziert ist, ist dies gleichbedeutend mit der Angabe `setcase all`.

posix
Nur beim Transfer aus POSIX ins Zielsystem bleiben Groß- und Kleinbuchstaben in den Dateinamen erhalten. Namen von Dateien, die aus BS2000/OSD ins Zielsystem transferiert werden, werden in Kleinbuchstaben geschrieben.

off
Die Dateinamen im Zielsystem werden in Kleinbuchstaben geschrieben.
off ist Voreinstellung, wenn keine *setcase*-Anweisung angegeben wird.

setcode - Code-Tabellen wechseln

Mit *setcode* können die aktuell eingestellten Code-Tabellen, mit denen der FTP-Client die Umsetzung von EBCDIC- auf ISO-Zeichen (erweiterter ASCII-Zeichensatz) durchführt, gewechselt werden. Dabei werden die Dienste von XHCS genutzt, d.h. es können nur Code-Tabellen angegeben werden, die in XHCS als kompatibel eingetragen sind (siehe Handbuch „XHCS“). Bei einer Verbindung zwischen zwei BS2000/OSD-Systemen ist darauf zu achten, dass beide Systeme die gleichen Code-Tabellen nutzen.

Die aktuell eingestellten Code-Tabellen können mit dem Kommando *status* ermittelt werden.
Voreinstellung: EDF041 für EBCDIC
ISO88591 für ASCII

setcode
<ebcdic-tabelle> <iso-tabelle>

<ebcdic-tabelle> <iso-tabelle>
Zwischen EBCDIC- und ISO-Tabelle wird eine Codekonvertierungs-Tabelle mit XHCS erzeugt, die vom FTP-Client für sämtliche Codekonvertierungen verwendet wird.

ACHTUNG!

Die Code-Tabellen dürfen keinesfalls in umgekehrter Reihenfolge eingegeben werden, weil dadurch falsche Konvertierungstabellen erzeugt werden und somit keine Dateiübertragung mehr möglich ist.

Beispiel

```
setcode EDF045 ISO88595
```

Anmerkungen

Es wird die Umwandlungstabelle zwischen der EBCDIC-Tabelle EDF045 und der ISO-Tabelle ISO88595 über XHCS bereitgestellt. Die verwendbaren Umsetztabellen sind über XHCS definiert. Aus dieser Umwandlungstabelle wird die entsprechende Umwandlungstabelle in Gegenrichtung erzeugt und die zuletzt gültige Konvertierungstabelle überschrieben.

setfile - Datei-Marker ein-/ausschalten

Mithilfe des Kommandos *setfile* können Sie das Verhalten von FTP beim Datentransfer in bestimmten Fällen festlegen:

- Der String "C-DATEIENDE" wird normalerweise benutzt, um das genaue Ende einer PAM-Datei zu markieren. Soll die Datei von Programmen weiter verarbeitet werden, bei denen diese Markierung zu Problemen führt, dann muss das Anfügen eines Markers abgeschaltet werden.
- Sätze ohne Inhalt (Leersätze) können von FTP mit einem Leerzeichen versehen werden, damit der EDT die Sätze bei der Ausgabe auf Terminal berücksichtigt.

setfile
[datend on | off] [pademptyrec on | off]

datend on | off
: schaltet "C-DATEIENDE ein bzw. aus.
on ist Voreinstellung.

pademptyrec on | off
: schaltet die Blank-Einfügung in Leersätze ein bzw. aus.
off ist Voreinstellung.

Wenn Sie *setfile* ohne Parameter angeben, gibt FTP die aktuellen Einstellungen für *datend* und *pademptyrec* aus.

Wenn Sie *datend* und *pademptyrec* beim FTP-Server einstellen wollen, geben Sie *quote site sfil* an. Die Parameter entsprechen den Parametern von *setfile*.

Das Kommando *status* (siehe Seite 197) liefert ebenfalls Informationen zu den Einstellungen von *datend* und *pademptyrec*.

settime - Einstellen der Überwachungszeit für Server-Antworten

Mit dem Kommando *settime* kann die Überwachungszeit für Server-Antworten (Standardeinstellung: 30 sec) modifiziert werden. Ist eine vom Client erwartete Server-Antwort nicht in der vorgegebenen Zeit eingetroffen, meldet der Client

```
time limit for server response exceeded.
```

Die aktuell eingestellte Überwachungszeit kann mit dem Kommando *status* ermittelt werden.

settime <sec>

status - Ausgabe von FTP-Status-Informationen

Das Kommando *status* gibt Informationen über die aktuelle Verbindung, eine eventuelle TLS-Sicherung dieser Verbindung sowie diverse lokal eingestellte Parameter aus. Zudem werden auch Informationen über die eingestellten Code-Tabellen, die Zeichenfolgen zum Wechsel zwischen DVS- und POSIX-Dateisystem und das aktuell eingestellte Dateisystem (DVS bzw. POSIX) ausgegeben.

status

Beispiel

```
→ status
   Not connected.
   No proxy connection.
   Passive Mode: off
   Mode: stream; Type: ascii; Form: non-print; Structure: file
   Copymode: off; Ftyp: textbin
   Verbose: on; Bell: off; Prompting: on; Globbing: on
   Filesystem is: BS2000
   The character to change the filesystem is: %
   ISO-codetable is ISO88591, EBCDIC-codetable is EDF041
   Time limit for server responses: 30 secs
   Time limit for file size determination: 60 secs
   Store unique: off; Receive unique: off
   Use special EOF marker (C-DATEIENDE): on
   Pad empty record with blank: off
   Case sensitivity: Off
   Used job variable: *NONE
   Used SDF-P variable: *NONE
   Used receive selector: *NONE
   Used send selector: *NONE
   Hash mark printing: off; Use of PORT cmds: on
   Protected control channel: off
   Private data channel: off
   Cipher: (not connected)
```

Die drei letzten Zeilen zeigen an, dass die TLS-Absicherung der Kontroll- und Datenverbindungen ausgeschaltet und aktuell keine Kontrollverbindung aufgebaut ist.

Nachfolgend sind zwei weitere mögliche Status-Ausgaben zur TLS-Absicherung aufgelistet:

- Status-Ausgabe bei ungesicherter Kontrollverbindung:

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

- Status-Ausgabe bei einer mit 3-DES abgesicherten Kontrollverbindung:

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

Bei diesen Beispielen ist die TLS-Absicherung der Datenverbindungen ausgeschaltet, erkennbar an der folgenden Anzeige: `Private data channel: off`

Auch im Fall „`Private data channel: on`“ wird die TLS-Absicherung der Datenverbindungen nur dann durchgeführt, wenn auch die Kontrollverbindung abgesichert ist.

struct - Übertragungsstruktur ändern oder abfragen

Das Kommando *struct* stellt die Übertragungsstruktur ein bzw. gibt die aktuell eingestellte Übertragungsstruktur aus. Die FTP-Norm stellt optional verschiedene Übertragungsstrukturen zur Verfügung; davon sind im BS2000/OSD FILE und RECORD implementiert. Ist kein Operand angegeben, wird die aktuelle Übertragungsstruktur am Bildschirm ausgegeben.

Nach dem Start von FTP steht *struct* standardmäßig auf *file*.

Über die aktuell eingestellte Übertragungsstruktur gibt das Kommando *status* Auskunft.

struct
[**file** | **record**]

file
: Stellt die Übertragungsstruktur FILE ein.

record
: Diese Angabe erlaubt es, Dateien unter Beibehaltung der Satzstruktur zu übertragen. Voraussetzung für die Angabe von *record* sind

  – Übertragungstyp EBCDIC bzw. ASCII und
  – die Dateieigenschaften RECFORM=V und FCBTYPE=SAM.

sunique - Eindeutiges Abspeichern ferner Dateien

Das Kommando *sunique* steuert die Behandlung von Dateien beim Abspeichern auf dem fernen Rechner, d.h. wenn mit *put*, *mput* oder *send* gearbeitet wurde. Die Angabe *sunique* schaltet diese Behandlung ein bzw. aus. Die festgelegte Einstellung gilt bis zur nächsten Angabe von *sunique*.

Nach dem Start von FTP ist *sunique* standardmäßig ausgeschaltet.

sunique ausgeschaltet: Existiert eine Datei gleichen Namens, wird sie - entsprechende Schreibberechtigung vorausgesetzt - überschrieben.

sunique eingeschaltet: Existiert die zu übertragende Datei bereits, wird sie, mit einem Suffix (.1 bis .99) versehen, neu angelegt. Sollten die Suffixe .1 bis .99 für diesen Dateinamen bereits belegt sein, wird eine Fehlermeldung ausgegeben und die Übertragung abgebrochen.

sunique

svar - Fehlerinformationen in einer SDF-P-Variablen hinterlegen

Mit dem Kommando *svar* können Sie dem FTP-Client mitteilen, ob er Kommando-Return-Information in einer SDF-P-Variablen hinterlegen soll.

Eine Alternative zur Hinterlegung von Fehlerinformationen bietet das Kommando *jobvar* (siehe Seite 141).

Tritt bei der Bearbeitung des *svar*-Kommandos ein Fehler auf und ist der Schalter 1 gesetzt (Batch-Berieb), dann wird der FTP-Client mit TERMJ beendet. Der Batch-Job bzw. die Prozedur wird somit erst nach der nächsten STEP-Anweisung fortgesetzt.

Wenn die Versorgung einer S-Variablen aktiviert ist, verändert sich das Verhalten von Kommandos, die mehrere Einzelaktionen ausführen (Kommandos *mdir*, *mls*, *mget*, *mput*, *mdelete*). Bei diesen Kommandos wird die Kommando-Verarbeitung nach der ersten fehlerhaften Aktion abgebrochen. Dagegen wird der FTP-Client im Batch-Betrieb nicht mit TERMJ beendet, wenn die Berechtigungsdaten beim *open*-Kommando fehlerhaft sind.

svar
<sv-name> | ***NONE**

<sv-name>
: Name des SDF-P-Variablen (S-Variable), die der FTP-Client nach Absetzen des *svar*-Kommandos mit den Kommando-Return-Informationen versorgen soll. Hierzu wird diese S-Variable neu erstellt. Falls bereits eine S-Variable mit dem Namen <sv-name> existiert, wird sie zunächst gelöscht.

***NONE**
: Bei Angabe von *NONE wird die Versorgung der S-Variablen gestoppt.

Layout der S-Variablen

Element	Elementbeschreibung
<sv-name>.STATUS	$S: Kommando erfolgreich ausgeführt $E: Kommando mit Fehler beendet $T: FTP-Client normal beendet $A: FTP-Client wegen Fehler abnormal beendet
<sv-name>.USERCMD	Name des FTP-Client-Kommandos
<sv-name>.CMDPARAM	FTP-Kommando-Parameter
<sv-name>.PROTCMD	vom FTP-Client an den FTP-Server gesendetes Kommando
<sv-name>.MESSAGE	lokale Meldung bzw. Server-Antwort

system - Ausgabe von Server-Informationen

Mit dem Kommando *system* wird Systeminformation vom Server angefordert. Dieser gibt dann folgende Meldung aus:

```
215: <operating system>  <additional information>
```

system

tenex - Übertragungstyp BINARY einschalten

Das Kommando *tenex* stellt den Übertragungstyp BINARY ein. Dieser Übertragungstyp sollte immer dann gewählt werden, wenn Binärdateien zu übertragen sind. Das Kommando *binary* ist äquivalent zum Kommando *tenex*.

Nach dem Start von FTP ist der Übertragungstyp ASCII voreingestellt.

tenex

trace - SOCKET-Trace-Ausgaben ein-/ausschalten

Die SOCKET-Trace-Ausgaben dienen vor allem dem Netzadministrator und dem Kundendienst zur Diagnose von Netzproblemen. Der normale Anwender benötigt in der Regel keine SOCKET-Trace-Ausgaben.

Sind die SOCKET-Trace-Ausgaben eingeschaltet, gibt FTP verschiedene Diagnose-Informationen am Bildschirm aus.

trace
[<trace-wert>]

<trace-wert>
Zulässig sind Werte zwischen 0 und 9. Je größer der eingestellte Wert ist, desto mehr Informationen werden ausgegeben.

0 Keine TRACE-Ausgaben (TRACE-Ausgaben ausschalten).

1 - 9 Ausgabe von Informationen der SOCKET-Traces.

Wird kein Operand angegeben, wird die TRACE-Ausgabe umgeschaltet, d.h. ist die TRACE-Ausgabe eingeschaltet, so wird sie ausgeschaltet, ist sie ausgeschaltet, so wird sie eingeschaltet (*trace-wert*=1). Ein Wert größer als 1 impliziert alle TRACE-Ausgaben mit kleinerem Level.

type - Übertragungstyp ändern oder abfragen

Das Kommando *type* stellt den Übertragungstyp (ASCII, BINARY oder EBCDIC) ein. Wird das Kommando ohne Operanden angegeben, gibt es den aktuell eingestellten Übertragungstyp aus.

Nach dem Start von FTP ist der Übertragungstyp ASCII voreingestellt.

Der Übertragungstyp kann auch mit den Kommandos *ascii* und *binary* geändert werden. Mit dem Kommando *status* kann der aktuell eingestellte Übertragungstyp ermittelt werden. Der Übertragungstyp ist genauer ab Seite 81 beschrieben. Eine Tabelle zur ASCII / EBCDIC-Übersetzung finden Sie im Handbuch „XHCS“.

Beim Abbau einer Verbindung und Aufbau einer neuen Verbindung muss für einen anderen Übertragungstyp als ASCII das Kommando *type* neuerlich abgesetzt werden.

type
[**ascii** | **binary** | **ebcdic** | **tenex**]

ascii
Stellt den Übertragungstyp ASCII ein. Dieser Übertragungstyp sollte immer für Textdateien gewählt werden.

binary
Stellt den Übertragungstyp BINARY ein. Dieser Übertragungstyp sollte immer bei Binärdateien gewählt werden.

ebcdic
Stellt den Übertragungstyp EBCDIC ein. Dieser Übertragungstyp sollte gewählt werden, wenn beide Partner mit EBCDIC arbeiten, d.h. keine Codeumsetzung gewünscht ist und im BS2000/OSD eine SAM-Datei erzeugt werden soll.

tenex
Entspricht dem Übertragungstyp BINARY.

Beispiel

1. Übertragungstyp abfragen.

```
→ type
   Using ascii mode to transfer files.
```

2. Übertragungstyp auf BINARY setzen.

```
→ type binary
   200 Type set to I.
```

3. Übertragungstyp abfragen.

```
→ type
   Using binary mode to transfer files.
```

user - Benutzerkennung am fernen Rechner angeben

Mit dem Kommando *user* können Sie in folgenden Fällen die Berechtigungsdaten nachträglich eingeben:

- Am fernen Rechner ist ein Mechanismus zur Zugangskontrolle implementiert.
- Der Auftragsschalter 1 ist nicht gesetzt, oder die Hinterlegung von Fehlerinformationen (Kommandos *jvar*, *svar*) ist aktiviert.
- Die notwendigen Berechtigungsdaten beim Kommando *open* wurden fehlerhaft eingegeben.

user
[<kennung> [<password> [<account>]]]

<kennung>
Benutzerkennung am fernen Rechner

<password>
Passwort am fernen Rechner
Kennwörter werden entsprechend den Konventionen des Partnerrechners angegeben. Kennwörter können für BS2000/OSD-Partner als C-Strings (c'...' sowie C'...') oder als Sedezimalzahlen (x'...' sowie X'...') angegeben werden. BS2000/OSD akzeptiert das Passwort sowohl in der Form <password> als auch '<password>'. Ist kein Passwort nötig, so wird "*NONE" angegeben.

<account>
Abrechnungsnummer am fernen Rechner

Eventuell zusätzlich erforderliche Berechtigungsdaten (Passwort, Abrechnungsnummer) werden bei Bedarf im Dialog abgefragt.

Beispiel

Der ferne Rechner ist ein Linux-Rechner.

1. Herstellung der Verbindung zu anlaged durch *open*.

```
→ open anlaged
   Connected to anlaged.
   220 anlaged FTP Server (vsFTPd 2.0.5).
```

2. Wahl der Login-Kennung *tcptest*.

```
   Name (anlaged:TCPTEST):
→ tcptest
```

3. Angabe eines falschen Passworts.

```
   331 Password required for tcptest.
   Password (anlaged:tcptest):
→ maewest

   530 Login failed.
   Login failed.
```

4. Wiederholung des Login.

```
→ user tcptest
```

5. Angabe des richtigen Passworts; erfolgreiches Login.

```
   331 Password required for tcptest.
   Password:
→ Kr!fm3(z
   230 Login successful.
```

verbose - Ein-/Ausschalten der Server-Antworten

Mit dem Kommando *verbose* wird die Ausgabe der Antworten des FTP-Servers ein- bzw. ausgeschaltet. Ist die Antwort-Ausgabe eingeschaltet, wird sie durch *verbose* ausgeschaltet und umgekehrt. Nach dem Start von FTP ist *verbose* eingeschaltet.

Der aktuell eingestellte Wert von *verbose* kann mit dem Kommando *status* ermittelt werden.

verbose

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Abfragen des fernen Arbeits-Dateiverzeichnisses mit Rückmeldung.

```
→ pwd
   257 "/usr/tcptest/man/sam/nach.bs2000" is current directory.
```

2. Übertragen mehrerer Dateien mit Server-Meldungen. Implizit wird die FTP-Server-Funktion PORT aufgerufen. Die Funktion *prompt* ist ausgeschaltet.

```
→ mget *
   200 PORT command okay.
   150 ASCII data for anton (89.16.100.0,1192).
   226 Transfer complete.
   242 bytes received in 0.06 seconds
   (3.69 Kbytes/s)
   200 PORT command okay.
   150 ASCII data for anton.1 (89.16.100.0,1193).
   226 Transfer complete.
   150 ASCII data for caesar (89.16.100.0,1199).
   226 Transfer complete.
   10845 bytes received in 0.78 seconds (13.56 Kbytes/s)
```

3. Umbenennen einer Datei mit Server-Meldungen.

```
→ rename anton zwaton
   350 File exists, ready for destination name
   200 RNTO command okay.
```

4. Ausschalten der Server-Meldungen.

```
→ verbose
   Verbose mode off.
```

5. *pwd*, *mget*, *rename* ohne Server-Meldungen.

```
pwd
mget *
rename anton zwaton
```

? - Information zu FTP-Kommandos

Das Kommando *?* gibt eine Liste aller FTP-Kommandos aus. Zu allen FTP-Kommandos sind auch Kurzinformationen erhältlich. Statt *?* kann synonym auch *help* verwendet werden.

?
<kommando>

Siehe Kommando *help* auf Seite 140.

! - In den Kommando-Modus von BS2000/OSD bzw. POSIX wechseln

In den BS2000/OSD-Kommando-Modus wechseln (BS2000/OSD-FTP)

Durch Eingabe des Kommandos *!* können während einer FTP-Sitzung BS2000/OSD-Kommandos eingegeben werden. Wenn Sie keinen Operanden angeben, wird in den BS2000/OSD-Kommando-Modus gewechselt, aus dem Sie mit dem BS2000/OSD-Kommando *RESUME* wieder in den FTP zurückgelangen. Bei Eingabe von *!* mit einem BS2000/OSD-Kommando als Operand befinden Sie sich nach Ausführung des Kommandos wieder im FTP-Client.

Ein Wechsel in den POSIX-Kommandomodus ist nicht möglich, es können aber POSIX-Kommandos eingegeben werden, wenn zuvor mit dem Kommando „lcd %POSIX“ in das POSIX-Dateiverzeichnis gewechselt wurde.

!
[<bs2000-kommando>]

<bs2000-kommando>
Beliebiges BS2000/OSD-Kommando. Die BS2000/OSD-Kommandos *LOAD*, *EXEC*, *LOGOFF* und *ABEND* dürfen weder direkt noch indirekt eingegeben werden, da sie den FTP bzw. die Task, in der FTP abläuft, beenden.
Statt des BS2000/OSD-FILE-Kommandos verwenden Sie bitte unbedingt das FTP-Kommando *file*, um Inkonsistenzen im Katalog zu vermeiden.

Beispiel

Absetzen des BS2000/OSD-Kommandos *sta p* aus dem FTP.

```
→ ! sta p
   NAME TSN TYPE SIZE CURR-CMD PROGRAM-NAME
   PETER 1447 3 DIALOG 87 EXECUTE :5:$TSOS.FTP

   ftp>

→ !

   x%BKPT AT xxxxxx

→ /sta p
   NAME TSN TYPE SIZE CURR-CMD PROGRAM-NAME
   PETER 1447 3 DIALOG 87 EXECUTE :5:$TSOS.FTP

   /

→ resume
   ftp>
```

In den POSIX-Kommando-Modus wechseln (POSIX-FTP)

Durch Eingabe des Kommandos *!* können während einer FTP-Sitzung POSIX-Kommandos eingegeben werden. Wenn Sie keinen Operanden angeben, wird in den POSIX-Kommando-Modus gewechselt, aus dem Sie mit dem POSIX-Kommando *exit* wieder in den FTP zurückgelangen. Bei Eingabe von *!* mit einem POSIX-Kommando als Operand befinden Sie sich nach Ausführung des Kommandos wieder im FTP-Client.

!
[<POSIX-kommando>]

<POSIX-kommando>
Beliebiges POSIX-Kommando

4.11 C-Unterprogramm-Schnittstelle YAPFAPI des FTP

Die C-Unterprogramm-Schnittstelle des FTP lehnt sich eng an die interaktive Schnittstelle an und wird durch die C-Funktion *YAPFAPI* realisiert. Das aufrufende Programm übergibt das FTP-Kommando als Zeichenkette an *YAPFAPI*. Als Ergebnis liefert *YAPFAPI* Meldung(en) des FTP-Servers sowie lokale Fehlerinformationen in entsprechende Puffer.

Möchte das aufrufende Programm die Ausgabe eines *dir*- oder *ls*-Kommandos nicht in eine Datei schreiben, sondern z.B. direkt interpretieren, so übergibt es mit dem Parameter *dirCallBackFct* (siehe unten) die Adresse einer entsprechenden Call-Back-Funktion mit folgender Signatur:

```
void dirCallBackFct(char *buffer, int bufLen);
```

Hierbei ist *buffer* ein Zeiger auf einen Puffer, der einen Teil der Ausgabe eines *dir*- bzw. *ls*-Kommandos enthält. *buflen* spezifiziert die Länge des Pufferinhalts.

Wenn bei TLS-gesicherten Verbindungen Client-Authentifizierung verwendet wird, kann das aufrufende Programm eine Passphrase für den privaten Schlüssel übergeben.

Beim Binden des Programms, das die FTP-Unterprogramm-Schnittstelle aufruft, müssen Nutzer der Unterprogramm-Schnittstelle, welche den Standard-C/C++- Compiler einsetzen, den Modul FTPAPI aus der folgenden Bibliothek hinzubinden:

SYSLIB.TCP-IP-AP.052	(bei /390-Rechnern)
SYSLIB.TCP-IP-AP.052	(bei S-Servern)
SPUOML.TCP-IP-AP.052	(bei SX-Rechnern)
SYSLIB.TCP-IP-AP.052	(bei SQ-Servern)

Außerdem muss die externe Referenz auf die benutzerdefinierte Exit-Routine aufgelöst werden (siehe Handbuch „interNet Services Administratorhandbuch“). Wenn Sie keine eigene Exit-Routine benötigen, können Sie hierfür den Dummy-Modul EXITFTP aus der folgenden Bibliothek verwenden:

SYSLNK.TCP-IP-AP.052	(bei /390-Rechnern)
SYSLNK.TCP-IP-AP.052	(bei S-Servern)
SPUOML.TCP-IP-AP.052	(bei SX-Rechnern)
SKMLNK.TCP-IP-AP.052	(bei SQ-Servern)

Da der FTP-Unterprogramm-Modul aus C-Quelltexten produziert wurde, müssen außerdem die C-Laufzeitbibiliotheken (CRTE) hinzugebunden werden.

Nur wer SPARC-Code erzeugen will und dazu den auf Sonderfreigabe erhältlichen C/C++-RS-Compiler einsetzt, sollte die Bibliotheken SPUOML.TCP-IP-AP.052 und SPMLNK.TCP-IP-AP.052 für das Einbinden der Module FTPAPI und EXITFTP verwenden.

Funktionsprototyp von YAPFAPI

Der Funktionsprototyp für den Unterprogramm-Aufruf lautet:

```
void YAPFAPI(struct YAPFAPI_pl_mdl *param)
```

param ist ein Zeiger auf eine Variable vom Typ `struct YAPFAPI_pl_mdl`.

Beschreibung der Struktur YAPFAPI_pl_mdl

Der nachfolgend beschriebene Datentyp `struct YAPFAPI_pl_mdl` ist in der Header-Datei YAPFAPI.H in der Bibliothek SYSLIB.TCP-IP-AP.052 deklariert.

```
struct YAPFAPI_pl_mdl {
   struct {
      int version;                /* Interface version                    */
      char *cmd;                  /* Contains command string              */
      char *serverMsg;            /* Buffer for server messages           */
      int maxServerMsgLen;        /* Length of buffer for server messages */
      char *localMsg;             /* Buffer for local messages            */
      int maxLocalMsgLen;         /* Length of buffer for local messages  */
      int combineMessages;        /* Write all messages into one buffer   */
      void (*dirCallBackFct)(char*, int);  /* Call back function */
      char *passPhrase;           /* Password for private key             */
      int passPhraseLen;          /* Length of password for private key   */
   } in_data;

   struct {
      int msgNumber;         /* Message number of last message from server */
      int serverMsgLen;      /* Length of messages from server             */
      int localMsgLen;       /* Length of local message                    */
      int rc;                /* Summary return code                        */
   } out_data;
}
/* Values for rc */
#define YAPFAPIrcOk                        0    /* Ok                      */
#define YAPFAPIrcVersionError              1    /* Wrong interface version */
#define YAPFAPIrcInitError                 2    /* Initialization Error    */
#define YAPFAPIrcLocalError                3    /* Local error             */
#define YAPFAPIrcFatalLocalError           4    /* Fatal local error       */
#define YAPFAPIrcRemoteError               5    /* Remote error            */
```

Beschreibung der Strukturelemente

version
Version der Schnittstelle. Dieser Parameter muss mit 1 versorgt werden.

cmd
Zeiger auf den Kommando-String, der das Kommando in derselben Syntax enthält, wie sie die interaktive Schnittstelle fordert (siehe Abschnitt „Kommandoübersicht (FTP-Client)“ auf Seite 112).

serverMsg
Zeiger auf einen vom Aufrufer bereitgestellten Puffer zur Aufnahme der Server-Meldungen.

maxServerMsgLen
Länge des Puffers, auf den *serverMsg* zeigt. Reicht die Puffergöße zur Aufnahme aller Meldungen nicht aus, dann werden die überzähligen Meldungen nur fragmentarisch oder überhaupt nicht ausgegeben.

localMsg
Zeiger auf einen vom Aufrufer bereitgestellten Puffer zur Aufnahme lokaler Meldungen.

maxLocalMsgLen
Länge des Puffers, auf den *localMsg* zeigt. Reicht die Puffergöße zur Aufnahme aller Meldungen nicht aus, dann werden die überzähligen Meldungen nur fragmentarisch oder überhaupt nicht ausgegeben.

combineMessages
Spezifiziert der Aufrufer für *combineMessages* einen von 0 verschiedenen Wert, dann werden sowohl Server-Meldungen als auch lokale Meldungen in dem von *localMsg* referenzierten Puffer bereitgestellt. Ein eigener Puffer für Server-Meldungen braucht in diesem Fall nicht bereitgestellt zu werden.

dirCallBackFct
Adresse einer Call-Back-Funktion. Soll keine Call-Back-Funktion verwendet werden, dann ist NULL zu spezifizieren.

passPhrase
Zeiger auf einen Puffer, der die Passphrase für den privaten Schlüssel enthält.

passPhraseLen
Länge der Passphrase

msgNumber
In *msgNumber* wird die Nummer der letzten Server-Meldung gemäß RFC 959 (FTP) zurückgeliefert.

serverMsgLen
Der in *serverMsgLen* zurückgelieferte Integer-Wert zeigt an, wieviele Zeichen aktuell in den durch *serverMsg* spezifizierten Puffer übertragen wurden.

localMsgLen
Der in *localMsgLen* zurückgelieferte Integer-Wert zeigt an, wieviele Zeichen aktuell in den durch *localMsg* spezifizierten Puffer übertragen wurden.

rc
In *rc* wird der insgesamt resultierende Return-Code zurückgeliefert.

Beispielprogramm für die Nutzung der FTP-Unterprogramm-Schnittstelle

Nachfolgend ist ein Beispielprogramm abgedruckt, das die FTP-Unterprogramm-Schnittstelle nutzt.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "yapfapi.h"

void printDir(char*, int);

main(int argc, char *argv[])
{
    struct YAPFAPI_pl_mdl param;
    char line[200];
    char serverMsg[2000];
    char localMsg[2000];
    FILE *fp_in;

    fp_in = fopen("(SYSDTA)", "r");

    for (;;) {
        printf("ftp> ");
        if (fgets(line, sizeof line, fp_in) == 0) {
        break;
        }
        param.in_data.version = 1;
        param.in_data.cmd = line;
        param.in_data.serverMsg = serverMsg;
        param.in_data.maxServerMsgLen = sizeof(serverMsg);
        param.in_data.localMsg = localMsg;
        param.in_data.maxLocalMsgLen = sizeof(localMsg);
        param.in_data.combineMessages = 1;
        param.in_data.dirCallBackFct = &printDir;
        param.in_data.passPhrase = "ABCDEFGH";
        param.in_data.passPhraseLen = 8;
        YAPFAPI(&param);
        printf("Ret: msgNumber:    %d\n", param.out_data.msgNumber);
        if (param.out_data.serverMsgLen != 0)
            printf("Ret: serverMsgLen: %d\n", param.out_data.serverMsgLen);
        printf("Ret: localMsgLen:  %d\n", param.out_data.localMsgLen);
```

```
        if (param.out_data.serverMsgLen != 0) {
            serverMsg[param.out_data.serverMsgLen] = '\0';
            printf("Ret: serverMsg: \n");
            printf("%s", serverMsg);
            printf("Ret: serverMsgEnd\n");
        }
        localMsg[param.out_data.localMsgLen] = '\0';
        printf("Ret: localMsg: \n");
        printf("%s", localMsg);
        printf("Ret: localMsgEnd\n");
        printf("Ret: rc:        %d\n", param.out_data.rc);
        if (param.out_data.rc == YAPFAPIrcInitError)
            break;
      }
}

void printDir(char* buffer, int bufLen)
{
    printf("%.*s", bufLen, buffer);
}
```

4.12 Fragen und Antworten (FAQ)

- **Frage:**
  Welche Bedeutung hat die Ausgabe der folgenden Meldung nach dem Laden von FTP- oder TELNET-Client bzw. von FTP- oder TELNET-Server?

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

  Antwort:
  Die Programme wurden entweder mit START-PROG ..... ohne PROG-MODE=*ANY gestartet oder das Socket-Subsystem SOC6 ist nicht gestartet.

- **Frage:**
  Der Verbindungsaufbau mit *open* im FTP- oder TELNET-Client dauert sehr lange. Was ist die Ursache?

  Antwort:
  Sowohl Client als auch Server benutzen beim Verbindungsaufbau DNS-Funktionen. Wenn die zugehörigen Resolver-Dateien nicht ordnungsgemäß eingestellt sind, können längere Wartezeiten die Folge sein. Dies ist beispielsweise der Fall, wenn der in den Resolverdateien spezifizierte DNS-Server nicht erreichbar ist.

  Der Name der Resolver-Datei im BS2000/OSD lautet:

  SYSDAT.SOCKETS.nnn.SOC6.RESOLV bzw.
  SYSDAT.LWRESD.nnn.RESOLV.CONF

- **Frage:**
  Welche Bedeutung hat die folgende FTP-Client-Meldung?

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

  Antwort:
  Der Client wartet auf eine Antwort des Servers, die nach einer voreingestellten Anzahl von Sekunden (30 Sekunden) nicht eingetroffen ist. Ursache hierfür kann z.B. eine zu hohe Netzbelastung sein.
  Das Problem können Sie oft dadurch lösen, dass Sie den voreingestellten Timeout-Wert von 30 Sekunden mit dem FTP-Benutzerkommando *settime* (siehe Seite 196) erhöhen.

- **Frage:**
  Wenn ich mit FTP eine Datei transferiere und sie im Zielsystem als PAM-Datei abspeichere, wird am Ende der String „C-DATEIENDE“ angefügt. Wozu dient das, und wie kann ich es verhindern?

  Antwort:
  Der String „C-DATEIENDE“ wird normalerweise benutzt, um das genaue Ende einer PAM-Datei zu markieren. Mit dem FTP-Benutzerkommando `setfile datend off` können Sie das Anhängen des Strings „C-DATEIENDE“ unterdrücken.

- **Frage:**
  In früheren FTP-Versionen (< V4.0) wurden Tabulatorzeichen in einer Textdatei automatisch in die entsprechende Anzahl von Leerzeichen umgesetzt. Jetzt ist dies nicht mehr der Fall. Was ist die Ursache?

  Antwort:
  Im Zuge der Unterstützung der Restart-Fähigkeit im BS2000/OSD-FTP ab V4.0 wurde die Voreinstellung für den Transfer von Textdateien von `ftyp text` auf `ftyp textbin` geändert. Daher werden Tabulatorzeichen standardmäßig nicht mehr umgesetzt.

  Wenn Sie die Umsetzung der Tabulatorzeichen dennoch nutzen wollen, können Sie dies wahlweise via *ftyp*-Kommando oder via Option-Datei einstellen:

  – *ftyp*-Kommando (siehe Seite 134). Spezifizieren Sie `ftyp text` im Client bzw. `quote site ftyp text` im Server.

  – Option-Datei des FTP-Clients (Option *initialCommand*, siehe Seite 92) bzw. Option-Datei des FTP-Servers (Option *initialChildCmds*, siehe Handbuch „interNetServices Administratorhandbuch“)

- **Frage:**
  In früheren FTP-Versionen (< V4.0) wurden SAM-Dateien mit fester Satzlänge bei *type binary* vollständig binär übertragen und ohne Satzende-Zeichen abgespeichert. In der aktuellen FTP-Version ist die Zieldatei mit Satzende-Zeichen versehen. Wie kann ich das verhindern?

  Antwort:
  Auch für SAM-Dateien mit fester Satzlänge werden jetzt die *ftyp*-Einstellungen berücksichtigt. Wenn *ftyp* ungleich „binary“ ist, bleibt die Satzstruktur der Ursprungsdatei bei der Übertragung erhalten. Um dies zu verhindern, müssen Sie explizit *ftyp binary* einstellen.

- **Frage:**
  Im BS2000/OSD-FTP und -TELNET ist es möglich mit den Kommandos *trace* und *debug* (bei den Clients) bzw. mit den Optionen *-T* und *-D* (bei den Servern) Diagnose-Information zu erzeugen. Worin besteht der Unterschied zwischen *trace* bzw. *-T* einerseits und *debug* bzw. *-D* andererseits?

  Antwort:
  debug bzw. *-D* erzeugen Diagnose-Informationen, die die Produkte FTP bzw. TELNET selbst betreffen. Der höchste sinnvolle Level ist hier 2.
  Dagegen geben *trace* bzw. *-T* Diagnose-Informationen aus, die von den Sockets erzeugt wurden. Der höchste sinnvolle Level ist hier 9.

- **Frage:**
  Ich habe einen zweiten FTP- bzw. TELNET-Server gestartet, kann zu diesem aber keine Verbindung herstellen. Was kann die Ursache sein?

  Antwort:
  Die häufigsten Ursachen des Problems sind:

  – Angabe einer bereits vergebenen Portnummer für den Server.
  – Verwendung eines schon vergebenen Applikationsnamens (Option *-A*) im Server.

- **Frage:**
  Mein FTP-Login wird vom (BS2000/OSD-) FTP-Server mit der Meldung `invalid login` abgewiesen. Ich kann aber keine Ursache für dieses Verhalten finden. Was ist zu tun?

  Antwort:
  Schalten Sie – falls möglich – im FTP-Server den FTP-Trace mit folgendem Konsol-Kommando ein:

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

  Wiederholen Sie das Login und speichern Sie die Traces mit dem Konsol-Kommando

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

  in die Datei SYSOUT.TCP-IP-AP.052.FTPD.<MMDDHHMMSS> ab.

  (MMDDHHMMSS ist Datum- und Zeitangabe der Form
  Month Day Hour Minute Second).

- **Frage:**
  Wie erreiche ich mit einem Webbrowser einen BS2000/OSD-FTP-Server?

  Antwort:
  Über die folgende URL erhalten Sie Zugang zum POSIX-Dateiverzeichnis der Kennung <userid>:

  ```
  ftp://<userid>,<accountnummer>@<rechnername>:<portnummer>/
  ```

  Damit ist zumindest die Ausgabe der Dateiverzeichnisse möglich. Ein Zugang zum BS2000/OSD-Dateiverzeichnis ist über den Webbrowser nicht möglich.

- **Frage:**
  Es gibt bei FTP die Möglichkeit, mit `quote <command>` das Kommando <command> an den Server zu schicken. Es gibt aber auch `quote site <command>` und `quote site exec <command>`. Worin besteht der Unterschied?

  Antwort:

  – Mit `quote <command>` schicken Sie standard-konforme FTP-Kommandos an den Server.

  – Mit `quote site <command>` schicken sie BS2000/OSD-spezifische („proprietäre“) Kommandos, nicht im Standard definierte Kommandos an den Server. Solche Kommandos sind u.a. *ftyp*, *cmod*, *modc*, *file*, *setc*, *sfil*.

  – Mit `quote site exec <command>` senden Sie BS2000/OSD-Kommandos zur Ausführung an den Server. Um die missbräuchliche Verwendung dieser Kommandos im Zielsystem zu verhindern, wird diese Variante beim Einsatz von FTAC (Option *-FTAClevel* >0) oder durch die Option *-disableSiteExecCommand* deaktiviert (siehe Handbuch „interNet Services Administratorhandbuch“).

- **Frage:**
  Beim Einsatz von FTP-Clients mit grafischer Benutzeroberfläche (GUI) besteht oft keine Möglichkeit, die für die Verbindung mit BS2000/OSD erforderliche Abrechnungsnummer anzugeben. Was ist in diesem Fall zu tun?

  Antwort:
  Geben Sie in diesen Fällen die Abrechnungsnummer bereits bei der Eingabe der Kennung wie folgt ein:

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

- **Frage:**
  Beim Abholen einer Datei von einem BS2000/OSD-FTP-Server bricht mein Nicht-BS2000/OSD-FTP-Client nach einer gewissen Zeit die Ausführung ab, ohne dass mit dem eigentlichen Datentransfer begonnen wurde.

  Antwort:
  Manche FTP-Clients zeigen den Transfer-Fortschritt durch einen Fortschrittsbalken an. Zu diesem Zweck erfragen die Clients zunächst mit dem FTP-Protokollkommando *SIZE* die Größe der Datei vom Server. Der Server muss i.A. für die Bearbeitung dieses Kommandos die betreffende Datei vollständig lesen. Dies kann bei sehr großen Dateien naturgemäß recht lange dauern, so dass die Zeitüberwachung des Clients die Verbindung abbricht.
  Leider kann diese Zeitüberwachung bei manchen Clients nicht auf längere Wartezeiten umkonfiguriert werden. Darüber hinaus wird in vielen Fällen auch bei Deaktivierung des Fortschrittsbalkens ein *SIZE*-Kommando an den Server abgesetzt.

  Wenn vom Hersteller des FTP-Clients keine Verbesserung der Konfigurationsmöglichkeiten zu erhalten ist, besteht beim BS2000/OSD-FTP-Server die Möglichkeit, mithilfe der Option *- disableSizeCommand* das *SIZE*-Kommando zu deaktivieren (siehe Handbuch „interNet Services Administratorhandbuch“).

  Da von einem Client die Unterstützung des *SIZE*-Kommandos durch den Server nicht vorausgesetzt werden darf, sollte der Transfer in jedem Fall funktionieren. Allerdings müssen Sie in Kauf nehmen, dass der Restart-Mechanismus nicht mehr funktioniert, da hierfür das *SIZE*-Kommando benötigt wird.

- **Frage:**

  Beim Transfer einer LMS-Datei von einem NK2- auf ein NK4-Pubset ist die Zieldatei keine gültige LMS-Datei mehr.

  Antwort:

  Transferieren Sie die Datei zunächst auf ein nicht-NK4-Pubset auf dem Zielrechner und kopieren Sie dann die LMS-Bibliothek mithilfe von LMS auf das NK4-Pubset.

  Alternativ können Sie auf dem Quellrechner die LMS-Bibliothek auf ein NK4-Pubset umsetzen und dann mit FTP auf das NK4-Pubset des Zielrechners transferieren.

5 FTAC-Schnittstelle

Mit der FTAC-Funktion von openFT haben Sie die Möglichkeit, Ihren BS2000-Server so sicher wie möglich und so sicher wie nötig zu machen. FTAC steht für „File Transfer Access Control“. Voraussetzung für die Nutzung von FTAC auf dem BS2000-Server ist der Einsatz von openFT mit openFT-AC.

Die in diesem Kapitel beschriebenen FTAC-Funktionen beziehen sich auf openFT V11.0. Wenn auf Ihrem BS2000-Server eine andere openFT-Version im Ensatz ist, kann der Funktionsumfang abweichen.

Die vollständige Beschreibung zu openFT-AC finden Sie in folgenden Handbüchern, welche für die verschiedenen openFT-Versionen zur Verfügung stehen:

– openFT für BS2000/OSD - Benutzerhandbuch
  Dieses Handbuch enthält die vollständige Beschreibung der FTAC-Funktionalität, der FTAC-Kommandos sowie die Beschreibung der zugehörigen CSV-Ausgaben und OPS-Variablen.
– openFT für BS2000/OSD - Meldungen
  Dieses Handbuch enthält alle FTAC-Meldungen.

5.1 FTAC-Funktionalität

FTAC bietet zum Schutz des BS2000-Servers folgende Möglichkeiten:

– Entkopplung von FTP-Zugangsberechtigung und Login-Berechtigung
– Zugriffsrechte abhängig von Partnersystemen
– benutzerspezifische Zugriffsrechte
– flexible Abstufung der Zugriffsrechte
– Protokollierung jeder Berechtigungsprüfung
– einfache Anwendung

5.1.1 Leistungen der FTAC-Funktion

Bei der Übertragung von Dateien unterscheidet man verschiedene Funktionen. Für den Zugangs- und Zugriffsschutz ist dabei ausschlaggebend, was das zu schützende System bei der Dateiübertragung macht:

– Senden einer Datei
– Empfangen einer Datei

Beim Senden einer Datei werden Daten aus dem zu schützenden System nach außen weitergegeben, beim Empfangen einer Datei gelangen Daten von außen in das zu schützende System. Nun besteht aber für den Datenschutz ein erheblicher Unterschied darin, wer von dem zu schützenden System eine Funktion verlangt. Im Sprachgebrauch des File Transfer heißt das, „wer Initiator (= Auftraggeber) eines Auftrages ist".

FTAC unterscheidet zwei Gruppen von Auftraggebern:

– Auftraggeber im zu schützenden System (Outbound-Aufträge)
– Auftraggeber in Partnersystemen (Inbound-Aufträge)

Die von FTP genutzte Funktionalität ist auf Inbound-Aufträge beschränkt. Daher können folgende Transfer-Funktionen unterschieden werden:

– **Inbound Senden**
– **Inbound Empfangen**

FTP-Partner (FTP-Clients) haben außerdem die Möglichkeit, mit den Dateimanagement-Funktionen sich in Ihrem lokalen System (BS2000-FTP-Server) Dateiverzeichnisse oder Dateiattribute anzusehen, Dateiattribute zu ändern sowie Dateien und Verzeichnisse zu löschen. Daraus ergibt sich eine weitere Funktion:

– **Inbound Dateimanagement**

Das Dateimanagement umfasst im Gegensatz zu den anderen Funktionen mehrere verschiedenartige Auftragsmöglichkeiten, die wiederum teilweise mit den Funktionen *inbound senden* und *inbound empfangen* gekoppelt sind:

- Ein FTP-Client darf lokale Dateien löschen, wenn die Funktion *inbound empfangen* zugelassen ist.
- Ein FTP-Client darf Dateiattribute lokaler Dateien ansehen, wenn die Funktion *inbound senden* zugelassen ist.
- Ein FTP-Client darf Verzeichnisse ansehen und löschen, wenn die Funktion *inbound Dateimanagement* zugelassen ist.
- Ein FTP-Client darf Dateiattribute lokaler Dateien ändern sowie Verzeichnisse erzeugen und umbenennen, wenn sowohl die Funktion *inbound empfangen* wie auch *inbound Dateimangement* zugelassen sind.

Die Schutzmechanismen, die die FTAC-Funktion bietet, werden in erster Linie durch den Einsatz von Berechtigungssätzen und Berechtigungsprofilen erreicht.

5.1.2 Berechtigungssatz

Der Berechtigungssatz enthält die grundsätzlichen Festlegungen, welche FTP-Funktionen erlaubt sind. Ein Berechtigungssatz gilt für genau eine Benutzerkennung im BS2000. Erfolgt ein Zugriff auf diese Benutzerkennung, so prüft FTAC, ob die im Berechtigungssatz eingestellten Werte eingehalten werden. Die Festlegungen des Berechtigungssatzes können Sie mit Berechtigungsprofilen entweder weiter einschränken oder durch Privilegierung erweitern. Über Berechtigungsprofile können Sie dann individuell eine oder mehrere Inbound-Funktionen zulassen. Sie können existierende Berechtigungssätze jederzeit ansehen und ändern, um sie dem aktuellen Bedarf anzupassen.

Nach der Installation von openFT-AC gelten zunächst für alle Benutzerkennungen die Angaben im Standardberechtigungssatz. Diesen Standardberechtigungssatz muss der FTAC-Verwalter nach der Installation so ändern, dass er für die meisten Benutzerkennungen den notwendigen Schutz bietet. Für einzelne Benutzerkennungen, die einen größeren Schutz erfordern, können speziell angepasste Berechtigungssätze anlegt werden.

5.1.3 Berechtigungsprofil

Mit einem Berechtigungsprofil definieren Sie die Zugangsberechtigung und die damit verbundenen Zugriffsrechte. Die Zugangsberechtigung ist sozusagen der Schlüssel für den Zugriff via FTP auf den BS2000-Server. Deshalb sollten Sie die Zugangsberechtigung wie ein Kennwort behandeln. Sie muss bei Übertragungsaufträgen anstelle einer Login-Berechtigung angegeben werden. Jeder, der diese Zugangsberechtigung kennt, hat zwar per FTP Zugang zu Ihrer Kennung auf dem BS2000-Server, aber er kann im Gegensatz zur Login-Berechtigung nicht machen, was er will. Welche Funktionen Sie zulassen, legen Sie mit den Zugriffsrechten für diese Zugangsberechtigung fest. Sie regeln damit z.B., auf welche Dateien unter welchen Voraussetzungen zugegriffen werden darf. Im Extremfall können Sie z.B. den Zugriff auf Ihre Benutzerkennung so einschränken, dass nur über ein einziges Profil auf nur eine Datei zugegriffen werden darf. Bei entsprechender Einstellung ist es möglich, ein Berechtigungsprofil gleichzeitig für openFT und auch für FTP zu nutzen.

FTAC überprüft bei jedem FTP-Auftrag, ob die Angaben im Auftrag im Widerspruch zu den Angaben im Berechtigungsprofil stehen. Ist das der Fall, wird der FTP-Auftrag abgelehnt. Im Client-System erscheint dann nur eine allgemein gehaltene Fehlermeldung. Dadurch wird verhindert, dass jemand die Definitionen des Berechtigungsprofils durch schrittweises Ausprobieren ermitteln kann. Im BS2000-Server wird ein Logging-Satz erstellt, der die genaue Ursache beschreibt.

Das Bild auf der nächsten Seite zeigt die Abläufe bei der Zugangsprüfung mit FTAC.

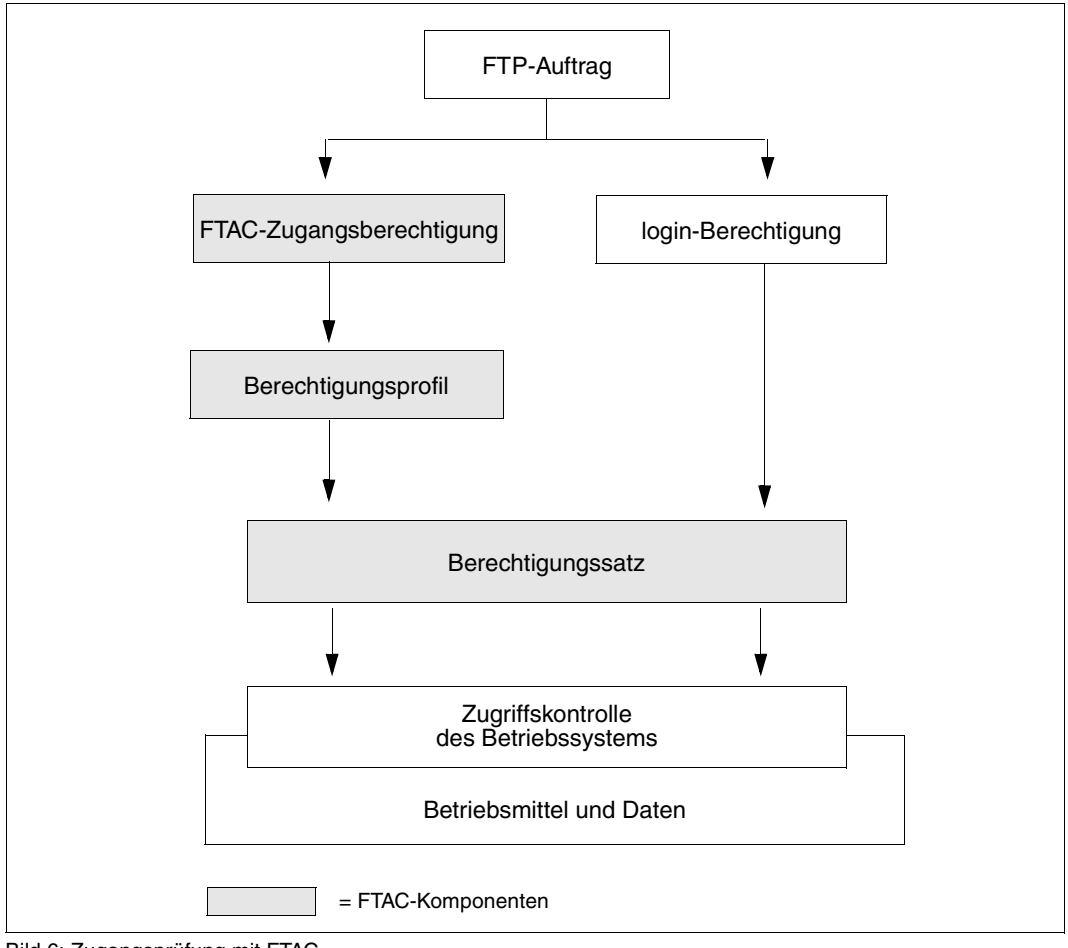


Bild 6: Zugangsprüfung mit FTAC

Ein Berechtigungsprofil enthält unter anderem

- eine Zugangsberechtigung. Diese Zugangsberechtigung muss eindeutig sein. Wenn ein Auftrag mit dem Berechtigungsprofil arbeiten soll, muss diese Zugangsberechtigung angegeben werden. FTAC erlaubt für diesen Auftrag dann nur die Zugriffsrechte, die im Berechtigungsprofil definiert sind. Um die Verantwortung für Aufträge eindeutig zuordnen zu können, empfiehlt es sich, eine Zugangsberechtigung für genau eine Person vorzusehen.
- gegebenenfalls Angaben, welche Partnersysteme auf dieses Berechtigungsprofil zugreifen dürfen.

- Angaben, welche Parameter in einem Auftrag in welchem Umfang benutzt werden dürfen. Damit werden die Zugriffsrechte für jeden eingeschränkt, der dieses Berechtigungsprofil für FTP benutzt.
- gegebenenfalls Angaben, ob oder bis wann das Berechtigungsprofil genutzt werden kann.
- ein Dateinamenspräfix. Dieses Präfix enthält einen Teil eines Pfadnamens. Der Benutzer des Profils kann sich dann nur unterhalb des angegebenen Pfadnamens bewegen, z.B. bedeutet /USR/HUGO/ als Dateinamenspräfix auf einem Unix-Rechner, dass der Benutzer dieses Profils nur auf Verzeichnisse unterhalb von /USR/HUGO/ zugreifen darf. Auf diese Weise wird ausgeschlossen, dass sich jemand durch Eingabe von „. .“ in gesperrte Verzeichnisse bewegen kann.

Sie können verschiedene Berechtigungsprofile abspeichern.

Berechtigungsprofile können Sie jederzeit:

- **ändern**
  und so dem aktuellen Bedarf anpassen.
- **sperren**
  In dem Fall wird ein Auftrag mit dem gesperrten Profil wegen ungültiger Zugangsberechtigung abgelehnt. Wollen Sie das Berechtigungsprofil wieder verwenden, müssen Sie das Berechtigungsprofil erst wieder entsperren.
- **löschen**
  Sie sollten die Zahl Ihrer Berechtigungsprofile in Grenzen halten, indem Sie Profile, die Sie nicht mehr benötigen, löschen.
- **privilegieren** (systemabhängig)
  In speziellen Fällen können Berechtigungsprofile auch eine Funktion nutzen, die im Berechtigungssatz gesperrt wurde. Dazu muss das Berechtigungsprofil vom FTAC-Administrator privilegiert werden.

Sie können sich jederzeit Informationen über Ihre Berechtigungsprofile ausgeben lassen.

5.1.4 Auswirkungen eines Berechtigungsprofils

Die folgende Tabelle enthält in der linken Spalte die möglichen Einschränkungen der Zugriffsrechte in einem Berechtigungsprofil und in der rechten Spalte die Angaben, die für den Übertragungsauftrag in Bezug auf das Partnersystem nötig sind.

Festlegung im Berechtigungsprofil	**Angaben für den Übertragungsauftrag**
Zugangsberechtigung	Es muss die Zugangsberechtigung angegeben werden: – Als Benutzerkennung geben Sie *$FTAC* oder eine andere von der Systemverwaltung fest vorgegebene Zeichenkettean. Als Passwort verwenden Sie die Zugangsberechtigung. – Wenn ein Berechtigungsprofil mit der Zugangsberechtigung „anonymousFTP“ definiert ist, ist auch ein Zugang mit der Benutzerkennung „FTP“ oder „ANONYMOUS“ möglich. In diesem Fall muss als Passwort eine Mailadresse angegeben werden. (Es wird lediglich dann geprüft, ob „@“ enthalten ist.)
Übertragungsrichtung eingeschränkt	Die Angabe muss spiegelbildlich zur Festlegung im Berechtigungsprofil erfolgen. Wenn im Profil die Übertragungsrichtung „Vom-Partner“ steht, darf das ferne System nur Daten zum lokalen Rechner schicken, bei „An Partner“ sind im lokalen Rechner nur lesende Zugriffe erlaubt.
Partnersysteme vorgegeben	Der Auftrag kann nur von den Partnersystemen gestellt werden, die im Profil eingetragen sind. Diese Option darf nur dann genutzt werden, wenn garantiert ist, dass die Namen der FTP-Clienten nicht länger als 8 Zeichen sind. Wenn DNS aktiviert ist, wird als Client-Name der vom DNS gelieferte Client-Name verwendet, andernfalls ist der zur IP-Adresse gehörende Eintrag in der SOCKET-HOST-TABLE relevant (siehe Handbuch „BCAM“). Falls kein entsprechender Eintrag vorhanden ist, wird der Eintrag in der PROCESSOR-TABLE verwendet. Bei lokalen Zugriffen wird als Name „LOOPBACK“ verwendet.
Präfix für den Dateinamen vorgegeben	Im Auftrag steht nur ein Teil des Dateinamens. FTAC ergänzt diese Angabe um das im Profil definierte Präfix zum vollständigen Dateinamen. Die Angabe absoluter Dateinamen oder das Verlassen des Verzeichnisses mittels „..“ wird von FTAC unterbunden.
Folgeverarbeitung	Diese FTAC-Funktion ist openFT vorbehalten und wird von FTP nicht unterstützt.
Einschränkung der Schreibregel	Der Auftrag wird nur dann durchgeführt, wenn er nicht gegen diese Schreibregel verstößt.

Berechtigungen auslagern

Sie können sowohl komplette Berechtigungen als auch einzelne Berechtigungssätze und Berechtigungsprofile in eine Datei sichern (auslagern). Aus dieser Datei können sie bei Bedarf wieder übernommen werden.

5.1.5 Überwachung des FTP-Servers durch FTAC

Mit der FTAC-Funktion findet eine Überprüfung der Server-Kommandos auf dem BS2000-FTP-Server statt. Die folgende Tabelle listet die von FTAC überprüften Server-Kommandos auf und stellt die Client-Kommandos vor, die diese Server-Kommandos nutzen.

Server-Kommando	**entsprechendes BS2000-FTP-Client-Kommando**
retr	*get*, *recv*, *mget*
stor	*put*, *send* (falls *sunique=off*), *mput*
stou	*put*, *send* (falls *sunique=on*)
appe	*append*
rfnr	*rename*
dele	*delete*, *mdelete*
site file	*quote site file*
cwd	*cd*
xcwd	*cd* (gleiche Funktion wie *cwd*)
cdup	*cdup*
xcup	*cdup* (gleiche Funktion wie *cdup*)
list	*dir*, *mdir*
nlst	*ls*, *mls*, *mget*, *mdelete*
site exec	*quote site exec* (wird bei FTAC-Nutzung immer abgewiesen)
mkd	*mkdir*
xmkd	*mkdir* (gleiche Funktion wie *mkd*)
rmd	*rmdir*
xrmd	*rmdir* (gleiche Funktion wie *rmd*)
pwd	*pwd*
xpwd	*pwd* (gleiche Funktion wie *pwd*)
size	*quote size*, *reget*, *reput*
mdtm	*quote mdtm*, *reget*, *reput*

Liste der per FTAC überprüften FTP-Serverkommandos

5.2 FTAC-Kommandoschnittstelle

Die FTAC-Kommandoschnittstelle bietet

- mit der funktionalen Kommandoübersicht eine schnelle Orientierung darüber, welche Kommandos Ihnen für welche Aufgaben zur Verfügung stehen,
- Information zur Eingabe der Kommandos,
- Erläuterungen zu Returncodes,
- die detaillierte Beschreibung aller für den Nutzer relevanten FTP-Kommandos in alphabetischer Reihenfolge.

5.2.1 Funktionale Kommandoübersicht

Die folgende Übersicht zeigt Ihnen aufgabenbezogen die FTAC-Benutzerkommandos. Voraussetzung für die Nutzung der folgenden Kommandos ist die Nutzung von openFT-AC für den BS2000-FTP-Server.

Kommandos für die FTAC-Berechtigungssätze

Der FTAC-Benutzer kann den Standard- und den eigenen Berechtigungssatz ansehen sowie den eigenen Berechtigungssatz ändern, sowie Partnersysteme anzeigen.

Berechtigungssatz ändern	MODIFY-FT-ADMISSION-SET	Seite 249
Berechtigungssatz anzeigen	SHOW-FT-ADMISSION-SET	Seite 267

Kommandos für die FTAC-Berechtigungsprofile

Der FTAC-Benutzer kann für seine eigene Kennung Berechtigungsprofile anlegen, ändern, sich anzeigen lassen und löschen.

Berechtigungsprofil anlegen	CREATE-FT-PROFILE	Seite 236
Berechtigungsprofil löschen	DELETE-FT-PROFILE	Seite 247
Berechtigungsprofil ändern	MODIFY-FT-PROFILE	Seite 253
Berechtigungsprofil anzeigen	SHOW-FT-PROFILE	Seite 279

Kommando zur Logging-Funktion

Logging-Sätze anzeigen	SHOW-FT-LOGGING-RECORDS	Seite 270

5.2.2 FTAC-Kommandos eingeben

Beachten Sie bitte folgende Hinweise zur Eingabe der Kommandos:

- Die einzelnen Operanden eines Kommandos müssen durch Kommata getrennt eingeben werden, z.B.

  ```
  /TRANSFER-FILE TRANSFER-DIRECTION=TO,PARTNER=ZENTRALE,LOCAL-PARAMETER=...
  ```

- In Hochkommata eingeschlossene Wertzuweisungen werden immer in Hochkommata eingeschlossen, d.h., dass sich die Anzahl der Hochkommata verdoppelt.
- Ist bei einem Operanden kein Standardwert gekennzeichnet (durch Unterstreichung), so muss er mit einem gültigen Wert angegeben werden (Pflichtoperand).
- Die Kommandos und Operanden können bei der Eingabe bis zur Eindeutigkeit abgekürzt werden. Außerdem können Sie mit Stellungsparametern arbeiten. Kurzform und Langform dürfen beliebig gemischt werden. Für openFT werden bestimmte Abkürzungsmöglichkeiten für Schlüsselwörter sowie einige Stellungsparameter garantiert. Das bedeutet, dass Sie diese Möglichkeiten auch in Folgeversionen in dieser Form haben werden. Wenn Sie sich also angewöhnen, die Kommandos in dieser Form einzugeben, sind Sie „auf der sicheren Seite“. Insbesondere sollten Sie in Prozeduren die garantierten Abkürzungen verwenden, um deren Ablauf für Folgeversionen sicherzustellen. In den Beispielen dieses Kapitels werden die empfohlenen Abkürzungen verwendet. Zudem sind in den einzelnen Kommandoformaten die Abkürzungsmöglichkeiten dargestellt.
- Ist einer Struktur ein struktureinleitender Operand vorangestellt (z.B. ist „*BS2000“ struktureinleitender Operand bei REM=*BS2000(...), so muss die einleitende Klammer diesem Operanden unmittelbar folgen. Struktureinleitende Operanden dürfen entfallen, wenn die Eindeutigkeit gewährleistet ist.
- Der Schlüsselwörtern vorangestellte Stern darf entfallen, wenn die Eindeutigkeit gewährleistet ist. Beachten Sie bitte, dass dies keine garantierte Abkürzung darstellt.

Sind Stellungsparameter explizit in der Syntax zugelassen, dürfen Wertzuweisungen in Stellungs- und Schlüsselwortform auch gemischt verwendet werden.

Dabei ist zu beachten:

- Die erste Wertzuweisung als Stellungsparameter wird dem ersten Operanden, die zweite Wertzuweisung in Stellungsparameterform dem zweiten Operanden zugeordnet.
- Für jeden weggelassenen Operanden vor einem Stellungsoperanden ist ein Komma einzugeben.

- Erfolgt zu einem Operanden je eine Wertzuweisung in Stellungs- und in Schlüsselwortform, gilt die innerhalb der entsprechenden Strukturklammer letztgenannte Zuweisung. Dies ist immer die Zuweisung per Schlüsselwortform, da die Stellungsparameter immer zuerst kommen müssen.
- Die Stellungsoperanden müssen zuerst aufgeführt werden. Der Übersichtlichkeit wegen sollten Doppelzuweisungen vermieden werden.
- Da nicht ausgeschlossen werden kann, dass sich in späteren Versionen die Reihenfolge der Operanden ändert, sollten in Prozeduren nur Schlüsselwortparameter benutzt werden.

5.2.3 Kommando-Returncodes

Die openFT-AC-Kommandos liefern Kommando-Returncodes zurück, die Sie bei Einsatz von SDF-P abfragen können. Jeder Returncode besteht aus einem Subcode1 (SC1), einem Subcode2 (SC2) und dem Maincode (MC).

Subcode1

Subcode1 beschreibt die Fehlerklasse. Er wird dezimal ausgegeben.

Folgende Fehlerklassen werden unterschieden:

- kein Fehler:
  Der Wert von Subcode1 ist 0.
- Syntaxfehler:
  Der Wert von Subcode1 liegt zwischen 1 und 31.
- interner Fehler (Systemfehler):
  Der Wert von Subcode1 ist 32.
- Fehler, die keiner anderen Klasse zugeordnet sind:
  Der Wert von Subcode1 liegt zwischen 64 und 127. In diesem Fall sollte zur Bestimmung der weiteren Vorgehensweise der Maincode ausgewertet werden.
- Kommando vorübergehend nicht ausführbar:
  Der Wert von Subcode1 liegt zwischen 128 und 130.

Subcode2

Subcode2 enthält entweder Zusatzinformationen zu Subcode1 oder ist gleich 0.

Maincode

Der Maincode entspricht dem Meldungsschlüssel der SYSOUT-Meldung. Mit dem Kommando /HELP-MSG-INFORMATION können Sie detaillierte Informationen abfragen. Zu einzelnen Meldungen finden Sie im Kapitel Meldungen Informationen zur Bedeutung der Meldung und zu den zu ergreifenden Maßnahmen.

Eine detaillierte Beschreibung der Kommando-Returncodes finden Sie im BS2000-Benutzerhandbuch „Benutzer-Kommandos (SDF-Format)“.

In den folgenden Kommandobeschreibungen wird für jedes Kommando einzeln angegeben, welche Kommando-Returncodes möglich sind und welche Bedeutung sie haben.

5.2.4 CREATE-FT-PROFILE - Berechtigungsprofil anlegen

Jeder FTAC-Benutzer kann auf seiner Kennung mit CREATE-FT-PROFILE eigene Berechtigungsprofile einrichten. Vom FTAC-Verwalter vorgegebene Berechtigungsprofile müssen vom Benutzer mit MODIFY-FT-PROFILE (siehe Seite 253ff) aktiviert werden, bevor sie verwendet werden können.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

CREATE-**FT-PROF**ILE - Darstellung der FTP-relevanten Operanden

NAME = <alphanum-name 1..8>

,**TRANS**FER-**AD**MISSION = ***NOT-SPEC**IFIED / <alphanum-name 8..32>(...) /
<c-string 8..32 with-low> (...) / <x-string 15..64>(...) / ***SEC**RET

<alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
VALID = ***YES** / ***NO**
,**USAGE** = ***PRIVATE** / ***PUB**LIC
,**EXP**IRATION-**DATE** = ***NOT-REST**RICTED / <date 8..10>

,**PRIV**ILEGED = ***NO**

,**IGN**ORE-**MAX-LEV**ELS = ***NO** / ***YES** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
,**IN**BOUND-**SEND** = ***NO** / ***YES**
,**IN**BOUND-**REC**EIVE = ***NO** / ***YES**
,**IN**BOUND-**MAN**AGEMENT = ***NO** / ***YES**

,**USER-AD**MISSION = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
USER-IDENTIFICATION = ***OWN** / <name 1..8>
,**ACCOUNT** = ***OWN** / ***NOT-SPEC**IFIED / <alphanum-name 1..8>
,**PASS**WORD = ***OWN** / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / ***NONE** / ***SEC**RET

,**INIT**IATOR = ***REM**OTE

,**TRANS**FER-**DIR**ECTION = ***NOT-REST**RICTED / **FROM**-PARTNER / **TO**-PARTNER

,**PART**NER = ***NOT-REST**RICTED / list-poss(50): <text 1..200 with-low>

,**MAX-PART**NER-**LEV**EL = ***NOT-REST**RICTED / <integer 0..100>

,**FILE-NAME** = ***NOT-REST**RICTED / ***EXP**ANSION(...)

,***EXP**ANSION(...)
PREFIX = <filename 1..53> / <partial-filename 2..53> / <c-string 1..511 with-low>

,**FILE-PASS**WORD = ***NOT-REST**RICTED / ***NONE** / <c-string 1...4> / <x-string 1...8> /
<integer -2147483648...2147483647> / ***SEC**RET

,**WRITE**-MODE = ***NOT-REST**RICTED / **NEW**-FILE / **REP**LACE-FILE / **EXT**END-FILE

,**FT-FUNC**TION = ***NOT-REST**RICTED / list-poss(4): ***TRANS**FER-**F**ILE / ***MOD**IFY-**F**ILE-**ATTR**IBUTES / ***READ-DIR**ECTORY / ***FILE-PROC**ESSING

Beschreibung der Operanden

NAME=<alphanum-name 1..8>
Mit NAME geben Sie dem Berechtigungsprofil einen Namen. Dieser Name muss unter allen Berechtigungsprofilen auf Ihrer Kennung eindeutig sein. Wenn bereits ein Berechtigungsprofil dieses Namens existiert, lehnt FTAC das Kommando mit folgender Meldung ab:

```
FTC0100 FT-Profil existiert bereits
```

Mit dem Kommando SHOW-FT-PROFILE (siehe Seite 279ff) können Sie sich über die bereits vergebenen Namen informieren. Für diese Information reicht es, das Kommando SHOW-FT-PROFILE ohne Operanden einzugeben.

TRANSFER-ADMISSION=
Mit TRANSFER-ADMISSION definieren Sie eine Zugangsberechtigung. Wenn diese Zugangsberechtigung in einem FTP-LOGON statt der LOGON-Berechtigung angegeben wird, dann gelten die in diesem Berechtigungsprofil definierten Zugriffsrechte. Diese Zugangsberechtigung muss in Ihrem gesamten openFT-System eindeutig sein, damit es keine Kollisionen mit Zugangsberechtigungen gibt, die andere FTAC-Benutzer für andere Zugriffsrechte definiert haben.

Wenn die von Ihnen gewählte Zugangsberechtigung bereits vergeben ist, lehnt FTAC das Kommando mit der folgenden Meldung ab:

```
FTC0101 Zugangsberechtigung existiert bereits
```

TRANSFER-ADMISSION=*NOT-SPECIFIED
Mit dieser Angabe richten Sie ein Profil ohne Zugangberechtigung ein. Ein derartiges Profil ist solange gesperrt, bis Sie eine gültige Zugangberechtigung vergeben.

TRANSFER-ADMISSION=<alphanum-name 8..32>(...) /
<c-string 8..32 with-low>(...) / <x-string 15..64>(...)
Die Zeichenkette muss im FTP-Auftrag als Zugangsberechtigung angegeben werden. Die alphanumerische Eingabe wird intern immer in Kleinbuchstaben abgelegt.

VALID=*YES
Die Zugangsberechtigung ist gültig.

VALID=*NO
Die Zugangsberechtigung ist nicht gültig. Mit dieser Angabe kann das Profil gesperrt werden.

USAGE=*PRIVATE
Ihr Profil wird aus Sicherheitsgründen gesperrt, sobald unter einer fremden Kennung versucht wird, die von Ihnen bereits verwendete TRANSFER-ADMISSION ein zweites Mal zu vergeben.

USAGE=*PUBLIC
Ihr Profil wird auch dann nicht gesperrt, wenn Ihre TRANSFER-ADMISSION durch einen anderen Nutzer „entdeckt“ wurde. „Entdeckt“ bedeutet, dass von einer zweiten Benutzerkennung versucht wurde, dieselbe TRANSFER-ADMISSION nochmals zu vergeben. Dies wird aus Sicherheitsgründen abgelehnt.

EXPIRATION-DATE=*NOT-RESTRICTED
Die Verwendung der Zugangsberechtigung ist zeitlich nicht eingeschränkt.

EXPIRATION-DATE=<date 8..10>
Die Verwendung der Zugangsberechtigung ist nur bis zu dem angegebenen Datum (max. 31.12. 2020) möglich. Die Angabe muss in der Form JJJJ-MM-TT bzw. JJ-MM-TT erfolgen.

TRANSFER-ADMISSION=*SECRET
Sie erhalten vom System die Aufforderung, die Zugangsberechtigung einzugeben, diese wird aber nicht auf dem Bildschirm sichtbar. Die Operanden VALID, USAGE und EXPIRATION-DATE können in diesem Fall ebenfalls verdeckt eingegeben werden.

PRIVILEGED=*NO
Das Berechtigungprofil ist nicht privilegiert.
FTP-Aufträge, die mit einem privilegierten Berechtigungsprofil abgewickelt werden, unterliegen nicht den Einschränkungen, die für MAX-ADM-LEVEL im Berechtigungssatz eingestellt sind. Nur der FTAC-Verwalter darf Profile privilegieren.

IGNORE-MAX-LEVELS=
Mit IGNORE-MAX-LEVELS kann festgelegt werden, für welche der Grundfunktionen die Einschränkungen des Berechtigungssatzes außer Kraft gesetzt werden sollen. Die eigenen MAX-USER-LEVELS können so überschritten werden. Die MAX-ADM-LEVELS im Berechtigungssatz können nur mit einem vom FTAC-Verwalter privilegierten Berechtigungsprofil wirksam überschritten werden. Der FTAC-Benutzer kann sich für spezielle Aufgaben (z.B. Senden einer bestimmten Datei in ein Partnersystem, mit dem er normalerweise keinen FTP betreiben darf) ein Berechtigungsprofil anlegen, in dem das Überschreiten des Berechtigungssatzes vorgesehen ist. Dieses Profil muss vom FTAC-Verwalter explizit privilegiert werden.
Wenn Sie IGNORE-MAX-LEVELS=*YES angeben, werden die Vorgaben für **alle** Grundfunktionen außer Kraft gesetzt. Wenn Sie für **bestimmte** Grundfunktion den Berechtigungssatz ignorieren wollen, müssen Sie mit den weiter unten folgenden Operanden arbeiten.

IGNORE-MAX-LEVELS=*NO
FTP-Aufträge, die mit diesem Berechtigungsprofil abgewickelt werden, unterliegen den Einschränkungen des Berechtigungssatzes.

IGNORE-MAX-LEVELS=*YES
*YES berechtigt dazu, mit Partnersystemen zu kommunizieren, deren Sicherheitsstufe die Angaben im Berechtigungssatz überschreitet. Solange Ihr Profil nicht privilegiert ist, können Sie sich nur über die MAX-USER-LEVELS im Berechtigungssatz hinwegsetzen, nicht aber über die MAX-ADM-LEVELS. Angaben zur aktuellen Einstellung der MAX-USER-LEVELS und MAX-ADM-LEVELS erhalten Sie mit dem Kommando SHOW-FT-ADMISSION-SET (siehe Beispiel auf Seite 268).

IGNORE-MAX-LEVELS=*PARAMETERS(...)
Mit den folgenden Operanden werden die Vorgaben für die einzelnen Grundfunktionen selektiv außer Kraft gesetzt.

INBOUND-SEND=*NO
Die mit der Grundfunktion „inbound senden“ maximal erreichbare Sicherheitsstufe wird durch den Berechtigungssatz festgelegt.

INBOUND-SEND=*YES
Für die Grundfunktion „inbound senden“ können Sie sich mit diesem Berechtigungsprofil über die MAX-USER-LEVELS hinwegsetzen. Wenn Ihr Profil privilegiert ist, sind Sie auch nicht an die Beschränkungen der MAX-ADM-LEVELS gebunden. Außerdem kann zusätzlich die Teilkomponente „Anzeigen von Dateiattributen“ der Grundfunktion „inbound Dateimanagement“ genutzt werden.

INBOUND-RECEIVE=*NO
Die mit der Grundfunktion „inbound empfangen“ maximal erreichbare Sicherheitsstufe wird durch Ihren Berechtigungssatz festgelegt.

INBOUND-RECEIVE=*YES
Sie können sich mit diesem Profil über Ihre Vorgabe für „inbound empfangen“ in den MAX-USER-LEVELS hinwegsetzen. Wenn Ihr Profil privilegiert ist, sind Sie auch nicht an die Beschränkungen der MAX-ADM-LEVELS gebunden. Außerdem können zusätzlich folgende Teilkomponenten der Grundfunktion „inbound Dateimanagement“ genutzt werden:
– Löschen von Dateien, sofern die Dateiattribute entsprechend gesetzt sind,
– Ändern von Dateiattributen, wenn die Grundfunktion „inbound Dateimanagement“ im Berechtigungssatz oder im Berechtigungsprofil zugelassen wurde.

INBOUND-MANAGEMENT = *NO
Die mit der Grundfunktion „inbound Dateimanagement“ maximal erreichbare Sicherheitsstufe wird durch den Berechtigungssatz festgelegt.

INBOUND-MANAGEMENT=*NO
Die mit der Grundfunktion „inbound Dateimanagement“ maximal erreichbare Sicherheitsstufe wird durch den Berechtigungssatz festgelegt.

INBOUND-MANAGEMENT=*YES
Für die Grundfunktion „inbound Dateimanagement“ können Sie sich mit diesem Berechtigungsprofil über die MAX-USER-LEVELS hinwegsetzen. Wenn Ihr Profil privilegiert ist, sind Sie auch nicht an die Beschränkungen der MAX-ADM-LEVELS gebunden. Die zur Grundfunktion „inbound Dateimanagement“ gehörende Teilkomponente „Ändern von Dateiattributen“ funktioniert nur dann, wenn im Berechtigungssatz oder im Berechtigungsprofil die Grundfunktion „inbound empfangen“ zugelassen wurde.

USER-ADMISSION=
Mit USER-ADMISSION geben Sie als Benutzer die Kennung an, unter der das Profil abgespeichert wird. FTP-Aufträge, die mit diesem Berechtigungsprofil arbeiten, greifen im lokalen System auf die angegebene Kennung zu.

USER-ADMISSION=*OWN
Für USER-IDENTIFICATION und ACCOUNT werden die Angaben für Ihre Benutzerkennung und Ihre Abrechnungsnummer aus Ihrer LOGON-Berechtigung übernommen. Ein etwaiges BS2000-Kennwort wird erst zu dem Zeitpunkt aus Ihrer LOGON-Berechtigung übernommen, zu dem ein FTP-Auftrag auf das Berechtigungsprofil zugreift.

USER-ADMISSION=*PARAMETERS(...)
Sie können die Benutzerkennung auch in ihren einzelnen Bestandteilen angeben. Damit können Sie beispielsweise erreichen, dass FTP-Aufträge, die mit diesem Berechtigungsprofil arbeiten, unter einer anderen Abrechnungsnummer abgerechnet werden. Ein anderer Einsatzfall ist das Vorgeben eines Kennwortes im Berechtigungsprofil. FTP-Aufträge, die mit diesem Berechtigungsprofil arbeiten, funktionieren nur dann, wenn Ihr aktuelles LOGON-Kennwort mit diesem vorgegebenen Kennwort übereinstimmt.

USER-IDENTIFICATION=*OWN / <name 1..8>
Mit USER-IDENTIFICATION geben Sie Ihre Benutzerkennung im BS2000 an. Beide Angaben haben die gleiche Wirkung.

ACCOUNT=
Mit ACCOUNT geben Sie die Abrechnungsnummer an, unter der ein FTP-Auftrag abgerechnet werden soll, wenn er mit diesem Berechtigungsprofil arbeitet.

ACCOUNT=*OWN
Die Abrechnungsnummer wird aus Ihrer LOGON-Berechtigung übernommen.

ACCOUNT=<alphanum-name 1..8>
Unter der angegebenen Abrechnungsnummer soll ein FTP-Auftrag abgerechnet werden, wenn er auf dieses Berechtigungsprofil zugreift. Sie können jede zu Ihrer Benutzerkennung gehörende Abrechnungsnummer angeben.

PASSWORD=
Mit PASSWORD geben Sie ein zu Ihrer Benutzerkennung gehörendes BS2000-Kennwort an.

PASSWORD=*OWN
Wenn ein FTP-Auftrag auf dieses Berechtigungsprofil Bezug nimmt, setzt FTAC das zu diesem Zeitpunkt gültige BS2000-Kennwort Ihrer Benutzerkennung ein. Damit wird verhindert, dass bei einer etwaigen Änderung des BS2000-Kennwortes auch das Berechtigungsprofil geändert werden muss.

PASSWORD=*NONE
Für die Benutzerkennung wird kein BS2000-Kennwort benötigt.

PASSWORD=<c-string 1..8> / <x-string 1..16>
Das angegebene Kennwort wird in dem Moment mit dem aktuellen LOGON-Kennwort verglichen, in dem ein FTP-Auftrag auf das Berechtigungsprofil zugreift. Widersprechen sich die Angaben, wird der FTP-Auftrag abgelehnt.

PASSWORD=*SECRET
Sie erhalten vom System die Aufforderung, das Kennwort einzugeben. Die Eingabe erscheint dann nicht auf dem Bildschirm.

INITIATOR=*REMOTE
FTP-Aufträge werden immer als *REMOTE behandelt, daher müssen Berechtigungsprofile für FTP mindestens die Einstellung *REMOTE aufweisen. Wird dasselbe Profil auch für openFT verwendet, ist beispielsweise auch die Einstellung (*LOCAL,*REMOTE) zulässig.

TRANSFER-DIRECTION=
Mit TRANSFER-DIRECTION legen Sie fest, welche Übertragungsrichtung mit diesem Berechtigungsprofil benutzt werden darf. Die Übertragungsrichtung ist immer von dem BS2000-FTP-Server aus zu sehen, in dem Sie das Berechtigungsprofil definiert haben.

TRANSFER-DIRECTION=*NOT-RESTRICTED
Mit diesem Berechtigungsprofil dürfen sowohl Daten vom Client zum Server als auch vom Server zum Client übertragen werden.

TRANSFER-DIRECTION=*FROM-PARTNER
Mit diesem Berechtigungsprofil dürfen nur Daten von einem Client zum Server übertragen werden. Damit ist auch kein Anzeigen von Dateiattributen bzw. Dateiverzeichnissen (Teilkomponenten des „inbound Dateimanagements“) möglich, d.h. folgende Server-Kommandos sind nicht zugelassen: *cdup*, *xcup*, *cwd*, *xcwd*, *list*, *nlst*, *pwd*, *xpwd*, *retr*.

TRANSFER-DIRECTION=*TO-PARTNER
Mit diesem Berechtigungsprofil dürfen nur Daten vom Server zu einem Client-System übertragen werden. Damit ist auch kein Modifizieren von Dateiattributen und kein Löschen von Dateien (Teilkomponenten des „inbound Dateimanagements“) möglich, d.h. folgende Server-Kommandos sind nicht zugelassen: *appe*, *dele*, *site file*, *mkd*, *xmkd*, *rmd*, *xrmd*, *rnfr*, *stor, stou*.

PARTNER=
Mit PARTNER können Sie festlegen, dass dieses Berechtigungsprofil nur für FTP-Aufträge benutzt werden kann, die mit einem bestimmten Client-System abgewickelt werden.

PARTNER=*NOT-RESTRICTED
Der Einsatzbereich dieses Berechtigungsprofils ist nicht auf FTP-Aufträge mit bestimmten Partnersystemen eingeschränkt.

PARTNER=list-poss(50): <text 1..200 with-low>
Das Berechtigungsprofil lässt nur solche FTP-Aufträge zu, die mit den angegebenen Client-Systemen abgewickelt werden. Maximal 50 Client-Systeme können angegeben werden. Die Gesamtlänge aller Partner darf 1000 Zeichen nicht überschreiten. Sie können den Namen aus der Partnerliste oder die Adresse des Partnersystems angeben, siehe auch openFT Benutzerhandbuch. Es wird empfohlen, den Namen aus der Partnerliste zu verwenden. Als Orientierung, wie eine Partneradresse in ein FTAC-Profil eingetragen werden soll, dient das jeweilige Format aus der Langform der Logging-Ausgabe.

MAX-PARTNER-LEVEL=
Mit MAX-PARTNER-LEVEL kann eine maximale Sicherheitsstufe angegeben werden. Bei FTP-Aufträgen wird dem Client-System eine von der System-Administration vorgegebene Sicherheitsstufe bzw. standardmäßig die Sicherheitsstufe 100 zugeordnet.

MAX-PARTNER-LEVEL=*NOT-RESTRICTED
Werden FTP-Aufträge mit diesem Berechtigungsprofil abgewickelt, so wird die maximal erreichbare Sicherheitsstufe durch den Berechtigungssatz festgelegt.

MAX-PARTNER-LEVEL=<integer 0..100>
Wenn Sie MAX-PARTNER-LEVEL kleiner als die von der System-Administration vorgegebene Sicherheitsstufe bzw. kleiner als den Standard-Wert 100 setzen, sperren Sie dadurch das Berechtigungsprofil (vorübergehend) für FTP-Aufträge.

FILE-NAME=
Mit FILE-NAME legen Sie fest, auf welche Dateien unter Ihrer Kennung FTP-Aufträge zugreifen dürfen, die mit diesem Berechtigungsprofil arbeiten.

FILE-NAME=*NOT-RESTRICTED
Das Berechtigungsprofil erlaubt uneingeschränkten Zugriff auf alle Dateien der Benutzerkennung.

FILE-NAME=*EXPANSION (PREFIX=<full-filename 1..53> / <partial-filename 2..53> / <c-string 1..511 with-low>)
Durch diese Angabe kann der Zugriff auf eine Menge von Dateien beschränkt werden, die alle mit demselben Präfix beginnen. Wenn in einem FTP-Auftrag, der mit diesem Berechtigungsprofil arbeitet, ein Dateiname angegeben wird, stellt FTAC vor diesen Dateinamen abhängig vom aktuell eingestellten Arbeitsverzeichnis das mit EXPANSION definierte Präfix. Der FTP-Auftrag darf dann auf die Datei *PräfixDateiname* zugreifen.

Ein Wechsel zwischen POSIX- und DVS-Dateisystem ist nicht möglich. Wenn das Präfix ein „/“ enthält oder mit „.“ beginnt, kann nur auf das POSIX-Dateisystem zugegriffen werden. In den übrigen Fällen ist der Zugriff nur auf das DVS-Dateisystem möglich.

Beispiel

PREFIX=DAGOBERT.; ein FTP-Auftrag, in dem als Dateiname BOERSE angegeben wurde, greift dann auf die Datei DAGOBERT.BOERSE zu.

Bitte beachten Sie, dass der Teil eines DVS-Dateinamens, der im FTP-Kommando angegeben wird, dennoch vom Typ <full-filename> sein muss.

FILE-PASSWORD=
Mit FILE-PASSWORD können Sie ein Kennwort für Dateien in das Berechtigungsprofil eintragen. Die FTAC-Funktionalität erlaubt dann nur Zugriffe auf Dateien, die mit diesem Kennwort geschützt sind, sowie auf nicht geschützte Dateien. Wenn in einem Berechtigungsprofil ein FILE-PASSWORD steht, darf das Kennwort nicht mehr in einem FTP-Auftrag angegeben werden, der mit diesem Berechtigungsprofil arbeitet. Sie können somit Anwendern in fernen Systemen einen Zugriff auf bestimmte Dateien erlauben, ohne deren Dateikennworte verraten zu müssen.

FILE-PASSWORD=*NOT-RESTRICTED
Das Berechtigungsprofil erlaubt Zugriffe auf alle Dateien; ist für eine Datei ein Kennwort vergeben, muss dieses im FTP-Auftrag angegeben werden.

FILE-PASSWORD=*NONE
Das Berechtigungsprofil erlaubt nur Zugriffe auf Dateien ohne Dateikennwort.

FILE-PASSWORD=<c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647>
Das Berechtigungsprofil erlaubt nur Zugriffe auf Dateien, die mit dem angegebenen Dateikennwort geschützt sind, sowie auf nicht geschützte Dateien. Im FTP-Auftrag darf das Kennwort, das bereits im Profil angegeben ist, nicht wiederholt werden.

FILE-PASSWORD=*SECRET
Sie erhalten vom System die Aufforderung, das Kennwort einzugeben. Die Eingabe erscheint dann nicht auf dem Bildschirm.

WRITE-MODE=
Mit WRITE-MODE legen Sie die für diesen FTP-Auftrag zulässige Schreibregel fest. WRITE-MODE wirkt nur, wenn die Empfangsdatei im selben System liegt, in dem auch dieses Berechtigungsprofil definiert ist. In FTP-Kommandos wird die Schreibregel nicht explizit angegeben, sondern ist impliziter Bestandteil des FTP-Kommandos:

appe	*EXTEND-FILE
stor, *rnfr*, *site file*, *dele*, *rmd*, *xrmd*	*REPLACE-FILE
stou	*NEW-FILE

WRITE-MODE=*NOT-RESTRICTED
In einem FTP-Auftrag, der auf dieses Berechtigungsprofil zugreift, sind alle FTP-Schreibkommandos zugelassen.

WRITE-MODE=*NEW-FILE
Die FTP-Kommandos *dele*, *rmd* und *xrmd* sind nicht zulässig.

WRITE-MODE=*REPLACE-FILE
Das FTP-Kommandos *stou* ist nicht zulässig.

WRITE-MODE=*EXTEND-FILE
Die FTP-Kommandos *stor*, *stou*, *dele*, *rmd* und *xrmd* sind nicht zulässig.

FT-FUNCTION=
Dieser Operand ermöglicht die Beschränkung der Gültigkeit des Profils auf bestimmte FTP-Funktionen (= Dateiübertragungs- und Dateimanagement-Funktionen).

FT-FUNCTION=*NOT-RESTRICTED
Die FTP-Funktionen stehen im vollen Umfang zur Verfügung.

FT-FUNCTION=(*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES,
***READ-DIRECTORY, *FILE-PROCESSING)**

Folgende Funktionen stehen zur Verfügung:

***TRANSFER-FILE**
Das Berechtigungsprofil darf für die Funktionen „Dateien übertragen“, „Dateiattribute ansehen“ und „Dateien löschen“ benutzt werden.

Folgende Server-Kommandos sind nicht zugelassen:

list, *nlist*, *pwd*, *xpwd*, *cwd*, *xcwd*, *cdup*, *xcup*, *rnfr*, *size*, *mdtm*

***MODIFY-FILE-ATTRIBUTES**
Das Berechtigungsprofil darf für die Funktionen „Dateiattribute ansehen“ und „Dateiattribute modifizieren“ benutzt werden.

Folgende Server-Kommandos sind nicht zugelassen:

retr, *stor*, *appe*, *stou*, *dele*, *list*, *nlist*, *pwd*, *xpwd*, *cwd*, *xcwd*, *cdup*, *xcup*, *size*, *mdtm*

***READ-DIRECTORY**
Das Berechtigungsprofil darf für die Funktionen „Dateiverzeichnisse ansehen“ und „Dateiattribute ansehen“ benutzt werden.

Folgende Server-Kommandos sind nicht zugelassen:

retr, *stor*, *appe*, *stou*, *dele*

***FILE-PROCESSING**
Das Berechtigungsprofil darf für die Funktionen „Vorverarbeitung“ und „Nachverarbeitung“ benutzt werden. Zusätzlich muss auch die Funktion „Dateien übertragen“ erlaubt sein. Die Angabe von *FILE-PROCESSING spielt nur bei FTAC-Profilen ohne Dateinamens-Präfix eine Rolle. Ansonsten entscheidet das erste Zeichen des Dateinamens-Präfix darüber, ob mit diesem FTAC-Profil nur normale Dateiübertragungen (kein Pipe-Zeichen |) oder nur Vor- bzw. Nachverarbeitungen (Pipe-Zeichen |) möglich sind.

Beispiel

Dagobert Duck will ein Berechtigungsprofil zu folgendem Zweck anlegen: Dussel Duck, Sachbearbeiter in der Duck'schen Goldmine, die ihren eigenen BS2000-Rechner besitzt, soll seinen Monatsbericht regelmäßig per FTP an seinen Chef Dagobert an den Rechner DAGODUCK schicken können. Die Datei soll dort immer mit dem Präfix MONATSBERICHTE. versehen werden. Da im Berechtigungssatz von Dagobert keine „inbound"-Aufträge erlaubt sind, muss Dagobert das Profil privilegieren (das darf er, weil er FTAC-Verwalter ist).

Das zum Anlegen eines solchen Berechtigungsprofils nötige Kommando lautet:

```
/CREATE-FT-PROFILE NAME=GOLDMOBE,                                    -
/                    TRANSFER-ADMISSION='MonatsberichtfuerdenChef',  -
/                    PRIVILEGED=*NO,                                 -
/                    IGNORE-MAX-LEVELS=*YES,                         -
/                    TRANSFER-DIRECTION=*FROM-PARTNER,               -
/                    FILE-NAME=*EXPANSION(PREFIX=MONATSBERICHTE.),   -
/                    WRITE-MODE=*REPLACE-FILE
```

Die Kurzform dieses Kommandos lautet:

```
/CRE-FT-PROF GOLDMOBE,TRANS-AD='MonatsberichtfuerdenChef',           -
/PRIV=*YES,IGN-MAX-LEV=*YES,TRANS-DIR=*FROM,                         -
/FILE-NAME=*EXP(PREF=MONATSBERICHTE.),WRITE=*REPL
```

Dussel Duck, der am BS2000-Rechner der Goldmine den Monatsbericht in der Datei LUEGENFUERDENALTEN stehen hat, kann diesen dann mit den folgenden FTP-Kommandos des BS2000-FTP-Clients an die Zentrale DAGODUCK senden:

```
→ ftp> open DAGODUCK
   Connected to DAGODUCK, port 21.
   220 DAGODUCK FTP server ... ready.
   Name (DAGODUCK:DUSSDUCK):

→ *$FTAC
   331 Send your FTAC transfer admission as password
   Password (DAGODUCK:$FTAC):

→ *MonatsberichtfuerdenChef
   230 $FTAC login ok, access restrictions apply.

→ ftp> put LUEGENFUERDENALTEN GOLDMINE
   200 PORT command successful.
   150 Opening data connection for GOLDMINE (139.25.24.2,4102).
   22595 bytes sent in 0.06 seconds (3.6e+02 Kbytes/s)
   226 Transfer complete.(SAM-IO)

→ ftp> bye
   221 Goodbye.
```

Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	0	FTC0051	Eine Userid mit demselben Namen existiert bereits im System.
0	0	FTC0056	Die Zugangsberechtigung ist gesperrt.
0	64	FTC0100	Es gibt bereits ein FT-Profil mit dem angegebenen Namen.
0	64	FTC0101	Es gibt bereits ein FT-Profil mit der angegebenen Transfer-Admission.
0	64	FTC0150	Das Kennwort zur Berechtigung fehlt.
0	64	FTC0153	Die angegebene Eigentuemeridentifikation ist nicht die eigene Benutzerkennung.
0	64	FTC0157	Keine Berechtigung zum Einrichten des Profils.
0	64	FTC0172	Die angegebene User-Admission existiert nicht im System.
0	64	FTC0173	Die angegebene Processing-Admission existiert nicht im System.
0	64	FTC0178	Der angegebene Partnername kommt mehrmals vor.
0	64	FTC0182	Verstoß gegen die maximale Länge für Partnernamen.
0	64	FTC0200	Die Summe der Laengen der beiden Folgeverarbeitungskommandos ist zu gross.
0	64	FTC0255	Ein Systemfehler ist aufgetreten.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im openFT-Handbuch „Meldungen“.

5.2.5 DELETE-FT-PROFILE - Berechtigungsprofil löschen

Mit dem Kommando DELETE-FT-PROFILE können Sie alle Berechtigungsprofile löschen, deren Eigentümer Sie sind. Durch gelegentliches Durchforsten Ihres Bestandes sollten Sie dafür sorgen, dass kein Speicherplatz durch überflüssige Berechtigungsprofile blockiert wird.
Mit SHOW-FT-PROFILE (siehe Seite 279ff) können Sie sich die Profile anschauen und dann darüber entscheiden, ob sie noch benötigt werden.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

DELETE**-FT-PROF**ILE - Darstellung der FTP-relevanten Operanden

NAME = ***ALL** / <alphanum-name 1..8> / *STD

,**SEL**ECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
TRANSFER**-AD**MISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name8..32> /
<c-string 8..32 with-low> / <x-string 15..64> / ***SEC**RET
,**OWN**ER**-ID**ENTIFICATION = ***OWN** / <name 1..8>

Beschreibung der Operanden

NAME=
Mit NAME können Sie das zu löschende Berechtigungsprofil über seinen Namen ansprechen.

NAME=*ALL
Mit *ALL können Sie alle Berechtigungsprofile löschen. Der FTAC-Benutzer entfernt mit Angabe dieses Operanden alle seine Berechtigungsprofile, wenn er nicht mit SELECT-PARAMETER ein spezielles Profil auswählt.

NAME=<alphanum-name 1..8>
Sie wollen das Berechtigungsprofil mit dem angegebenen Namen löschen.

NAME = *STD
Löscht das Standard-Berechtigungsprofil für die eigene Kennung.

SELECT-PARAMETER=
Mit SELECT-PARAMETER können Sie Auswahlkriterien für die zu löschenden Berechtigungsprofile angeben.

SELECT-PARAMETER=*OWN
Mit *OWN löschen Sie Ihre eigenen Berechtigungsprofile.

SELECT-PARAMETER=*PARAMETERS(...)
Mit dieser Struktur können Sie die einzelnen Auswahlkriterien angeben.

TRANSFER-ADMISSION=
Mit TRANSFER-ADMISSION können Sie die Zugangsberechtigung eines Berechtigungsprofils als Auswahlkriterium zum Löschen heranziehen.

TRANSFER-ADMISSION=*ALL
Sie wollen Berechtigungsprofile unabhängig von der TRANSFER-ADMISSION löschen.

TRANSFER-ADMISSION=*NOT-SPECIFIED
Sie wollen Berechtigungsprofile löschen, für die keine Zugangsberechtigung definiert ist.

TRANSFER-ADMISSION=<alphanum-name 8..32> /
<c-string 8..32 with-low> / <x-string 15..64>
Sie wollen das Berechtigungsprofil löschen, das mit dieser Zugangsberechtigung angesprochen wird. Der FTAC-Benutzer kann nur Zugangsberechtigungen seiner eigenen Berechtigungsprofile angeben.

TRANSFER-ADMISSION=*SECRET
Sie erhalten vom System die Aufforderung, die Zugangsberechtigung einzugeben, diese wird aber nicht auf dem Bildschirm sichtbar.

OWNER-IDENTIFICATION =*OWN / <name 1..8>
OWNER-IDENTIFICATION berechtigt den FTAC-Benutzer, seine eigenen Berechtigungsprofile unter dieser USER-ID zu löschen. Beide Angaben haben die gleiche Wirkung.

Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
1	0	FTC0053	Es existiert kein FT-Profil zu den angegebenen Kriterien.
0	64	FTC0150	Das Kennwort zur Berechtigung fehlt.
0	64	FTC0153	Die angegebene Eigentuemeridentifikation ist nicht die eigene Benutzerkennung.
0	64	FTC0255	Ein Systemfehler ist aufgetreten.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im openFT-Handbuch „Meldungen“.

5.2.6 MODIFY-FT-ADMISSION-SET - Berechtigungssatz ändern

Mit dem Kommando MODIFY-FT-ADMISSION-SET können Sie den Berechtigungssatz Ihrer Benutzerkennung ändern. Dabei dürfen Sie auf zwei Bestandteile des Berechtigungssatzes zugreifen:

- Sie können ein Kennwort definieren, das anschließend bei fast allen FTAC-Kommandos (Ausnahme: die /SHOW...-Kommandos) angegeben werden muss. Damit verhindern Sie, dass ein anderer Benutzer, der auf Ihrer Kennung arbeitet, FTAC-Kommandos eingeben kann.

  Es gibt keine Möglichkeit, ein FTAC-Kennwort ausgegeben zu bekommen. Wenn ein FTAC-Benutzer sein FTAC-Kennwort vergisst, kann nur der FTAC-Verwalter das Kennwort löschen oder ändern.

- Sie dürfen die Grenzwerte für die von Ihrer Kennung aus maximal erreichbaren Sicherheitsstufen (die MAX-USER-LEVELS) ändern. Die vom FTAC-Verwalter festgelegten Grenzwerte (MAX-ADM-LEVELS) können Sie nicht ausser Kraft setzen.
  Die MAX-USER-LEVELS wirken nur dann, wenn sie höher, d.h. restriktiver, als die MAX-ADM-LEVELS sind.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

MODIFY-**FT-ADM**ISSION-SET - Darstellung der FTP-relevanten Operanden

USER-IDENTIFICATION = ***OWN** / <alphanum-name 1...8>

,**SEL**ECT-PARAMETER = ***ALL**

,**MAX-LEV**ELS = ***UNCH**ANGED / ***STD** / <integer 0...100> / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

INBOUND-**SEND** = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**IN**BOUND-**REC**EIVE = ***UNCH**ANGED / ***STD** / <integer 0...100>

,**IN**BOUND-**MAN**AGEMENT = ***UNCH**ANGED / ***STD** / <integer 0...100>

Beschreibung der Operanden

USER-IDENTIFICATION=
Gibt die Benutzerkennung an, deren Berechtigungssatz geändert werden soll.

USER-IDENTIFICATION=*OWN
Der Berechtigungssatz der Kennung, unter der Sie gerade arbeiten, soll geändert werden.

USER-IDENTIFICATION=<alphanum-name 1..8>
Der Berechtigungssatz dieser Benutzerkennung soll geändert werden. Sie können als FTAC-Benutzer an dieser Stelle nur Ihre eigene USER-ID angeben.

SELECT-PARAMETER=*ALL
An dieser Stelle wird in späteren FTAC-Versionen die Angabe zusätzlicher Auswahlkriterien möglich sein.

MAX-LEVELS=
Mit diesem Operanden legen Sie fest, welche Sicherheitsstufe(n) Sie von der Kennung dieses Berechtigungssatzes aus mit welcher Grundfunktion erreichen können. Sie können entweder pauschal eine Sicherheitsstufe für alle Grundfunktionen angeben oder für die einzelnen Grundfunktionen verschiedene Sicherheitsstufen festlegen. Angaben des FTAC-Benutzers legen die MAX-USER-LEVELS fest, Angaben des FTAC-Verwalters definieren die MAX-ADM-LEVELS dieses Berechtigungssatzes. FTAC führt Berechtigungsprüfungen auf der Basis der jeweils kleinsten angegebenen Sicherheitsstufe durch. Der FTAC-Benutzer kann die vom FTAC-Verwalter für ihn vorgegebenen Werte nur unter-, nicht aber überschreiten.

MAX-LEVELS=*UNCHANGED
Die in diesem Berechtigungssatz festgelegten Sicherheitsstufen sollen nicht geändert werden.

MAX-LEVELS=*STD
Für diesen Berechtigungssatz sollen die Festlegungen des Standardberechtigungssatzes gelten. Damit wird der Berechtigungssatz aus der Berechtigungsdatei gelöscht. Dies ist auch dann möglich, wenn die Benutzerkennung bereits gelöscht wurde.

MAX-LEVELS=<integer 0..100>
Mit diesem Wert können Sie pauschal eine maximale Sicherheitsstufe für alle Grundfunktionen festlegen. Ein Wert kleiner als die von der System-Administration vorgegebene Sicherheitsstufe bzw. kleiner als der Standard-Wert 100 bedeutet, dass auf dieser Kennung bis auf weiteres (erneutes Ändern des Berechtigungssatzes) kein per FTAC überprüfter FTP-Zugriff möglich ist.

MAX-LEVELS=*PARAMETERS(...)
Mit dieser Struktur können Sie für jede einzelne Grundfunktion eine maximale Sicherheitsstufe festlegen. FTP-Partner haben immer die von der System-Administration vorgegebene Sicherheitsstufe, die standardmäßig 100 beträgt.

INBOUND-SEND=
Legt die maximal erreichbare Sicherheitsstufe für die Grundfunktion „inbound Senden“ fest. Alle Partnersysteme, die höchstens diese Sicherheitsstufe haben, dürfen Dateien vom Eigentümer des Berechtigungssatzes anfordern.

INBOUND-SEND=*UNCHANGED
Der Wert für INBOUND-SEND bleibt unverändert.

INBOUND-SEND=*STD
Für INBOUND-SEND wird der Wert aus dem Standardberechtigungssatz eingesetzt.

INBOUND-SEND=<integer 0..100>
Für INBOUND-SEND wird diese maximale Sicherheitsstufe in den Berechtigungssatz eingetragen. Ein Wert kleiner als die von der System-Administration vorgegebene Sicherheitsstufe bzw. kleiner als der Standard-Wert 100 bedeutet, dass auf dieser Kennung INBOUND-SEND nicht möglich ist. In diesem Fall ist das Server-Kommando *retr* nicht zulässig.

INBOUND-RECEIVE=
Legt die maximal erreichbare Sicherheitsstufe für die Grundfunktion „inbound Empfangen“ fest.

INBOUND-RECEIVE=*UNCHANGED
Der Wert für INBOUND-RECEIVE bleibt unverändert.

INBOUND-RECEIVE=*STD
Für INBOUND-RECEIVE wird der Wert aus dem Standardberechtigungssatz eingesetzt.

INBOUND-RECEIVE=<integer 0..100>
Für INBOUND-RECEIVE wird diese maximale Sicherheitsstufe in den Berechtigungssatz eingetragen. Ein Wert kleiner als die von der System-Administration vorgegebene Sicherheitsstufe bzw. kleiner als der Standard-Wert 100 bedeutet, dass auf dieser Kennung INBOUND-RECEIVE nicht möglich ist. In diesem Fall sind folgende Server-Kommandos nicht zulässig: *stor*, *stou*, *appe*, *rnfr*, *dele*, *site file*.

INBOUND-MANAGEMENT=
legt die maximal erreichbare Sicherheitsstufe für die Grundfunktion „inbound Dateimanagement“ fest. Alle Partnersysteme, die höchstens diese Sicherheitsstufe haben, dürfen in einem FTP-Auftrag die Änderung von Dateiattributen veranlassen und Verzeichnisse abfragen.

INBOUND-MANAGEMENT=*UNCHANGED
Der Wert für INBOUND-MANAGEMENT bleibt unverändert.

INBOUND-MANAGEMENT=*STD
Für INBOUND-MANAGEMENT wird der Wert aus dem Standardberechtigungssatz eingesetzt.

INBOUND-MANAGEMENT=<integer 0..100>
Für INBOUND-MANAGEMENT wird diese maximale Sicherheitsstufe in den Berechtigungssatz eingetragen. Ein Wert kleiner als die von der System-Administration vorgegebene Sicherheitsstufe bzw. kleiner als der Standard-Wert 100 bedeutet, dass auf dieser Kennung INBOUND-MANAGEMENT nicht möglich ist. In diesem Fall sind folgende Server-Kommandos nicht zulässig: *cwd*, *xcwd*, *list*, *nlst*, *mkd*, *xmkd*, *rmd*, *xrmd*, *pwd*, *xpwd*, *cdup*, *xcup*, *rnfr, size*

Beispiel

Donald informiert sich über seine Berechtigungssätze:

```
→ /SHOW-FT-ADMISSION-SET
```

Kurzform:

```
→ /SHOW-FT-ADM
```

Er erhält folgende Ausgabe:

```
   %                MAX. USER LEVELS              MAX. ADM LEVELS        ATTR
   % USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
   % DUCKTAIL 100  100  100  100  100  100   80   80  100  100   80  100
```

Donald verbietet die Grundfunktion „inbound Senden“ und damit das Lesen von Dateien auf seiner Kennung.

```
→ /MODIFY-FT-ADMISSION-SET MAX-LEVELS=*PARAMETERS(INBOUND-SEND=99)
```

Die Kurzform dieses Kommandos lautet

```
→ /MOD-FT-ADM MAX-LEV=(IN-SEND=99)
```

Zur Kontrolle lässt er sich die Berechtigungssätze noch einmal ausgeben.

```
→ /SHOW-FT-ADM
   %                MAX. USER LEVELS              MAX. ADM LEVELS        ATTR
   % USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
   % DUCKTAIL 100  100   99  100  100  100   80   80  100  100   80  100
```

Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	0	FTC0050	Eingestellte Sicherheitsstufe ueberschreitet Grenzwert des Verwalters und bleibt solange unwirksam,bis der Grenzwert des Verwalters entsprechend heraufgesetzt wurde.
0	64	FTC0150	Das Kennwort zur Berechtigung fehlt.
0	64	FTC0151	Die Aenderung ist dem Verwalter oder dem Eigentuemer vorbehalten.
0	64	FTC0152	Die angegebene Benutzerkennung ist nicht die eigene Benutzerkennung.
0	64	FTC0175	Der Operand „NEW-PASSWORD“ darf fuer *STD nicht angegeben werden.
0	64	FTC0176	Die angegebene Userid existiert nicht im System.
0	64	FTC0255	Ein Systemfehler ist aufgetreten.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im openFT-Handbuch „Meldungen“.

5.2.7 MODIFY-FT-PROFILE - Berechtigungsprofil ändern

Mit dem Kommando MODIFY-FT-PROFILE kann jeder FTAC-Benutzer seine Berechtigungsprofile ändern.

In einem privilegierten Berechtigungsprofil kann ein FTAC-Benutzer nur die Operanden TRANSFER-ADMISSION und PRIVILEGED ändern.

Sobald ein Berechtigungsprofil modifiziert wird, wird auch der Zeitstempel aktualisiert. Der Zeitstempel wird bei SHOW-FT-PROFILE INF=*ALL ausgegeben (LAST-MODIF). Der Zeitstempel wird auch dann aktualisiert, wenn Sie die Eigenschaften des Profils nicht ändern, d.h. MODIFY-FT-PROFILE mit dem Parameter NAME angeben, darüber hinaus aber keine weiteren Parameter.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

MODIFY**-FT-PROF**ILE - Darstellung der FTP-relevanten Operanden

NAME = ***ALL** / ***STD** / <alphanum-name 1..8>

,**SEL**ECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
| **TRANSF**ER**-AD**MISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name 8..32> /
| <c-string 8..32 with-low> / <x-string 15..64> / ***SEC**RET
| ,**OWN**ER**-ID**ENTIFICATION = ***OWN** / <name 1..8>

,**NEW-NAME** = ***OLD** / <alphanum-name 1..8>

,**TRANS**FER**-AD**MISSION = ***UNCH**ANGED / ***NOT-SPEC**IFIED / ***OLD-ADM**ISSION(...) /
<alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...) / ***SEC**RET

***OLD-ADM**ISSION(...)

<alphanum-name 8..32>(...) / <c-string 8..32 with-low>(...) / <x-string 15..64>(...)
| **VAL**ID = ***YES** / ***NO** / ***UNCH**ANGED
| ,**USAGE** = ***PRIVATE** / ***PUB**LIC / ***UNC**HANGED
| ,**EXP**IRATION**-DATE** = ***NOT-REST**RICTED / <date 8..10> / ***UNCH**ANGED

,**PRIV**ILEGED = ***UNCH**ANGED / ***NO**

,**IGN**ORE**-MAX-LEV**ELS = ***UNCH**ANGED / ***NO** / ***YES** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)
| ,**IN**BOUND**-SEND** = ***UNCH**ANGED / ***NO** / ***YES**
| ,**IN**BOUND**-REC**EIVE = ***UNCH**ANGED / ***NO** / ***YES**
| ,**IN**BOUND**-MAN**AGEMENT = ***UNCH**ANGED / ***NO** / ***YES**

```
,USER-ADMISSION = *UNCHANGED / *OWN / *PARAMETERS(...)

    *PARAMETERS(...)
        | USER-IDENTIFICATION = *OWN / <name 1..8>
        | ,ACCOUNT = *OWN / *NOT-SPECIFIED / <alphanum-name 1..8>
        | ,PASSWORD = *OWN / *NOT-SPECIFIED / <c-string 1..8> / <x-string 1..16> / *NONE / *SECRET

,INITIATOR = *UNCHANGED / *REMOTE

,TRANSFER-DIRECTION = *UNCHANGED / *NOT-RESTRICTED / FROM-PARTNER / TO-PARTNER

,PARTNER = *UNCHANGED / *NOT-RESTRICTED / *ADD(...) / *REMOVE(...) /
                        list-poss(50): <text 1..200 with-low>

    *ADD(...)
        | NAME = list-poss(50): <text 1..200 with-low>

    *REMOVE(...)
        | NAME = list-poss(50): <text 1..200 with-low>

,MAX-PARTNER-LEVEL = *UNCHANGED / *NOT-RESTRICTED / <integer 0..100>

,FILE-NAME = *UNCHANGED / *NOT-RESTRICTED / *EXPANSION(...)

    *EXPANSION(...)
        | PREFIX = <full-filename 1..53> / <partial-filename 2..53> / <c-string 1..511 with-low>

,FILE-PASSWORD = *UNCHANGED / *NOT-RESTRICTED / *NONE / <c-string 1...4> / <x-string 1...8> /
                        <integer -2147483648...2147483647> / *SECRET

,WRITE-MODE = *UNCHANGED / *NOT-RESTRICTED / *NEW-FILE / *REPLACE-FILE / *EXTEND-FILE

,FT-FUNCTION = *UNCHANGED / *NOT-RESTRICTED / list-poss(4):
                    *TRANSFER-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY /
                      *FILE-PROCESSING
```

Beschreibung der Operanden

NAME=
Mit NAME geben Sie den Namen des Berechtigungsprofils an, das Sie ändern wollen.

NAME=*ALL
Sie wollen alle Ihre Berechtigungsprofile gleichzeitig ändern.

NAME = *STD
Ändert das Standard-Berechtigungsprofil Ihrer Kennung.

NAME=<alphanum-name 1..8>
Sie wollen das Berechtigungsprofil mit diesem Namen ändern.

SELECT-PARAMETER=
Mit SELECT-PARAMETER können Sie eine Zugangsberechtigung angeben. Sie ändern dann das Berechtigungsprofil, das mit dieser Zugangsberechtigung angesprochen wird.

SELECT-PARAMETER=*OWN
Sie wollen Ihre eigenen Berechtigungsprofile ändern.

SELECT-PARAMETER=*PARAMETERS(...)
Mit dieser Struktur können Sie Auswahlkriterien für die Profile angeben, die Sie ändern wollen.

TRANSFER-ADMISSION=
Die Angabe von TRANSFER-ADMISSION an dieser Stelle wirkt als Auswahlkriterium für die Berechtigungsprofile, die Sie ändern wollen.

TRANSFER-ADMISSION=*ALL
Sie ändern alle Ihre Berechtigungsprofile, unabhängig von der Zugangsberechtigung.

TRANSFER-ADMISSION=*NOT-SPECIFIED
Es werden nur Berechtigungsprofile ohne definierte Zugangsberechtigung geändert.

TRANSFER-ADMISSION=<alphanum-name 8..32> /
<c-string 8..32 with-low> / <x-string 15..64>
Sie ändern das Berechtigungsprofil, das mit dieser Zugangsberechtigung angesprochen wird.

TRANSFER-ADMISSION=*SECRET
Sie erhalten vom System die Aufforderung, die Zugangsberechtigung einzugeben, diese wird aber nicht auf dem Bildschirm sichtbar.

OWNER-IDENTIFICATION=*OWN / <name 1..8>
OWNER-IDENTIFICATION berechtigt den FTAC-Benutzer, seine eigenen Berechtigungsprofile zu ändern. Beide Angaben haben die gleiche Wirkung.

NEW-NAME=
Mit NEW-NAME geben Sie Ihren Berechtigungsprofilen neue Namen (oder nicht).

NEW-NAME=*OLD
Der Name des Berechtigungsprofils bleibt unverändert.

NEW-NAME=<alphanum-name 1..8>
So soll der neue Name des Berechtigungsprofiles lauten. Dieser Name muss unter allen Berechtigungsprofilen auf Ihrer Kennung eindeutig sein. Wenn bereits ein Berechtigungsprofil dieses Namens existiert, lehnt FTAC das Kommando mit folgender Meldung ab:

```
FTC0100 FT-Profil existiert bereits
```

Mit dem Kommando SHOW-FT-PROFILE (siehe Seite 279ff) können Sie sich über die bereits vergebenen Namen informieren. Für diese Information reicht es, wenn Sie SHOW-FT-PROFILE ohne Operanden angeben.

TRANSFER-ADMISSION=
Mit TRANSFER-ADMISSION können Sie an dieser Stelle die mit einem Berechtigungsprofil verbundene Zugangsberechtigung ändern. Sie müssen darauf achten, dass die Zugangsberechtigung in Ihrem openFT (BS2000)-System eindeutig bleibt. Wenn die von Ihnen gewählte Zugangsberechtigung bereits vergeben ist, lehnt FTAC das Kommando mit der folgenden Meldung ab:

```
FTC0101 Zugangsberechtigung existiert bereits
```

TRANSFER-ADMISSION=*UNCHANGED
Die Zugangsberechtigung bleibt unverändert.

TRANSFER-ADMISSION=*NOT-SPECIFIED
Es wird keine Zugangsberechtigung vergeben und eine eventuell schon vorhandene Zugangsberechtigung ist nicht mehr gültig. Das Profil ist somit gesperrt.

TRANSFER-ADMISSION=*OLD-ADMISSION(...)
Die Zugangsberechtigung selbst bleibt unverändert. Die Optionen können jedoch im Gegensatz zur Angabe TRANSFER-ADMISSION=*UNCHANGED geändert werden. Die Beschreibung der Werte in der Klammer (VALID=, USAGE= und EXPIRATION-DATE=) finden Sie in den nächsten Absätzen.

TRANSFER-ADMISSION=<alphanum-name 8..32>(...) /
<c-string 8..32 with-low>(...) / <x-string 15..64>(...)
Die Zeichenkette muss im FTP-Auftrag als Zugangsberechtigung angegeben werden. Die alphanumerische Eingabe wird intern immer in Kleinbuchstaben abgelegt.

VALID=*UNCHANGED
Der Wert bleibt unverändert.

VALID=*YES
Die Zugangsberechtigung ist gültig.

VALID=*NO
Die Zugangsberechtigung ist nicht gültig. Mit dieser Angabe kann das Profil gesperrt werden.

USAGE=*UNCHANGED
Der Wert bleibt unverändert.

USAGE=*PRIVATE
Ihr Profil wird aus Sicherheitsgründen gesperrt, sobald unter einer fremden Kennung versucht wird, die von Ihnen bereits verwendete TRANSFER-ADMISSION ein zweites Mal zu vergeben.

USAGE=*PUBLIC
Ihr Profil wird auch dann nicht gesperrt, wenn Ihre TRANSFER-ADMISSION durch einen anderen Nutzer „entdeckt“ wurde. „Entdeckt“ bedeutet, dass von einer zweiten Benutzerkennung versucht wurde dieselbe TRANSFER-ADMISSSION nochmals zu vergeben. Dies wird aus Sicherheitsgründen abgelehnt.

EXPIRATION-DATE=*UNCHANGED
Der Wert bleibt unverändert.

EXPIRATION-DATE=*NOT-RESTRICTED
Die Verwendung der Zugangsberechtigung ist zeitlich nicht eingeschränkt.

EXPIRATION-DATE=<date 8..10>
Die Verwendung der Zugangsberechtigung ist nur bis zu dem angegebenen Datum (max. 31.12. 2020) möglich. Die Angabe muss in der Form JJJJ-MM-TT bzw. JJ-MM-TT erfolgen.

TRANSFER-ADMISSION=*SECRET
Sie erhalten vom System die Aufforderung, die Zugangsberechtigung einzugeben, diese wird aber nicht auf dem Bildschirm sichtbar. Die Operanden VALID, USAGE und EXPIRATION-DATE können in diesem Fall ebenfalls verdeckt eingegeben werden.

PRIVILEGED=
Mit PRIVILEGED kann der FTAC-Verwalter ein Berechtigungsprofil eines beliebigen FTAC-Benutzers privilegieren. FTP-Aufträge, die mit einem privilegierten Berechtigungsprofil abgewickelt werden, unterliegen nicht den Einschränkungen, die für MAX-ADM-LEVEL im Berechtigungssatz eingestellt sind.
Der FTAC-Benutzer kann nur eine eventuell vergebene Privilegierung wieder zurücknehmen.

PRIVILEGED=*UNCHANGED
Der Status dieses Berechtigungsprofils bleibt unverändert.

PRIVILEGED=*NO
Mit *NO können Sie eine Privilegierung zurücknehmen.

IGNORE-MAX-LEVELS=
Mit IGNORE-MAX-LEVELS kann festgelegt werden, für welche der sechs Grundfunktionen die Einschränkungen des Berechtigungssatzes außer Kraft gesetzt werden sollen. Die MAX-ADM-LEVELS im Berechtigungssatz können nur mit einem vom FTAC-Verwalter privilegierten Berechtigungsprofil wirksam überschritten werden. Der FTAC-Benutzer kann sich für spezielle Aufgaben (z.B. Senden einer bestimmten Datei in ein Partnersystem, mit dem er normalerweise keinen FTP betreiben darf) ein Berechtigungsprofil anlegen, in dem das Überschreiten des Berechtigungssatzes vorgesehen ist. Dieses Profil muss anschließend vom FTAC-Verwalter privilegiert werden.
Wenn Sie IGNORE-MAX-LEVELS=*YES angeben, werden die Vorgaben für alle Grundfunktionen außer Kraft gesetzt. Wenn Sie nur für eine Grundfunktion den Berechtigungssatz ingnorieren wollen, müssen Sie mit den weiter unten folgenden Operanden arbeiten.

IGNORE-MAX-LEVELS=*UNCHANGED
Mit dem Berechtigungsprofil können Sie nach der Änderung dieselben Sicherheitsstufen erreichen wie vorher auch (es sei denn, Sie haben eine Privilegierung des Profils mit PRIVILEGED=*NO zurückgenommen).

IGNORE-MAX-LEVELS=*NO
FTP-Aufträge, die mit diesem Berechtigungsprofil abgewickelt werden, unterliegen den Einschränkungen des Berechtigungssatzes.

IGNORE-MAX-LEVELS=*YES
*YES berechtigt dazu, mit Partnersystemen zu kommunizieren, deren Sicherheitsstufe die Angaben im Berechtigungssatz überschreitet. Solange Ihr Profil nicht privilegiert ist, können Sie sich nur über die MAX-USER-LEVELS im Berechtigungssatz hinwegsetzen, nicht aber über die MAX-ADM-LEVELS.
Angaben zur aktuellen Einstellung der MAX-USER-LEVELS und MAX-ADM-LEVELS erhalten Sie mit dem Kommando SHOW-FT-ADMISSION-SET (siehe Beispiel auf Seite 268).

IGNORE-MAX-LEVELS=*PARAMETERS(...)

INBOUND-SEND=*UNCHANGED
Die mit der Grundfunktion „inbound senden“ maximal erreichbare Sicherheitsstufe bleibt unverändert.

INBOUND-SEND=*NO
Die mit der Grundfunktion „inbound senden“ maximal erreichbare Sicherheitsstufe wird durch den Berechtigungssatz festgelegt.

INBOUND-SEND=*YES
Für die Grundfunktion „inbound senden“ können Sie sich mit diesem Berechtigungsprofil über die MAX-USER-LEVELS hinwegsetzen. Wenn Ihr Profil privilegiert ist, sind Sie auch nicht an die Beschränkungen der MAX-ADM-LEVELS gebunden. Außerdem kann zusätzlich die Teilkomponente „Anzeigen von Dateiattributen“ der Grundfunktion „inbound Dateimanagement“ genutzt werden.

INBOUND-RECEIVE=*UNCHANGED
Die mit der Grundfunktion „inbound empfangen“ maximal erreichbare Sicherheitsstufe bleibt unverändert.

INBOUND-RECEIVE=*NO
Die mit der Grundfunktion „inbound empfangen“ maximal erreichbare Sicherheitsstufe wird durch Ihren Berechtigungssatz festgelegt.

INBOUND-RECEIVE=*YES
Sie können sich mit diesem Profil über Ihre Vorgabe für „inbound empfangen“ in den MAX-USER-LEVELS hinwegsetzen. Wenn Ihr Profil privilegiert ist, sind Sie auch nicht an die Beschränkungen der MAX-ADM-LEVELS gebunden. Außerdem können zusätzlich folgende Teilkomponenten der Grundfunktion „inbound Dateimanagement“ genutzt werden:
– Löschen von Dateien, sofern die Dateiattribute entsprechend gesetzt sind,
– Ändern von Dateiattributen, wenn die Grundfunktion „inbound Dateimanagement“ INBOUND-MANAGEMENT=*UNCHANGED
Die mit der Grundfunktion „inbound Dateimanagement“ maximal erreichbare Sicherheitsstufe bleibt unverändert.

INBOUND-MANAGEMENT=*NO
Die mit der Grundfunktion „inbound Dateimanagement“ maximal erreichbare Sicherheitsstufe wird durch den Berechtigungssatz festgelegt.

INBOUND-MANAGEMENT=*YES
Für die Grundfunktion „inbound Dateimanagement“ können Sie sich mit diesem Berechtigungsprofil über die MAX-USER-LEVELS hinwegsetzen. Wenn Ihr Profil privilegiert ist, sind Sie auch nicht an die Beschränkungen der MAX-ADM-LEVELS gebunden. Die zur Grundfunktion „inbound Dateimanagement“ gehörende Teilkomponente „Ändern von Dateiattributen“ funktioniert nur dann, wenn im Berechtigungssatz oder im Berechtigungsprofil die Grundfunktion „inbound empfangen“ zugelassen wurde.

USER-ADMISSION=
Mit USER-ADMISSION geben Sie als Benutzer die Kennung an, unter der das geänderte Profil abgespeichert wird. FTP-Aufträge, die mit diesem Berechtigungsprofil arbeiten, greifen im lokalen System auf die angegebene Kennung zu.
Wenn der FTAC-Verwalter ein Berechtigungsprofil für einen Benutzer angelegt hat, können die Angaben ACCOUNT und PASSWORD im Operanden USER-ADMISSION vom Benutzer mit MODIFY-FT-PROFILE eingetragen werden (da sie auch nur ihm bekannt sein sollten), bevor das Profil tatsächlich verwendet werden kann (siehe das Kommando CREATE-FT-PROFILE, Seite 236).

USER-ADMISSION=*UNCHANGED
Die USER-ADMISSION dieses Berechtigungsprofils bleibt unverändert.

USER-ADMISSION=*OWN
Für USER-IDENTIFICATION und ACCOUNT werden die Angaben für Ihre Benutzerkennung und Ihre Abrechnungsnummer aus Ihrer LOGON-Berechtigung übernommen. Ein etwaiges BS2000-Kennwort wird erst zu dem Zeitpunkt aus Ihrer LOGON-Berechtigung übernommen, zu dem ein FTP-Auftrag auf das Berechtigungsprofil zugreift.

USER-ADMISSION=*PARAMETERS(...)
Sie können die Benutzerkennung auch in ihren einzelnen Bestandteilen angeben. Damit können Sie beispielsweise erreichen, dass FTP-Aufträge, die mit diesem Berechtigungsprofil arbeiten, unter einer anderen Abrechnungsnummer abgerechnet werden. Ein anderer

Einsatzfall ist das Vorgeben eines Kennwortes im Berechtigungsprofil. FTP-Aufträge, die mit diesem Berechtigungsprofil arbeiten, funktionieren nur dann, wenn Ihr aktuelles LOGON-Kennwort mit diesem vorgegebenen Kennwort übereinstimmt.

USER-IDENTIFICATION=
USER-IDENTIFICATION identifiziert Ihre Benutzerkennung im BS2000.

USER-IDENTIFICATION=*OWN
Ihre Benutzerkennung wird aus der LOGON-Berechtigung übernommen.

USER-IDENTIFICATION=<name 1..8>
<name 1..8> ist die Benutzerkennung, der das Profil gehören soll.

ACCOUNT=
Mit ACCOUNT geben Sie die Abrechnungsnummer an, unter der ein FTP-Auftrag abgerechnet werden soll, wenn er mit diesem Berechtigungsprofil arbeitet.

ACCOUNT=*OWN
Die Abrechnungsnummer wird aus Ihrer LOGON-Berechtigung übernommen.

ACCOUNT=*NOT-SPECIFIED
Die Abrechnungsnummer wird erst vom Eigentümer des Berechtigungsprofils angegeben. Diese Funktion ermöglicht dem FTAC-Verwalter das Einrichten von Profilen für Benutzerkennungen, deren Abrechnungsnummer er nicht kennt.

ACCOUNT=<alphanum-name 1..8>
Unter der angegebenen Abrechnungsnummer soll ein FTP-Auftrag abgerechnet werden, wenn er auf dieses Berechtigungsprofil zugreift. Sie können jede zur Benutzerkennung gehörende Abrechnungsnummer angeben.

PASSWORD=
Mit PASSWORD geben Sie ein zu Ihrer Benutzerkennung gehörendes BS2000-Kennwort an.

PASSWORD=*OWN
Wenn ein FTP-Auftrag auf dieses Berechtigungsprofil Bezug nimmt, setzt FTAC das zu diesem Zeitpunkt gültige BS2000-Kennwort Ihrer Benutzerkennung ein. Damit wird verhindert, dass bei einer etwaigen Änderung des BS2000-Kennwortes auch das Berechtigungsprofil geändert werden muss.

PASSWORD=*NOT-SPECIFIED
Das Kennwort wird erst vom Eigentümer des Berechtigungsprofils angegeben. Diese Funktion ermöglicht dem FTAC-Verwalter das Einrichten von Profilen für fremde Benutzerkennungen.

PASSWORD=*NONE
Für die Benutzerkennung wird kein BS2000-Kennwort benötigt.

PASSWORD=<c-string 1..8> / <x-string 1..16>
Das angegebene Kennwort wird in dem Moment mit dem aktuellen LOGON-Kennwort verglichen, in dem ein FTP-Auftrag auf das Berechtigungsprofil zugreift. Widersprechen sich die Angaben, wird der FTP-Auftrag abgelehnt.

PASSWORD=*SECRET
Sie erhalten vom System die Aufforderung, das Kennwort einzugeben. Die Eingabe erscheint dann nicht auf dem Bildschirm.

INITIATOR=
Mit INITIATOR legen Sie fest, dass Auftraggeber im fernen System dieses Berechtigungsprofil für ihre FTP-Aufträge benutzen dürfen.

INITIATOR=*UNCHANGED
Die Festlegungen in diesem Berechtigungsprofil bleiben unverändert.

INITIATOR=*REMOTE
FTP-Aufträge werden immer als *REMOTE behandelt. Daher müssen Berechtigungsprofile für FTP mindestens die Einstellung *REMOTE aufweisen. Wird dasselbe Profil auch für openFT verwendet, ist beispielsweise auch die Einstellung (*LOCAL,*REMOTE) zulässig.

TRANSFER-DIRECTION=
Mit TRANSFER-DIRECTION legen Sie fest, welche Übertragungsrichtung mit diesem Berechtigungsprofil benutzt werden darf. Die Übertragungsrichtung ist immer von dem BS2000-FTP-Server aus zu sehen, in dem Sie das Berechtigungsprofil definiert haben.

TRANSFER-DIRECTION=*UNCHANGED
Die Angabe im Berechtigungsprofil soll unverändert bleiben.

TRANSFER-DIRECTION=*NOT-RESTRICTED
Mit diesem Berechtigungsprofil dürfen sowohl Daten vom Client zum Server als auch vom Server zum Client übertragen werden.

TRANSFER-DIRECTION=*FROM-PARTNER
Mit diesem Berechtigungsprofil dürfen nur Daten von einem Client zum Server übertragen werden. Damit ist auch kein Anzeigen von Dateiattributen bzw. Dateiverzeichnissen (Teilkomponenten des „inbound Dateimanagements“) möglich, d.h. folgende Server-Kommandos sind nicht zugelassen: *cdup*, *xcup*, *cwd*, *xcwd*, *list*, *nlst*, *pwd*, *xpwd*, *retr*, *size*, *mdtm*

TRANSFER-DIRECTION=*TO-PARTNER
Mit diesem Berechtigungsprofil dürfen nur Daten vom Server zu einem Client-System übertragen werden. Damit ist auch kein Modifizieren von Dateiattributen und kein Löschen von Dateien (Teilkomponenten des „inbound Dateimanagements“) möglich, d.h. folgende Server-Kommandos sind nicht zugelassen: *appe*, *dele*, *site file*, *mkd*, *xmkd*, *rmd*, *xrmd*, *rnfr*, *stor, stou*.

PARTNER=
Mit PARTNER können Sie festlegen, dass dieses Berechtigungsprofil nur für FTP-Aufträge benutzt werden kann, die mit einem bestimmten Client-System abgewickelt werden.

PARTNER=*UNCHANGED
Ein eventueller PARTNER im Berechtigungsprofil soll unverändert bleiben.

PARTNER=*NOT-RESTRICTED
Der Einsatzbereich dieses Berechtigungsprofils ist nicht auf FTP-Aufträge mit bestimmten Partnersystemen eingeschränkt.

PARTNER=list-poss(50): <text 1..200 with-low>
Das Berechtigungsprofil lässt nur solche FTP-Aufträge zu, die mit den angegebenen Client-Systemen abgewickelt werden. Maximal 50 Client-Systeme können angegeben werden. Sie können bei PARTNER den Namen aus der Partnerliste oder die Adresse des Partnersystems angeben, siehe auch „openFT für BS2000/OSD - Benutzerhandbuch“. Es wird empfohlen, den Namen aus der Partnerliste zu verwenden.

PARTNER=*ADD(list-poss(50): <text 1..200 with-low>
Mit dieser Angabe können Sie einer vorhandenen Menge von Client-Systemen weitere Elemente hinzufügen. Maximal 50 Client-Systeme können angegeben werden.

PARTNER=*REMOVE(list-poss(50): <text 1..200 with-low>)
Mit dieser Angabe können Sie Elemente aus einer vorhandenen Liste von Client-Systemen löschen. Maximal 50 Client-Systeme können angegeben werden.

MAX-PARTNER-LEVEL=
Mit MAX-PARTNER-LEVEL kann eine maximale Sicherheitsstufe angegeben werden. Bei FTP-Aufträgen wird dem Client-System eine von der System-Administration vorgegebene Sicherheitsstufe bzw. standardmäßig die Sicherheitsstufe 100 zugeordnet.
MAX-PARTNER-LEVEL wirkt im Zusammenhang mit den Werten im Berechtigungssatz. Bei Verwendung nicht privilegierter Berechtigungsprofile wird die Zugangsprüfung auf der Basis des kleinsten vorgegebenen Wertes durchgeführt.

MAX-PARTNER-LEVEL=*UNCHANGED
Die Angabe zu MAX-PARTNER-LEVEL in diesem Berechtigungsprofil soll unverändert bleiben.

MAX-PARTNER-LEVEL=*NOT-RESTRICTED
Werden FTP-Aufträge mit diesem Berechtigungsprofil abgewickelt, so wird die maximal erreichbare Sicherheitsstufe durch den Berechtigungssatz festgelegt.

MAX-PARTNER-LEVEL=<integer 0..100>
Wenn Sie MAX-PARTNER-LEVEL kleiner als die von der System-Administration vorgegebene Sicherheitsstufe bzw. kleiner als den Standard-Wert 100 setzen, sperren Sie dadurch das Berechtigungsprofil (vorübergehend) für FTP-Aufträge.

FILE-NAME=
Mit FILE-NAME legen Sie fest, auf welche Dateien unter Ihrer Kennung FTP-Aufträge zugreifen dürfen, die mit diesem Berechtigungsprofil arbeiten.

FILE-NAME=*UNCHANGED
Die Angaben für FILE-NAME in diesem Berechtigungsprofil werden nicht geändert.

FILE-NAME=*NOT-RESTRICTED
Das Berechtigungsprofil erlaubt uneingeschränkten Zugriff auf alle Dateien der Benutzerkennung.

FILE-NAME =*EXPANSION(PREFIX = <filename 1..53> /
<partial-filename 2..53> / <c-string 1..511 with-low>)
Der Zugriff kann auf eine Menge von Dateien beschränkt werden, die alle mit demselben Präfix beginnen. Wenn in einem FT-Auftrag, der mit diesem Berechtigungsprofil arbeitet, ein *Dateiname* angegeben wird, stellt FTAC vor diesen Dateinamen das mit EXPANSION definierte *Präfix*. Der FT-Auftrag wird dann auf die Datei *PräfixDateiname* zugreifen.
Durch diese Angabe kann der Zugriff auf eine Menge von Dateien beschränkt werden, die alle mit demselben Präfix beginnen. Wenn in einem FTP-Auftrag, der mit diesem Berechtigungsprofil arbeitet, ein Dateiname angegeben wird, stellt FTAC vor diesen Dateinamen abhängig vom aktuell eingestellten Arbeitsverzeichnis das mit EXPANSION definierte Präfix. Der FTP-Auftrag darf dann auf die Datei *PräfixDateiname* zugreifen.

Ein Wechsel zwischen POSIX- und DVS-Dateisystem ist nicht möglich. Wenn das Präfix ein „/“ enthält oder mit „.“ beginnt, kann nur auf das POSIX-Dateisystem zugegriffen werden. In den übrigen Fällen ist der Zugriff nur auf das DVS-Dateisystem möglich.

Beispiel

PREFIX=DAGOBERT.; ein FTP-Auftrag, in dem als Dateiname BOERSE angegeben wurde, greift dann auf die Datei DAGOBERT.BOERSE zu.

FILE-PASSWORD=
Mit FILE-PASSWORD können Sie ein Kennwort für Dateien in das Berechtigungsprofil eintragen. FTAC erlaubt dann nur Zugriffe auf Dateien, die mit diesem Kennwort geschützt sind sowie auf nicht geschützte Dateien. Wenn in einem Berechtigungsprofil ein FILE-PASSWORD steht, darf das Kennwort nicht mehr in einem FTP-Auftrag angegeben werden, der mit diesem Berechtigungsprofil arbeitet. Sie können somit Anwendern in fernen Systemen einen Zugriff auf bestimmte Dateien erlauben, ohne deren Dateikennworte verraten zu müssen.

FILE-PASSWORD=*UNCHANGED
Die Angaben zum FILE-PASSWORD sollen in diesem Berechtigungsprofil unverändert bleiben.

FILE-PASSWORD=*NOT-RESTRICTED
Das Berechtigungsprofil erlaubt Zugriffe auf alle Dateien; ist für eine Datei ein Kennwort vergeben, muss dieses im FTP-Auftrag angegeben werden.

FILE-PASSWORD=*NONE
Das Berechtigungsprofil erlaubt nur Zugriffe auf Dateien ohne Dateikennwort.

FILE-PASSWORD=<c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647>
Das Berechtigungsprofil erlaubt nur Zugriffe auf Dateien, die mit dem angegebenen Dateikennwort geschützt sind, sowie auf nicht geschützte Dateien. Im FTP-Auftrag darf das Kennwort, das bereits im Profil angegeben ist, nicht wiederholt werden.

FILE-PASSWORD=*SECRET
Sie erhalten vom System die Aufforderung, das Kennwort einzugeben. Die Eingabe erscheint dann nicht auf dem Bildschirm.

WRITE-MODE=
Mit WRITE-MODE legen Sie die für diesen FTP-Auftrag zulässige Schreibregel fest. WRITE-MODE wirkt nur, wenn die Empfangsdatei im selben System liegt, in dem auch dieses Berechtigungsprofil definiert ist. In FTP-Kommandos wird die Schreibregel nicht explizit angegeben, sondern ist impliziter Bestandteil des FTP-Kommandos:

appe	*EXTEND-FILE
stor, *rnfr*, *site file*, *dele*, *rmd*, *xrmd*	*REPLACE-FILE
stou	*NEW-FILE

WRITE-MODE=*UNCHANGED
Die Angaben zu WRITE-MODE sollen in diesem Berechtigungsprofil unverändert bleiben.

WRITE-MODE=*NOT-RESTRICTED
In einem FTP-Auftrag, der auf dieses Berechtigungsprofil zugreift, sind alle FTP-Schreibkommandos zugelassen.

WRITE-MODE=*NEW-FILE
Die FTP-Kommandos *dele*, *rmd* und *xrmd* sind nicht zulässig.

WRITE-MODE=*REPLACE-FILE
Das FTP-Kommandos *stou* ist nicht zulässig.

WRITE-MODE=*EXTEND-FILE
Die FTP-Kommandos *stor*, *stou*, *dele*, *rmd* und *xrmd* sind nicht zulässig.

FT-FUNCTION=
Dieser Operand ermöglicht die Beschränkung der Gültigkeit des Profils auf bestimmte FTP-Funktionen (= Dateiübertragungs- und Dateimanagement-Funktionen).

FT-FUNCTION=*UNCHANGED
Die Dateimanagement-Funktionen bleiben im bisherigen Umfang erhalten.

FT-FUNCTION=*NOT-RESTRICTED
Die FTP-Funktionen stehen im vollen Umfang zur Verfügung.

FT-FUNCTION=(*TRANSFER-FILE, *MODIFY-FILE-ATTRIBUTES,
***READ-DIRECTORY, *FILE-PROCESSING)**
Folgende Funktionen stehen zur Verfügung:

***TRANSFER-FILE**
Das Berechtigungsprofil darf für die Funktionen „Dateien übertragen“, „Dateiattribute ansehen“ und „Dateien löschen“ benutzt werden.
Folgende Server-Kommandos sind nicht zugelassen:

list, *nlist*, *pwd*, *xpwd*, *cwd*, *xcwd*, *cdup*, *xcup*, *rnfr*, *size*, *mdtm*

***MODIFY-FILE-ATTRIBUTES**
Das Berechtigungsprofil darf für die Funktionen „Dateiattribute ansehen“ und „Dateiattribute modifizieren“ benutzt werden.
Folgende Server-Kommandos sind nicht zugelassen:

retr, *stor*, *appe*, *stou*, *dele*, *list*, *nlist*, *pwd*, *xpwd*, *cwd*, *xcwd*, *cdup*, *xcup*, *size*, *mdtm*

***READ-DIRECTORY**
Das Berechtigungsprofil darf für die Funktionen „Dateiverzeichnisse ansehen“ und „Dateiattribute ansehen“ benutzt werden.
Folgende Server-Kommandos sind nicht zugelassen:

retr, *stor*, *appe*, *stou*, *dele*, *rnfr*.

***FILE-PROCESSING**
Das Berechtigungsprofil darf für die File-Transfer-Funktionen „Vorverarbeitung“ und „Nachverarbeitung“ benutzt werden. Zusätzlich muss auch die Funktion „Dateien übertragen“ erlaubt sein. Die Angabe von *FILE-PROCESSING spielt nur bei FTAC-Profilen ohne Dateinamens-Präfix eine Rolle. Ansonsten entscheidet das erste Zeichen des Dateinamens-Präfix darüber, ob mit diesem FTAC-Profil nur normale Dateiübertragungen (kein Pipe-Zeichen |) oder nur Vorverarbeitungen und Nachverarbeitungen (Pipe-Zeichen |) möglich sind.

Beispiel

Nachdem Donald Duck ein Berechtigungsprofil mit dem Namen *profil1* eingerichtet hat, das anderen Benutzern den Zugriff auf seine Kennung ohne Kenntnis der LOGON-Berechtigung ermöglicht, möchte er nunmehr dieses Profil soweit einschränken, dass nur mehr FTP-Zugriffe auf Dateien möglich sind, die mit dem Präfix *FILIALE.* beginnen. Dazu setzt er folgendes Kommando ab:

```
→ /MODIFY-FT-PROFILE␣NAME=profil1,FILE-NAME=*EXPANSION(PREFIX=filiale.)
```

Eine mögliche Kurzform des Kommandos lautet:

```
→ /MOD-FT-PROF␣profil1,FILE-N=(PRE=filiale.)
```

Damit ist das Berechtigungsprofil nun stark eingeschränkt. Die übrigen Angaben darin bleiben unverändert.

Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	0	FTC0051	Eine Userid mit demselben Namen existiert bereits im System.
0	64	FTC0053	Es existiert kein FT-Profil zu den angegebenen Kriterien.
0	64	FTC0055	Die Partnereinschraenkung wurde aufgehoben.
0	0	FTC0056	Die Zugangsberechtigung ist gesperrt.
0	64	FTC0100	Es gibt bereits ein FT-Profil mit dem angegebenen Namen.
0	64	FTC0101	Es gibt bereits ein FT-Profil mit der angegebenen Transfer-Admission.
0	64	FTC0150	Das Kennwort zur Berechtigung fehlt.
0	64	FTC0151	Die Aenderung ist dem Verwalter oder dem Eigentuemer vorbehalten.
0	64	FTC0153	Die angegebene Eigentuemeridentifikation ist nicht die eigene Benutzerkennung.
0	64	FTC0170	Der angegebene Partner ist unbekannt innerhalb der fuer diesen Benutzer moeglichen Partnersysteme.
0	64	FTC0171	Das angegebene Profil existiert nicht.
0	64	FTC0172	Die angegebene User-Admission existiert nicht im System.
0	64	FTC0173	Die angegebene Processing-Admission existiert nicht im System.
0	64	FTC0174	Die Parameter „NEW-NAME“ und „TRANSFER-ADMISSION“ duerfen nur zusammen mit eindeutigen Auswahlkriterien ("NAME" oder „TRANSFER-ADMISSION“) verwendet werden.
0	64	FTC0178	Der angegebene Partnername kommt mehrmals vor.
0	64	FTC0179	Verstoss gegen die maximale Anzahl von Partnereinschraenkungen.
0	64	FTC0182	Verstoß gegen die maximale Länge für Partnernamen.
0	64	FTC0200	Die Summe der Laengen der beiden Folgeverarbeitungskommandos ist zu gross.
0	64	FTC0255	Ein Systemfehler ist aufgetreten.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im openFT-Handbuch „Meldungenn.

5.2.8 SHOW-FT-ADMISSION-SET - Berechtigungssätze anzeigen

Mit dem Kommando SHOW-FT-ADMISSION-SET können Sie sich Berechtigungssätze anschauen. Sie erhalten folgende Informationen wahlweise auf SYSOUT oder SYSLST ausgegeben:

– ob der Berechtigungssatz privilegiert ist (wenn ja, sind Sie FTAC-Verwalter),

– ob auf dieser Benutzerkennung ein Kennwort zur Abgabe von FTAC-Kommandos nötig ist. Das Kennwort selbst wird nicht angezeigt,

– welche Grenzwerte der Eigentümer dieser Kennung für die erreichbaren Sicherheitszonen eingestellt hat,

– welche Grenzwerte der FTAC-Verwalter für die erreichbaren Sicherheitszonen vorgegeben hat.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

SHOW-**FT-ADM**ISSION-**S**ET - Darstellung der FTP-relevanten Operanden
USER-IDENTIFICATION = ***OWN** / ***ALL** / ***STD** / <alphanum-name 1...8> ,**OUT**PUT = ***SYSOUT (LAYOUT = *STD** / ***CSV)** / ***SYSLST (LAYOUT = *STD** / ***CSV)**

Beschreibung der Operanden

USER-IDENTIFICATION=
Benutzerkennung, deren Berechtigungssatz Sie sich anschauen wollen. FTAC-Benutzer können sich nur über ihren Berechtigungssatz und den Standardberechtigungssatz informieren.

USER-IDENTIFICATION=*OWN
FTAC gibt den Berechtigungssatz aus, der zu Ihrer Benutzerkennung gehört.

USER-IDENTIFICATION=*ALL
FTAC gibt den Standardberechtigungsatz und den Berechtigungssatz aus, der zu Ihrer Benutzerkennung gehört.

USER-IDENTIFICATION=*STD
FTAC gibt nur den Standardberechtigungssatz aus.

USER-IDENTIFICATION=<alphanum-name 1..8>
FTAC gibt den Berechtigungssatz aus, der zu dieser Benutzerkennung gehört. Der Operand steht für die USER-ID der angegebenen Kennung. Der FTAC-Benutzer kann hier nur seine eigene Kennung angeben.

OUTPUT=
bestimmt das Ausgabemedium für die angeforderte Information.

OUTPUT=*SYSOUT(...)
Die Ausgabe erfolgt auf SYSOUT.

OUTPUT=*SYSLST(...)
Die Ausgabe erfolgt auf SYSLST.

LAYOUT=*STD
Die Ausgabe wird in eine vom Anwender leicht lesbare Form gebracht.

LAYOUT=*CSV
Die Ausgabe erfolgt im Comma Separated Value Format. Dies ist ein speziell im PC-Umfeld weit verbreitetes, tabellenartiges Format, bei dem die einzelnen Felder durch das Separatorenzeichen Semikolon ";" getrennt sind.

Beispiel

Dagobert Duck, der FTAC-Verwalter des Bankhauses Duck, will sich über die Berechtigungssätze in seinem System informieren. Er gibt das Kommando

```
→ /SHOW-FT-ADMISSION-SET␣USER-IDENTIFICATION=*ALL
```

Kurzform:

```
→ /SHOW-FT-AD␣*ALL
```

ein und erhält folgende Ausgabe:

```
%                MAX. USER LEVELS              MAX. ADM LEVELS        ATTR
% USER-ID  OBS  OBR  IBS  IBR  IBP  IBF  OBS  OBR  IBS  IBR  IBP  IBF
% *STD      10   10  100  100    0    0   10   10  100  100    0    0
% DAGOBERT 100  100    0   99    0*   0* 100  100    0   99    0*   0* PRIV
% DAISY     50   50   99  100   50   50   50   50  100  100   50   50  PW
% DANIEL     0   10   99   99    0    0   10   10  100  100    0    0  PW
% DONALD    50  100   99  100*   0    0   50  100  100  100    0    0
```

Die Angaben haben folgende Bedeutung:

In der Spalte USER-ID steht die Benutzerkennung, zu der der jeweilige Berechtigungssatz gehört. Es gibt in diesem Beispiel außer dem Standardberechtigungssatz noch Berechtigungssätze für die Kennungen DAGOBERT, DAISY, DANIEL und DONALD.

In der Spalte ATTR wird der privilegierte Berechtigungssatz gekennzeichnet.
Also ist DAGOBERT der FTAC-Verwalter.

Ebenfalls wird in ATTR angezeigt (mit PW), ob auf der Kennung ein FTAC-Kennwort definiert wurde. DAGOBERT, DAISY und DANIEL verhindern auf diese Weise, dass ein anderer auf ihrer Kennnung FTAC-Kommandos abgibt, die Änderungen hervorrufen.

In den sechs Spalten im Bereich MAX-USER-LEVELS sind die Grenzwerte angegeben, die die FTAC-Benutzer für ihren Berechtigugnssatz festgelegt haben. Entsprechend enthalten die sechs Spalten im Bereich MAX-ADM-LEVELS die Grenzwerte, die der FTAC-Verwalter engestellt hat. Der kleinere der Grenzwerte gibt an, bis zu welcher Sicherheitsstufe der Eigentümer des Berechtigungssates die jeweilige Grundfunktion nutzen darf. Die für den FTP-Nutzer relevanten Grundfunktionen sind in der Ausgabe folgendermaßen abgekürzt:

IBS = **I**N**B**OUND-**S**END

IBR = **I**N**B**OUND-**R**ECEIVE

IBP = **I**N**B**OUND-**P**ROCESSING

IBF = **I**N**B**OUND-**F**ILEMANAGEMENT

Zunächst ist zu beachten, dass FTP-Nutzer (unabhängig von der ihrem System zugeordneten Sicherheitsstufe) eine von der System-Administration vorgegebene Sicherheitsstufe bzw. standardmäßig die Sicherheitsstufe 100 zugeordnet bekommen. In diesem Beispiel gehen wir dabei von der Sicherheitsstufe 100 aus.

Der Standardberechtigungssatz ist also so eingerichtet, dass er für FTP-Nutzer das Senden von Dateien zum FTP-Server und das Holen von Dateien vom FTP-Server zulässt, aber keine Dateimanagement-Aktionen.

DAGOBERT lässt keine FTP-Zugriffe von außen auf seine Kennung zu (IBS=0, IBR=99, IBP=0).

Die Dateien der Kennung DAISY dürfen nicht per FTP gelesen werden (IBS=99), aber es dürfen Dateien auf diese Kennung transferiert werden (IBR=100). Auf die Kennung DANIEL darf nicht mit FTP zugegriffen werden (IBS, IBR und IBF sind kleiner als 100).

Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	64	FTC0052	Die Ausgabe der Information wurde unterbrochen.
0	64	FTC0152	Die angegebene Benutzerkennung ist nicht die eigene Benutzerkennung.
0	64	FTC0181	Der angegebene FT-Profilname kommt mehrmals vor.
0	64	FTC0255	Ein Systemfehler ist aufgetreten.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im openFT-Handbuch „Meldungen.

5.2.9 SHOW-FT-LOGGING-RECORDS - Logging-Sätze anzeigen

Bei Nutzung der FTAC-Funktionalität können Sie sich mit SHOW-FT-LOGGING-RECORDS auch die FTAC-Logging-Sätze Ihrer Benutzerkennung ansehen. Auch FT-Verwalter können sich nur die FTAC-Logging-Sätze ansehen, die sich auf ihre eigene Benutzerkennung beziehen. Der FTAC-Verwalter ist der einzige, der sich alle FTAC-Logging-Sätze im System ansehen kann.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

SHOW-FT-LOGGING-**REC**ORDS - Darstellung der FTP-relevanten Operanden

```
SELECT = *OWN / *ALL / *PARAMETERS(...)

  *PARAMETERS(...)
    LOGGING-ID = *ALL / <alphanum-name 1..12> / *INTERVAL(...)
      *INTERVAL(...)
        FROM = 1 / <alphanum-name 1..12>
        ,TO = *HIGHEST-EXISTING / <alphanum-name 1..12>
    ,OWNER-IDENTIFICATION = *OWN / *ALL / <name 1..8>
    ,CREATION-TIME = *INTERVAL (...) / *DAYS(...)
      *INTERVAL(...)
        FROM = 1970-01-01 (...) / <date 8..10> (...)
          (...)
            TIME = 00:00 / <time 1..8>
        ,TO = *TOMORROW (...) / *TODAY (...) / <date 8..10> (...)
          (...)
            TIME = 00:00 / <time 1..8>
      *DAYS(...)
        NUMBER = <integer 1..1000>
    ,RECORD-TYPE = *ALL / *PARAMETERS(...)
      *PARAMETERS(...)
        ,FTAC = (*TRANSFER-FILE,*READ-FILE-ATTRIBUTES,*DELETE-FILE,
                 *CREATE-FILE,*MODIFY-FILE-ATTRIBUTES,
                 *READ-DIRECTORY,*MOVE-FILE,*CREATE-DIRECTORY,
                 *DELETE-DIRECTORY,*MODIFY-DIRECTORY. *LOGIN) / *NONE /
                 list-poss(11): *TRANSFER-FILE / *READ-FILE-ATTRIBUTES / *DELETE-FILE /
                 *CREATE-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY /
                 *MOVE-FILE / *CREATE-DIRECTORY / *DELETE-DIRECTORY /
                 *MODIFY-DIRECTORY / *LOGIN
```

```
        ,INITIATOR = (*LOCAL,*REMOTE) / *REMOTE
        ,PARTNER = *ALL / <text 1..200 with-low>
        ,FILE-NAME = *ALL / <filename 1..54> / <filename-prefix 2..53> /
                     <c-string 1..512 with-low> / *DIRECTORY(...) / *POSIX(NAME=<posix-pathname 1..510>)
          *DIRECTORY(...)
            | NAME = *ALL / <partial-filename 1..53> / <c-string 1..512 with-low>
        ,REASON-CODE = *ALL / *FAILURE / <text 1..4>
        ,TRANSFER-ID = *ALL / <integer 1.. 2147483639>
,NUMBER = 1 / *ALL / <integer 1..99999999>
,INFORMATION = *STD / *ALL
,OUTPUT = *SYSOUT (LAYOUT = *STD / *CSV ) /
          *SYSLST (LAYOUT = *STD / *CSV)
```

Beschreibung der Operanden

SELECT=
Dient zur Auswahl einer Gruppe von Logging-Sätzen.

SELECT=*OWN / *ALL
Wählt die Logging-Sätze der eigenen Kennung aus. Wenn kein weiteres Auswahlkriterium angegeben wird, werden Logging-Sätze der eigenen Kennung ausgegeben. Die Angaben *OWN und *ALL erzeugen für Nutzer die gleiche Ausgabe.

SELECT=*PARAMETERS(...)

LOGGING-ID=
Nummer des Logging-Satzes.

LOGGING-ID=*ALL
Die Nummer des Logging-Satzes ist nicht Auswahlkriterium.

LOGGING-ID=<alphanum-name 1..12>
Nummer des Logging-Satzes, der ausgegeben werden soll. Der Wertebereich für die Logging-Id erstreckt sich von 1 bis 999999999999.

LOGGING-ID=*INTERVAL(...)
Bereich der Logging-Sätze, die ausgegeben werden sollen.

FROM = <alphanum-name 1..12>
Erster Logging-Satz, der ausgegeben wird. Der Wertebereich für die Logging-Id erstreckt sich von 1 bis 999999999999.

TO = *HIGHEST-EXISTING / <alphanum-name 1..12>
Letzter Logging-Satz, der ausgegeben wird. Der Wertebereich für die Logging-Id erstreckt sich von 1 bis 999999999999.

OWNER-IDENTIFICATION=
Benutzerkennung, zu der Logging-Sätze ausgegeben werden sollen.

OWNER-IDENTIFICATION=*OWN
Es werden Logging-Sätze zu der eigenen Kennung ausgegeben.

OWNER-IDENTIFICATION=*ALL
Es werden Logging-Sätze aller Benutzerkennungen ausgegeben. Der FTAC-Verwalter kann sich die FTAC-Logging-Sätze beliebiger Benutzerkennungen ansehen.
Der FTP-Benutzer erhält auch mit dieser Angabe nur Informationen über Logging-Sätze, die sich auf seine Benutzerkennung beziehen.

OWNER-IDENTIFICATION=<name 1..8>
Beliebige Benutzerkennung, zu der Logging-Sätze ausgegeben werden sollen.
Der FT-Benutzer darf nur seine eigene Kennung angeben.

CREATION-TIME=*INTERVAL(...)
Über Erzeugungsdatum ausgewählter Bereich der Logging-Sätze, die ausgegeben werden sollen.

FROM=1970-01-01(...) / < date 8..10>(...)
Datum im Format *yyyy-mm-dd* oder *yy-mm-dd*, z.B. 2004-01-29 bzw. 04-01-29 für den 29.01.2004. FT gibt dann alle Logging-Sätze aus, die ab dem spezifizierten Zeitpunkt geschrieben wurden.

TIME=00:00 / <time 1..8>
Zeitpunkt für den mit CREATION-TIME spezifizierten Tag. FT gibt alle Logging-Sätze aus, die ab diesem Zeitpunkt geschrieben wurden. Den Zeitpunkt geben Sie im Format *hh:mm:ss* an, z.B. 14:30:10.

TO=*TOMORROW / *TODAY(...) / <date 8..10>(...)
Datum im Format *yyyy-mm-dd* oder *yy-mm-dd*, z.B. 2004-01-29 bzw. 04-01-29 für den 29.01.2004. FT gibt dann alle Logging-Sätze aus, die bis zu dem spezifizierten Zeitpunkt geschrieben wurden.

TIME=00:00 / <time 1..8>
Zeitpunkt für den mit CREATION-TIME spezifizierten Tag. FT gibt alle Logging-Sätze aus, die bis zu diesem Zeitpunkt geschrieben wurden. Den Zeitpunkt geben Sie im Format *hh:mm:ss* an, z.B. 14:30:10.

CREATION-TIME = *DAYS(NUMBER=<integer 1..1000>)
Der Bereich wird in Anzahl von Tagen angegeben. Es werden alle Logging-Sätze ausgegeben, die in den letzten n Kalendertagen einschließlich heute erzeugt wurden.

RECORD-TYPE=
Satztyp, zu dem die Logging-Sätze ausgegeben werden sollen.

RECORD-TYPE=*ALL
Der Satztyp ist nicht Auswahlkriterium.

RECORD-TYPE=*PARAMETERS(...)
Typ des Logging-Satzes.

FTAC=
(*TRANSFER-FILE, *READ-FILE-ATTRIBUTES, *DELETE-FILE, *CREATE-FILE, *MODIFY-FILE-ATTRIBUTES, *READ-DIRECTORY, *MOVE-FILE, *CREATE-DIRECTORY, *DELETE-DIRECTORY, *MODIFY-DIRECTORY / *LOGIN) / *NONE / list-poss(11): *TRANSFER-FILE / *READ-FILE-ATTRIBUTES / *DELETE-FILE / *CREATE-FILE / *MODIFY-FILE-ATTRIBUTES / *READ-DIRECTORY / *MOVE-FILE / *CREATE-DIRECTORY / *MODIFY-DIRECTORY / *DELETE-DIRECTORY / *LOGIN
Gibt an, ob FTAC-Logging-Sätze ausgegeben werden sollen oder nicht. Falls ja, kann zusätzlich differenziert werden, zu welcher FTP-Funktion FTAC-Logging-Sätze ausgegeben werden sollen. Hierbei bedeuten:

*TRANSFER-FILE
Es werden alle Logging-Sätze zur Funktion „Dateien übertragen“ ausgegeben. Das entspricht den Server-Kommandos *retr*, *stor*, *stou* und *appe*.

*READ-FILE-ATTRIBUTES
Alle Logging-Sätze zur Funktion „Dateiattribute lesen“ werden ausgegeben.

*DELETE-FILE
Es werden alle Logging-Sätze zur Funktion „Dateien löschen“ ausgegeben. Das entspricht dem Server-Kommando *dele*.

*CREATE-FILE
Es werden alle Logging-Sätze zur Funktion „Dateien anlegen“ ausgegeben. Das entspricht dem Server-Kommando *site file*.

*MODIFY-FILE-ATTRIBUTES
Es werden alle Logging-Sätze zur Funktion „Dateiattribute ändern“ ausgegeben. Das entspricht dem Server-Kommando *rnfr*.

*READ-DIRECTORY
Es werden alle Logging-Sätze zur Funktion „Dateiverzeichnisse lesen“ ausgegeben. Das entspricht den Server-Kommandos *cwd*, *xcwd*, *list*, *nlst*, *pwd*, *xpwd*, *cdup*, *xcup, size* und *mdtm*.

*MOVE-FILE
Alle Logging-Sätze zur Funktion „Dateien kopieren und anschließend löschen“ werden ausgegeben.

*CREATE-DIRECTORY
Es werden alle Logging-Sätze zur Funktion „Verzeichnis anlegen“ ausgegeben. Das entspricht den Server-Kommandos *mkd* und *xmkd*.

*DELETE-DIRECTORY
Es werden alle Logging-Sätze zur Funktion „Verzeichnis löschen“ ausgegeben. Das entspricht den Server-Kommandos *rmd* und *xrmd*.

*MODIFY-DIRECTORY
Alle Logging-Sätze zur Funktion „Verzeichnis ändern“ werden ausgegeben.

*LOGIN
Alle Logging-Sätze zur Funktion „Inbound FTP-Zugang“ werden ausgegeben. Logging-Sätze vom Typ *LOGIN werden nur im Falle einer falschen Zugangsberechtigung geschrieben.

INITIATOR=
Wählt die Logging-Sätze nach Initiator aus.

INITIATOR=(*LOCAL,*REMOTE)
Der Initiator ist nicht Auswahlkriterium.

INITIATOR=*REMOTE
Entspricht dem Standardwert, da FTP-Aufträge immer Remote-Aufträge sind.

PARTNER=
Client-System

PARTNER=*ALL
Das Client-System ist nicht Auswahlkriterium.

PARTNER= <text 1..200 with-low>
Client-System, zu dem Sie sich Logging-Sätze ausgeben lassen wollen. Näheres zur Adressangabe siehe openFT für BS2000/OSD Benutzerhandbuch.

FILE-NAME=
Dateiname

FILE-NAME=*ALL
Der Dateiname ist nicht Auswahlkriterium.

FILE-NAME=<full-filename 1..54> / <c-string 1..512 with-low>
***POSIX(NAME=posix-pathname_1..510)**
Vollqualifizierter Name der Datei, zu der Logging-Sätze ausgegeben werden.

FILE-NAME=<partial-filename 2..53>
Teilqualifizierter Name der Dateien, zu denen Sie sich Logging-Sätze ausgeben lassen wollen.

FILE-NAME = *DIRECTORY(...)
Name des Dateiverzeichnisses

***DIRECTORY(...)**
Die Angabe des Dateiverzeichnisses bezieht sich auf die entsprechende Angabe im Benutzerkommando SHOW-REMOTE-FILE-ATTRIBUTES (siehe openFT für BS2000/OSD - Benutzerhandbuch).

NAME=*ALL
Der Name des Dateiverzeichnisses ist nicht Auswahlkriterium.

NAME=<partial-filename 1..53> / <c-string 1..512 with-low>
Name des Dateiverzeichnisses. Dateiverzeichnisse werden im DVS durch teilqualifizierte Dateinamen dargestellt.

REASON-CODE=
Auswahl nach dem REASON-Code des Logging-Satzes.

REASON-CODE=*ALL
Der REASON-Code ist nicht Auswahlkriterium, es werden alle Sätze ausgegeben.

REASON-CODE=*FAILURE
Es werden alle fehlerhaften Sätze ausgegeben

REASON-CODE=<text 1..4>
Definiert die auszugebenden Sätze anhand der Fehlermeldungsnummer, wobei führende Nullen entfallen dürfen (z.B. 14 für 0014).

TRANSFER-ID =
Auswahl nach der Auftragsidentifikation.

TRANSFER-ID = *ALL
Die Auftragsidentifikation ist nicht Auswahlkriterium.

TRANSFER-ID = <integer 1..2147483639>
Gibt nur Logging-Sätze zu der angegebenen Auftragsidentifikation aus.

NUMBER=
Maximale Anzahl der auszugebenden Logging-Sätze.

NUMBER=1 / <integer 1..99999999>
Maximale Anzahl der Logging-Sätze, die angezeigt werden sollen. Standardmäßig wird ein Logging-Satz ausgegeben.

NUMBER=*ALL
Sie bekommen alle Ihre Logging-Sätze ausgegeben.

INFORMATION=
Umfang der gewünschten Information.

INFORMATION=*STD
Die Logging-Sätze werden in einem Standardumfang ausgegeben.

INFORMATION=*ALL
Die Logging-Sätze werden in ausführlicher Form ausgegeben.

OUTPUT=
Legt das Ausgabemedium fest. .

OUTPUT=*SYSOUT(...)
Die Ausgabe erfolgt auf SYSOUT.

OUTPUT=*SYSLST(...)
Die Ausgabe erfolgt auf SYSLST.

LAYOUT=*STD
Die Ausgabe wird in eine vom Anwender leicht lesbare Form gebracht.

LAYOUT=*CSV
Die Ausgabe erfolgt im Comma Separated Value Format. Dies ist ein speziell im PC Umfeld weit verbreitetes, tabellenartiges Format, bei dem die einzelnen Felder durch das Separatorenzeichen Semikolon ";" getrennt sind.

Beschreibung der Ausgabe

Kurze Ausgabeform eines FT-Logging-Satzes (Beispiel)

```
→ /SHOW-FT-LOGGING-RECORDS NUMBER=2
   %TYP LOGG-ID TIME    RC    PARTNER  INITIATOR INIT USER-ADM FILENAME
   %2010-06-22
   %PM    3283 18:26:59 0000 <G133H301 *REMOTE        FT2V292  TEST2
   %P     3212 11:33:53 0000 >G133H301 *REMOTE        FT2V292  TEST1
```

Lange Ausgabeform (Beispiel)

```
→ /SHOW-FT-LOGGING-RECORDS NUMBER=2,INFORMATION=*ALL
   %LOGGING-ID  = 00003283  RC     = 0000       TIME    = 2010-06-22 18:26:59
   %   INITIATOR= *REMOTE   PARTNER = G133H301   REC-TYPE = FTAC(FTP)
   %   INITSN   =           TRANS   = FROM       FUNCTION = MODIFY-FILE-ATTR
   %   USER-ADM = FT2V292   PROFILE = Z          PRIV     = NO
   %   FILENAME = TEST2

   %LOGGING-ID  = 00003212  RC     = 0000       TIME    = 2010-06-22 11:33:53
   %   INITIATOR= *REMOTE   PARTNER = G133H301   REC-TYPE = FTAC(FTP)
   %   INITSN   =           TRANS   = TO         FUNCTION = TRANSFER-FILE
   %   USER-ADM = FT2V292   PROFILE = Z          PRIV     = NO
   %   FILENAME = TEST1
```

Erläuterung

Bezeichnung	**Erläuterung**	
TYP (Spalte 1) bzw. REC-TYPE	Angabe, ob es sich um einen FT- oder FTAC-Logging-Satz handelt. In der kurzen Ausgabeform kennzeichnet in der ersten Spalte T den FT-, C den FTAC-Loggingsatz und P den FTP-spezifischen FTAC-Loggingsatz, in der langen Ausgabeform (REC-TYPE) finden Sie die Angaben im Klartext.	
TYP (Spalte 2-3) bzw. FUNCTION	Definition der FT-Funktion:	
	␣	Datei übertragen
	V	Datei übertragen mit Löschen der Sendedatei (nur Inbound möglich)
	A	Dateiattribute lesen
	D	Datei löschen
	C	Datei erzeugen
	M	Dateiattribute ändern
	R	Dateiverzeichnis lesen
	CD	Dateiverzeichnis einrichten
	MD	Dateiverzeichnis ändern
	DD	Dateiverzeichnis löschen
	L	Login (inbound FTP-Zugang)
LOGG-ID bzw. LOGGING-ID	maximal zwölfstellige Nummer des Logging-Satzes	
TIME	Zeitpunkt, wann der Logging-Satz geschrieben wurde	
RC	Reason-Code. Er gibt an, ob ein Auftrag erfolgreich ausgeführt oder warum er abgelehnt bzw. abgebrochen wurde. Wenn ein FTP-Auftrag aus „FTAC-Gründen“ (z.B. 0014) abgelehnt wurde, können Sie den genauen Grund für den Abbruch nur dem FTAC-Logging-Satz entnehmen. Weitere Information zum Reason-Code können Sie mit dem BS2000-Kommando HELP-MSG-INFORMATION abfragen.	
PARTNER	informiert über das beteiligte Client-System. Ausgegeben wird der ggf. gekürzte, maximal achtstellige, symbolische Name. In der Kurzform ist dem Namen des Client-Systems eine Kennzeichnung vorangestellt, der Sie die Richtung des Auftrags entnehmen können.	
TRANS=TO bzw. > bei PARTNER	Die Übertragungsrichtung ist zum Client-System. Diese Richtung wird angegeben bei einem – Sendeauftrag – Auftrag zum Ansehen von fernen Dateiattributen – Auftrag zum Ansehen von fernen Dateiverzeichnissen	
TRANS=FROM bzw. < bei PARTNER	Die Übertragungsrichtung ist zum lokalen System. Diese Richtung wird angegeben bei einem – Empfangsauftrag – Auftrag zum Ändern von fernen Dateiattributen – Auftrag zum Löschen von fernen Dateien	
TRANS	BOTH	Die Übertragungsrichtung ist zum Client-System und zum lokalen System.

Bezeichnung		Erläuterung
INITIATOR		Initiator des Auftrags; bei FTP immer *REMOTE
INIT bzw. INITSN		Bei FTP ist dieses Feld immer leer.
USER-ADM		Benutzerkennung, auf die sich die Aufträge im lokalen System beziehen
FILENAME		Dateiname im lokalen System
PROFILE		verwendetes Berechtigungsprofil
PRIV	*NO	nicht privilegiertes Berechtigungsprofil
	*YES	privilegiertes Berechtigungsprofil

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
0	0	CMD0001	Keine Logging-Sätze zum Auswahlkriterium gefunden.
33	32	CMD0221	Auftrag abgewiesen. Interner Fehler.
36	32	CMD0221	Auftrag abgewiesen. Inkonsistente Auftragsdaten.
83	32	CMD0221	Interner Fehler.
88	32	CMD0221	Fehler bei OPS-Ausgabe.
36	64	FTR1036	Benutzer nicht für andere Kennung berechtigt.
2	0	FTR2225	Informationsausgabe abgebrochen.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im im openFT-Handbuch „Meldungen.

Beispiel

Sie wollen sich alle Logging-Sätze zu Ihrer Benutzerkennung ansehen, die vor dem 1.1.2011 protokolliert wurden.

```
/SHOW-FT-LOGGING-RECORDS␣SELECT=*PARAMETERS(LOGGING-DATE=2011-01-01), -
/                        NUMBER=*ALL
```

Sie erhalten alle Logging-Sätze ausgegeben, die vor 00:00 Uhr des 1.1.2011 geschrieben wurden.

Den ersten Satz der Ausgabe wollen Sie sich ausführlich ansehen.

```
/SHOW-FT-LOG-REC␣(LOG-DATE=2011-01-01),INF=*ALL
```

5.2.10 SHOW-FT-PROFILE - Berechtigungsprofile anzeigen

Mit dem Kommando SHOW-FT-PROFILE können sich FTAC-Benutzer über ihre Berechtigungsprofile informieren. Sie erhalten entweder den Inhalt der ausgewählten Berechtigungsprofile oder nur die Namen der Profile ausgegeben. Grundsätzlich gibt es keine Möglichkeit, mit SHOW-FT-PROFILE an im Profil definierte Kennwörter oder an die Zugangsberechtigung heranzukommen! Wenn also eine Zugangsberechtigung vergessen wurde, muss mit MODIFY-FT-PROFILE eine neue angegeben werden.

Bei einem *HELP* auf die SDF-Kommandosyntax erscheinen Operanden, die in der folgenden Übersicht nicht dargestellt sind, da die Beschreibung in diesem Abschnitt auf die Operanden beschränkt ist, die für FTP relevant sind.

SHOW-FT-PROFILE - Darstellung der FTP-relevanten Operanden

NAME = ***ALL** / <alphanum-name 1..8> / *STD

,**SEL**ECT-PARAMETER = ***OWN** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

TRANSFER-**ADM**ISSION = ***ALL** / ***NOT-SPEC**IFIED / <alphanum-name 8..32> / <c-string 8..32 with-low> / <x-string 15..64> / ***SEC**RET

,**OWN**ER-**ID**ENTIFICATION = ***OWN** / ***ALL** / <name 1..8>

,**INF**ORMATION = **ONLY-**NAMES / ***ALL**

,**OUT**PUT =***SYSOUT (LAYOUT = *STD** / ***CSV)** /

***SYSLST (LAYOUT = *STD** / ***CSV)**

Beschreibung der Operanden

NAME=
Mit NAME geben Sie an, welche Berechtigungsprofile Sie sich anschauen wollen. NAME greift auf den benutzerweit eindeutigen Namen eines Berechtigungsprofils zu.

NAME=*ALL
Sie wollen sich alle Berechtigungsprofile Ihrer Kennung anschauen.

NAME=<alphanum-name 1..8>
Sie wollen sich das Berechtigungsprofil mit dem angegebenen Namen anschauen.

NAME = *STD
Zeigt das Standard-Berechtigungsprofil der eigenen Kennung an.

SELECT-PARAMETER=
Mit SELECT-PARAMETER können Sie Auswahlkriterien angeben, welche Berechtigungsprofile Sie sich anschauen wollen.

SELECT-PARAMETER=*OWN
Mit *OWN können Sie sich alle Berechtigungsprofile anschauen, deren Eigentümer Sie sind. Das bedeutet, dass Sie sich alle Berechtigungsprofile anschauen können, die Ihrer Benutzerkennung zugeordnet sind.

SELECT-PARAMETER=*PARAMETERS(...)
Diese Struktur enthält die Auswahlkriterien, mit denen Sie auf Ihre Berechtigungsprofile zugreifen können.

TRANSFER-ADMISSION=
Mit TRANSFER-ADMISSION können Sie die in einem Berechtigungsprofil definierte Zugangsberechtigung als Auswahlkriterium angeben.

TRANSFER-ADMISSION=*ALL
Sie nutzen die TRANSFER-ADMISSION nicht als Auswahlkriterium.

TRANSFER-ADMISSION=*NOT-SPECIFIED
Es werden nur Berechtigungsprofile ohne definierte Zugangsberechtigung angezeigt.

TRANSFER-ADMISSION=<alphanum-name 8..32> /
<c-string 8..32 with-low> / <x-string 15..64>
Sie wollen sich Ihr Berechtigungsprofil anschauen, das mit dieser Zugangsberechtigung angesprochen werden kann.

TRANSFER-ADMISSION=*SECRET
Sie erhalten vom System die Aufforderung, die Zugangsberechtigung einzugeben, diese wird aber nicht auf dem Bildschirm sichtbar.

OWNER-IDENTIFICATION =*OWN / *ALL / <name 1..8>
OWNER-IDENTIFICATION berechtigt den FTAC-Benutzer, auf seine eigenen Berechtigungsprofile zuzugreifen. Alle drei Angaben haben die gleiche Wirkung.

INFORMATION=
Mit INFORMATION bestimmen Sie den Umfang der gewünschten Information.

INFORMATION=*ONLY-NAMES
FTAC gibt Ihnen nur die Namen der Berechtigungsprofile und ein Kennzeichen, ob diese Profile privilegiert sind, aus.

INFORMATION= *ALL
FTAC gibt Ihnen den Inhalt der Berechtigungsprofile mit Ausnahme der Kennwörter und der Zugangsberechtigung aus.

OUTPUT=
Mit OUTPUT können Sie das Ausgabemedium für die Informationen bestimmen.

OUTPUT=*SYSOUT(...)
Die Ausgabe erfolgt auf SYSOUT.

OUTPUT=*SYSLST(...)
Die Ausgabe erfolgt auf SYSLST.

LAYOUT=*STD
Die Ausgabe wird in eine vom Anwender leicht lesbare Form gebracht.

LAYOUT=*CSV
Die Ausgabe erfolgt im Comma Separated Value Format. Dies ist ein speziell im PC Umfeld weit verbreitetes, tabellenartiges Format, bei dem die einzelnen Felder durch das Separatorenzeichen Semikolon “;“ getrennt sind.

Beispiel

Der FTAC-Verwalter schaut sich das Berechtigungsprofil UMSAWARE mit dem Kommando SHOW-FT-PROFILE an, um festzustellen, ob das Profil keine Gefährdung für den Datenschutz darstellt. Das ist möglich mit folgendem Kommando:

```
/SHOW-FT-PROFILE␣NAME=UMSAWARE,
          SELECT-PARAMETER=(OWNER-IDENTIFICATION=DONALD),INFORMATION=*ALL
```

Kurzform:

```
/SHOW-FT-PROF␣UMSAWARE,(,DONALD),INF=*ALL
```

Die Ausgabe hat dann die Form:

```
%UMSAWARE
% IGN-MAX-LEV = (IBR)
% FILE-NAME   = (PREFIX=UMSATZ.)
% USER-ADM    = (DONALD,M4711DON,OWN)
% PROC-ADM    = SAME
```

Die erste Zeile der Ausgabe zeigt den Namen des Berechtigungsprofils. Die beiden folgenden Zeilen zeigen die Festlegungen, die Donald im Kommando CREATE-FT-PROFILE durch die Operanden IGNORE-MAX-LEVELS=(INBOUND-RECEIVE=*YES) und FILE-NAME=(PREFIX=UMSATZ.) getroffen hat. Die Werte für USER-ADMISSION und PROCESSING-ADMISSION hat Donald hingegen nicht festgelegt, daher sind für sie Standardwerte eingesetzt worden.

Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	64	FTC0052	Die Ausgabe der Information wurde unterbrochen.
0	64	FTC0053	Es existiert kein FT-Profil zu den angegebenen Kriterien.
0	0	FTC0054	Es gibt keine Informationen zu den angegebenen Kriterien.
0	64	FTC0153	Die angegebene Eigentuemeridentifikation ist nicht die eigene Benutzerkennung.
0	64	FTC0171	Das angegebene Profil existiert nicht.
0	64	FTC0255	Ein Systemfehler ist aufgetreten.

SC1/2 = Subcode 1/2 in Dezimal-Darstellung
Zusätzliche Informationen finden Sie im openFT-Handbuch „Meldungen.

6 TELNET

Dieser Abschnitt beschreibt sowohl den TELNET-Client in BS2000/OSD, als auch (ab Seite 336) den TELNET-Server.

Grundfunktion

Wenn ein Anwender mit dem Programm TELNET die Verbindung zu einem Partner-Rechner hergestellt hat, verhält sich seine Datenstation ähnlich einer Zeilendatenstation, die direkt an das Partner-System angeschlossen ist.

Mit TELNET ist ein Dialog nur im Line-Modus möglich. Ist der Partnerrechner ein BS2000-System, so sind zum Beispiel FHS und EDOR nicht verwendbar, SDF und EDT sind nur im Line-Modus verwendbar. Ist der Partnerrechner ein Unix-System, so ist zum Beispiel CED nicht verwendbar.

Sonstige Funktionen

- Information für den Anwender
  In einer Sitzung hat der Anwender die Möglichkeit, sich
  - über die zur Verfügung stehenden Kommandos und deren Bedeutung,
  - über den Partner-Rechner, mit dem er gerade verbunden ist und die Bedingungen (Optionen) der Datenübertragung

  zu informieren.
- Überwachungsfunktionen
  Zu Wartungs- und Diagnosezwecken sind im Client Trace-Funktionen realisiert, die jenen Server-Trace-Funktionen entsprechen, die der Administrator über das Kommando */INTR* des Systembedieners anstoßen kann.
- Schnittstelle zum lokalen und fernen Betriebssystem
  Damit der Anwender zur Abgabe eines BS2000-Kommandos den Programmlauf (und zuvor die Verbindung zum Partner-Rechner) nicht unterbrechen muss, steht eine Schnittstelle zum lokalen Betriebssystem zur Verfügung.

6.1 TELNET-Client im BS2000/OSD

Jeder Anwender von TELNET im BS2000 eröffnet eine eigene TELNET-Client-Task. Beim Kommando *open* stellt der Client die Verbindung zum gewünschten TELNET-Server an einem fernen Rechner her.

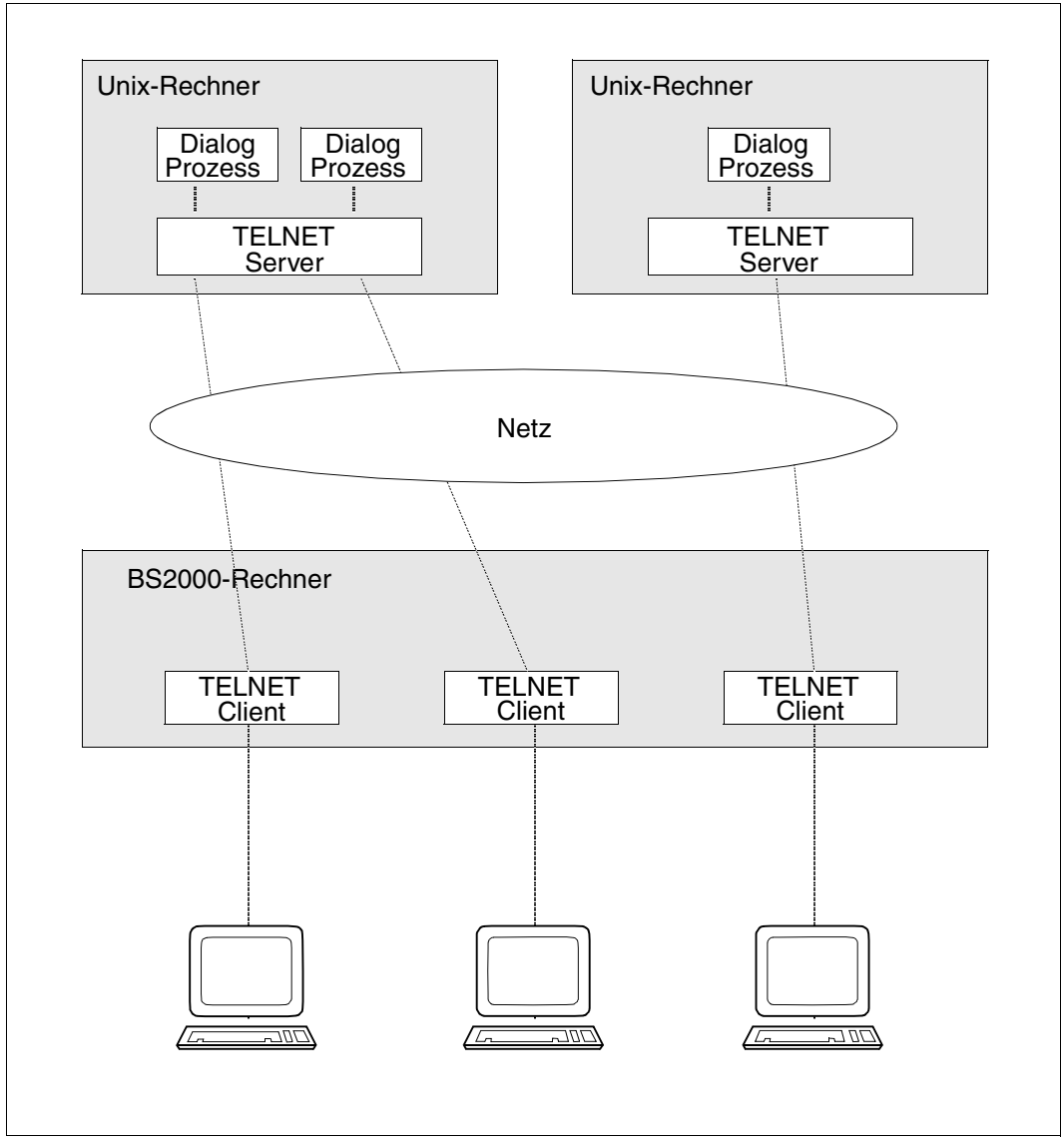


Bild 7: TELNET-Clients im BS2000/OSD

6.1.1 Kommando-Modus, Eingabe-Modus

Über TELNET kann an Rechnern des lokalen Netzes im Dialog gearbeitet werden. Um zu entscheiden, ob Eingaben an den TELNET-Client oder an das ferne Dialog-System gerichtet sind, kennt TELNET zwei Modi.

- Im Kommando-Modus werden Kommandos an den TELNET-Client eingegeben.
- Im Eingabe-Modus werden Daten und Kommandos an das ferne Dialog-System eingegeben.

Mit TELNET ist nur zeilenorientierter Dialog möglich. Ist der Partnerrechner ein Unix-System, so ist beispielsweise das Arbeiten mit CED nicht möglich. Ist der Partnerrechner ein BS2000-System, so sind z.B. FHS gar nicht, SDF und EDT nur im Line-Modus verwendbar.

Mit dem TELNET-Kommando *!* kann aus dem TELNET-Kommando-Modus in den BS2000/OSD- bzw. POSIX-Kommando-Modus gewechselt werden.

Das Bild auf Seite 286 zeigt die drei Modi zur Eingabe von Daten und Kommandos und Übergänge von einem Modus zum anderen. (Exaktes Verhalten beim Wechsel in den BS2000/OSD- bzw. POSIX-Kommando-Modus durch das Kommando *!* bzw. Wechsel in den TELNET-Kommando-Modus durch Eingabe des ESCAPE-Symbols sind in der Detailbeschreibung der Kommandos nachzulesen.)

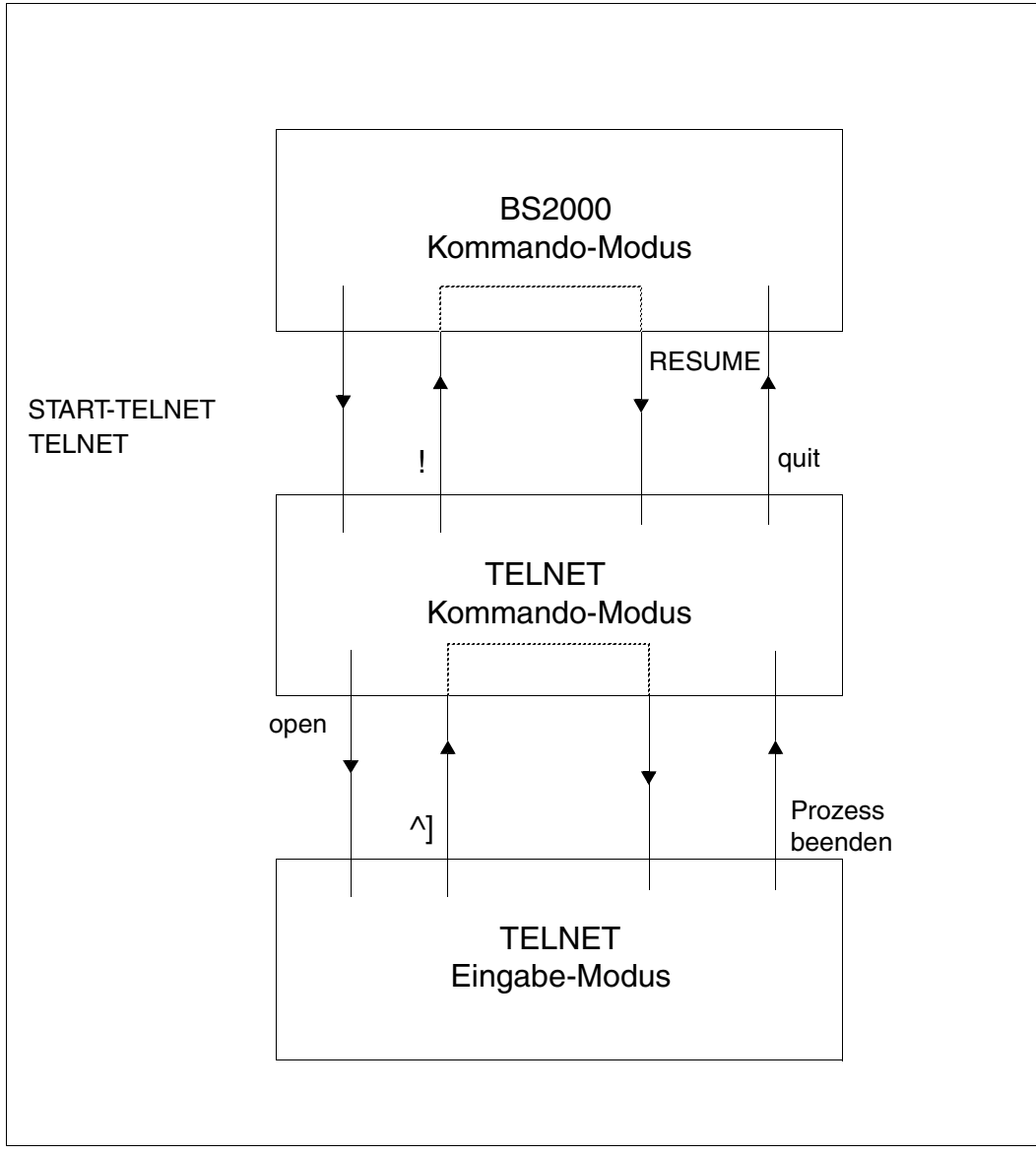


Bild 8: Dateneingabe-Modi

Einstieg

Den TELNET-Client starten Sie wie folgt:

START-TELNET bzw. TELNET

Das Beenden des Client erfolgt mit dem Kommando *quit*.

Im Dialog meldet sich TELNET mit einer Versionsnummer und der Eingabe-Aufforderung *telnet*>. Der Auftragsschalter 1 darf nicht gesetzt sein. Nach dem Aufruf von TELNET befinden Sie sich im TELNET-Kommando-Modus. In diesem Modus kann TELNET gesteuert und Information eingeholt werden.

Mit dem Kommando *open* <*ipadr*> werden Sie mit einem fernen Rechner verbunden. Sie befinden sich nun im TELNET-Eingabe-Modus und können sämtliche zeilenorientierte Kommandos an das Betriebssystem des fernen Rechners richten. Die Verbindung zum fernen Rechner wird gelöst, wenn im fernen Betriebssystem das jeweilige Kommando zur Beendigung des laufenden Prozesses gegeben wird.

Zur Abgabe eines TELNET-Kommandos kann der Eingabe-Modus durch Eingabe des ESCAPE-Symbols unterbrochen werden. Mit dem Kommando *close* kann im TELNET-Kommando-Modus die Verbindung zum fernen Rechner gelöst werden.

Die Voreinstellung für das ESCAPE-Symbol ist ^]. Mit dem Kommando *escape* kann das ESCAPE-Symbol gewechselt werden. Der Funktion der Taste CONTROL auf asynchronen Terminals (Unix-Welt) entspricht das Zeichen ^. Soll das Zeichen ^ übertragen werden, ist es zu verdoppeln.

6.1.2 TELNET-Client in POSIX

Wenn der TELNET-Client unter POSIX abläuft, gibt er die folgende Startmeldung aus:

```
POSIX-TELNET <vers> <date> <time>
```

<date> und <time> spezifizieren hier Datum und Uhrzeit zum Zeitpunkt der Übersetzung des Main-Moduls. Zum Zeitpunkt des Starts wechselt der TELNET-Client in das lokale POSIX-Dateiverzeichnis.

Dem Batch-Betrieb in BS2000/OSD entspricht in POSIX der TELNET-Aufruf mit Schalter -n.

Die Eingabe-Aufforderung für das Logon-Passwort erscheint bei gesonderter Eingabe erst nach einer kurzen Wartezeit.

Sicherer TELNET-Client in POSIX

Der sichere Einsatz (via Authentifizierung und Verschlüsselung) des TELNET-Client in POSIX erfolgt analog zum sicheren Einsatz des TELNET-Client in BS2000/OSD (siehe Abschnitt „Sicherheit im TELNET-Client“ auf Seite 289).

Die private Option-Datei hat den Dateinamen $HOME/.telnet.options. Die Dateinamen beziehen sich auf das POSIX-Dateisystem. Wenn Sie z.B. eine BS2000/OSD-Datei als CertificateFile verwenden wollen, müssen Sie deshalb dem Dateinamen den Präfix „/BS2/“ voranstellen.

6.1.3 Sicherheit im TELNET-Client

Es gibt drei Verfahren, um den sicheren Einsatz von TELNET via Authentifizierung und Verschlüsselung zu gewährleisten:

- START-TLS-Option

  Die START-TLS-Option wurde ausschließlich für TLS/SSL implementiert und wird im BS2000/OSD durch die Client-Option *-Z tls-required* bzw. das Client-Kommando *tls* unterstützt.

- „Telnet Authentication Option“ (RFC 2941) für das Aushandeln eines Authentifizierungsverfahrens

  Im BS2000/OSD wird derzeit nur TLS/SSL unterstützt. Eingestellt wird die „Telnet Authentication Option“ durch die Option *-A* bzw. das Client-Kommando *auth*. Die „Telnet Authentication Option“ wird in Zukunft möglicherweise an Bedeutung gewinnen, weil mit ihr die verschiedensten Authentifizierungsverfahren unterstützt werden können, u.a. auch Kerberos. Im Folgenden wird die „Telnet Authentication Option“ als AUTHENTICATION-Option bezeichnet.

- „Telnet Data Encryption Option“ (RFC 2946) für aus Aushandeln eines symmetrischen Verschlüsselungsverfahrens und des zugehörigen Schlüssels

  Im BS2000/OSD wird derzeit nur DES 64 (RFC 2952, RFC 2953) unterstützt. Eingestellt wird die „Telnet Data Encryption Option“ durch die Option *-E* bzw. durch das Client-Kommando *encrypt*. Die „Telnet Data Encryption Option“ wird im Folgenden als ENCRYPTION-Option bezeichnet.

START-TLS-Option (siehe Seite 292), AUTHENTICATION-Option (siehe Seite 309) und ENCRYPTION-Option (siehe Seite 310) sind in den nachfolgenden Abschnitten ausführlich beschrieben.

6.1.3.1 Einstellung der Options via Option-Datei

Die Option-Einstellungen hinterlegen Sie in einer oder mehreren Option-Dateien. Eine dieser Option-Dateien wird beim Start des TELNET-Client eingelesen. Sie können jedoch auch zu einem späteren Zeitpunkt die Option-Einstellungen den aktuellen Erfordernissen anpassen, indem Sie mit dem TELNET-Client-Kommando *readopt* eine Option-Datei einlesen.

Ermittlung der relevanten Option-Datei durch den TELNET-Client

Bei der Ermittlung der relevanten Option-Dateien verfährt der TELNET-Client wie folgt:

1. Zunächst sucht der TELNET-Client nach einer zentral abgelegten Option-Datei mit dem Standarddateinamen $.SYSDAT.TCP-IP-AP.052.T-GOPT.
2. Unabhängig von der unter 1.) genannten zentralen Datei sucht der TELNET-Client zusätzlich eine anwendereigene Option-Datei unter dem Namen SYSDAT.TCP-IP-AP.052.TEL.OPT. Sofern diese Datei existiert, liest der TELNET-Client aus ihr die Options.

Zu den Besonderheiten bei der TLS-Unterstützung des TELNET-Client in POSIX siehe Seite 288.

Falls eine Option in beiden Dateien mit unterschiedlichen Werten vorkommt, gilt der in der anwendereigenen Datei definierte Wert.

Falls Sicherheitseinstellungen via TELNET-Client-Kommando vorgenommen werden, haben diese Vorrang vor den in den Option-Dateien spezifizierten Angaben.

Notation der Options in der Option-Datei

Die einzelnen Options müssen gemäß den folgenden Regeln in die Option-Datei eingetragen sein:

- Jede fortzusetzende Zeile muss mit dem Fortsetzungszeichen „\“ abgeschlossen werden.
- Eine Zeile, die mit dem Zeichen „#“ in Spalte 1 beginnt, wird beim Einlesen ignoriert.
- Bei den Option-Namen wird Klein-/Großschreibung nicht unterschieden.

6.1.3.2 Steuerung der Sicherheitseinstellungen mithilfe von TELNET-Client-Kommandos

Für die Steuerung der Sicherheitseinstellungen im TELNET-Client gibt es die folgenden TELNET-Client-Kommandos:

- *tls* - TLS-Unterstützung ein-/ausschalten (siehe Seite 330)
- *auth* - Einstellungen der Authentication-Option modifizieren, Status ausgeben (siehe Seite 313)
- *encrypt* - Einstellungen der Encryption-Option modifizieren, Status ausgeben (siehe Seite 317)

Die via TELNET-Client-Kommando vorgenommenen Sicherheitseinstellungen haben Vorrang vor den in den Option-Dateien spezifizierten Angaben.

6.1.3.3 START-TLS-Option

Mit dieser Option können Sie die TLS-Unterstützung im TELNET-Client steuern. Das Aushandeln der Modalitäten von Authentifizierung übernimmt in diesem Fall TLS, so dass TELNET nicht davon belastet wird.

Die Options für die Nutzung der TLS-Unterstützung geben Sie wie folgt in der/den oben genannten Option-Datei(en) an:

```
-Z <option>
```

Zeitpunkt, zu dem Options bzw. Option-Änderungen wirksam werden

Nach dem Start des TELNET-Client können Sie mit dem Client-Kommando *readopt* (siehe Seite 325) die gewünschte Option-Datei einlesen. Diesen Vorgang können Sie beliebig oft wiederholen.

Die Option *-Z OpenSSLLibname* (siehe Seite 307) wird während einer TELNET-Sitzung nur einmal ausgewertet, und zwar zum Zeitpunkt des Ladens der OpenSSL-Bibliothek. Alle anderen Optionen werden nach dem Aufbau der Verbindung zum Server wirksam.

Beschreibung der -Z-Options

Nachfolgend sind die -Z-Options beschrieben. Dabei ist zu beachten:

– Mit Ausnahme der Option *-Z tls-required* (siehe Seite 293) können alle beschriebenen Z-Options auch für die AUTHENTICATION-Option (siehe Seite 309) verwendet werden, sofern mit der AUTHENTICATION-Option eine TLS/SSL-Unterstützung verbunden ist.

– Die parallele Unterstützung von START-TLS-Option (*-Z tls-required*) und AUTHENTICATION-Option (*-A*) ist nicht sinnvoll. Deshalb wird bei gleichzeitiger Angabe beider Options folgende Fehlermeldung ausgegeben:

  ```
  Both START-TLS and AUTHENTICATION-Option not allowed
  ```

– Die Option *-Z OpenSSLlibName* (siehe Seite 307) ist auch für die Unterstützung der ENCRYPTION-Option (siehe Seite 310) von Bedeutung, da ausschließlich Encryption-Routinen aus dieser OpenSSL-Bibliothek verwendet werden.

-Z tls-required - TLS-Absicherung im TELNET-Client ein-/ausschalten

Mit der Option *-Z tls-required* wird die TLS-Absicherung via START-TLS-Option im TELNET-Client ein- oder ausgeschaltet.

-Z tls-required
[**=** { **yes** | **no** }]

yes
START-TLS-Unterstützung wird eingeschaltet.

no
START-TLS-Unterstützung wird ausgeschaltet.

-Z tls-required ohne Operanden spezifiziert
Es gilt *-Z tls-required = yes* (START-TLS-Unterstützung wird eingeschaltet).

-Z tls-required nicht angegeben
START-TLS-Unterstützung wird nicht eingeschaltet.

-Z CertificateFile - Datei mit X.509-Client-Zertifikat spezifizieren

Mit der Option *-Z CertificateFile* wird eine Datei spezifiziert, die das X.509-Client-Zertifikat im PEM-Format enthält. Diese Datei kann auch den privaten Schlüssel des Client enthalten. In der Regel werden aber Zertifikat und Schlüssel in getrennten Dateien abgelegt. In diesem Fall wird die Schlüsseldatei mithilfe der Option *-Z KeyFile* (siehe Seite 295) spezifiziert.

-Z CertificateFile
=\{<dateiname 1 .. 54> | ***NONE**\}

<dateiname 1 .. 54>
Name der Datei, die das X.509-Client-Zertifikat im PEM-Format enthält.

***NONE**
Es wird kein Client-Zertifikat (und somit auch keine Client-Authentifizierung) verwendet. *NONE ist Voreinstellung.

-Z KeyFile - Datei mit Client-Schlüssel im PEM-Format spezifizieren

Mit der Option *-Z KeyFile* wird eine Datei spezifiziert, die den privaten Client-Schlüssel im PEM-Format enthält.

Wenn sowohl X.509-Client-Zertifikat als auch privater Client-Schlüssel innerhalb derselben Datei enthalten sind (siehe Option *-Z CertificateFile* auf Seite 294), braucht die Option *-Z KeyFile* nicht angegeben zu werden.

Wenn der Client-Schlüssel mit einer Passphrase geschützt ist, dann muss diese Passphrase nach dem Start des TELNET-Client beim Aufbau einer mit TLS gesicherten TELNET-Verbindung eingegeben werden.

-Z KeyFile
={<dateiname 1 .. 54> | ***NONE**}

<dateiname 1 .. 54>
Name der Datei, die den privaten Client-Schlüssel enthält.

***NONE**
Es wird keine separate Client-Schlüssel-Datei verwendet.

Voreinstellung ist der bei der Option *-Z CertificateFile* (siehe Seite 294) spezifizierte Dateiname.

-Z CACertificateFile - Datei mit Server-Authentifizierung spezifizieren

Mit der Option *-Z CACertificateFile* wird eine Datei spezifiziert, die die für die Authentifizierung des TELNET-Servers erforderlichen CA-Zertifikate im PEM-Format enthält. Die einzelnen PEM-Zertifikate sind sequenziell in der Datei angeordnet.

Zum Hinzufügen oder Löschen von Zertifikaten können Sie die Datei mit einem beliebigen Text-Editor bearbeiten. Die einzelnen Zertifikate sind wie folgt in der Datei notiert:

```
-----BEGIN CERTIFICATE-----
< CA-Zertifikat in Base64-Codierung >
-----END CERTIFICATE-----
```

Text außerhalb dieser Sequenzen wird vom TELNET-Client ignoriert und kann deshalb zum Kennzeichnen der Zertifikate verwendet werden, die wegen der Base64-Codierung in nicht lesbarer Form vorliegen.

-Z CACertificateFile
={<dateiname 1 .. 54> | ***NONE**}

<dateiname 1 .. 54>
Name der Datei, die die für die Authentifizierung des TELNET-Servers erforderlichen Zertifikate im PEM-Format enthält.

***NONE**
Es wird keine Datei mit CA-Zertifikaten angegeben.
*NONE ist Voreinstellung.

-Z CARevocationFile - Datei mit CRL spezifizieren

Mit der Option *-Z CARevocationFile* wird eine Datei spezifiziert, die die CRLs (Certificate Revocation List) der Zertifizierungsinstanzen enthält. (Zertifikate, die von einer Zertifizierungsinstanz herausgegeben wurden, können durch Veröffentlichung einer so genannten Certificate Revocation List (CRL) für ungültig erklärt werden.)

-Z CARevocationFile
={<dateiname 1 .. 54> | ***NONE**}

<dateiname 1 .. 54>
Name der Datei, die die CRLs der Zertifizierungsinstanzen enthält.

***NONE**
Es wird keine Datei mit CRLs angegeben.
*NONE ist Voreinstellung.

-Z VerifyServer - TELNET-Server-Zertifikat verifizieren (ja/nein)

Mit der Option *-Z VerifyServer* wird festgelegt, ob das TELNET-Server-Zertifikat verifiziert werden soll.

-Z VerifyServer
={**YES** | **NO**}

YES
Das TELNET-Server-Zertifikat soll verifiziert werden.
YES ist Voreinstellung.

NO
Das TELNET-Server-Zertifikat soll nicht verifiziert werden.
Mit dieser Einstellung wird man anfällig für „man in the middle“-Angriffe.

-Z VerifyDepth - Verifizierungstiefe festlegen

Mit der Option *-Z VerifyDepth* wird die so genannte Verifizierungstiefe festgelegt, d.h. die Anzahl der maximal zulässigen Zertifikate zwischen dem TELNET-Server-Zertifikat und dem Zertifikat, das dem TELNET-Client bekannt ist.

Im Einzelnen ist zu beachten:

- Wird für die maximale Tiefe der Wert 1 spezifiziert (Default), dann muss das Server-Zertifikat direkt von einer dem TELNET-Client bekannten CA (Certificate Authority) signiert worden sein, damit es akzeptiert wird.
- Wird die maximale Tiefe überschritten, dann wird die Verbindung abgebrochen, sofern nicht aufgrund von *-Z VerifyServer NO* (siehe Seite 298) die Verifizierung des TELNET-Server-Zertifikats ausgeschaltet ist.
- Die Spezifikation der Tiefe 0 ist nicht sinnvoll. In diesem Fall wären nur selbstsignierte Zertifikate zulässig.

-Z VerifyDepth
=<tiefe>

<tiefe>
Anzahl der maximal zulässigen Zertifikate zwischen dem TELNET-Server-Zertifikat und dem Zertifikat, das dem TELNET-Client bekannt ist (einschließlich dem TELNET-Server-Zertifikat).
Voreinstellung: 1

-Z CipherSuite - Spezifikation einer Verschlüsselungsverfahren-Vorzugsliste

Mit der Option *-Z CipherSuite* wird eine Verschlüsselungsverfahren-Vorzugsliste spezifiziert. Falls diese Option nicht angegeben wird, wird eine voreingestellte Vorzugsliste verwendet.

-Z CipherSuite
=<spezifikation>

<spezifikation>
Spezifikation einer Verschlüsselungsverfahren-Vorzugsliste (Näheres siehe nachfolgende Beschreibung des Aufbaus einer Spezifikation).
ALL: !EXP: !ADH ist Voreinstellung.

Beschreibung des Aufbaus einer Spezifikation

Die Spezifikation besteht aus einem oder mehreren Chiffre-Mnemonics, die durch einen Doppelpunkt (:) getrennt sind.

Ein Chiffre-Mnemonic kann folgende Formen annehmen:

- Ein Chiffre-Mnemonic kann aus einer einzelnen Verschlüsselungs-Suite wie z.B. DES-CBC-SHA bestehen.
- Ein Chiffre-Mnemonic kann repräsentieren:
  - Liste von Verschlüsselungs-Suites, die einen bestimmten Algorithmus enthalten
  - Verschlüsselungs-Suites eines bestimmten Typs

  Beispielsweise repräsentiert SHA1 alle Verschlüsselungs-Suiten, die den Digest-Algorithmus SHA1 benutzen und SSLv3 repräsentiert alle SSL-Version 3-Algorithmen.
- Listen von Verschlüsselungs-Suiten können mithilfe des „+“-Zeichens zu einem einzelnen Chiffre-Mnemonic kombiniert werden. Dies wird dann als logische UND-Operation interpretiert. So repräsentiert SHA1+DES alle Verschlüsselungs-Suiten, die die SHA1- und DES-Algorithmen enthalten.

- Jedem Chiffre-Mnemonic kann optional eines der Zeichen „!“, „-“ oder „+“ vorangestellt werden:
  - Bei Voranstellen von „!“ werden die betreffenden Verschlüsselungs-Suiten dauerhaft aus der Vorzugsliste gelöscht. Sie erscheinen auch dann nicht wieder in der Vorzugsliste, wenn sie explizit angegeben werden.
  - Bei Voranstellen von „-“ werden die betreffenden Verschlüsselungs-Suiten aus der Vorzugsliste gelöscht, aber einige von ihnen oder alle können durch nachfolgende Optionen wieder hinzugefügt werden.
  - Bei Voranstellen von „+“ werden die betreffenden Verschlüsselungs-Suiten an das Ende der Vorzugsliste verschoben. Somit werden keine Verschlüsselungs-Suiten zur Vorzugsliste hinzugefügt, sondern nur existierende verschoben.
  - Wenn keines der drei Zeichen „!“, „-“ oder „+“ vorangestellt ist, wird der Chiffre-Mnemonic als eine Liste von Verschlüsselungs-Suiten interpretiert, die an die aktuelle Vorzugsliste angehängt wird. Wenn dies eine Verschlüsselungs-Suite einschließt, die schon in der aktuellen Vorzugsliste enthalten ist, dann wird diese ignoriert. Sie wird nicht an das Ende der Vorzugsliste verschoben.
- Der Chiffre-Mnemonic @STRENGTH kann an beliebiger Stelle eingefügt werden, um die aktuelle Vorzugsliste nach der Länge der Verschlüsselungsschlüssel zu sortieren.

Zulässige Chiffre-Mnemonics

Nachfolgend sind die zulässigen Chiffre-Mnemonics beschrieben.

ALL
: Alle Verschlüsselungs-Suiten mit Ausnahme der eNULL-Chiffren. Letztere müssen explizit aktiviert werden.

HIGH
: Verschlüsselungs-Suiten mit Schlüssellängen größer 128 Bit. Da 3DES mit 168 Bit Länge bewertet wird (anstatt mit 112 Bit, wie von manchen Kryptographen), zählt es zu dieser Suiten-Klasse.

MEDIUM
: Verschlüsselungs-Suiten mit Schlüssellänge 128 Bit.

LOW
: Verschlüsselungs-Suiten mit 64 oder 56 Bit Schlüssellänge, ausgenommen Export-Verschlüsselungs-Suiten.

EXP, EXPORT
: Export-Verschlüsselungs-Algorithmen einschließlich 40- und 56-Bit-Algorithmen.

EXPORT40
: 40-Bit-Export-Verschlüsselungs-Algorithmen.

EXPORT56
: 56-Bit-Export-Verschlüsselungs-Algorithmen.

eNULL, NULL
: „NULL“-Verschlüsselungs-Algorithmen, d.h. solche ohne Verschlüsselung. Da diese keine Verschlüsselung bieten und damit ein Sicherheitsrisiko darstellen, werden sie standardmäßig deaktiviert und müssen gegebenenfalls explizit angegeben werden.

aNULL
: Verschlüsselungs-Suiten ohne Authentifizierung. Dies sind im Augenblick die anonymen Diffie-Hellman-Algorithmen. Diese Algorithmen sind anfällig für „man in the middle“-Angriffe, so dass von ihrer Benutzung abgeraten wird.

kRSA, RSA
: Verschlüsselungs-Suiten mit RSA-Schlüsselaustausch.

kEDH
: Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüsselvereinbarung.

aRSA
: Verschlüsselungs-Suiten mit RSA-Authentifizierung, d.h. die Zertifikate enthalten RSA-Schlüssel.

aDSS, DSS
: Verschlüsselungs-Suiten mit DSS-Authentifizierung, d.h. die Zertifikate enthalten DSS-Schlüssel.

TLSv1, SSLv3, SSLv2
: TLS v1-, SSL v3- bzw. SSL v2-Verschlüsselungs-Suiten. Die TLSv1-Suiten und die SSLv3-Suiten sind identisch.

AES
: Verschlüsselungs-Suiten mit AES-Verschlüsselung (128 und 256 Bit Schlüssellänge).

DH
: Verschlüsselungs-Suiten mit Diffie-Hellman-Schlüsselaustausch, einschließlich anonymem Austausch.

ADH
Verschlüsselungs-Suiten mit anonymem Diffie-Hellman-Schlüsselaustausch.

3DES
Verschlüsselungs-Suiten mit Triple-DES-Verschlüsselung.

DES
Verschlüsselungs-Suiten mit DES-Verschlüsselung (kein Triple-DES).

RC4
Verschlüsselungs-Suiten mit RC4-Verschlüsselung.

RC2
Verschlüsselungs-Suiten mit RC2-Verschlüsselung.

MD5
Verschlüsselungs-Suiten mit MD5-Hash-Funktion.

SHA1, SHA
Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.

In der nachfolgenden Tabelle sind die verfügbaren Verschlüsselungs-Suiten zusammengefasst.

Name	**Version**	**Schlüssel-Austausch**	**Authentifizierung**	**Veschlüsselung**	**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	

Verfügbare Verschlüsselungs-Suiten

Name	Version	Schlüssel-Austausch	Authentifi-zierung	Veschlüsse-lung	Digest	Export
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	keine	AES(256)	SHA1	
ADH-AES128-SHA	SSLv3	DH	keine	AES(128)	SHA1	
ADH-DES-CBC3-SHA	SSLv3	DH	keine	3DES(168)	SHA1	
ADH-DES-CBC-SHA	SSLv3	DH	keine	DES(56)	SHA1	
ADH-RC4-MD5	SSLv3	DH	keine	RC4(128)	MD5	
EXP-ADH-DES-CBC-SHA	SSLv3	DH(512)	keine	DES(40)	SHA1	export
EXP-ADH-RC4-MD5	SSLv3	DH(512)	keine	RC4(40)	MD5	export
NULL-SHA	SSLv3	RSA	RSA	keine	SHA1	
NULL-MD5	SSLv3	RSA	RSA	keine	MD5	

Verfügbare Verschlüsselungs-Suiten

-Z RandomSeed - Pseudo-Zufallszahlengenerator initialisieren

Mit der Option *-Z RandomSeed* wird spezifiziert, wie der von TLS verwendete Pseudo-Zufallszahlengenerator initialisiert wird. Eine gute Initialisierung mit möglichst zufälligen, nicht vorhersagbaren Werten ist für die TLS-Absicherung von entscheidender Bedeutung. Falls auf dem Rechner, auf dem TELNET-Client abläuft, das BS2000/OSD-Subsystem PRNGD (**P**seudo **R**andom **N**umber **G**enerator **D**emon) aktiv ist, wird PRNGD für die Initialisierung verwendet, so dass die Einstellung der Option *-Z RandomSeed* praktisch ohne Bedeutung ist. Das Subsystem PRNGD ist beschrieben im Handbuch „interNet Services Administratorhandbuch“.

-Z RandomSeed
=\{**PROGRAM** | **USER**\}

PROGRAM
Es werden programm-interne Funktionen verwendet. Diese nutzen vor allem die Schwankungen der Echtzeituhr in Relation zum Taktgeber der Rechner-CPU, um Zufallszahlen für die Initialisierung zu generieren.

USER
Ein Teil der Initialisierung erfolgt analog der Initialisierung bei der Angabe PROGRAM. Zusätzlich wird der Anwender wiederholt aufgefordert, Zeichen in möglichst zufälliger Auswahl via Tastatur einzugeben und/oder die ENTER-Taste zu betätigen. Für die Initialisierung wird zum einen der Zeitstempel der Eingabe verwendet. Darüber hinaus werden die eingegebenen Zeichen ebenfalls für die Initialisierung verwendet. Da jedoch die Zufälligkeit der eingegebenen Zeichen nicht bekannt ist und diese Zeichen außerdem in vielen Fällen abgehört werden können, werden sie nicht berücksichtigt bei der Abschätzung, ob schon genügend Initialisierungsmaterial vorliegt. In diese Abschätzung geht nur die Anzahl der Betätigungen der ENTER-Taste ein.
*USER ist Voreinstellung.

Wenn der TELNET-Client im Batch-Modus betrieben wird, empfiehlt sich in der Regel die Einstellung PROGRAM, da im Batch-Modus kein Anwender für die Eingabe von zufälligen Zeichen zur Verfügung steht.

-Z Protocol - TLS/SSL-Protokollauswahl

OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-Protokoll in der Version 1. Mit der Option *-Z Protocol* können einige dieser Protokolle selektiv aktiviert werden.

-Z Protocol
={**+** | **-**} {**SSLv2** | **SSLv3** | **TLSv1** | **All**} ...

+
Das nachfolgend spezifizierte Protokoll ist zugelassen.

-
Das nachfolgend spezifizierte Protokoll ist nicht zugelassen.

SSLv2
SSL-Protokoll der Version 2

Das SSL-Protokoll in der Version 2 weist einige sicherheitstechnische Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

SSLv3
SSL-Protokoll der Version 3

TLSv1
TLS-Protokoll der Version 1

ALL
Alle Protokolle sollen aktiviert werden.

All-SSLv2 ist Voreinstellung.

Beispiel

Die Angaben `-Z Protocol SSLv3+TLSv1` und `-Z Protocol All-SSLv2` haben dieselbe Wirkung.

-Z OpenSSLlibName - LMS-Datei für OpenSSL-Bibliothek festlegen

Mit der Option *-Z OpenSSLlibName* wird festgelegt, aus welcher LMS-Datei die OpenSSL-Bibliothek nachgeladen werden soll. Die OpenSSL-Bibbliothek wird nur nachgeladen, falls mindestens eine der beiden Options *-Z tls-required* oder *-A on* spezifiziert wurde.

Eine vom Standardnamen abweichende Angabe kann z.B. erforderlich sein, wenn die OpenSSL-Bibliothek auch von anderen Produkten verwendet wird.

Das Nachladen der OpenSSL-Bibliothek lässt sich durch Cache-Speichern mithilfe von DAB beschleunigen. Bei gemeinsamer Verwendung der OpenSSL-Bibliothek durch mehrere Produkte wird die Größe des verwendeten DAB-Puffers verringert.

-Z OpenSSLlibName
=<openssl-libname>

<openssl-libname>
Name der LMS-Datei, aus der die OpenSSL-Bibliothek nachgeladen werden soll.
Voreinstellung: $.SYSLNK.TCP-IP-AP.052

-Z UseCryptoHardware - Krypto-Hardware verwenden (ja/nein)

Mit der Option *-Z UseCryptoHardware* wird festgelegt, ob zur Berechnung kryptographischer Algorithmen Krypto-Hardware, z.B. eine openCRYPT^{TM}-Box (siehe Handbuch „open-Crypt“), verwendet werden soll.

-Z UseCryptoHardware
={**YES** | **NO**}

YES
Es wird Krypto-Hardware verwendet.

NO
Es wird keine Krypto-Hardware verwendet.
NO ist Voreinstellung.

6.1.3.4 Option -A - AUTHENTICATION-Option aktivieren / deaktivieren

Mit der Option *-A* kann die Unterstützung der AUTHENTICATION-Option aktiviert bzw. deaktiviert werden. In BS2000/OSD ist die AUTHENTICATION-Option derzeit nur für TLS/SSL realisiert. Die für den SSL-Betrieb notwendigen Einstellungen können somit mithilfe der *-Z*-Options (siehe Seite 292 ff) vorgenommen werden.

Die parallele Unterstützung von START-TLS-Option (*-Z tls-required*) und AUTHENTICATION-Option ist nicht sinnvoll. Deshalb wird bei gleichzeitiger Angabe der Options *-A on* und *-Z tls-required* folgende Fehlermeldung ausgegeben:

```
Both START-TLS and AUTHENTICATION-Option not allowed
```

Alternativ können Sie die Unterstützung der AUTHENTICATION-Option auch mithilfe des TELNET-Client-Kommandos *auth* (siehe Seite 313) einstellen.

-A
on | **off** | **debug** | **status**

on
Die AUTHENTICATION-Option wird unterstützt.

off
Die AUTHENTICATION-Option wird nicht unterstützt.
off ist Voreinstellung.

debug
Der Authentication-Trace wird eingeschaltet.

status
Gibt den aktuellen Status der AUTHENTICATION-Option aus.

6.1.3.5 Option -H - ENCRYPTION-Option aktivieren / deaktivieren

Mit der Option *-H* kann die Unterstützung der ENCRYPTION-Option aktiviert bzw. deaktiviert werden, die zum Aushandeln des Verschlüsselungsverfahrens sowie des verwendeten Schlüssels verwendet wird. Derzeit wird in TELNET nur DES64 in den Varianten DES_CFB64 und DES_OFB64 unterstützt.

Da ausschließlich Encryption-Routinen aus der OpenSSL-Bibliothek verwendet werden, können Sie diese Bibliothek − sofern ihr Name vom Standard (SYSLNK.TCP-IP-AP.052) abweicht − mithilfe der Option *-Z OpenSSLlibname* (siehe Seite 307) spezifizieren.

Die Option *-H* wird nur dann wirksam, wenn nicht gleichzeitig die Options *-Z tls-required* oder *-A on* spezifiziert wurden.

Alternativ können Sie die Unterstützung der ENCRYPTION-Option auch mithilfe des TELNET-Client-Kommandos *encrypt* (siehe Seite 317) einstellen.

-H
on | **off** | **debug** | **key** <x-string 1..16>

on
Es wird ein Verschlüsselungsverfahren und ein Schlüssel ausgehandelt.

off
Es wird kein Verschlüsselungsverfahren und kein Schlüssel ausgehandelt.
off ist Voreinstellung.

debug
Der Encryption-Trace wird eingeschaltet.

key <x-string 1..16>
Encryption-Key für DES

Beachten Sie, dass nicht zwischen dem Schlüssel für Verschlüsselung und dem Schlüssel für die Entschlüsselung unterschieden wird. TELNET-Client und TELNET-Server verwenden denselben Schlüssel.

6.1.3.6 Option -X - Code-Tabellen wechseln

Mit dieser Option können die aktuell eingestellten Code-Tabellen, mit denen der TELNET-Client die Umsetzung von EBCDIC- auf ISO-Zeichen (erweiterter ASCII-Zeichensatz) durchführt, gewechselt werden. TCP-IP-AP nutzt dabei die Dienste von XHCS, d.h. es dürfen nur Code-Tabellen angegeben werden, die in XHCS als kompatibel eingetragen sind (siehe Handbuch „XHCS“). Bei einer Verbindung zwischen zwei BS2000-Systemen ist darauf zu achten, dass beide Systeme die gleichen Code-Tabellen nutzen.

Alternativ können Sie die Einstellungen zur Option *-X* auch mithilfe des TELNET-Client-Kommandos *setcode* (siehe Seite 328) vornehmen.

-X
<ebcdic-tabelle>:<iso-tabelle>

<ebcdic-tabelle> <iso-tabelle>
Zwischen EBCDIC- und ISO-Tabelle wird eine Codekonvertierungs-Tabelle mit XHCS erzeugt, die vom TELNET-Client für sämtliche Codekonvertierungen verwendet wird.

ACHTUNG!

Die Code-Tabellen dürfen keinesfalls in umgekehrter Reihenfolge eingegeben werden, weil dadurch falsche Konvertierungstabellen erzeugt werden und somit keine Dateiübertragung mehr möglich ist.

Der BS2000/OSD-TELNET-Server unterstützt nur einfache 7-bit-Terminals, daher ist das Kommando bei Verbindungen zum BS2000/OSD-TELNET-Server nicht sinnvoll.

Beispiel

```
-X EDF045:ISO88595
```

6.1.4 Kommandoübersicht

TELNET-Kommandos können bis zur Eindeutigkeit abgekürzt werden. Operanden werden durch Leerzeichen getrennt.

Kommando	Funktion	Seite
open	eröffnen der Verbindung zu einem fernen Rechner	321
close	schließen der Verbindung zum fernen Rechner	314
!	in den BS2000-/POSIX-Kommando-Modus wechseln	334
crmod	einfügen von Carriage Return ein-/ausschalten	315
escape	ändern des ESCAPE-Symbols	318
exit	Client-Exit ein-/ausschalten	319
setcode	wechseln der Code-Tabellen	328
^]	wechseln in den TELNET-Kommando-Modus	332
quit	beenden des TELNET-Clients	324
rexit	Server-Exits ein-/ausschalten	326
send	senden spezieller Kommandos *ao* / *ip* / *ayt* / *nop*	327

TELNET steuern

Kommando	Funktion	Seite
?	informieren über TELNET-Kommandos	333
help	wie ?, informieren über TELNET-Kommandos	320
status	ausgeben von TELNET-Status-Information	329
options	Optionen anzeigen/nicht anzeigen	323
trace	setzen des Socket-Trace-Level	331
debug	einschalten des TELNET-Trace	316

Information über TELNET

Kommando	Funktion	Seite
auth	AUTHENTICATION-Option aktivieren / deaktivieren	313
encrypt	ENCRYPTION-Option aktivieren / deaktivieren	317
readopt	Option-Datei lesen	325
tls	TLS-Unterstützung im TELNET-Client ein-/ausschalten	330

Information über TELNET

auth - AUTHENTICATION-Option aktivieren / deaktivieren

Mit *auth* können Sie die Unterstützung der AUTHENTICATION-Option aktivieren bzw. deaktivieren. Die Einstellungen des *auth*-Kommandos können Sie alternativ auch mithilfe der Option *-A* (siehe Seite 309) vornehmen.

In BS2000/OSD ist Authentifizierung derzeit nur für SSL realisiert. Deshalb können Sie die erforderlichen Einstellungen mithilfe der *-Z*-Options (siehe Seite 292 ff) vornehmen.

Die Einstellungen des *auth*-Kommandos werden erst bei der nächsten TELNET-Sitzung wirksam.

auth
disable | **enable** | **status**

disable
Deaktiviert die AUTHENTICATION-Option.

enable
Aktiviert die AUTHENTICATION-Option.

status
Zeigt den aktuellen Status der unterstützten Authentication-Typen an.

close - Schließen der Verbindung zum fernen Rechner

Mit dem Kommando *close* wird die Verbindung zum fernen Rechner geschlossen.

Die Verbindung zu einem fernen Rechner wird mit dem Kommando *open* eröffnet. Das Kommando *status* gibt den Namen des Rechners aus, zu dem eine Verbindung existiert.

close

Nach Ausführung des Kommandos *close* befinden Sie sich im TELNET-Kommando-Modus.

Beispiel

1. Die Abfrage des Status zeigt an, dass eine Verbindung zu *anlaged* besteht.

```
→ status
   Connected to anlaged.
   Escape character is '^]'.
   ISO-codetable is ISO88591, EBCDIC-codetable is EDF041.
```

2. Umschalten in den TELNET-Kommando-Modus

```
→ ^]
```

3. Verbindung zu *anlaged* wird geschlossen

```
→ close
   Connection closed
```

crmod - Einfügen von Carriage Return ein-/ausschalten

Werden vom fernen Rechner Nachrichten gesendet, die kein Carriage Return enthalten, so kann mit dem Kommando *crmod* (Einfügen von Carriage Return einschalten) der TELNET-Client veranlasst werden, einen Zeilenvorschub durchzuführen. Als Voreinstellung ist Einfügen von Carriage Return ausgeschaltet.

Unix-System sendet Carriage Return; wird das Einfügen von Carriage Return eingeschaltet, führt dies zu einer Verdoppelung des Zeilenvorschubs.

crmod

Beispiel

Der ferne Rechner ist ein Unix-Rechner. Verdoppelter Zeilenvorschub wird eingeschaltet.

```
→ crmod
   Will map carriage return on output
```

Anmerkung

Verdoppelter Zeilenvorschub wird eingeschaltet.

debug - DEBUG-Ausgaben ein-/ausschalten

Die DEBUG-Ausgaben dienen vor allem Netzadministratoren und Kundendienst-Mitarbeitern zur Diagnose von Problemen im Netz. Der Anwender benötigt normalerweise keine DEBUG-Ausgaben.

debug
<debug-wert>

<debug-wert>
Zulässig sind Werte zwischen 0 und 9.

0 keine Debug-Ausgaben

1 Ausgabe aller Nachrichten zwischen TELNET-Client und -Server sowie zwischen TELNET-Client und Terminal.

Wird kein Operand angegeben, dann wird die DEBUG-Ausgabe umgeschaltet, d.h. ist die DEBUG-Ausgabe eingeschaltet, so wird sie ausgeschaltet. Ist sie ausgeschaltet, so wird sie eingeschaltet. Angaben größer 1 werden wie 1 behandelt.

encrypt - ENCRYPTION-Option aktivieren / deaktivieren

Mit *encrypt* können Sie die Unterstützung der ENCRYPTION-Option aktivieren bzw. deaktivieren, die zum Aushandeln des Verschlüsselungsverfahrens sowie des verwendeten Schlüssels verwendet wird. Derzeit wird in TELNET nur DES64 in den Varianten DES_CFB64 und DES_OFB64 unterstützt. *encrypt* ist nur wirksam, wenn nicht gleichzeitig die Options *-Z tls-required* oder *-A on* gelten.

Da ausschließlich Encryption-Routinen aus der OpenSSL-Bibliothek verwendet werden, können Sie diese Bibliothek − sofern ihr Name vom Standard (SYSLNK.TCP-IP-AP.052) abweicht − mithilfe der Option *-Z OpenSSLlibname* (siehe Seite 307) spezifizieren.

Die Einstellungen des *encrypt*-Kommandos können Sie alternativ auch mithilfe der Option *-H* (siehe Seite 310) vornehmen.

Die Einstellungen des *encrypt*-Kommandos werden erst bei der nächsten TELNET-Sitzung wirksam.

encrypt
disable | **enable** | **key** <x-string 1..16> | **status**

disable
Deaktiviert die Encryption für beide Richtungen (Input/Output).

enable
Aktiviert die Encryption für beide Richtungen (Input/Output).

key <x-string 1..16>
Encryption-Key für DES

status
Zeigt die aktuellen Einstellungen für die Encryption-Option an. Die Anzeige umfasst im Wesentlichen die unterstützten Verschlüsselungsalgorithmen.

escape - ESCAPE-Symbol ändern

Bei Eingabe des ESCAPE-Symbols findet ein Wechsel vom Eingabe-Modus in den TELNET-Kommando-Modus statt.

Das ESCAPE-Symbol besteht aus zwei Zeichen. Das erste Zeichen ist ^, das zweite Zeichen ist vom Benutzer mit dem Kommando *escape* frei wählbar.

Der Wechsel vom Eingabe-Modus in den TELNET-Kommando-Modus ist beim Kommando ^*]* auf Seite 332 beschrieben.

escape

exit - Client-Exit ein-/ausschalten

Exit-Routinen für den TELNET-Client stellen Sie mit dem Kommando *exit* ein.

Eine genauere Beschreibung des Exit-Mechanismus in TELNET und des Kommandos *exit* finden Sie im Handbuch „interNet Services Administratorhandbuch“.

exit
[receive : < selector 1 >] [send : < selector 2 >]

help - Über TELNET-Kommandos informieren

Über ein bestimmtes TELNET-Kommando oder über alle TELNET-Kommandos wird informiert.

help
[<kommando>]

<kommando>
TELNET-Kommando, über das Information gewünscht wird. Fehlt dieser Operand, werden alle zulässigen TELNET-Kommandos aufgelistet.

Beispiel

```
telnet>  help
Commands may be abbreviated.  Commands are:

open            connect to a site
close           close current connection
quit            exit telnet
!               BS2000 MCLP
escape          set escape character
status          print status information
options         toggle viewing of options processing
crmod           toggle mapping of received carriage returns
trace           set socket trace level
help            print help information
setcode         change the codetables
send            send special commands ao/ip/ayt/nop
exit            set local exit
rexit           set remote exit
debug           set telnet trace
?               print help information
tls             switch on/off START-TLS Option
readopt         read Option File
auth            switch on/off AUTHENTICATION Option
encrypt         switch on/off ENCRYPTION Option
telnet>
```

open - Eröffnen der Verbindung zu einem fernen Rechner

Mit dem Kommando *open* wird eine Verbindung zu einem fernen Rechner eröffnet. Es muss entweder der Name oder die Internet-Adresse dieses Rechners bekannt sein. Der Rechner muss entweder zum lokalen Netz gehören oder über einen Gateway-Rechner erreichbar sein. Namen und Adressen der mit *open* erreichbaren Rechner können vom Netzverwalter erfragt werden.

Bei einem fernen BS2000/OSD-Rechner können Sie sich nun mit LOGON einloggen. Wenn Sie das LOGON-Kommando ohne Passwort eingeben, fordert BS2000/OSD mit der Meldung JMS0151 das Passwort nachträglich an. In TELNET ist das Eingabefeld für das Passwort dunkelgesteuert.

Nach erfolgreichem Verbindungsaufbau können zeilenorientierte Kommandos des fernen Betriebssystems eingegeben werden (man befindet sich im Eingabe-Modus). Durch Eingabe des jeweiligen Kommandos zur Beendigung des laufenden Prozesses im fernen Betriebssystem wird die Verbindung abgebaut.

open
<ipadr> | <remotehost> | localhost loopback [<port>]

<ipadr>
: Internet-Adresse (IPv4- oder IPv6-Adresse) des fernen Rechners, zu dem die Verbindung aufgebaut werden soll:

  – Eine IPv4-Adresse muss in der üblichen „decimal-dotted“-Notation angegeben werden.

  – Eine IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

<remote-host>
: Symbolischer Name des fernen Rechners, zu dem die Verbindung aufgebaut werden soll

localhost
: Für *localhost* ist eine eigene Internet-Adresse generiert, die nicht mit der Adresse des eigenen Rechners im lokalen Netz identisch ist. Diese Adresse ist nur dem lokalen Rechner bekannt; andere Rechner können sie nicht verwenden.

loopback
: *loopback* steht für die Internet-Adresse, mit der der lokale Rechner tatsächlich im LAN erreichbar ist.

<port>
: Port-Nummer des TELNET-Servers. Der TELNET-Server hat standardmäßig die Port-Nummer 23.

Beispiel

Der ferne Rechner ist ein Unix-Rechner.

1. Die Verbindung zum Unix-Rechner *anlaged* wird eröffnet.

```
→ open anlaged
   Trying...
   Connected to anlaged.
   Escape character is '^]'.
   ISO-codetable is ISO88591, EBCDIC-codetable is EDF041.
   local exits defined:  receive: -, send: -
   remote exits defined: receive: -, send: -
   Host name: anlaged
```

2. Login auf der Kennung *gast*

```
→ gast
   gast
   Continue by entering RETURN or MENUE
```

options - Optionen-Anzeige einschalten/ausschalten

Zwischen TELNET-Client und TELNET-Server werden eine Reihe von Optionen ausgehandelt (z.B. beim Verbindungsaufbau). Im Zustand „eingeschaltet“ werden diese TELNET-Optionen am Bildschirm ausgegeben. Optionen, die vom lokalen Client gesendet werden, sind durch SENT gekennzeichnet; Optionen, die vom lokalen Client empfangen werden, sind durch RCVD gekennzeichnet.

Als Voreinstellung werden Optionen nicht angezeigt.

options

Beispiel

1. Kommando *option* mit Rückmeldung

```
→ options
   Will show option processing.
```

2. Kommando *open* mit Ausgabe der Optionen

   Die Option ECHO wird zwischen Sender und Empfänger ausgehandelt.

```
→ open anlaged
   Trying...
   Connected to anlaged.
   Escape character is '^]'.
   ISO-codetable is ISO88591, EBCDIC-codetable is EDF041.
   RCVD will ECHO (reply) SENT dont ECHO (reply)
   RCVD wont ECHO (don't reply)
   Host name: anlaged
```

quit - TELNET beenden

Das Kommando *quit* beendet das Programm TELNET. Falls noch eine Verbindung zu einem fernen Rechner existiert, wird diese geschlossen (implizierter *close*).

quit

Beispiel

```
→ quit
   Connection closed.
   Verbrauchte CPU-Zeit 4.7232 Sekunden
```

readopt - Option-Datei einlesen

Mit dem Kommando *readopt* wird die Option-Datei eingelesen. Das Kommando ergänzt/ersetzt die Optionen, die beim Start des TELNET-Client aus globaler und lokaler Option-Datei gelesen wurden (sofern diese Dateien zum Start-Zeitpunkt existierten). Das Kommando *readopt* können Sie beliebig oft ausführen, um z.B. Option-Sätze zusammenzusetzen, die auf verschiedene Dateien verteilt sind, oder um zwischen unterschiedlichen Option-Sätzen zu wechseln.

Die Option *-Z OpenSSLlibname* (siehe Seite 307) wird erst nach einem Neustart des TELNET-Client wirksam. Alle anderen Optionen werden nach dem Aufbau der Verbindung zum TELNET-Server wirksam.

readopt
<option-datei>

<option-datei 1 .. 54>
Name der Option-Datei

rexit - Server-Exits ein-/ausschalten

Im Client haben Sie die Möglichkeit, TELNET-Server-Exits mit dem Kommando *rexit* (remote exit) einzustellen. Dies betrifft allerdings nur die Exits für *send* und *receive* und wirkt sich nur auf die Verbindung dieses speziellen Clients zum Server aus.

Eine genauere Beschreibung des exit-Mechanismus in TELNET und des Kommandos *exit* finden Sie im „interNet Services Administratorhandbuch“.

rexit
[receive : < selector 1>] [send : < selector 2>]

send - Sende spezielle Kommandos

send
ao | ip | ayt | nop | ?

ao
: unterbricht den aktuell auf dem fernen Rechner laufenden Prozess (interrupt process).

ip
: leert die gesamte Ausgabe des fernen Rechners und gibt sie auf dem Terminal des Benutzers aus (abort output).

ayt
: Überprüfung, ob der Server noch „lebt“. Im Positiv-Fall antwortet der Server mit *YES* (are you there).

nop
: sendet *No operation* an den Server.

?
: gibt Hilfsinformation zum Kommando *send*.

setcode - Code-Tabellen wechseln

Mit *setcode* können die aktuell eingestellten Code-Tabellen, mit denen der TELNET-Client die Umsetzung von EBCDIC- auf ISO-Zeichen (erweiterter ASCII-Zeichensatz) durchführt, gewechselt werden. TCP-IP-AP nutzt dabei die Dienste von XHCS, d.h. es dürfen nur Code-Tabellen angegeben werden, die in XHCS als kompatibel eingetragen sind (siehe Handbuch „XHCS“). Bei einer Verbindung zwischen zwei BS2000-Systemen ist darauf zu achten, dass beide Systeme die gleichen Code-Tabellen nutzen.

Die Einstellungen des *setcode*-Kommandos können Sie alternativ auch mithilfe der Option *-X* (siehe Seite 311) vornehmen.

setcode
<ebcdic-tabelle> <iso-tabelle>

<ebcdic-tabelle> <iso-tabelle>

Zwischen EBCDIC- und ISO-Tabelle wird eine Codekonvertierungs-Tabelle mit XHCS erzeugt, die vom TELNET-Client für sämtliche Codekonvertierungen verwendet wird.

ACHTUNG!

Die Code-Tabellen dürfen keinesfalls in umgekehrter Reihenfolge eingegeben werden, weil dadurch falsche Konvertierungstabellen erzeugt werden und somit keine Dateiübertragung mehr möglich ist.

Beispiel

```
setcode EDF045 ISO88595
```

Anmerkungen

Der BS2000-TELNET-Server unterstützt nur einfache 7-bit-Terminals, daher ist das Kommando bei Verbindungen zum BS2000-TELNET-Server nicht sinnvoll.

status - Ausgabe von TELNET-Status-Informationen

Die Status-Meldung informiert,

- ob und zu welchem Rechner eine Verbindung besteht,
- wie das aktuelle ESCAPE-Symbol aussieht.

Das Kommando *escape* legt das ESCAPE-Symbol fest.

status

Beispiel

Es werden die momentan eingestellten Parameter des TELNET-Client angezeigt.

```
→ status
   No connection.
   Escape character is'‘^]‘.
   ISO-codetable is ISO88591 , EBCDIC-codetable is EDF041
   local exits defined:  receive: -, send: -
   remote exits defined: receive: -, send: -
   TLS off
   Authentication off
   Encryption off
   Decryption off

→ status
   Connected to loopback.
   Escape character is'‘^]‘.
   ISO-codetable is ISO88591 , EBCDIC-codetable is EDF041
   local exits defined:  receive: -, send: -
   remote exits defined: receive: -, send: -
   TLS off
   Authentication off
   Encryption off
   Decryption off
```

tls - TLS-Unterstützung im TELNET-Client ein-/aussschalten

Mit *tls* können Sie TLS-Absicherung im TELNET-Client ein- und ausschalten. Bei eingeschalteter TLS-Unterstützung übernimmt SSL das Aushandeln der Authentifizierungsmodalitäten, so dass TELNET nicht dadurch belastet ist.

Die Einstellungen des *tls*-Kommandos können Sie alternativ auch mithilfe der START-TLS-Option (*-Z tls-required*, siehe Seite 292 und Seite 293) vornehmen. Die für den TLS/SSL-Betrieb erforderlichen Einstellungen können Sie mithilfe der *-Z*-Options vornehmen. Die *-Z*-Options sind ab Seite 294 beschrieben.

Die Einstellungen des *tls*-Kommandos werden erst bei der nächsten TELNET-Sitzung wirksam.

tls
[**on** | **off** | **status**]

on
Schaltet die Unterstützung der START-TLS-Option im TELNET-Client ein.

off
Schaltet die Unterstützung der START-TLS-Option im TELNET-Client aus.

status
Zeigt den aktuellen Status der TLS-Unterstützung an.

tls-Kommando ohne Operanden spezifiziert
Schaltet die Unterstützung der START-TLS-Option im TELNET-Client ein.

Wenn die TLS-Absicherung nicht eingeschaltet ist, werden außer eventuell der Option *-Z OpenSSLlibName* (siehe Seite 307) alle anderen *-Z*-Options ignoriert.

trace - SOCKET-Trace-Ausgaben ein-/ausschalten

Die SOCKET-Trace-Ausgaben dienen vor allem dem Netzadministrator und dem Kundendienst zur Diagnose von Netzproblemen. Der normale Anwender benötigt in der Regel keine SOCKET-Trace-Ausgaben.

Sind die SOCKET-Trace-Ausgaben eingeschaltet, gibt TELNET verschiedene Diagnose-Informationen am Bildschirm aus.

trace
[<trace-wert>]

<trace-wert>
Zulässig sind Werte zwischen 0 und 9. Je größer der eingestellte Wert ist, desto mehr Informationen werden ausgegeben.

0 Keine TRACE-Ausgaben (TRACE-Ausgaben ausschalten).

1 - 9 Ausgabe von Informationen der SOCKET-Traces.

Wird kein Operand angegeben, wird die TRACE-Ausgabe umgeschaltet, d.h. ist die TRACE-Ausgabe eingeschaltet, so wird sie ausgeschaltet. Ist sie ausgeschaltet, wird sie eingeschaltet (*trace-wert*=1). Ein Wert größer als 1 impliziert alle TRACE-Ausgaben mit kleinerem Level.

^] - In den TELNET-Kommando-Modus wechseln

Durch Eingabe des ESCAPE-Symbols wird vom Eingabe- in den TELNET-Kommando-Modus gewechselt.

^]

^] ist die Vorbelegung für das ESCAPE-Symbol. Nach Ausführung eines korrekten TELNET-Kommandos wird im Regelfall in den Eingabe-Modus zurückgekehrt. Bei fehlerhaftem Kommando bleibt der TELNET-Kommando-Modus erhalten.

Mit dem Kommando *escape* kann statt] ein anderes Zeichen als zweites Zeichen des ESCAPE-Symbols gewählt werden. Nach Ausführung des Kommandos *?* bleibt der TELNET-Kommando-Modus erhalten. Durch das Kommando *!* wird in den BS2000/OSD-Kommando-Modus bzw. in den POSIX-Kommando-Modus gewechselt. Nach Ausführung des Kommandos *close* bleibt der TELNET-Kommando-Modus erhalten.

? - Über TELNET-Kommandos informieren

Über ein bestimmtes TELNET-Kommando oder über alle TELNET-Kommandos wird informiert.

?
[<kommando>]

Siehe Kommando *help* auf Seite 320.

! - In den BS2000/OSD- bzw. POSIX-Kommando-Modus wechseln

In den BS2000/OSD-Kommando-Modus wechseln (TELNET-Client im BS2000/OSD)

Das Kommando *!* ermöglicht die Eingabe von BS2000/OSD-Kommandos. Wird kein Operand angegeben, dann wird in den BS2000/OSD-Kommando-Modus gewechselt, aus dem man mit dem BS2000/OSD-Kommando *RESUME* wieder in TELNET zurückgelangt. Bei Eingabe von *!* mit einem BS2000/OSD-Kommando als Operand befindet man sich nach Ausführung des Kommandos wieder in TELNET.

!
[<bs2000-kommando>]

<bs2000-kommando>
Beliebiges zeilenorientiertes BS2000/OSD-Kommando.
Die BS2000/OSD-Kommandos *LOAD*, *EXEC*, *LOGOFF* und *ABEND* dürfen weder direkt noch indirekt eingegeben werden, weil sie TELNET bzw. die Task, in der TELNET abläuft, beenden.

Beispiel

1. Wechsel in den BS2000/OSD-Kommando-Modus

```
→ !
% PROGRAM BREAK AT 0013E8, AMODE = 24
```

2. Eingabe des BS2000-Kommandos *STATUS PROG*

```
→ STA P
   NAME     TSN TYPE          SIZE CURR-CMD  PROGRAM-NAME
   PETER    2726 3 DIALOG     79 STA         :5:$TSOS.TELNET
   BRONCO   2716 3 DIALOG     87 EXECUTE     :5:$TSOS.FTP
```

3. Rückkehr in den TELNET-Client

```
→ R
```

In den POSIX-Kommando-Modus wechseln (TELNET-Client in POSIX)

Das Kommando *!* ermöglicht die Eingabe von POSIX-Kommandos. Wird kein Operand angegeben, dann wird in den POSIX-Kommando-Modus gewechselt, aus dem man mit dem POSIX-Kommando *exit* wieder in TELNET zurückgelangt. Bei Eingabe von *!* mit einem POSIX-Kommando als Operand befindet man sich nach Ausführung des Kommandos wieder in TELNET.

!
[<posix-kommando>]

<posix-kommando>
Beliebiges zeilenorientiertes POSIX-Kommando.

6.2 TELNET-Server

Der TELNET- Server in BS2000/OSD nimmt Verbindungswünsche von TELNET-Clients am lokalen Netz entgegen und vermittelt sie an *$DIALOG* weiter.

Der TELNET-Server ist im Adminsitratorhandbuch zu InterNet Services beschreiben.

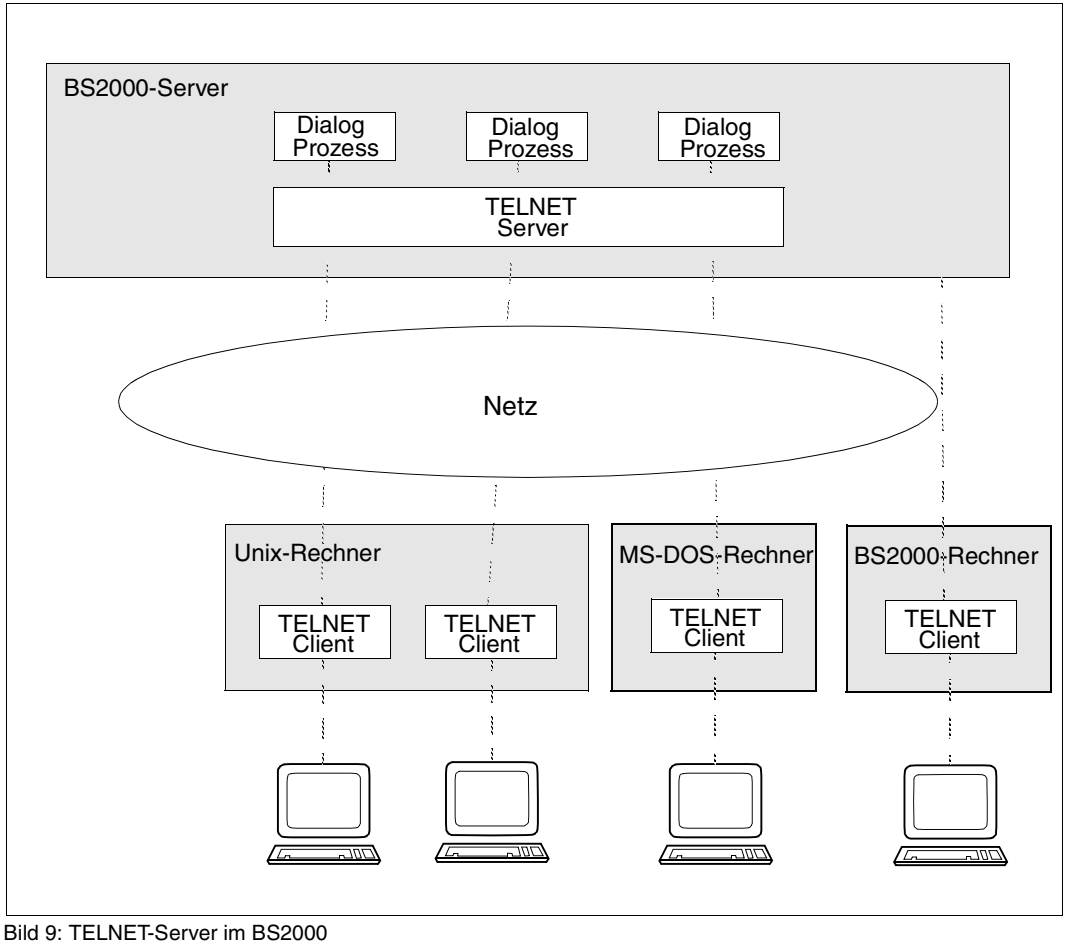


Bild 9: TELNET-Server im BS2000

Arbeitsweise

Der Server nimmt Verbindungswünsche entgegen und baut eine DCAM-Verbindung zu *$DIALOG* auf. Gegenüber *$DIALOG* wird eine einfache Schreibstation simuliert, d.h. es ist nur zeilenweiser Dialog möglich. Es findet keine Überlaufkontrolle statt.

7 OpenSSH

Dieses Kapitel basiert auf den Manual Pages (Man Pages) von OpenSSH und beschreibt den OpenSSH Client. Der OpenSSH Server ist im Handbuch „interNet Services Administratorhandbuch“ beschrieben.

Die hier vorliegende Beschreibung ist auf die BS2000/OSD-relevanten Teile gekürzt. An entsprechenden Stellen des Kapitels, wie z.B. bei der Beschreibung von Options, wird auf die Man Pages von OpenSSH verwiesen, da nur dort die aktuellsten Versionen beschrieben sind. Sie finden die Man Pages von OpenSSH im Internet unter *http://www.openssh.org/manual.html* oder nach der Installation der Komponente OpenSSH auf Ihrem Server

- unter <*installationspfad*>*/readme/TCP-IP-SV.openssh/html/* als HTML-Datei,
- unter <*installationspfad*>*/readme/TCP-IP-SV.openssh/pdf/* als PDF-Datei,
- unter <*installationspfad*>*/readme/TCP-IP-SV.openssh/text/* als Text-Datei.

Der Standard-Installationspfad lautet: */opt/TCP-IP-SV/openssh*

Wenn im weiteren Verlauf dieses Kapitel wiederholt auf die „Man Pages von OpenSSH“ verwiesen wird, sind im Einzelnen diese Quellen gemeint. Dabei sind die mit dem Produkt ausgelieferten Man Pages zu bevorzugen, da diese Man Pages die BS2000/OSD-spezifischen Anpassungen enthalten (geänderte Pfadnamen, erweiterte Funktionalität etc.).

SSH (**S**ecure **Sh**ell) ist ein kryptographisches Protokoll für die Durchführung der folgenden Aufgaben:

- Login auf einen fernen Rechner
- interaktive / nicht interaktive Kommandoausführung auf einem fernen Rechner
- File Transfer zwischen verschiedenen Rechnern eines Netzes

SSH bezeichnet nicht nur das Protokoll selbst, sondern auch die konkreten Implementierungen.

Ausführliche Informationen zum Konzept von OpenSSH finden Sie im Handbuch „interNet Services Administratorhandbuch“.

7.1 Bestandteile der OpenSSH Protokoll-Suite

Die OpenSSH Protokoll-Suite umfasst folgende Programme bzw. Kommandos:

- Auf der Server-Seite: Server-Programm *sshd* (siehe Handbuch „interNet Services Administratorhandbuch“)
- Auf der Client-Seite:
  - Client-Programm *ssh* bzw. *slogin* (siehe Seite 339): ersetzt *rlogin*, *rsh* und *telnet*.
  - *scp* (siehe Seite 351): ersetzt *rcp*.
  - *sftp* (siehe Seite 352): ersetzt *ftp*.
- Administrations-Utilities:
  - *ssh-agent* (siehe Seite 357)
  - *ssh-add* (siehe Seite 359)
  - *ssh-keygen* (siehe Seite 361)
  - *ssh-keyscan* (siehe Seite 363)

Für die sichere Kommunikation via SSH von Windows-Systemen mit BS2000/OSD stellt Ihnen z.B. die Open Source Stoftware *PuTTY* (siehe *http://www.chiark.greenend.org.uk/~sgtatham/putty/*) die Funktionalität der Client-Seite von OpenSSH zur Verfügung.

7.2 OpenSSH Client-Anwendung ssh (slogin)

Mit der OpenSSH Client-Anwendung *ssh* können Sie

- sich auf einem fernen Rechner einloggen,
- Kommados auf einem fernen Rechner ausführen.

ssh ist die sichere Alternative zu *rlogin, rsh* und *telnet* und sollte anstelle dieser Programme verwendet werden. Im Gegensatz zu *rlogin* und *rsh* gewährleistet der OpenSSH Client sichere, verschlüsselte Kommunikation über ein unsicheres Netz.

7.2.1 OpenSSH Client konfigurieren

Der OpenSSH Client *ssh* liest seine Konfigurations-Options der Reihe nach aus folgenden Quellen:

1. Kommandozeilen-Argumente, die Sie beim Aufruf von *ssh* (siehe Seite 341), *scp* (siehe Seite 351) und *sftp* (siehe Seite 352) angeben.
2. benutzerspezifischen Konfigurationsdatei (*$HOME/.ssh/config*)

   Obwohl diese Datei in der Regel keine sicherheitskritischen Informationen enthält, sollte Lese-/Schreibberechtigung lediglich für den Eigentümer bestehen. Allen anderen Benutzern sollte der Zugriff verweigert werden.
3. systemweite Konfigurationsdatei (*/etc/ssh/ssh_config*)

   Diese Datei enthält Default-Werte für Konfigurationsparameter,

   – wenn keine benutzerspezifische Konfigurationsdatei existiert oder

   – wenn die entsprechenden Parameter in der benutzerspezifischen Konfigurationsdatei nicht spezifiziert sind.

Für jede Option gilt der zuerst gefundene Wert.

Syntax und Semantik der Konfigurationsdateien von ssh

Die Konfigurationsdateien von *ssh* müssen wie folgt aufgebaut sein:

- Die Konfigurationsdatei ist in einen oder mehrere logische Abschnitte unterteilt. Jeder Abschnitt wird durch eine `Host`-Option eingeleitet, auf die weitere, von der `Host`-Option verschiedene Konfigurations-Options folgen. Die nächste `Host`-Option markiert dann den Anfang des nächsten Abschnitts etc. Die Konfigurations-Options eines Abschnitts sind nur für Rechner relevant, deren Namen in der zugehörigen `Host`-Option spezifiziert sind.
- Die Konfigurationsdatei enthält pro Zeile: optionales Leerzeichen, gefolgt von Schlüsselwort und zugehörigem Argument bzw. zugehöriger Argumentenliste.

  Schlüsselwort und Argument(enliste) können getrennt sein durch:

  – Leerzeichen

  – optionales Leerzeichen und genau ein „=“

  Bei Schlüsselwörtern wird nicht zwischen Groß- und Kleinschreibung unterschieden. Bei Argumenten muss Groß-/Kleinschreibung beachtet werden, sogar „yes“ und „no“ müssen in Kleinbuchstaben angegeben werden.

  – Leere Zeilen sowie Zeilen, die mit „#“ beginnen, werden als Kommentar interpretiert.

  Die ausführliche Beschreibung der Konfigurations-Options finden Sie auf den Man Pages von OpenSSH.

7.2.2 OpenSSH Client starten

ssh starten Sie mit dem folgenden Kommando:

```
ssh [-1246AaCfgkMNnqsTtVvXxY] [-b <bind_address>] [-c <cipher_spec>]
    [-D <port>] [-e <escape_char>] [-F <configfile>] [-i <identity_file>]
    [-L <port>:<host>:<hostport>] [-l <login_name>] [-m <mac_spec>]
    [-o <option>] [-p <port>] [-R <port>:<host>:<hostport>] [-S <ctl>]
    [<user>@]<hostname> [<command>]
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Das *ssh* Client-Programm können Sie nur innerhalb einer Remote-Login-Session starten, d.h. Sie müssen in POSIX per *rlogin*, *ssh*, oder *slogin* eingeloggt sein. In $DIALOG oder TELNET-Sitzungen können Sie *ssh* nur benutzen, wenn *stdin*, *stdout* und *stderr* auf eine Datei oder eine POSIX-Pipe umgelenkt wurden.

ssh stellt die Verbindung zum Rechner `hostname` her und meldet sich dort unter dem (optionalen) Benutzernamen `user` an. Der Benutzer muss seine Identität dem fernen Rechner gegenüber nachweisen. Näheres hierzu finden Sie im nachfolgenden Abschnitt „Authentifizierung zwischen OpenSSH Client und Server“ auf Seite 342. Wenn `command` spezifiziert ist, wird dieses Kommando in der remote Shell anstatt in der Login Shell des Benutzers ausgeführt.

Returnwert

Bei Erfolg: Exit Status des remote ausgeführten Kommandos

Bei Fehler: 255

7.2.3 Authentifizierung zwischen OpenSSH Client und Server

Je nachdem, ob SSH-Protokoll Version 1 oder Version 2 eingesetzt wird, werden bei der Authentifizierung des OpenSSH Client gegenüber dem OpenSSH Server unterschiedliche Verfahren angewendet.

Client-Authentifizierung in der SSH-Protokoll Version 1

SSH-Protokoll Version1 unterstützt folgende Verfahren für die Client-Authentifizierung, wobei je nach Konfiguration einige Verfahren gezielt zugelassen oder ausgeschlossen werden können:

1. rhosts-Authentifizierung
2. rhosts-RSA-Authentifizierung
3. RSA-basierte Authentifizierung
4. Passwort-Authentifizierung

Die in der konkreten Konfiguration verfügbaren Verfahren werden der Reihe nach angewendet, bis ein Verfahren erfolgreich authentifiziert hat bzw. bis alle Verfahren gescheitert sind. Die Authentifizierungsverfahren sind nachfolgend im Einzelnen erläutert.

1. rhosts-Authentifizierung

Die rhosts-Authentifizierung entspricht der Authentifizierung bei den r-Utilities *rlogin* und *rsh*. *Ssh* verwendet dabei (anders als beim *rlogin*-Kommando) DNS-Hostnamen, nicht BCAM-Hostnamen. Die DNS-Hostnamen werden dabei durch Reverse-Lookup aus der IP-Adresse bestimmt.

ACHTUNG!

Die rhosts-Authentifizierung ist äußerst unsicher und sollte vom Administrator unbedingt deaktiviert werden. In der Standard-Konfiguration des OpenSSH Servers ist die rhosts-Authentifizierung deshalb bereits deaktiviert.

Eine erfolgreiche rhosts-Authentifizierung benötigt lediglich einen entsprechenden Eintrag des Rechners, von dem aus sich der Benutzer anmeldet, in einer der folgenden Dateien:

- */etc/hosts.equiv* bzw. */etc/ssh/shosts.equiv*

  Beide Dateien sind syntaktisch identisch aufgebaut und werden in der angegebenen Reihenfolge durchsucht.

  Existiert für den Rechner, von dem aus sich der Benutzer anmeldet, ein Eintrag in einer dieser Dateien auf dem fernen Rechner und stimmen außerdem die Benutzernamen auf beiden Seiten überein, dann darf sich der Benutzer unmittelbar einloggen.

- *$HOME/.rhosts* bzw. *$HOME/.shosts*

  Beide Dateien sind syntaktisch identisch aufgebaut und werden in der angegebenen Reihenfolge durchsucht.

  Existiert eine der beiden Dateien im $HOME-Verzeichnis des Benutzers auf dem fernen Rechner und enthält diese Datei einen Eintrag mit dem Namen des Client-Rechners sowie dem Benutzernamen auf diesem Rechner, dann darf sich der Benutzer unmittelbar einloggen.

2. **rhosts-Authentifizierung in Verbindung mit RSA-basierter Authentifizierung**

   Bei dieser Methode wird die rhosts-Authentifizierung durch eine RSA-basierte Authentifizierung des Client-Rechners ergänzt. Die bekannten öffentlichen Schlüssel des Client-Rechners sind in den Dateien */etc/ssh/known_hosts* und *$HOME/.ssh/known_hosts* des OpenSSH Servers *sshd* abgelegt.

   Die rhosts-Authentifizierung in Verbindung mit RSA-basierter Authentifizierung verhindert IP- und DNS-Spoofing.

3. **RSA-basierte Authentifizierung**

   Bei der RSA-basierten Authentifizierung erzeugt jeder Benutzer für Authentifizierungszwecke ein Paar bestehend aus öffentlichem und privatem Schlüssel. Der OpenSSH Server kennt den öffentlichen Schlüssel des Benutzers, wenn dieser Schlüssel in den Server-Dateien */etc/ssh/known_hosts* und *$HOME/.ssh/known_hosts* hinterlegt ist. Nur der Benutzer selbst kennt seinen eigenen privaten Schlüssel.

   ssh wickelt das RSA-Authentifizierungsprotokoll automatisch ab. Der Benutzer generiert das Schlüsselpaar durch Aufruf des OpenSSH Basistools *ssh-keygen* (siehe Seite 361). Am komfortabelsten lässt sich RSA-Authentifizierung mithilfe des Authentifizierungsagenten *ssh-agent* (siehe Seite 357) durchführen.

4. **Passwort-Authentifizierung**

   Wenn die anderen Authentifizierungsmethoden versagen, fordert *ssh* ein Benutzer-Passwort an und sendet dieses verschlüsselt zur Prüfung an den fernen Rechner.

Wenn die Authentifizierung mit keiner der genannten Methoden erfolgreich ist, wird der Verbindungswunsch des OpenSSH Client abgewiesen.

SSH-Protokoll Version 1 unterstützt keine Mechanismen, die eine starke Integrität der Verbindung sicherstellen.

Client-Authentifizierung in der SSH-Protokoll Version 2

SSH-Protokoll Version 2 unterstützt ähnliche Authentifizierungsmethoden wie SSH-Protokoll Version 1. Bei Verwendung des Default-Wertes für die Option *PreferredAuthentication* in der Server-Konfigurationsdatei *sshd_config* (siehe Handbuch „interNet Services Administratorhandbuch“) führt der OpenSSH Client der Reihe nach folgende Authentifizierungsmethoden durch:

1. rhosts-Authentifizierung
2. Public Key-Authentifizierung
3. Passwort-Authentifizierung

Die Verfahren werden der Reihe nach angewendet, bis ein Verfahren erfolgreich authentifiziert hat bzw. bis alle Verfahren gescheitert sind.

Die Public Key-Authentifizierung ähnelt der RSA-Authentifizierung und gestattet die Verwendung von RSA- und DSA-Algorithmen. Der OpenSSH Client signiert die Session-Id mit seinem privaten Schlüssel (*$HOME/.ssh/id_dsa* oder *$HOME/.ssh/id_rsa*) und sendet das Ergebnis an den OpenSSH Server. Der Server prüft, ob in seiner Datei *$HOME/.ssh/authorized_keys* der passende öffentliche Schlüssel enthalten ist. Wenn beide Schlüssel vorhanden sind und die Signatur korrekt ist, nimmt der Server die Verbindung an.

SSH-Protokoll Version 2 unterstützt zusätzliche Mechanismen, die Vertraulichkeit und Integrität der Verbindung sicherstellen:

- Vertraulichkeit ist gewährleistet durch Verschlüsselung des Datenverkehrs mithilfe von 3DES, Blowfish, CAST128 oder Arcfour.
- Integrität ist gewährleistet durch Verschlüsselung des Datenverkehrs mit hmac-md5 oder hmac-sha1.

Server-Authentifizierung

Der OpenSSH Client *ssh* pflegt und prüft automatisch eine Datenbank, in der die Host-Ids aller Rechner gespeichert sind, zu denen *ssh* jemals Kontakt hatte. Host-Keys werden benutzer-spezifisch in der Datei *$HOME/.ssh/known_hosts* abgelegt. Zusätzlich prüft *ssh* die Datei */etc/ssh/ssh_known_hosts* auf bekannte Rechner. Jeder neue Rechner wird nach Bestätigung durch den Benutzer automatisch in der benutzer-spezifischen Datei gespeichert.

Sobald sich die Host-Id eines Rechners ändert, meldet *ssh* dies und sperrt die vorhandenen Passwörter. Auf diese Weise wird verhindert, dass ein Trojanisches Pferd die Passwörter der Benutzer abrufen kann. Außerdem können so „man-in-the-middle“-Angriffe abgewehrt werden, die sonst die Verschlüsselung umgehen könnten.

Die Option *StrictHostKeyChecking* in der Konfigurationsdatei *ssh_config* wird verwendet, um das Einloggen in Rechner zu unterbinden, deren Host-Key unbekannt ist oder verändert wurde.

Eine ausführliche Beschreibung der Option *StrictHostKeyChecking* finden Sie auf den Man Pages von OpenSSH.

7.2.4 Kommandoausführung und Datenweiterleitung (Data Forwarding)

Sobald sich der OpenSSH Client (*ssh*) erfolgreich authentifiziert hat, beginnt ein Dialog zur Vorbereitung der OpenSSH Session. Zu diesem Zeitpunkt kann der Client z.B. folgende Aktionen anfordern:

- Einrichtung einer Pseudo-tty
- Forwarding von TCP/IP-Verbindungen
- Forwarding der Verbindung zu einem Authentifizierungsagenten (*ssh-agent*, siehe Seite 357) über einen sicheren Kanal

Schließlich fordert *ssh* entweder eine Shell oder die Ausführung eines Kommandos an. Client und Server befinden sich im Session-Modus. In diesem Modus kann jede Seite jederzeit Daten senden. Diese Daten werden dann wie folgt weitergeleitet:

- vom Benutzerterminal auf der Client-Seite an die Shell / das Kommando auf der Server-Seite bzw.
- von der Shell / vom Kommando auf der Server-Seite an das Benutzerterminal auf der Client-Seite.

7.2.5 Login Session und Kommando-Ausführung auf einem fernen Rechner

Sobald der OpenSSH Server die Identität des Benutzers akzeptiert hat (siehe hierzu auch Kapitel OpenSSH Abschnitt „Login-Prozess“ im Handbuch „interNet Services Administratorhandbuch“), führt der Server entweder das aktuelle Kommando auf dem fernen Rechner aus oder er stellt dem Benutzer die normale Shell des fernen Rechners zur Verfügung. Dabei wird die gesamte Kommunikation mit dem remote ausgeführten Kommando bzw. mit der remote Shell automatisch verschlüsselt.

Folgende Fälle sind zu unterscheiden:

- Wenn kein Pseudo-Terminal (Pseudo tty) zugewiesen wurde, ist die Session transparent und kann für den sicheren Transfer binärer Daten verwendet werden.
- Wenn ein Pseudo-Terminal (Pseudo tty) zugewiesen wurde (normales Login), kann der Benutzer Escape Characters verwenden (siehe nachfolgender Abschnitt „Escape Characters“).

Auf den meisten Systemen führt das Deaktivieren der Escape Characters zu einer transparenten Session, selbst wenn ein Pseudo-tty verwendet wird. (Die Verwendung von Escape Characters wird mit der Option *EscapeChar* in der Client-Konfigurationsdatei *ssh_config* gesteuert.)

Die Session wird beendet, wenn die Kommandoausführung oder die Shell auf dem fernen Rechner beendet werden und alle TCP/IP-Verbindungen beendet sind. Der Exit Status des remote ausgeführten Programms wird als Exit Status von *ssh* zurückgeliefert.

7.2.6 Escape Characters

Wenn ein Pseudo tty angefordert wird, unterstützt *ssh* eine Reihe von Funktionen durch Verwendung von Escape Characters.

Folgende Escape Characters werden unterstützt (Default: „~“):

~.	Verbindung trennen.
~^Z	*ssh* im Hintergrund ausführen.
~#	Weitergeleitete Verbindungen auflisten.
~&	*ssh* beim Logout im Hintergrund ausführen, so lange weitergeleitete Verbindungen noch aktiv sind.
~?	Liste der Escape Characters ausgeben.
~B	BREAK an das ferne System senden. (Nur sinnvoll in SSH-Protokoll Version 2 und nur, wenn der Kommunikationspartner das SSH-Protokoll Version 2 unterstützt.)
~C	Kommandozeile öffnen. (Nur sinnvoll für zusätzliche Port Forwardings mit den Options -L und -R.) OpenSSH selbst „promptet“ mit einer Mini-Eingabezeile, in die Sie z.B. `-L ... <RETURN>` eingeben können.
~R	Neuverschlüsselung der Session anfordern. (Nur sinnvoll in SSH-Protokoll Version 2 und nur, wenn der Kommunikationspartner SSH-Protokoll Version 2 unterstützt.)

Ein einzelnes „~“ (Tilde) Zeichen kann gesendet werden als „~~“ oder als „~“, gefolgt von einem anderen als den oben erläuterten Zeichen. Damit ein Escape Character als Sonderzeichen interpretierbar ist, muss das Escape Character immer in einer neuen Zeile beginnen. Das Escape Character können Sie ändern mit der Option *EscapeChar* in der Konfigurationsdatei *ssh_config* des OpenSSH Client *ssh* oder über den Kommandozeilen-Parameter *-e* beim *ssh*-Start-Aufruf.

7.2.7 Port Forwarding (TCP Forwarding)

Das Forwarding beliebiger TCP/IP Verbindungen über einen sicheren Kanal können Sie wahlweise als Kommandozeilen-Parameter beim *ssh*-Start-Aufruf oder in der Konfigurationsdatei *ssh_config* (siehe Seite 340) spezifizieren. Mögliche Anwendungen des Port Forwarding sind z.B. eine sichere Verbindung zum Konto beim Electronic Banking oder bei Tunnelling durch eine Firewall.

7.2.8 Umgebungsvariable von ssh

Der OpenSSH Client *ssh* setzt eine Reihe von Umgebungsvariablen. Nähere Informationen zu diesen Variablen finden Sie auf den Man Pages von OpenSSH.

7.2.9 Dateien von ssh

Neben den Konfigurationsdateien (siehe Seite 339) verwendet der OpenSSH Client *ssh* weitere Dateien, die nachfolgend beschrieben sind.

$HOME/.ssh/known_hosts

Diese Datei enthält für alle Hosts, auf denen sich der Benutzer eingeloggt hat, die zugehörigen Host-Keys, sofern diese nicht in der Datei */etc/ssh/ssh_known_hosts* enthalten sind.

$HOME/.ssh/identity, $HOME/.ssh/id_dsa, $HOME/.ssh/id_rsa

Diese Dateien enthalten die Authentifizierungsdaten des Benutzers. Bei SSH-Protokoll Version 1 sind dies die RSA-Schlüssel und bei SSH-Protokoll Version 2 die DSA-Schlüssel. Die Dateien enthalten sicherheitsrelevante Daten und dürfen nur für den Eigentümer lesbar sein. Alle anderen Benutzer dürfen für die Dateien weder Lese-, Schreib- noch Ausführungsberechtigung besitzen. Beachten Sie, dass *ssh* private Schlüssel ignoriert, wenn diese für andere Benutzer zugänglich sind. Beim Erzeugen des Schlüssels können Sie eine Passphrase spezifizieren, durch die der sensible Teil der Dateien mit 3DES verschlüsselt wird.

$HOME/.ssh/identity.pub, $HOME/.ssh/id_dsa.pub, $HOME/.ssh/id_rsa.pub

Diese Dateien enthalten im Klartext den öffentlichen Schlüssel für die Authentifizierung, also den öffentlichen Teil der Dateien *$HOME/.ssh/identity, $HOME/.ssh/id_dsa* und *$HOME/.ssh/id_rsa*:

- Der Inhalt der Datei *$HOME/.ssh/identity.pub* sollte auf allen Rechnern, auf denen sich der Benutzer mit SSH-Protokoll Version 1 via RSA authentifizieren und einloggen möchte, zur Datei *$HOME/.ssh/authorized_keys* hinzugefügt werden.
- Der Inhalt der Dateien *$HOME/.ssh/id_dsa.pub* und *$HOME/.ssh/id_rsa.pub* sollte auf allen Rechnern, auf denen sich der Benutzer mit SSH-Protokoll Version 2 via DSA/RSA authentifizieren und einloggen möchte, zur Datei *$HOME/.ssh/authorized_keys* hinzugefügt werden.

Die Dateien *$HOME/.ssh/identity.pub, $HOME/.ssh/id_dsa.pub* und *$HOME/.ssh/id_rsa.pub* enthalten keine sensiblen Daten und können (müssen aber nicht) allgemein lesbar sein. Da diese Dateien nie automatisch verwendet werden, sind sie nicht zwingend erforderlich, sondern werden dem Benutzer lediglich zur Vereinfachung des Handlings angeboten.

$HOME/.ssh/authorized_keys

Diese Datei enthält alle öffentlichen Schlüssel (RSA/DSA), die der Benutzer für das Login verwenden kann. Das Dateiformat ist unter sshd (8) auf den Man Pages von OpenSSH beschrieben. In der einfachsten Form entspricht das Format von *$HOME/.ssh/authorized_keys* dem Format der Dateien *$HOME/.ssh/identity.pub, $HOME/.ssh/id_dsa.pub* und *$HOME/.ssh/id_rsa.pub.*

Obwohl der Inhalt der Datei *$HOME/.ssh/authorized_keys* nicht hoch sensibel ist, empfiehlt es sich, die Datei nur für den Benutzer zugänglich zu machen (lesen und schreiben). Für alle anderen Benutzer sollte die Datei unzugänglich sein.

/etc/ssh/ssh_known_hosts

Diese Datei enthält eine Liste aller systemweit bekannten Host-Keys. Der System-Administrator sollte die Datei für die Aufnahme der öffentlichen Host-Keys aller in der Organisation verfügbaren Rechner vorbereiten. Die Datei sollte allgemein lesbar sein.

Jede Datei-Zeile enthält folgende Bestandteile, die durch Leerzeichen voneinander getrennt sind:

– Name(n) des Host; mehrere Namen für denselben Host werden durch Kommas getrennt angegeben
– öffentlicher Schlüssel
– Kommentar (optional).

Das Dateiformat ist unter sshd (8) auf den Man Pages von OpenSSH beschrieben.

OpenSSH verwendet den kanonischen System Namen (wie er vom Name Server geliefert wird), um den Client-Host beim Einloggen zu überprüfen. Weitere Namen werden benötigt, weil OpenSSH den vom Benutzer verwendeten Namen erst nach Überprüfung der Schlüssel in einen kanonischen Namen umwandelt. Andernfalls wäre es mit Zugang zum Name Server möglich, die Authentifizierung des Host zu manipulieren.

/etc/ssh/ssh_host_key, /etc/ssh/ssh_host_dsa_key, /etc/ssh/ssh_host_rsa_key

Diese Dateien enthalten die privaten Bestandteile der Host-Keys und werden für die rhosts-RSA-Authentifizierung (RhostsRSA Authentication) sowie für die rhost-Authentifizierung (Hostbased Authentication) verwendet:

- Wenn die rhosts-RSA-Authentifizierung (siehe Seite 342) mit SSH-Protokoll Version 1 verwendet wird, muss die effektive (setuid) Benutzerkennung von *ssh* die Root-Berechtigung sein, da der Host-Key nur mit Root-Berechtigung gelesen werden kann.
- Bei rhosts-Authentifizierung (siehe Seite 344) mit SSH-Protokoll Version 2 verwendet *ssh* das Utility *ssh-keysign* für den Zugriff auf den Host-Key. (ssh-keysign (8) ist auf den Man Pages von OpenSSH beschrieben.) Die effektive (setuid) Benutzerkennung von *ssh* muss dann nicht zwingend die Root-Berechtigung sein. Standardmäßig hat *ssh* nicht die Root-Rerechtigung als effektive Berechtigung.

$HOME/.rhosts

Diese Datei enthält eine Auflistung der für ein Login zulässigen Host-/Benutzer-Paare, die bei der rhosts-Authentifizierung benötigt wird. Beachten Sie, dass diese Datei auch von *rlogin* und *rsh* verwendet wird und deshalb nicht sicher ist.
Jede Zeile der Datei *$HOME/.rhosts* enthält einen Host-Namen in der kanonischen Form (wie er vom Name Server geliefert wird) und einen Benutzernamen auf diesem Host. Host-Name und Benutzername sind durch ein Leerzeichen getrennt.
Wenn das Home-Verzeichnis des Benutzers auf einer NFS-Partition liegt, kann es sein, dass die Datei auf manchen Rechnern allgemein lesbar sein muss, da der Server-Dämon *sshd* die Datei als Root liest. Zusätzlich muss der Benutzer Eigentümer dieser Datei sein und kein anderer Benutzer darf Schreibrechte für diese Datei besitzen. Für die meisten Rechner ist die empfohlene Berechtigung lesen/schreiben für den Benutzer und kein Zugriffsrecht für alle anderen Benutzer.

sshd ist standardmäßig so eingestellt, dass er die rhosts-Authentifizierung erst nach erfolgreichem Absolvieren einer RSA-Host-Authentifizierung zulässt. Wenn der Host-Key des Client nicht in der Datei */etc/ssh/ssh_known_hosts* auf dem Server-Rechner vorhanden ist, kann der Host-Key in der Datei *$HOME/.ssh/known_hosts* gespeichert werden. Dies erfolgt am einfachsten dadurch, dass mithilfe von *ssh* eine Verbindung vom Server-Rechner zurück zum Client eingerichtet wird. Dadurch wird der Host-Key automatisch in *$HOME/.ssh/known_hosts* gespeichert.

$HOME/.shosts

Diese Datei wird genau so verwendet wie *$HOME/.rhosts*. Mit der Datei *$HOME/.shosts* kann man die rhost-Authentifizierung anwenden, ohne damit ein Login via *rlogin* oder *rsh* zuzulassen.

/etc/hosts.equiv

Diese Datei wird während der rhosts-Authentifizierung verwendet. Sie enthält die Host-Namen in der kanonischen Form. Pro Zeile ist ein Host-Name enthalten. Eine vollständige Beschreibung finden Sie unter sshd (8) auf den Man Pages von OpenSSH. Wenn der Host des Client in der Datei verzeichnet ist und der Benutzername bei Client und Server übereinstimmt, wird automatisch das Login erlaubt, sofern nicht zusätzlich eine erfolgreiche RSA-host-Authentifizierung erforderlich ist. Letzeres ist meist der Fall. Die Datei sollte nur für Benutzer mit Root-Berechtigung schreibbar sein.

/etc/ssh/shosts.equiv

Diese Datei wird identisch verwendet wie die Datei */etc/hosts.equiv*. */etc/ssh/shosts.equiv* kann hilfreich sein, um ein Login mit *ssh*, nicht jedoch mit *rsh* oder *rlogin* zu erlauben.

/etc/ssh/sshrc

Diese Datei enthält Kommandos, die *ssh* beim Einloggen des Benutzers ausführt, bevor die Benutzer-Shell (oder das Benutzerkommando) gestartet wird (siehe hierzu auch Abschnitt „Login Session und Kommando-Ausführung auf einem fernen Rechner“ auf Seite 345). Eine vollständige Beschreibung der Datei */etc/ssh/sshrc* finden Sie unter sshd(8) auf den Man Pages von OpenSSH.

$HOME/.ssh/rc

Diese Datei enthält Kommandos, die *ssh* beim Einloggen des Benutzers ausführt, bevor die Benutzer-Shell (oder das Benutzerkommando) gestartet wird (siehe hierzu auch Abschnitt „Login Session und Kommando-Ausführung auf einem fernen Rechner“ auf Seite 345). Eine vollständige Beschreibung der Datei *$HOME/.ssh/rc* finden Sie unter sshd (8) auf den Man Pages von OpenSSH.

$HOME/.ssh/environment

Die Datei enthält zusätzliche Definitionen für Umgebungsvariablen.

7.3 OpenSSH Client-Anwendungen scp und sftp

Mit den OpenSSH Client-Anwendungen *scp* und *sftp* stehen Ihnen Programme für das sichere Kopieren von Dateien sowie für den sicheren Datei-Transfer zwischen Rechnern in einem Netz zur Verfügung.

7.3.1 scp - sicheres Kopieren von Dateien zwischen Rechnern im Netz

Syntax

```
scp [-1246BCpqrv] [-c <cipher>] [-F <ssh_config>] [-i <identity_file>]
    [-l <limit>] [-o <ssh_option>] [-P <port>] [-S <program>] [-X binary]
    [[<user>@]<host1>:]<file1> [...] [[<user>@]<host2>:]<file2>
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Beschreibung

Mit *scp* können Sie Dateien zwischen Rechnern in einem Netz kopieren. Im Unterschied zum r-Utility *rcp* fragt *scp* Passphrasen und Passwörter ab, wenn diese für die Authentifizierung erforderlich sind.

Jeder Dateiname kann Angaben zu Rechner und Benutzer enthalten, um anzuzeigen, dass die Datei von dem bzw. auf den betreffenden Rechner kopiert werden soll.

Der Schalter `-X binary` kann bei Verbindungen zu ASCII-Servern verwendet werden, um Dateien binär (bit-identisch) zu übertragen.

Returnwert

Bei Erfolg: 0

Bei Fehler: < 0

7.3.2 sftp - sicherer Datei-Transfer zwischen Rechnern im Netz

Syntax

```
sftp [-1Cv] [-B <buffer_size>] [-b <batchfile>] [-F <ssh_config>]
     [-o <ssh_option>] [-P <sftp_server_path>] [-R <num_requests>]
     [-S <program>] [-s <subsystem> | <sftp_server>] [-X binary] <host>

sftp [-X binary] [[<user>@]<host>[:<file> [<file>]]]

sftp [-X binary] [[<user>@]<host>[:<dir>[/]]]

sftp [-X binary] -b <batchfile> [<user>@]<host>
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Beschreibung

sftp ist ein interaktives Programm für den File-Transfer (ähnlich wie FTP), das alle Operationen über einen verschlüsselten *ssh*-Kanal auf TCP-Ebene abwickelt. Darüber hinaus kann *sftp* viele Eigenschaften von *ssh* nutzen, wie z.B. Public Key-Authentifizierung und Datenkomprimierung. *sftp* stellt die Verbindung zum angegebenen Rechner her und loggt sich dort ein. Danach befindet sich *sftp* im interaktiven Modus für die Kommando-Eingabe.

Der Schalter `-X binary` kann bei Verbindungen zu ASCII-Servern benutzt werden, um Dateien binär (bit-identisch) zu übertragen. Im interaktiven Modus kann dazu auch das Kommando `type [text|binary]` (siehe Seite 355) verwendet werden.

Dateien aus einem remote Verzeichnis transferieren

Mit dem Kommando-Format

```
sftp [-X binary] [[<user>@]<host>[:<file> [<file>]]]
```

können Sie *sftp* verwenden, um Dateien automatisch von einem Verzeichnis auf einem fernen Rechner ins lokale Verzeichnis zu transferieren. Voraussetzung hierfür ist, dass eine nicht-interaktive Authentifizierung verwendet wird. Andernfalls ist der Datei-Transfer erst nach erfolgter interaktiver Authentifizierung möglich.

sftp in einem remote Verzeichnis starten

Mit dem folgenden Kommando-Format können Sie *sftp* in einem Verzeichnis auf einem fernen Rechner starten:

```
sftp [-X binary] [[<user>@]<host>[:<dir>[/]]]
```

Automatisierte sftp-Sessions

Das Kommando-Format

```
sftp [-X binary] -b <batchfile> [<user>@]<host>
```

ermöglicht automatische *sftp*-Sessions unter Verwendung der Option `-b`. In diesem Fall ist in der Regel Public Key-Authentifizierung erforderlich, um die Notwendigkeit einer Passwort-Eingabe beim Verbindungsaufbau zu umgehen (siehe Abschnitt „Authentifizierung zwischen OpenSSH Client und Server“ auf Seite 342).

Interaktive Kommandos

Im interaktiven Modus stellt Ihnen *sftp* eine Reihe interaktiver Kommandos zur Verfügung, die denen des FTP ähnlich sind.

Bei der Notation der Kommandos ist zu beachten:

- Groß-/Kleinschreibung ist nicht relevant.
- Pfadnamen, die Leerzeichen enthalten, müssen in einfache oder doppelte Hochkommata gesetzt werden.

`bye`
: *sftp* beenden.

`cd <path>`
: Verzeichnis auf dem fernem Rechner in <path> ändern.

`chgrp <group> <path>`
: Group Id der Datei <path> in <group> ändern.
<group> muss eine numerische Group Id sein.

`chmod <mode > <path>`
: Zugriffsrechte der Datei <path> auf <mode> ändern..

`chown <own> <path>`
: Eigentümer der Datei <path> zu <own> ändern. <own> muss eine numerische UID sein.

`exit`
: *sftp* verlassen.

`get [<flags>] <remote-path> [<local-path>]`
: Datei <remote-path> vom fernen Rechner auf den lokalen Rechner übertragen und unter <local-path> abspeichern. Wenn <local-path> nicht spezifiziert ist, wird die Datei unter dem gleichen Namen wie auf dem fernen Rechner abgespeichert.
Wenn das Flag `-P` spezifiziert ist, werden auch alle Zugriffsrechte und die Zugangszeit der Datei kopiert.

`help`
Hilfetext (Kurzreferenz der Kommandos) anzeigen.

`lcd <path>`
Lokales Verzeichnis in <path> ändern.

`lls [<ls-options> [<path>]]`
Lokales Verzeichnis anzeigen. Entweder <path> anzeigen, oder, wenn <path> nicht spezifiziert ist, aktuelles Verzeichnis anzeigen.

`lmkdir <path>`
Das durch <path> spezifizierte lokale Verzeichnis erstellen.

`ln <oldpath> <newpath>`
Symbolischen Link von <oldpath> zu <newpath> erstellen.

`lpwd`
Lokales Arbeitsverzeichnis drucken.

`ls [<flags>] [<path>]`
Inhalt des fernen Verzeichnisses anzeigen. Entweder Inhalt von <path> anzeigen, oder, wenn <path> nicht spezifiziert ist, Inhalt des aktuellen Verzeichnisses anzeigen. Wenn das Flag `-I` nicht spezifiziert ist, zusätzliche Information inklusive Berechtigungen und Besitz anzeigen.

`lumask <mask>`
Lokale umask (Zugriffsrechte) in <mask> ändern.

`mkdir <path>`
Fernes, durch <path> spezifiziertes Verzeichnis erstellen.

`progress`
Fortschrittsanzeige ein-/ausschalten.

`put [<flags>] <local-path> [<remote-path>]`
Datei <local-path> hochladen und auf den fernen Rechner übertragen. Wenn <remote-path> nicht spezifiziert ist, wird die Datei auf dem fernen Rechner genauso benannt wie auf dem lokalen Rechner. Wenn das Flag `-P` spezifiziert ist, werden auch alle Zugriffsrechte der Datei und die Zugangszeit der Datei kopiert.

`pwd`
Fernes Arbeitsverzeichnis anzeigen.

`quit`
sftp verlassen.

`rename <oldpath> <newpath>`
Ferne Datei von <oldpath> in <newpath> umbenennen.

`rm <path>`
Ferne Datei <path> löschen.

`rmdir <path>`
Fernes Verzeichnis <path> entfernen.

`symlink <oldpath> <newpath>`
Symbolischen Link von <oldpath> zu <newpath> erstellen.

`type [binary|text]`
Übertragungsmodus (bei Verbindungen zu ASCII Rechnern) auf „binär“ setzen bzw. auf „text“ zurücksetzen. Wenn kein Argument angegeben ist, wird der aktuelle Modus angezeigt (siehe auch Aufrufschalter `-X binary` Seite 352).

`version`
Protokoll-Version von *sftp* anzeigen.

`! <command>`
Kommando <command> in der lokalen Shell ausführen.

`!`
Lokale Sub-Shell erzeugen.

`?`
Hilfetext (Kurzreferenz der Kommandos) anzeigen (Synonym für `help`, siehe oben).

7.4 OpenSSH Basis-Utilities

Für die Unterstützung einer komfortablen und automatisierten Client-/Server-Authentifizierung stellt OpenSSH folgende Basis-Utilities zur Verfügung:

- *ssh-agent* - Authentifizierungsagent

  Da bei der Client-Authentifizierung (siehe Seite 342) die privaten Schlüssel aus Sicherheitsgründen stets verschlüsselt in Dateien abgelegt werden sollten, müssen Sie bei jeder Verwendung des privaten Schlüssels das zugehörige Passwort eingeben. In diesen und weiteren Fällen vereinfacht der Authentifizierungsagent die Handhabung der privaten Schlüssel.

- *ssh-add* - private RSA/DSA-Schlüssel in den Authentifizierungsagenten laden

  Nach dem Start enthält der Authentifizierungsagent zunächst keine privaten RSA/DSA-Schlüssel. Diese laden Sie mit dem Utility *ssh-add* in den Authentifizierungsagenten.

- *ssh-keygen* - benutzer-spezifisches Key-Paar (RSA/DSA) erzeugen und administrieren

  Bei der RSA/DSA-basierten Client-Authentifizierung benötigt jeder Benutzer ein Paar aus privatem und öffentlichem RSA- bzw. DSA-Schlüssel. Mit dem Utility *ssh-keygen* können Sie solche Schlüsselpaare erzeugen und verwalten.

- *ssh-keyscan* - *ssh_known_host*-Dateien erstellen und überprüfen

  Mit dem Utility *ssh-keyscan* können Sie die Public SSH-Host-Keys verschiedener Hosts von jedem OpenSSH Client aus abfragen und in die *ssh_known_hosts*-Dateien übernehmen. Außerdem unterstützt Sie *ssh-keyscan* bei der Überprüfung bestehender *ssh_known_hosts*-Dateien.

7.4.1 ssh-agent - Authentifizierungsagent

Syntax

```
ssh-agent [-a <bind_address>] [-c | -s] [-t <life>]
          [-d] [<command> <[args> ...]]

ssh-agent [-c | -s] -k
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Beschreibung

Der Authentifizierungsagent *ssh-agent* ist ein Utility zur Verwaltung der privaten Schlüssel für die Public Key Authentifizierung (RSA, DSA). Zunächst verfügt *ssh-agent* über keinerlei Schlüssel. Mit dem Utility *ssh-add* (siehe Seite 359) können Sie *ssh-agent* mit Schlüsseln versorgen. *ssh-add* fügt dem Authentifizierungsagenten die Dateien *$HOME/.ssh/id_rsa*, *$HOME/.ssh/id_dsa* und *$HOME/.ssh/identity* hinzu.

Sinnvollerweise starten Sie *ssh-agent* zu Beginn einer Login-Session auf Ihrem lokalen PC, Notebook oder Terminal. Alle anderen Fenster und Programme werden dann als Clients von *ssh-agent* ausgeführt. Mithilfe von Umgebungsvariablen können die Clients den *ssh-agent* lokalisieren und für die automatische Authentifizierung nutzen, wenn sie sich mit dem OpenSSH Client *ssh* an fernen Rechnern einloggen.

Die Ausführung von *ssh-agent* auf Ihrem lokalen PC, Notebook oder Terminal hat den Vorteil, dass *ssh-agent* Authentifizierungsdaten niemals über den Kanal sendet, über den er die Anforderungen erhält. Vielmehr führt *ssh-agent* Operationen, die einen privaten Schlüssel benötigen, selbst durch und sendet das Ergebnis zurück an den Rechner, von dem die Anforderung kam. Dieses Verhalten wird als Agent Forwarding bezeichnet.

Wenn Sie Authentifizierungsagenten mit dem Format

```
eval ‘ssh-agent [-c | -s]‘
```

aufrufen, wird *ssh-agent* als Hintergrundprozess (Dämon) ausgeführt und erzeugt Shell-Kommandos als Ausgabe. Mithilfe dieser Kommandos können die Shell und ihre Child-Prozesse die Dienste des Authentifizierungsagenten nutzen.

Den *ssh-agent*-Dämon beenden Sie mit dem Kommando:

```
eval ‘ssh-agent -k [-c | -s]‘
```

Wenn Sie Authentifizierungsagenten mit dem Format

```
ssh-agent [<command> [<args> ...]]
```

aufrufen, kann das Kommando <command> und alle Child-Prozesse von <command> den Authentifizierungsagenten nutzen. Nach Ausführung von <command> wird der Authentifizierungsagent automatisch beendet.

Wenn Sie den Authentifizierungsagenten ohne Parameter aufrufen, erzeugt dieser einen Unix-Domain Socket, und lauscht dort im Hintergrund auf Benutzeranforderungen.

Dateien

ssh-agent verwendet die folgenden Dateien:

$HOME/.ssh/identity

Enthält den privaten RSA-Schlüssel des Benutzers für die Authentifizierung gemäß SSH-Protokoll Version 1.

$HOME/.ssh/id_dsa

Enthält den privaten DSA-Schlüssel des Benutzers für die Authentifizierung gemäß SSH-Protokoll Version 2.

$HOME/.ssh/id_rsa

Enthält den privaten RSA-Schlüssel des Benutzers für die Authentifizierung gemäß SSH-Protokoll Version 2.

/tmp/ssh-XXXXXXXX/agent.<ppid>

Unix-Domain Sockets, die die Verbindungen zum *ssh-agent* enthalten. Für diese Sockets sollten nur für den Eigentümer Lese- und Schreibberechtigung bestehen, alle anderen Benutzer sollten weder lese- noch schreibberechtigt sein. Bei Beendigung des *ssh-agent* werden die Sockets automatisch entfernt.

7.4.2 ssh-add - Private Keys in den Authentifizierungsagenten laden

Syntax

```
ssh-add [-lLdDxXc] [-t <life>] [<file> ...]
ssh-add -s <reader>
ssh-add -e <reader>
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Beschreibung

ssh-add versorgt den Authentifizierungsagenten *ssh-agent* mit privaten RSA- bzw. DSA-Schlüsseln. Ohne Argumente aufgerufen, fügt *ssh-add* dem Authentifizierungsagenten die Dateien *$HOME/.ssh/id_rsa*, *$HOME/.ssh/id_dsa* und *$HOME/.ssh/identity* hinzu. Alternative Dateinamen können Sie in der Kommandozeile spezifizieren.

Wenn eine Datei durch eine Passphrase geschützt ist, fordert *ssh-add* den Benutzer zur Eingabe einer Passphrase auf, die *ssh-add* dann vom Terminal des Benutzers einliest. Bei mehreren geheimen RSA- oder DSA-Schlüsseln versucht *ssh-add*, die zuletzt eingelesene Passphrase erneut zu verwenden.

Mit `ssh-add -l` können Sie sich die von *ssh-agent* aktuell verwalteten Schlüssel anzeigen lassen.

Die Ausführung von *ssh-add* setzt voraus, das der Authentifizierungsagent *ssh-agent* gestartet ist und in der Umgebungsvariable *SSH_AUTH_SOCK* der Name von dessen Socket abgelegt ist. Die Umgebungsvariable *SSH_AUTH_SOCK* wird automatisch beim Start von *ssh-agent* gesetzt.

Returnwert

Bei Erfolg: 0

Das spezifizierte Kommando konnte nicht ausgeführt werden: 1

ssh-add konnte keine Verbindung zu *ssh-agent* aufnehmen: 2

Umgebungsvariable von ssh-add

`SSH_AUTH_SOCK`
: Identifiziert den Pfadnamen des Socket einer Domäne des Unix-Systems, der für die Kommunikation mit dem Authentifizierungsagenten *ssh-agent* verwendet wird.

Dateien von ssh-add

ssh-add verwendet die folgenden Dateien:

$HOME/.ssh/identity

Enthält den privaten RSA-Schlüssel des Benutzers für die Authentifizierung gemäß SSH-Protokoll Version 1.

$HOME/.ssh/id_dsa

Enthält den privaten DSA-Schlüssel des Benutzers für die Authentifizierung gemäß SSH-Protokoll Version 2.

$HOME/.ssh/id_rsa

Enthält den privaten RSA-Schlüssel des Benutzers für die Authentifizierung gemäß SSH-Protokoll Version 2.

ssh-add ignoriert die genannten Dateien, wenn diese von anderen Benutzerkennungen zugreifbar sind. Mit `chmod go-rwx ...` können Sie die Dateien dem Zugriff anderer Benutzer entziehen.

7.4.3 ssh-keygen - RSA/DSA-Schlüssel-Paar generieren und administrieren

Syntax

```
ssh-keygen [-q] [-b <bits>] -t <type> [-N <new_passphrase>] [-C <comment>]
           [-f <output_keyfile>]
ssh-keygen -p [-P <old_passphrase>] [-N <new_passphrase>] [-f <keyfile>]
ssh-keygen -i [-f <input_keyfile>]
ssh-keygen -e [-f <input_keyfile>]
ssh-keygen -y [-f <input_keyfile>]
ssh-keygen -c [-P <passphrase>] [-C <comment>] [-f <keyfile>]
ssh-keygen -l [-f <input_keyfile>]
ssh-keygen -B [-f <input_keyfile>]
ssh-keygen -D <reader>
ssh-keygen -U <reader> [-f <input_keyfile>]
ssh-keygen -r <hostname> [-f <input_keyfile>] [-g]
ssh-keygen -G <output_file> [-v] [-b <bits>] [-M <memory>] [-S <start_point>]
ssh-keygen -T <output_file> -f <input_file> [-v] [-a <num_trials>]
           [-W <generator>]
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Beschreibung

Jeder Authentifikationsalgorithmus benötigt ein eigenes Paar aus privatem und öffentlichem Schlüssel. Mit dem Utility *ssh-keygen* können Sie ein solches Schlüsselpaar (RSA oder DSA) erstellen. Den Verschlüsselungsalgorithmus spezifizieren Sie mit dem Parameter `-t`.

ssh-keygen erzeugt, verwaltet und konvertiert Authentifizierungsschlüssel für den OpenSSH Client *ssh*. *ssh-keygen* kann RSA-Schlüssel für die Verwendung mit SSH-Protokoll Version 1 und RSA- oder DSA-Schlüssel für die Verwendung mit SSH-Protokoll Version 2 generieren.

In der Regel wird ein Benutzer, der OpenSSH mit RSA- oder DSA-Authentifizierung einsetzen möchte, *ssh-keygen* starten, um den Authentifizierungsschlüssel in *$HOME/.ssh/identity*, *$HOME/.ssh/id_dsa* oder *$HOME/.ssh/id_dsa* zu erzeugen. Zusätzlich kann der System-Administrator mit *ssh_keygen* Host-Keys generieren.

Private Schlüssel (Private Keys) werden unter *$HOME/.ssh/i<dentity>*, öffentliche Schlüssel (Public Keys) unter *$HOME/.ssh/<identity>.pub* im Verzeichnis des Benutzers abgelegt. Dabei steht <*identity*> für

id_dsa (`-t dsa`) bei DSA bzw.
id_rsa (`-t rsa`) bei RSA bzw.
identity (`-t rsa1`) bei RSA1.

Sie können die Datei auch anders benennen:

- *ssh-keygen* fragt bei Erstellung der Schlüssel den Namen der Datei ab, in der der Private Key hinterlegt werden soll.
- Der Dateiname für den Public Key wird automatisch angepasst und mit dem Appendix „.pub“ erweitert.

Weitere Informationen zu *ssh-keygen* finden Sie auf den Man Pages von OpenSSH.

Dateien

Die Beschreibung der Dateien von *ssh-keygen* finden Sie auf den Man Pages von OpenSSH.

7.4.4 ssh-keyscan

Syntax

```
ssh-keyscan [-v46] [-p <port>] [-T <timeout>] [-t <type>] [-f <file>]
            [<host> | <addrlist> <namelist>] [...]
```

Die ausführliche Beschreibung der Operanden finden Sie auf den Man Pages von OpenSSH.

Beschreibung

Mit dem Utility *ssh-keyscan* können Sie die Public SSH-Host-Keys verschiedener Hosts von jedem OpenSSH Client aus abfragen und direkt in die *ssh_known_hosts*-Dateien übernehmen. Darüber hinaus unterstützt Sie *ssh-keyscan* bei der Verifizierung bestehender *ssh_known_hosts*-Dateien. *ssh-keyscan* stellt eine minimale Schnittstelle bereit, die für Shell- und Perl-Skripts gleichermaßen geeignet ist.

ssh-keyscan verwendet nicht-blockierende Socket-Ein-/Ausgabe, um mit möglichst vielen Rechnern gleichzeitig Verbindung aufnehmen zu können.

Für das Scannen der Rechner nach Host-Keys benötigt *ssh-keyscan* keine Login-Berechtigung auf diesen Rechnern. Auch umfasst der Scanning-Prozess keinerlei Verschlüsselung.

Sicherheitsaspekte

Wenn Sie eine *ssh_known_hosts*-Datei mithilfe von *ssh-keyscan* erstellen, sind Sie anfällig für „man-in-the-middle“-Angriffe. Lässt andererseits das zugrunde liegende Security-Modell solche Risiken zu, so kann *ssh-keyscan* Sie beim Entdecken verfälschter Host-Key-Dateien unterstützen. Ebenso erleichtert *ssh-keyscan* Ihnen das Aufspüren von „man-in-the-middle“-Angriffen, die nach Erstellung der *ssh_known_hosts*-Datei gestartet wurden.

7.5 BS2000/OSD-spezifische Einschränkungen

Beim Arbeiten mit OpenSSH im BS2000/OSD-Umfeld sind die nachfolgend beschriebenen Besonderheiten zu beachten.

Eingabe-Aufforderung bei leerem Passwort

Gewöhnliche Unix-Rechner verlangen kein Passwort, wenn Sie sich per *login* oder *slogin* an einer Kennung ohne Passwort anmelden. Der POSIX *rlogin* verlangt jedoch auch bei einer Kennung ohne Passwort eine Passwort-Eingabe. Dieses Verhalten führt jedoch nicht zu einer höheren Sicherheit, da es gleichzeitig möglich ist, für dieselbe Kennung ein *rsh*-Kommando ohne Passwort abzusetzen.

OpenSSH verhält sich hier wie die anderen Unix-Systeme und fragt nicht nach einem leeren Passwort. Da in OpenSSH jedoch standardmäßig ein Login zu Kennungen ohne Passwort gesperrt ist, müssen Sie in diesem Fall in der Konfigurationsdatei */etc/ssh/sshd_config* die Direktive *PermitEmptyPasswords* auf „yes“ setzen.

Groß-/Kleinschreibung des Benutzernamens

In BS2000/OSD und OSD/POSIX wird, anders als in Unix-Betriebssystemen, beim Benutzernamen nicht zwischen Groß- und Kleinschreibung unterschieden. Somit kann sich der Benutzer „Benutzername“ in BS2000/OSD und POSIX als „benutzername“, „BENUTZERNAME“ oder auch „beNuTzErNaMe“ anmelden. Der Name des angemeldeten Benutzers wird in der Datei */var/adm/utmp* registriert. Mit dem `who`-Kommando können Sie sich den Benutzernamen anzeigen lassen.

Während rlogin den Benutzernamen in Großbuchstaben einträgt, verwendet OpenSSH den Benutzernamen in Kleinbuchstaben (wie auf Unix-Betriebssystemen üblich).

Einschränkungen bei der Verwendung der BS2000/OSD Dialog-Schnittstelle

Das *ssh* Client-Programm können Sie nur innerhalb einer Remote-Login-Session starten, d.h. Sie müssen in POSIX per *rlogin*, *ssh*, oder *slogin* eingeloggt sein.
In $DIALOG oder TELNET-Sitzungen können Sie *ssh* nur benutzen, wenn *stdin*, *stdout* und *stderr* auf eine Datei oder eine POSIX-Pipe umgelenkt wurden. Andernfalls bricht *ssh* mit folgender Meldung ab:

```
select: Invalid fileno
```

8 Mail-Reader in BS2000/OSD

Mit dem Mail-Reader können Sie in BS2000/OSD Mails über die Zugriffs-Services (POP3 und IMAP) abholen und weiterverarbeiten.

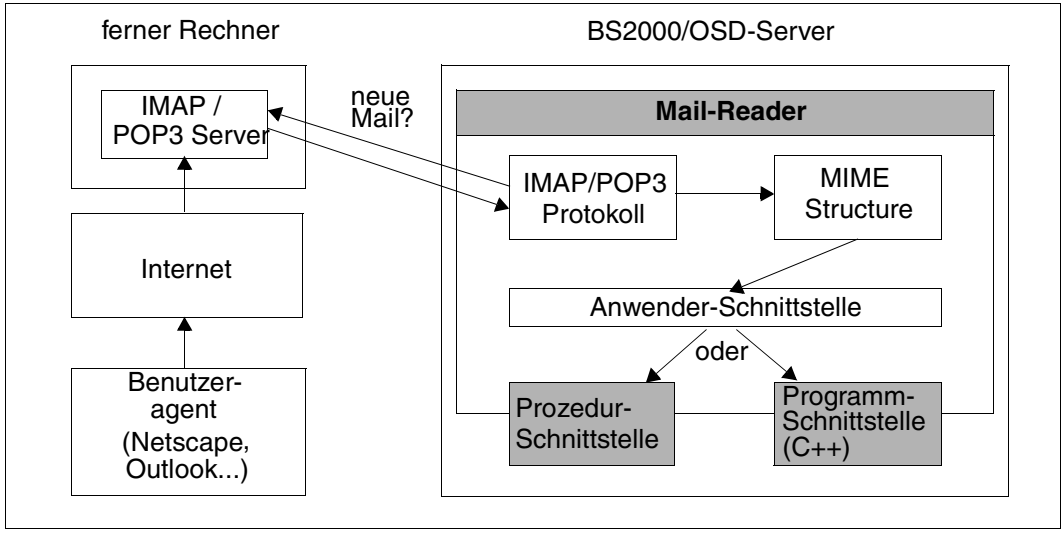


Bild 10: Mail-Reader in BS2000/OSD

In BS2000/OSD bietet Ihnen der Mail-Reader zwei Möglichkeiten zur Weiterverarbeitung:

– über die Prozedur-Schnittstelle
– über die Programm-Schnittstelle

Prozedur-Schnittstelle

Der Mail-Reader ermöglicht es, mithilfe von Prozeduren auf die Nachrichtenköpfe (Header), auf die Nachrichten (Messagebody) und auf die Anhänge (Attachments) einer Mail zuzugreifen.

Programm-Schnittstelle

Über die C++-Schnittstelle wird einer Funktion eine Instanz einer C++-Klasse `DwMessage` übergeben, die die gesamte Mail beinhaltet. Dazu müssen Sie ein C++-Unterprogramm schreiben und dieses mit den Modulen des Mail-Readers binden.

Zur softwaremäßigen Darstellung einer Mail wird die Open-Source-Bibliothek „mimelib“ verwendet, die Klassen zur Verfügung stellt, die einen bequemen Zugriff auf die Mail gestatten. Diese Bibliothek ist ein unter der GP-Lizenz stehender Bestandteil von KDE (Paket *kdepim*) und basiert auf „mime++“ von Doug Sauder.

8.1 Mail-Reader starten/beenden

Mail-Reader starten

Verwenden Sie eines der beiden folgenden Kommandos, um den Mail-Reader zu starten:

```
/START-MAILREADER
```

oder

```
/START-PROGRAM *MODULE($.SYSPRG.MAIL.033,READER,*ANY,*ADVANCED)
```

Mail-Reader beenden

Verwenden Sie eines der beiden folgenden Kommandos, um den Mail-Reader zu beenden:

```
/INTR [<tsn>,]SHUTDOWN
```

oder

```
/INTR [<tsn>,]CANCEL
```

Diese Kommandos beenden den Mail-Reader geordnet, d.h. nachdem ein eventuell gerade laufender Zugriff auf einen Postkorb beendet ist.

8.2 Konfigurationsdatei

Über die Konfigurationsdatei *SYSDAT.MAIL.033.READER* steuern Sie das Verhalten des Mail-Readers. Die Konfigurationssdatei wird beim Start des Mail-Readers gelesen.

Aufbau der Konfigurationsdatei

Mit der Konfigurationsdatei bestimmen Sie:

– Verhalten des Mail-Reader bezüglich der Server POP3 oder IMAP
– Ereignislogging
– Verarbeitung der Mail im BS2000

Dementsprechend ist die Konfigurationsdatei in folgende Parameterbereiche gegliedert:

– SERVER
– TRACE
– MAILHANDLING

In den nachfolgenden Abschnitten (ab Seite 368) wird jeweils die Syntax dieser Parameterbereiche dargestellt und anhand von Beispielen erläutert.

Zeilen in der Konfigurationsdatei, die mit # beginnen, werden als Kommentare ignoriert.

Konfiguration des Mail-Readers via Konfigurationsdatei ändern

Änderungen in der Konfigurationsdatei *SYSDAT.MAIL.033.READER* wirken sich auf die Konfiguration des Mail-Readers erst nach einem Neustart des Mail-Readers aus.

Wenn Sie die Konfiguration des Mail-Readers im laufenden Betrieb ändern wollen, können Sie ein erneutes Einlesen der Konfigurationsdatei wie folgt veranlassen:

- Falls der Mail-Reader als Batch-Task ausgeführt wird:
  - ► Geben Sie folgendes Kommando ein:

    ```
    /INTR <TSN der Batch-Task>,RECONFIG
    ```

- Falls der Mail-Reader als Dialog-Task ausgeführt wird (z.B. zu Testzwecken):
  - ► Unterbrechen Sie die Ausführung des Mail-Readers durch Drücken der K2-Taste.
  - ► Geben Sie folgendes Kommando ein:

    ```
    /INTR <TSN der Dialog-Task>,RECONFIG
    ```

  - ► Geben Sie folgendes Kommando ein, um die Ausführung des Mail-Readers fortzusetzen:

    ```
    /RESUME-PROG
    ```

8.2.1 Mail-Verarbeitung: Parameterbereich MAILHANDLING

Im Parameterbereich MAILHANDLING legen Sie fest, wie mit der erhaltenen Mail verfahren werden soll.

Für jede Mail werden zwei Dateien angelegt:

- Prozedur-Datei
  enthält die Anweisungen, die beim Erhalt einer Mail ausgeführt werden
- Nachrichten-Datei
  enthält die erhaltene Nachricht

Entsprechend legen Sie mit dem Parametersatz PROCEDURE fest, wie die Prozedur-Datei aufgebaut ist. Mit dem Parametersatz MESSAGEBODY legen Sie fest, wie die Datei aufgebaut ist, die die eigentliche Nachricht enthält.

8.2.1.1 Syntax

```
MAILHANDLING = PARAMETERS ( ... )
    PARAMETERS ( ... )
        |  PREFIX=<string>
        |  ,ENTER=JOB / PROCEDURE
        |  ,DELETE=YES / NO
        |  ,SMIME=PARAMETERS( ... )
        |      PARAMETERS( ... )
        |          |  KEY=<private key file>
        |          |  ,CERTIFICATE=<certificate file>
        |          |  ,CRL=<CRL file>
        |          |  ,CA_CERTIFICATES=<CA certificates file>
        |          |  ,VERIFY_SIGNATURE=YES / NO
        |          |  ,VERIFY_DEPTH=<depth>
        |          |  ,USE_CRYPTO_HARDWARE=NO / YES
        |  ,BODY=PARAMETERS( ... )
        |      PARAMETERS( ... )
        |          |  PROCEDURE=PARAMETERS( ... )
        |          |      PARAMETERS( ... )
        |          |          |  SUFFIX=<string>
        |          |          |  ,TEXT=(<textlist>)
        |          |          |  ,ATTACHMENT=(<textlist>)
        |          |  ,MESSAGEBODY=PARAMETERS( ... )
        |          |      PARAMETERS( ... )
        |          |          |  TEXT=(<textlist>)
        |  ,ATTACHMENT=PARAMETERS( ... )
        |      PARAMETERS( ... )
        |          |  PROCEDURE=(<textlist>)
        |          |  ,MESSAGEBODY=(<textlist>)
```

Beschreibung der Operanden

PREFIX=<string>
Namens-Präfix für die erzeugten Dateien.
Die Dateinamen werden gemäß dem Schema *PREFIX.JJJJ-MM-DD.HHMMSS*[*.#nnn*] gebildet. Wenn innerhalb einer Sekunde mehrere Dateien erzeugt werden, so wird mit *#nnn* hochgezählt.
Voreinstellung: MAIL

ENTER=JOB / PROCEDURE
legt fest, ob die Prozedur-Datei mit /*ENTER-JOB-* oder mit */ENTER-PROCEDURE* gestartet werden soll.

ENTER=JOB
startet die Enter-Prozedur mit /*ENTER-JOB*.

ENTER=PROCEDURE
startet die Enter-Prozedur mit */ENTER-PROCEDURE*.

DELETE=YES / NO
gibt an, ob nach Ausführung der Enter-Prozedur die Prozedurdatei gelöscht werden soll.

DELETE=YES
Die Prozedurdatei wird nach Ausführung der Enter-Prozedur gelöscht.

DELETE=NO
Die Prozedurdatei wird nach Ausführung der Enter-Prozedur **nicht** gelöscht.

SMIME=PARAMETERS(...)
bestimmt die Behandlung von gemäß S/MIME-Standard signierten und/oder verschlüsselten Mails. Es wird nur S/MIME Version 2 unterstützt.

KEY=<private key file>
benennt die Datei, die den für die Entschlüsselung von verschlüsselten Mails notwendigen privaten Schlüssel enthält. Dieser private Schlüssel muss unbedingt geheim bleiben. Der Schlüssel muss im PEM-Format abgelegt sein.

CERTIFICATE=<certificate file>
benennt die Datei, die das zum privaten Schlüssel zugehörige X.509-Zertifikat enthält. Das Zertifikat muss im PEM-Format abgelegt sein.

CRL=<CRL file>
benennt die Datei, die eine Certificate Revocation List (CRL) enthält. Mit der CRL wird bei signierten Mails überprüft, ob die bei der Signatur verwendeten Zertifikate noch gültig sind. Hierfür sind in regelmässigen Abständen (von typischerweise wenigen Wochen) die CRLs von den betroffenen CAs zu besorgen und in einer gemeinsamen Datei abzulegen, deren Name mit diesem Operand spezifiziert wird.

Rein technisch funktioniert die Verifizierung auch ohne Verwendung einer CRL. Wenn man aber nicht auf andere Weise sicherstellen kann, dass die verwendeten Zertifikate noch gültig sind, sollte man wie eben beschrieben eine CRL-Datei erzeugen und spezifizieren.

CA_CERTIFICATES=<CA certificates file>
benennt eine Datei, die die für die Verifizierung von S/MIME-Signaturen benötigten CA-Zertifikate enthält. Die X.509-Zertifikate müssen im PEM-Format abgelegt sein und sind sequenziell in der Datei angeordnet.

Zum Hinzufügen oder Löschen von Zertifikaten kann die Datei mit einem beliebigen Text-Editor bearbeitet werden.
Die einzelnen Zertifikate sind wie folgt in der Datei notiert:

```
-----BEGIN CERTIFICATE-----
< CA-Zertifikat in Base64-Kodierung >
-----END CERTIFICATE-----
```

Text außerhalb dieser Sequenzen wird vom Mail-Reader ignoriert und kann deshalb zum Kennzeichnen der Zertifikate verwendet werden, die wegen der ASN.1/Base64-Kodierung in nicht lesbarer Form vorliegen.

VERIFY_SIGNATURE=<u>YES</u> / NO
gibt an, ob eine evtl. vorhandene Signatur mithilfe der CA-Zertifikate und der CRL überprüft werden soll.

VERIFY_SIGNATURE=<u>YES</u>
Vorhandene Signatur wird überprüft.

VERIFY_SIGNATURE=NO
Vorhandene Signatur wird **nicht** überprüft .

VERIFY_DEPTH=<depth>
legt die so genannte Verifizierungstiefe fest, d.h. die Anzahl der maximal zulässigen Zertifikate zwischen dem Zertifikat des S/MIME-Signierers und dem Zertifikat, das dem Mail-Reader bekannt ist. Wird die maximale Verifizierungstiefe überschritten, wird die Verifizierung als fehlgeschlagen betrachtet.

Im Einzelnen ist zu beachten:

1 Signierer-Zertifikat wird nur akzeptiert, wenn es direkt von einer dem Mail-Reader bekannten Certificate Authority (CA) signiert worden ist.

0 Nicht sinnvoll, da in diesen Fällen nur selbstsignierte Zertifikate zulässig sind.

Voreinstellung: 9

USE_CRYPTO_HARDWARE=NO / YES
gibt an, ob zur Berechnung kryptographischer Algorithmen Krypto-Hardware, z.B. eine openCRYPT™-BOX, verwendet werden soll.

USE_CRYPTO_HARDWARE=NO
Es wird keine Krypto-Hardware verwendet.

USE_CRYPTO_HARDWARE=YES
Es wird Krypto-Hardware verwendet.

BODY=PARAMETERS(...)
Die folgenden Parameter bestimmen die Behandlung des Mail-Bodys.
Als Mail-Bodys werden alle Nachrichten-Teile behandelt, bei denen im Content-Type-Header der *name*-Parameter nicht versorgt ist.

PROCEDURE=PARAMETERS(...)
bestimmt den Inhalt der zu erzeugenden Prozedur-Datei.

SUFFIX=<string>
legt den Suffix fest, der die Prozedur-Datei von der Nachrichten-Datei unterscheidet.
Standardeinstellung: PROC

TEXT=(<textlist>)
Der Operanden-Wert ist eine durch Kommas getrennte Liste von Texten, die jeweils in einfache Anführungszeichen eingeschlossen sind. Für jeden non-Multipart-Mail-Bestandteil und jeden Bestandteil einer Multipart-Mail, bei dem der *name*-Operand des Content-Type-Headers nicht versorgt ist, wird diese Textliste am Anfang der Prozedur-Datei eingefügt (jedes Element der Liste steht in einer eigenen Zeile). Dabei werden die im Abschnitt „Mail-spezifische Parameter substituieren“ auf Seite 390 definierten, in %-Zeichen eingeschlossenen Parameter substituiert. Insbesondere wird %FILE-NAME% durch den Namen der geschriebenen Nachrichten-Datei ersetzt. Als Voreinstellung ist diese Text-Liste leer.

ATTACHMENT=(<textlist>)
Der Operanden-Wert ist eine durch Kommas getrennte Liste von Texten, die jeweils in einfache Anführungszeichen eingeschlossen sind. Für jeden Teil einer Multipart-Mail, bei dem der *name*-Operand des Content-Type-Headers versorgt ist, wird diese Textliste in die Prozedur-Datei eingefügt (jedes Element der Liste steht in einer eigenen Zeile). Dabei werden die im Abschnitt „Mail-spezifische Parameter substituieren“ auf Seite 390 definierten, in %-Zeichen eingeschlossenen Parameter substituiert. Insbesondere wird %ATTACHMENT-FILE-NAME% durch den Namen der erzeugten Attachment-Nachrichten-Datei und %ATTACHMENT-PROCEDURE-NAME% durch den Namen der erzeugten Prozedur-Datei für das jeweilige Attachment ersetzt. Als Voreinstellung ist diese Text-Liste leer.

MESSAGEBODY=PARAMETERS(...)
bestimmt den Aufbau der Nachrichten-Datei.

TEXT=(<textlist>)
Der Operanden-Wert ist eine durch Kommas getrennte Liste von Texten, die jeweils in einfache Anführungszeichen eingeschlossen sind. Für einen nicht-Multipart-Mail-Bestandteil auf der obersten Ebene und für jeden Teil einer Multipart-Mail, bei dem der *name*-Operand des Content-Type-Headers nicht versorgt ist, wird diese Textliste in die Nachrichten-Datei eingefügt (jedes Element der Liste steht in einer eigenen Zeile). Dabei werden die im Abschnitt „Mail-spezifische Parameter substituieren“ auf Seite 390 definierten, in %-Zeichen eingeschlossenen Parameter substituiert. Insbesondere wird %TEXT% durch den Nachrichtentext substituiert. Als Voreinstellung ist diese Text-Liste leer.

ATTACHMENT=PARAMETERS(...)
bestimmt den Aufbau der für jedes Attachment zu generierenden Prozedur-Datei und Nachrichten-Datei.
Der Name der Dateien wird derart generiert, dass an den Namen des Mail-Bodys der Wert des *name*-Parameters aus dem Content-Type-Header-Feld angehängt wird. Für die Attachment-Prozedur-Datei wird noch zusätzlich das Suffix angehängt (siehe Operand SUFFIX).

PROCEDURE=(<textlist>)
bestimmt den Aufbau der für jedes Attachment zu generierenden Prozedur-Datei.
Der Operanden-Wert ist eine durch Kommas getrennte Liste von Texten, die jeweils in einfache Anführungszeichen eingeschlossen sind. Diese Textliste wird in die Prozedur-Datei eingefügt, wobei jedes Element der Liste in einer eigenen Zeile steht. Dabei werden die im Abschnitt „Mail-spezifische Parameter substituieren“ auf Seite 390 definierten, in %-Zeichen eingeschlossenen Parameter substituiert. Als Voreinstellung ist diese Text-Liste leer.

MESSAGEBODY=(<textlist>)
bestimmt den Aufbau der für jedes Attachment zu generierenden Nachrichten-Datei.
Der Operanden-Wert ist eine durch Kommas getrennte Liste von Texten, die jeweils in einfache Anführungszeichen eingeschlossen sind. Diese Textliste wird in die Nachrichten-Datei eingefügt, wobei jedes Element der Liste in einer eigenen Zeile steht. Dabei werden die im Abschnitt „Mail-spezifische Parameter substituieren“ auf Seite 390 definierten, in %-Zeichen eingeschlossenen Parameter substituiert. Insbesondere wird %TEXT% durch den Nachrichtentext und %HEADER% durch die Header-Zeilen substituiert. Als Voreinstellung ist diese Text-Liste leer.

8.2.1.2 Beispiel

Das folgende Beispiel stellt den Parameterbereich MAILHANDLING der Konfigurationsdatei dar.

```
MAILHANDLING=PARAMETERS(
    ENTER=PROCEDURE,
    BODY=PARAMETERS(
        PROCEDURE=PARAMETERS(
            TEXT=(
                '/WRITE-TEXT ''To: %To%''',
                '/WRITE-TEXT ''Subject: %SUBJECT%''',
                '/INCLUDE-PROCEDURE MAILPROC(FILENAME=%FILE-NAME%)'
            ),
            ATTACHMENT=(
                '/REMARK ****************************',
                '/REMARK * Now we call the procedure ',
                '/REMARK * handling the attachment   ',
                '/REMARK ****************************',
                '/INCLUDE-PROCEDURE %ATTACHMENT-PROCEDURE-NAME%,(-',
                '/                    CONTENTTYPE=''%CONTENT-TYPE%'')'
            )
        )
    ),
    ATTACHMENT=PARAMETERS(
        PROCEDURE=(
            '/BEG-PAR-DECL',
            '/DECLARE-PARAMETER CONTENTTYPE',
            '/END-PAR-DECL',
            '/CALL-PROCEDURE ATTPROC,(CT=''&(CONTENTTYPE)'')'
        )
    )
)
```

Wenn Sie das oben dargestellte Beispiel verwenden, um eine Mail mit Attachment zu empfangen, erzeugt der Mail-Reader folgende Prozedur-Datei:

– MAIL.2010-06-27.134758.PROC nach folgendem Muster, um die Nachricht zu verarbeiten:

```
/WRITE-TEXT 'To: mail01@system'
/WRITE-TEXT 'Subject: This is a funny test'
/INCLUDE-PROCEDURE MAILPROC,(FILENAME=MAIL.2010-06-27.134758)
/REMARK ****************************
/REMARK * Now we call the procedure
/REMARK * handling the attachment
/REMARK ****************************
```

```
/INCLUDE-PROCEDURE MAIL.2010-06-27.134758.testtxt.PROC,(-
/                    CONTENTTYPE='text/plain; name="test.txt"')
```

– MAIL.2010-06-27.134758.TESTTXT.PROC nach folgendem Muster, um den Anhang zu verarbeiten:

```
/BEG-PAR-DECL
/DECLARE-PARAMETER CONTENTTYPE
/END-PAR-DECL
/CALL-PROCEDURE ATTPROC,(CT='&(CONTENTTYPE)')
```

Um die Nachricht zu verarbeiten, ruft die mit dem Kommando ENTER-PROCEDURE gestartete Prozedur MAIL.2010-06-27.134758.PROC die anderweitig definierte Prozedur MAILPROC mit dem Namen der Nachrichten-Datei als Parameter auf. Um den Anhang zu verarbeiten, wird anschließend die Prozedur MAIL.2010-06-27.134758.TESTTXT.PROC aufgerufen, die wiederum die anderweitig definierte Prozedur ATTPROC mit dem Content-Type-Header des Attachments als Parameter aufruft. Weitere Beispiele finden Sie in Abschnitt „Mails mit BS2000/OSD-Prozeduren verarbeiten“ auf Seite 394.

8.2.2 Ereignislogging: Parameterbereich TRACE

Mit dem TRACE-Parametersatz bestimmen Sie, welche Ereignisse protokolliert und wohin die Trace-Einträge geschrieben werden.

8.2.2.1 Syntax

```
TRACE = PARAMETERS( ... )
        | SOCKET_TRACE = <level>
        | ,TLS = <level>
        | ,PROTOCOL = <level>
        | ,IO = <level>
        | ,SMIME = <level>
        | ,HEADER = <level>
        | ,LEVEL = <level>
        | ,PARSE = <level>
        | ,STRING = <level>
        | ,FILENAME = <trace file>
```

Beschreibung der Operanden

SOCKET_TRACE=<level>
Hiermit kann das Logging des SOCKETS-Subsystem aktiviert werden. <level> gibt dabei den Trace-Umfang an. Je größer <level> ist, umso mehr bzw. ausführlichere Trace-Daten werden auf SYSOUT ausgegeben. Der Level 0 deaktiviert den Trace. Der Operand SOCKET_TRACE **i**st unabhängig vom Operanden LEVEL.
Voreingestellt ist ein deaktivierter Socket-Trace.

TLS=<level>
steuert das Logging bestimmter Aktionen/Ereignisse bei der Abwicklung des TLS/SSL-Protokolls.

0	Logging deaktiviert (voreingestellt).
1	Logging aktiviert. Eine Logging-Ausgabe erfolgt nur dann, wenn der Operand LEVEL den Wert 0 hat.

PROTOCOL=<level>
steuert das Logging bestimmter Ereignisse des POP3/IMAP-Protokolls wie z.B. Abholen oder Löschen einer Mail.
Dabei gibt <level> den Trace-Umfang an: Je größer <level> ist, umso mehr Trace-Einträge werden geschrieben. Ein Level von 0 unterdrückt alle Trace-Einträge (voreingestellt). Der maximale Logging-Umfang wird mit dem Wert 2 erreicht. Eine Logging-Ausgabe erfolgt nur dann, wenn der Operand LEVEL den Wert 0 hat.

IO=<level>
erlaubt die Aktivierung eines Mitschnitts des Datenverkehrs mit dem POP3/IMAP-Server.

0	Mitschnitt deaktiviert (voreingestellt).
1	Mitschnitt aktiviert. Eine Logging-Ausgabe erfolgt nur dann, wenn der Operand LEVEL den Wert 0 hat.

SMIME=<level>
steuert das Logging bestimmter Aktionen bei der Abwicklung der S/MIME-Entschlüsselung und/oder -Verifizierung.
Dabei gibt <level> den Trace-Umfang an. Je größer <level> ist, umso mehr Trace-Einträge werden geschrieben. Ein Level von 0 unterdrückt alle Trace-Einträge.
Voreingestellt ist der Wert 0.
Der maximale Logging-Umfang wird mit dem Wert 2 erreicht. Eine Logging-Ausgabe erfolgt nur dann, wenn der Operand LEVEL den Wert 0 hat.

HEADER=<level>
bestimmt den Umfang des Logging von Ereignissen, die beim Erstellen der Prozedur- und Nachrichten-Dateien auftreten.
Dabei gibt <level> den Trace-Umfang an. Je größer <level> ist, umso mehr Trace-Einträge werden geschrieben. Ein Level von 0 unterdrückt alle Trace-Einträge.
Voreingestellt ist der Wert 0.
Der maximale Logging-Umfang wird mit dem Wert 3 erreicht. Eine Logging-Ausgabe erfolgt nur dann, wenn der Operand LEVEL den Wert 0 hat.

PARSE=<level>
bestimmt den Umfang des Trace für Ereignisse beim Einlesen der Konfigurationsdatei.
Dabei gibt <level> den Trace-Umfang an. Je größer <level> ist, umso mehr Trace-Einträge werden geschrieben. Ein Level von 0 unterdrückt alle Trace-Einträge.
Voreingestellt ist der Wert 0.
Der maximale Logging-Umfang wird mit dem Wert 2 erreicht.

LEVEL=<level>
steuert die Ausgabe der PROTOCOL-, IO- und HEADER-Traces und sonstiger Ereignisse aus verschiedenen Meldungsklassen gemäß folgender Tabelle:

0	TLS-, PROTOCOL-, IO-, SMIME- und HEADER-Traces, INFO-, WARNING-, ERROR- und FATAL-Ereignisse werden protokolliert.
1	INFO-, WARNING-, ERROR- und FATAL-Ereignisse werden protokolliert. TLS-, PROTOCOL-, IO-, SMIME- und HEADER-Traces werden **nicht** protokolliert.
2	WARNING-, ERROR- und FATAL-Ereignisse werden protokolliert. TLS-, PROTOCOL-, IO-, SMIME- und HEADER-Traces werden **nicht** protokolliert.
3	ERROR-, FATAL-Ereignisse werden protokolliert. TLS-, PROTOCOL-, IO-, SMIME- und HEADER-Traces werden **nicht** protokolliert.

FILENAME=<trace file>
bestimmt die Ausgabedatei für die Trace-Einträge. Wird dieser Parameter nicht angegeben, dann werden die Trace-Einträge auf SYSOUT ausgegeben.

8.2.2.2 Beispiel

In diesem Beispiel werden alle Ereignisse des POP3-/IMAP-Protokolls mit dem Level 1 auf SYSOUT geschrieben.

```
TRACE = PARAMETERS (
    PROTOCOL = 1,
    LEVEL = 0
)
```

8.2.3 POP3/IMAP-Server: Parameterbereich SERVER

Mit dem SERVER-Parametersatz geben Sie an, von welchem POP3- oder IMAP-Server der Mail-Reader die Mail abholt.

Hierfür werden folgende Angaben benötigt:

– Host-Name des Rechners, auf dem der POP3- bzw. IMAP-Server abläuft

– Benutzerkennung und Passwort für diesen Rechner

– Bei einem IMAP-Server: zusätzlich der MAILBOX-Name

Optional können Sie angeben, wie oft der Mail-Reader kontrolliert, ob neue Nachrichten vorliegen.

Darüber hinaus können Sie eine TLS/SSL-Verschlüsselung der Verbindung zum Server anfordern, damit das zu übertragende Passwort nicht mitgelesen werden kann.

8.2.3.1 Syntax

```
SERVER = POP3 ( ... ) / IMAP (...)
    POP3 ( ... )
        HOSTNAME=<hostname>
        ,PORT=<port number>
        ,USER=<user-id>
        ,PASSWORD=<password>
        ,INTERVAL=<int>
        ,TLS=NO / YES
            YES(. . . )
                REQUIRED=NO / YES
                ,MODE=IMPLICIT / EXPLICIT
                ,PROTOCOL=<protocol spec>
                ,CIPHER_SUITE=<cipher suite spec>
                ,CERTIFICATE=<certificate file>
                ,KEY=<private key file>
                ,CA_CERTIFICATES=<CA certificates file>
                ,CRL=<CRL file>
                ,VERIFY_SERVER=YES / NO
                ,VERIFY_DEPTH=<depth>
                ,USE_CRYPTO_HARDWARE=NO / YES
    ,IMAP ( ... )
        HOSTNAME=<hostname>
        ,PORT=<port number>
        ,USER=<user-id>
        ,PASSWORD=<password>
        ,KEEP=NO / YES
        ,MAILBOX=<mailbox>
        ,INTERVAL=<int>
        ,TLS=NO / YES
            YES(. . . )
                REQUIRED=NO / YES
                ,MODE=IMPLICIT / EXPLICIT
                ,PROTOCOL=<protocol spec>
                ,CIPHER_SUITE=<cipher suite spec>
                ,CERTIFICATE=<certificate file>
                ,KEY=<private key file>
                ,CA_CERTIFICATES=<CA certificates file>
                ,CRL=<CRL file>
                ,VERIFY_SERVER=YES / NO
                ,VERIFY_DEPTH=<depth>
                ,USE_CRYPTO_HARDWARE=NO / YES
```

Beschreibung der Operanden

SERVER=POP3(. . .) / IMAP(. . .)
gibt an, welches Protokoll für den Zugriff auf die Mails verwendet werden soll.

SERVER=POP3
Das POP3-Protokoll soll verwendet werden.

SERVER=IMAP
Das IMAP-Protokoll soll verwendet werden.

HOSTNAME=<hostname>
Der Server läuft auf dem Host <hostname>. Der <hostname> kann dabei der BCAM-Prozessor- oder der DNS-Name des Servers sein.

PORT=<port number>
gibt die Port-Nummer an, die beim Aufbau der TCP-Verbindung zum Server verwendet werden soll. Wird dieser Parameter verwendet, dann wird bei Nutzung von TLS/SSL die automatische Port-Nummern-Wahl in Abhängigkeit vom MODE-Parameter außer Kraft gesetzt und in jedem Fall die angegebene Port-Nummer verwendet.

USER=<user-id>
User-ID des Benutzers, auf dessen Mailbox zugegriffen werden soll.
Die ID ist in einfache Anführungszeichen eingeschlossen.
Je nach Server ist die Groß-/Kleinschreibung relevant.

PASSWORD=<password>
Passwort für den Zugriff auf die Mailbox.
Das Passwort ist in einfache Anführungszeichen eingeschlossen.
Je nach Server ist die Groß-/Kleinschreibung relevant.

KEEP=<u>NO</u> / YES
(nur für IMAP-Server)
gibt an, ob die jeweilige Mail nach dem Herunterladen durch den Mail-Reader auf dem Server gelöscht werden soll.

KEEP=<u>NO</u>
(nur für IMAP-Server)
Die Mail wird nach dem Herunterladen auf dem Server gelöscht.

KEEP=YES
(nur für IMAP-Server)
Nach dem Herunterladen wird die jeweilige Mail als gesehen markiert, bleibt aber auf dem Server stehen. Sie kann dann z.B. mit einem Standard-Mail-Client wieder als nicht gesehen markiert und damit einer erneuten Verarbeitung zugeführt werden.
Dieser Parameter-Wert bietet sich daher vor allem für Tests an. Im Produktiv-Betrieb muss durch geeignete Maßnahmen sichergestellt werden, dass die Mails rechtzeitig gelöscht werden, da andernfalls nach einer gewissen Zeit ein Resourcen-Engpass auf dem IMAP-Server auftritt.

MAILBOX=<mailbox>
(nur für IMAP-Server)
IMAP-Mailbox, auf die zugegriffen werden soll.
Voreinstellung: INBOX

INTERVAL=<int>
Zeitintervall in Sekunden, nach dem der Mail-Reader kontrolliert, ob neue Nachrichten vorliegen. Voreinstellung: 900 Sekunden.
0: genau einmal kontrollieren

TLS=NO / YES
gibt an, ob die Verbindung zum Server mit TLS/SSL abgesichert werden soll.

TLS=NO
Die Verbindung zum Server soll nicht abgesichert werden.

TLS=YES
Die Verbindung zum Server soll abgesichert werden.

REQUIRED=NO / YES
gibt an, ob die Verbindung zum Server abgebrochen werden soll, wenn der Server kein TLS/SSL unterstützt.

REQUIRED=NO
Auch wenn der Server kein TLS/SSL unterstützt, wird der Mail-Transfer durchgeführt. Wird beim Vergleich des Host-Namens mit dem Server-Namen im X.509-Zertifikat eine Nicht-Übereinstimmung festgestellt, dann wird dennoch ein Mail-Transfer durchgeführt.

REQUIRED=YES
Wenn der Server kein TLS/SSL unterstützt, wird die Verbindung abgebrochen. Wird beim Vergleich des Host-Namens mit dem Server-Namen im X.509-Zertifikat eine Nicht-Übereinstimmung festgestellt, so wird der Mail-Reader mit einer Fehlermeldung beendet.

MODE=IMPLICIT / EXPLICIT
gibt an, ob die TLS/SSL-Verbindung zum Server implizit nach dem TCP-Verbindungsaufbau aufgebaut werden soll.

MODE=IMPLICIT
Direkt nach dem TCP-Verbindungsaufbau erfolgt ein impliziter Aufbau einer TLS/SSL-Verbindung. Wenn der PORT-Parameter nicht verwendet wurde, wird für den TCP-Verbindungsaufbau die Port-Nummer 995 (POP3-Server) bzw. 993 (IMAP-Server) verwendet.

MODE=EXPLICIT
Falls der PORT-Parameter nicht verwendet wurde, erfolgt zunächst ein TCP-Verbindungsaufbau unter Verwendung der Standard-Portnummer 110 (POP3-Server) bzw. 143 (IMAP-Server). Danach wird mit dem POP3-Kommando STLS bzw. dem IMAP-Kommando STARTTLS der Aufbau einer TLS/SSL-Verbindung initiiert. Dabei ist zu beachten, dass manche Server (z.B. der UW POP3/IMAP-Server) nur den Aufbau einer TLSv1-, nicht aber einer SSLv2- oder SSLv3-Verbindung zulassen.

PROTOCOL=<protocol spec>
Sie können die verwendeten Protokolle einschränken. Grundsätzlich werden SSL Version 2 und 3 und TLS Version 1 unterstützt.
Erlaubt sind die Angaben SSLv2, SSLv3, TLSv1 und All.
Je nachdem, ob das entsprechende Protokoll aktiviert oder deaktiviert werden soll, können Sie dem jeweiligen Protokoll ein Plus- oder Minus-Zeichen voranstellen. Die Eingaben „SSLv3 TLSv1“ und „All -SSLv2“ haben den gleichen Effekt.
Voreinstellung: SSLv3 TLSv1
Manche Server unterstützen nur TLS, wenn MODE=EXPLICIT verwendet wird. In diesem Fall muss PROTOCOL=TLSv1 verwendet werden, da andernfalls der Aufbau der TLS/SSL-Verbindung scheitert.

CIPHER_SUITE=<cipher suite spec>
Die Cipher-Suite-Spezifikation wird in eine Vorzugsliste für Verschlüsselungsalgorithmen umgewandelt. Die Spezifikation besteht aus einer oder mehreren Chiffre-Mnemonics, die durch einen Doppelpunkt (:) getrennt sind.

Ein Chiffre-Mnemonic kann folgende Formen annehmen:

- Ein Chiffre-Mnemonic kann aus einer einzelnen Verschlüsselungs-Suite wie z.B. DES-CBC-SHA bestehen.
- Ein Chiffre-Mnemonic kann repräsentieren:
  - eine Liste von Verschlüsselungs-Suites, die einen bestimmten Algorithmus enthalten.
  - Verschlüsselungs-Suites eines bestimmten Typs.

  Beispiele

  SHA1 repräsentiert alle Verschlüsselungs-Suiten, die den Digest-Algorithmus SHA1 benutzen.

  SSLv3 repräsentiert alle SSL Version 3 Algorithmen.

- Listen von Verschlüsselungs-Suiten können mithilfe des „+“-Zeichens zu einem einzelnen Chiffre-Mnemonic kombiniert werden. Dieses wird dann als logische UND-Operation interpretiert.

  Beispiel

  SHA1+DES repräsentiert alle Verschlüsselungs-Suiten, die die SHA1- und DES-Algorithmen enthalten.

- Jedem Chiffre-Mnemonic kann optional eines der Zeichen „!“, „-“ oder „+“ vorangestellt werden:

  – Bei Voranstellen von „!“ werden die betreffenden Verschlüsselungs-Suiten dauerhaft aus der Vorzugsliste gelöscht. Sie erscheinen auch dann nicht wieder in der Vorzugsliste, wenn sie explizit angegeben werden.

  – Bei Voranstellen von „-“ werden die betreffenden Verschlüsselungs-Suiten aus der Vorzugsliste gelöscht, aber einige von ihnen oder alle können durch nachfolgende Optionen wieder hinzugefügt werden.

  – Bei Voranstellen von „+“ werden die betreffenden Verschlüsselungs-Suiten an das Ende der Vorzugsliste verschoben. Hiermit werden keine Verschlüsselungs-Suiten zur Vorzugsliste hinzugefügt, sondern nur existierende verschoben.

  – Wenn keines der drei Zeichen „!“, „-“ oder „+“ vorangestellt ist, wird der Chiffre-Mnemonic als eine Liste von Verschlüsselungs-Suiten interpretiert, die an die aktuelle Vorzugsliste angehängt wird. Schließt dies eine Verschlüsselungs-Suite ein, die schon in der aktuellen Vorzugsliste enthalten ist, dann wird sie ignoriert und nicht an das Ende der Vorzugsliste verschoben.

- Der Chiffre-Mnemonic @STRENGTH kann an beliebiger Stelle eingefügt werden, um die aktuelle Vorzugsliste nach der Stärke der Verschlüsselungsalgorithmen (d.h. vor allem nach der Länge der Verschlüsselungsschlüssel) zu sortieren.

Nachfolgend sind die zulässigen Chiffre-Mnemonics beschrieben:

ALL Alle Verschlüsselungs-Suiten mit Ausnahme der eNULL-Chiffren. eNULL-Chiffren müssen explizit aktiviert werden.

HIGH Verschlüsselungs-Suiten mit Schlüssellänge größer 128 Bit. Da 3DES mit 168 Bit Länge (anstatt 112 Bit wie von manchen Kryptologen) bewertet wird, zählt es zu dieser Suiten-Klasse.

MEDIUM
Verschlüsselungs-Suiten mit Schlüssellänge 128 Bit.

LOW Verschlüsselungs-Suiten mit 64 oder 56 Bit Schlüssellänge, ausgenommen Export-Verschlüsselungs-Suiten.

EXP, EXPORT
Export-Verschlüsselungs-Algorithmen einschließlich 40- und 56-Bit-Algorithmen.

EXPORT40
40-Bit-Export-Verschlüsselungs-Algorithmen.

EXPORT56
56-Bit-Export-Verschlüsselungs-Algorithmen.

eNULL, NULL
NULL-Verschlüsselungs-Algorithmen, d.h. solche ohne Verschlüsselung. Da diese keine Verschlüsselung bieten und damit ein Sicherheitsrisiko sind, werden sie standardmäßig deaktiviert und müssen gegebenenfalls explizit angegeben werden.

aNULL
Verschlüsselungs-Suiten ohne Authentizierung. Dies sind im Augenblick Diffie-Hellman-Algorithmen. Diese Algorithmen sind anfällig für „man in the middle“-Angriffe, so dass von ihrer Benutzung abgeraten wird.

kRSA, RSA
Verschlüsselungs-Suiten mit RSA-Schlüsselaustausch.

kEDH Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüsselvereinbarung.

aRSA Verschlüsselungs-Suiten mit RSA-Authentizierung, d.h. die Zertifikate enthalten RSA-Schlüssel.

aDSS, DSS
Verschlüsselungs-Suiten mit DSS-Authentizierung, d.h. die Zertifikate enthalten DSS-Schlüssel.

TLSv1, SSLv3, SSLv2
: TLS Version 1, SSL Version 3 bzw. SSL Version 2 Verschlüsselungs-Suiten. Die TLSv1-Suiten und die SSLv3-Suiten sind identisch.

DH Verschlüsselungs-Suiten mit Diffie-Hellman-Schlüsselaustausch, einschließlich anonymem Austausch.

ADH Verschlüsselungs-Suiten mit anonymem Diffie-Hellman-Schlüsselaustausch.

AES Verschlüsselungs-Suiten mit AES-Verschlüsselung (128 und 256 Bit Schlüssellänge).

3DES Verschlüsselungs-Suiten mit Triple-DES-Verschlüsselung.

DES Verschlüsselungs-Suiten mit DES-Verschlüsselung (kein Triple-DES).

RC4 Verschlüsselungs-Suiten mit RC4-Verschlüsselung.

RC2 Verschlüsselungs-Suiten mit RC2-Verschlüsselung.

MD5 Verschlüsselungs-Suiten mit MD5-Hash-Funktion.

SHA1, SHA
: Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.

In der nachfolgenden Tabelle sind die verfügbaren Verschlüsselungs-Suiten zusammengefasst.

Name	Version	Schlüsselaustausch	Authentifizierung	Verschlüsselung	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)	SHA	
DHE-RSA-AES128-SHA	SSLv3	DH	RSA	AES(128)	SHA	
DHE-DSS-AES128-SHA	SSLv3	DH	DSS	AES(128)	SHA	
AES128-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	

Verfügbare Verschlüsselungs-Suiten

Name	Version	Schlüssel-austausch	Authentifi-zierung	Verschlüs-selung	Digest	Export
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	keine	AES(256)	SHA1	
ADH-AES128-SHA	SSLv3	DH	keine	AES(128)	SHA1	
ADH-DES-CBC3-SHA	SSLv3	DH	keine	3DES(168)	SHA1	
ADH-DES-CBC-SHA	SSLv3	DH	keine	DES(56)	SHA1	
ADH-RC4-MD5	SSLv3	DH	keine	RC4(128)	MD5	
EXP-ADH-DES-CBC-SHA	SSLv3	DH(512)	keine	DES(40)	SHA1	export
EXP-ADH-RC4-MD5	SSLv3	DH(512)	keine	RC4(40)	MD5	export
NULL-SHA	SSLv3	RSA	RSA	keine	SHA	
NULL-MD5	SSLv3	RSA	RSA	keine	MD5	

Verfügbare Verschlüsselungs-Suiten

CERTIFICATE=<certificate file>
spezifiziert eine Datei, die das X.509-Client-Zertifikat zur Client-Authentizierung im PEM-Format enthält. Diese Datei kann auch den privaten Schlüssel des Client enthalten. In der Regel werden aber Zertifikat und Schlüssel in getrennten Dateien abgelegt.

KEY=<private key file>
spezifiziert eine Datei, die den privaten Client-Schlüssel im PEM-Format enthält. Wenn sowohl X.509-Zertifikat als auch privater Client-Schlüssel innerhalb derselben Datei enthalten sind (siehe Operand CERTIFICATE), dann braucht dieser Operand nicht angegeben zu werden.

Der Client-Schlüssel sollte nicht mit einer Passphrase geschützt werden, da sonst bei jedem Mail-Reader-Start die Passphrase eingegeben werden müsste.
Durch entsprechende Dateiattribute muss allerdings dafür gesorgt werden, dass Unbefugte keinen Zugriff auf den privaten Schlüssel erhalten.

CA_CERTIFICATES=<CA certificates file>
spezifiziert eine Datei, die die für die Authentifizierung des Servers erforderlichen CA-Zertifikate im PEM-Format enthält. Die einzelnen PEM-Zertifikate sind sequenziell in der Datei angeordnet.

Zum Hinzufügen oder Löschen von Zertifikaten kann die Datei mit einem beliebigen Text-Editor bearbeitet werden. Die einzelnen Zertifikate sind wie folgt in der Datei notiert:

```
-----BEGIN CERTIFICATE-----
< CA-Zertifikat in Base64-Kodierung >
-----END CERTIFICATE-----
```

Text außerhalb dieser Sequenzen wird vom Mail-Reader ignoriert und kann deshalb zum Kennzeichnen der Zertifikate verwendet werden, die wegen der ASN.1/Base64-Kodierung in nicht lesbarer Form vorliegen.

CRL=<CRL file>
spezifiziert eine Datei, die die CRLs (Certificate Revocation List) der Zertifizierungsinstanzen (Certificate Authority, CA) enthält. (Zertifikate, die von einer Zertifizierungsinstanz herausgegeben wurden, können durch Veröffentlichung in einer CRL für ungültig erklärt werden.)

VERIFY_SERVER=YES / NO
legt fest, ob das Zertifikat des Servers verifiziert werden soll.

VERIFY_SERVER=YES
Das Zertifikat des Servers soll verifiziert werden.

VERIFY_SERVER=NO
Das Zertifikat des Servers soll nicht verifiziert werden. Diese Einstellung ist anfällig für „man in the middle“-Angriffe.

VERIFY_DEPTH=<depth>
legt die sogenannte Verifizierungstiefe fest, d.h. die Anzahl der maximal zulässigen Zertifikate zwischen dem Zertifikat des Servers und dem Zertifikat, das dem Mail-Reader bekannt ist. Wird die maximale Verifizierungstiefe überschritten, dann wird die Verbindung abgebrochen, sofern nicht aufgrund von VERIFY-SERVER=NO die Verifizierung des Server-Zertifikats ausgeschaltet ist.

1 Server-Zertifikat wird nur akzeptiert, wenn es direkt von einer dem Mail-Reader bekannten Certificate Authority (CA) signiert worden ist.

0 Nicht sinnvoll, da in diesen Fällen nur selbstsignierte Zertifikate zulässig sind.

Standardeinstellung: 9

USE_CRYPTO_HARDWARE=NO / YES
legt fest, ob zur Berechnung kryptographischer Algorithmen Krypto-Hardware, z.B. eine openCRYPT™-BOX, verwendet werden soll.

USE_CRYPTO_HARDWARE=NO
Es wird keine Krypto-Hardware verwendet.

USE_CRYPTO_HARDWARE=YES
Es wird Krypto-Hardware verwendet.

8.2.3.2 Beispiel

In diesem Beispiel pollt der Mail-Reader unter der Benutzerkennung *inetvalu* mit dem Passwort *geheim* alle 600 Sekunden auf den Server *localhost*.

```
SERVER=POP3(
     HOSTNAME=localhost,
     USER='inetvalu',
     PASS='geheim',
     INTERVAL=600)
)
```

8.3 Mail-spezifische Parameter substituieren

Mit den Substitutionsregeln können in den Prozedur-Dateien und Nachrichten-Dateien Mail-spezifische Parameter substituiert werden.

Die Regeln legen insbesondere folgende Substitutionen fest:

- Namen der generierten Dateien
- Mail-Bestandteile, insbesondere Header-Felder, zum Beispiel wird *%SUBJECT%* durch die Betreff-Zeile der Mail ersetzt.

Groß-/Kleinschreibung der Schlüsselwörter wird nicht ausgewertet. Außerdem gilt die Regel, dass %% durch % und \' bzw. '' durch ' ersetzt wird. In den nachfolgenden Unterabschnitten werden die verwendbaren Schlüsselwörter erläutert.

8.3.1 Schlüsselwörter zur Substitution von Dateienamen

Dateinamen werden entsprechend der folgenden Schlüsselwörter substituiert:

FILE-NAME
: Name der erzeugten Nachrichten-Datei.

PROCEDURE-NAME
: Name der erzeugten Prozedur-Datei für die Verarbeitung der Nachrichten-Datei.

ATTACHMENT-FILE-NAME
: Name der erzeugten Attachment-Datei.

ATTACHMENT-PROCEDURE-NAME
: Name der erzeugten Prozedur-Datei für die Verarbeitung der Attachment-Datei.

Die Dateinamen werden aus mehreren Bestandteilen gebildet, die mit eingeschobenen Punkten zu einem Dateinamen zusammengesetzt werden:

- Dem in der Kongfiurationsdatei oder per Voreinstellung definierten Präfix
- Datum und Uhrzeit in der Form JJJJ-MMDD.HHMMSS
- Eine aus dem *name*-Operanden des Content-Type-Headers gebildeten Zeichenfolge, die wie folgt gebildet wird: Es werden vom Anfang max. soviele Zeichen genommen, bis die Länge des gesamten Dateinamens 38 Zeichen erreicht, dabei werden nur alphanumerische Zeichen (Buchstaben und Ziffern) berücksichtigt, d.h. aus einem `datei1.txt` wird ein `datei1txt`.
- Dem in der Konfigurationsdatei oder per Voreinstellung definierten Suffix (bei den Prozedurdateien).

8.3.2 Schlüsselwörter zur Substitution von Mail-Bestandteilen

Mail-Bestandteile werden entsprechend der folgenden Schlüsselwörter substituiert.

Ersetzt werden die im Folgenden aufgelisteten, standardmäßigen Mail-Header, wenn sie von „%“ eingeschlossen sind (%...%).

Die zugehörige Beschreibung entnehmen Sie folgenden RFCs:
- RFC 822 bzw RFC 2822 (Standard Mail)
- RFC 1036 (USENET Messages)
- RFC 2045 (MIME Messages)

Bcc	Sender
Cc	Subject
Comments	To
Date	Approved
Encrypted	Control
From	Distribution
In-Reply-To	Expires
Keywords	Followup-To
Message-Id	Lines
Received	Newsgroups
References	Organization
Reply-To	Path
Resent-Bcc	Summary
Resent-Cc	Xref
Resent-Date	Content-Description
Resent-From	Content-Id
Resent-Message-Id	Content-Transfer-Encoding
Resent-Reply-To	Cte (synonym: Content-Transfer-Encoding)
Resent-Sender	Content-Type
Resent-To	Mime-Version
Return-Path	Content-Disposition

Die Mail-Header werden nur ersetzt, wenn die Header-Felder in der betreffenden Mail enthalten sind.

Das Schlüsselwort „Text“ wird durch den Nachrichtentext der betreffenden Mail ersetzt. Das Schlüsselwort „Header“ wird durch die Header-Zeilen ersetzt.

Beachten Sie außerdem, dass

- alle enthaltenen Zeilenumbrüche aus dem Header entfernt werden,
- Groß- und Kleinschreibung nicht ausgewertet wird,
- „%%“ durch „%“ ersetzt wird,
- „\’ “ und „ ” “ durch „’ “ ersetzt wird.

Parameter ausgewählter Header-Felder

Um den Zugriff auf ausgewählte Mail-Header zu erleichtern, werden folgende zusätzlichen Substituierungen durchgeführt:

Content-Type-Type
: das Feld *Type* des Content-Type Headers.

  Beispiel:
  Bei *Content-Type: text/plain;name="filename.txt"* lautet es *text*.

Content-Type-Subtype
: das Feld *Subtype* des Content-Type Headers.

  Beispiel:
  Bei *Content-Type: text/plain;name="filename.txt"* lautet es *plain*.

Content-Type-Name
: das Feld *name* des Content-Type Headers.

  Beispiel:
  Bei *Content-Type: text/plain;name="filename.txt"* lautet es *"filename.txt"*.

Content-Disposition-Name
: das Feld *Filename* des Content-Disposition Headers

  Beispiel:
  Bei *Content-Disposition: inline; filename="filename.txt"* lautet es *"filename.txt"*.

Disposition-Type-Name
: das Feld *Filename* des Content-Disposition Headers

  Beispiel:
  Bei *Content-Disposition: inline; filename="filename.txt"* lautet es *"filename.txt"*.

8.3.3 S/MIME

Damit bei S/MIME-signierten und/oder -verschlüsselten Mails der Erfolgsstatus der Entschlüsselung bzw. der Verifizierung der Signatur zurückgeliefert werden kann, werden folgende Schlüsselwörter unterstützt:

SMime-Decryption-Status
: „NOT-ENCRYPTED“, wenn die Mail nicht verschlüsselt war,
„OK“, wenn die Mail erfolgreich entschlüsselt werden konnte,
ansonsten eine kurze Fehlerbeschreibung

SMime-Verification-Status
: „NOT-SIGNED“, wenn die Mail nicht signiert war,
„OK“, wenn die Mail-Signatur erfolgreich verifiziert werden konnte,
ansonsten eine kurze Fehlerbeschreibung

SMime-Signer-DN
: DN (Distinguished Name) des Mail-Signierers.

SMime-Signer-CN
: CN (Common Name) des Mail-Signierers.

SMime-Signer-E-Mail
: Mail-Adresse(n) des Mail-Signierers, durch Komma getrennt; mehrere Mail-Adressen können bei X.509v3-Zertifikaten auftreten, wenn sowohl im DN als auch im SubjectAltName eine Mail-Adresse angegeben ist

SMime-Issuer-DN
: DN (Distinguished Name) der Certificate Authority (CA), die das Mail-Signierer-Zertifikat ausgestellt hat.

Bei multipart/mixed-Nachrichten werden die Informationen über die Entschlüsselung und/oder Signatur-Verifizierung nur dem Nachrichten-Text-Teil, nicht aber den Attachment-Teilen zugeordnet. Bei letzteren wird also *%SMime-Decryption-Status%* zu *NOT-ENCRYPTED* und *%SMime-Verication-Status%* zu *NOT-SIGNED* expandiert, es sei denn, das Attachment ist wiederum eine S/MIME-signierte oder -verschlüsselte Mail.

8.4 Mail prozedural verarbeiten

Innerhalb der BS2000/OSD-Prozeduren werden unabhängig vom Aufbau der Mail alle Mail-Bestandteile (Header, Attachments) zur Verfügung gestellt.

8.4.1 Aufbau einer Mail

Eine Mail kann unterschiedlich komplex aufgebaut sein:

- Im einfachsten Fall besteht sie aus Kopfzeilen (Headern) und einer Nachricht.
- Die Mail kann aber auch Anhänge (Attachments) enthalten. Dann besteht die Mail aus den allgemeinen Headern, wie „Subject" und „From", aus der Nachricht (mit ihren eigenen Headern) und den Anhängen jeweils wieder mit eigenen Headern.
- Der komplizierteste Fall liegt vor, wenn die Mail als Anhang eine weitere Mail enthält. Diese kann wieder Anhänge und/oder andere Mails als Anhänge enthalten usw.

Der Mail-Reader analysiert die Mail und unterteilt die Mail für die prozedurale Verarbeitung in kleinere Pakete und stellt diese einzeln verarbeitbaren Pakete den Prozeduren zur Verfügung. Dies erlaubt innerhalb der Prozeduren den Zugriff auf die Header der Mail und auf die einzelnen Attachments, unabhängig davon wie kompliziert die Mail aufgebaut ist.

8.4.2 Mails mit BS2000/OSD-Prozeduren verarbeiten

8.4.2.1 Mail ohne Anhang

Bei einer Mail ohne Anhang erzeugt der Mail-Reader die beiden folgenden Dateien:

- Eine Datei enthält die Nachricht der Mail.
- Eine Datei enthält die Prozedur für die Verarbeitung der Mail.
  Der Aufbau der erzeugten Prozeduren ist in weiten Bereichen frei konfigurierbar.

Außerdem generiert der Mail-Reader für jede erhaltene Mail einen neuen Namen.

Die Prozedur wird anschließend mit einem `/ENTER-JOB`-Kommando bzw. `/ENTER-PROCEDURE`-Kommando aufgerufen.

Beispiel 1

Das nachfolgende Beispiel zeigt, wie Sie den Mail-Reader so konfigurieren, dass die erzeugte Prozedur nur eine Zeile enthält. Alle erhaltenen Mails werden ausgedruckt.

```
/PRINT-FILE MAIL.2010-06-27.073001,DELETE-FILE=*YES
```

MAIL.2010-06-27.073001 ist dabei der eindeutige Name, unter dem der Nachrichtentext abgelegt wird.

Der Name der Prozedur lautet entsprechend *MAIL.2010-06-27.073001.PROC*.

Der Mail-Reader ruft anschließend die Prozedur *MAIL.2010-06-27.073001.PROC* auf, und diese wiederum startet den Ausdruck mit */PRINT-FILE*.

Der entsprechende Ausschnitt aus der Konfigurationsdatei muss folgendermaßen aussehen:

```
MAILHANDLING = PARAMETERS(
    ENTER = PROCEDURE,
    BODY= PARAMETERS (
        PROCEDURE = PARAMETERS (
           TEXT = ('/PRINT-FILE %FILE-NAME%,DELETE-FILE=*YES'
           )
        )
    )
)
```

Der Schlüsselwort-Ausdruck *%FILE-NAME%* wird vom Mail-Reader durch den jeweils aktuellen Namen der Nachrichten-Datei substituiert.

Beispiel 2

Ebenso können die meisten Header durch solche Variablen substituiert werden. Im nachfolgenden Beispielausschnitt aus der Konfigurationsdatei wird %SUBJECT% durch den Betreff der Mail ersetzt.

```
    TEXT = ('/PRINT-FILE %FILE-NAME%,DELETE-FILE=*YES,',
            '/  COVER-PAGES=PAR(HEADER-PAGE-TEXT=''Betreff: %SUBJECT%'')'
    )
```

Dabei wird ′′ zu ′.

Es entsteht beispielsweise folgende Prozedur:

```
/PRINT-FILE MAIL.2010-06-27.073001,DELETE-FILE=*YES,-
/  COVER-PAGES=PAR(HEADER-PAGE-TEXT='Betreff:Mail-Reader 2010')
```

Beispiel 3

Für jede erhaltene Mail wird die Prozedur *HANDLEMAIL* aufgerufen. Der Betreff und der Dateiname der Mail werden dabei als Parameter übergeben.

Der entsprechende Ausschnitt aus der Konfigurationsdatei lautet beispielsweise:

```
MAILHANDLING = PARAMETERS(
    ENTER = PROCEDURE,
    BODY = PARAMETERS (
        PROCEDURE = PARAMETERS (
           TEXT = ('/CALL-PROCEDURE HANDLEMAIL,(FILENAME=%FILE-NAME%,-',
                   '/   Subject=''%SUBJECT%'')'
           )
        )
    )
)
```

8.4.2.2 Mail mit Anhängen

Eine Mail enthält mehrere Anhänge.
Die Mail besteht also aus folgenden Komponenten:

- Header
- Nachricht mit Headern
- Anhang 1 mit Headern
- Anhang 2 mit Headern
- usw.

Der Mail-Reader erzeugt dementsprechend folgende Dateien:

- zwei Dateien für die Nachricht:
  - eine Datei für die Prozedur zur Verarbeitung der Nachricht
  - eine Datei für den Messagebody
- jeweils zwei Dateien für jeden Anhang :
  - Eine Datei beinhaltet jeweils das Attachment.
  - Die andere Datei enthält eventuell eine Prozedur zum Verarbeiten des Attachments.

  Der Name der Dateien, die für die Anhänge erzeugt werden, setzt sich zusammen aus dem Präfix, das auch für die Inline-Nachricht verwendet wird, und einem von den Content-Type oder Content-Disposition Headern vorgeschlagenen Dateinamen.

Beispiel 1: Komplette Mail verarbeiten

Die Beispiel-Mail hat zwei Anhänge, *datei1.txt* und *datei2.txt*.
Wenn Sie die Mail vollständig ausdrucken wollen inklusive der Anhänge *datei1.txt* und *datei2.txt*, lautet der entsprechende Auszug aus der Konfigurationsdatei beispielsweise folgendermaßen:

```
MAILHANDLING = PARAMETERS(
    ENTER = PROCEDURE,
    BODY = PARAMETERS(
      PROCEDURE = PARAMETERS (
        TEXT = ('/PRINT-FILE %FILE-NAME%'),
        ATTACHMENT = ('/PRINT-FILE %ATTACHMENT-FILE-NAME%,-',
                      '/ COVER-PAGES=PAR(HEADER-PAGE-TEXT=''%CONTENT-TYPE%'')'
          )
       )
    )
)
```

Dies führt zu folgender Prozedur *MAIL.2010-06-27.073001.PROC*:

```
/PRINT-FILE MAIL.2010-06-27.073001
/PRINT-FILE MAIL.2010-06-27.073002.datei1txt,-
/ COVER-PAGES=PAR(HEADER-PAGE-TEXT='text/plain; name:“datei1.txt“')
/PRINT-FILE MAIL.2010-06-27.073002.datei2txt,-
/ COVER-PAGES=PAR(HEADER-PAGE-TEXT='text/plain; name:“datei2.txt“')
```

Der Ausdruck *%CONTENT-TYPE%* in der Konfigurationsdatei wird durch den erhaltenen Header der Mail substituiert.

Beispiel 2: Anhänge verarbeiten

Die Beispiel-Mail hat zwei Anhänge, *datei1.txt* und *datei2.txt*.
Für die Anhänge *datei1.txt* und *datei2.txt* können Sie eigene Prozeduren erzeugen, z. B. mit nachfolgender Konfigurationsdatei:

```
MAILHANDLING = PARAMETERS(
    ENTER = PROCEDURE,
    BODY = PARAMETERS(
        PROCEDURE = PARAMETERS (
            TEXT = ('/PRINT-FILE %FILE-NAME%'
            ),
            ATTACHMENT = ('/CALL-PROC %ATTACHMENT-PROCEDURE-NAME%,-',
                          '/  (FILENAME=%ATTACHMENT-FILE-NAME%)'
            )
        )
    ),
    ATTACHMENT = PARAMETERS (
        PROCEDURE = ('/BEG-PAR-DECL',
                     '/DECL-PAR FILENAME',
                     '/END-PAR-DECL',
                     '/CALL-PROCEDURE ATTPROC,(FILENAME=&(FILENAME))'
        )
    )
)
```

Dies führt zu folgender Prozedur *MAIL.2010-06-27.073001.PROC*:

```
/PRINT-FILE MAIL.2010-06-27.073001
/CALL-PROC MAIL.2010-06-27.073002.datei1txt.PROC,-
/  (FILENAME=MAIL.2010-06-27.073002.datei1txt)
/CALL-PROC MAIL.2010-06-27.073002.datei2txt.PROC,-
/  (FILENAME=MAIL.2010-06-27.073002.datei2txt)
```

Zusätzlich werden noch die Prozeduren *MAIL.2010-06-27.073002.datei1txt.PROC* und *MAIL.2010-06-27.073002.datei2txt.PROC* erzeugt.

```
/BEG-PAR-DECL
/DECL-PAR FILENAME
/END-PAR-DECL
/CALL-PROCEDURE ATTPROC,(FILENAME=&(FILENAME))
```

8.5 Programm-Schnittstelle (C++)

Eine komplette Beschreibung der Programmschnittstelle finden Sie als PDF-Datei in der LMS-Bibliothek *SYSLIB.MAIL.nnn* (X-Element mit dem Namen *MIMELIB.PDF*).

Um die C++-Schnittstelle zu verwenden, muss der Benutzer eine Funktion folgenden Prototyps zur Verfügung stellen:

```
int handle_mail(DwMessage *);
```

Diese Funktion wird vom Mail-Reader beim Erhalt einer Mail aufgerufen, und die Mail wird als `DwMessage` übergeben. In der Funktion kann der Benutzer die Mail verarbeiten. Wird anschließend 0 zurückgegeben, so löscht der Mail-Reader die Mail aus dem Postkorb, ansonsten verbleibt die Mail im Postkorb und es wird später versucht, diese Mail erneut zu verarbeiten.

9 Mail-Sender in BS2000/OSD

Der Mail-Sender ist ein Benutzeragent (Mail User Agent, MUA), mit dem Sie im BS2000/OSD Daten als Mail versenden können.

Der BS2000/OSD Mail-Sender bietet Ihnen folgende Möglichkeiten:

- Aus BS2000/OSD-Prozeduren heraus automatisiert Listen als Mail an den lokalen Mail-Server in POSIX oder an ferne Mail-Server versenden.
- Benachrichtigungen in Fehlersituationen übermitteln.
- Text- oder Binärdateien vom BS2000/OSD aus versenden.
- Transfer-Codierung (Transfer-Encoding) spezifizieren (7-Bit- und 8-Bit-Codierung, eine Quoted-printable-Codierung sowie Base64-Codierung).
- to-, cc-und bcc-Empfänger spezifizieren.
- Zusätzliche Übersetzungstabellen für nationale Zeichensätze spezifizieren.
- Mails über eine mit TLS abgesicherte SMTP-Verbindung zum SMTP-Server senden.
- Mails korrekt nach S/MIME-Standard verschlüsseln und signieren.
- Mails asynchron versenden.

Die zuletzt genannte Funktion basiert auf dem BS2000-Subsystem ASTI (**A**ssistant for **S**ervice **T**ask **I**ntegration). Das Frontend des Mail-Senders, das als DSSM-Subsystem MAILCLNT realisiert ist, nimmt die Mails an der SDF-Kommando-, SVC- oder TPR-Schnittstelle entgegen und erstellt Mail-Aufträge, die es an ASTI übergibt. Das Mail-Sender-Backend ist eine TU Service Task, die die Mails von ASTI übernimmt und über eine SMTP-Verbindung an einen Mail-Server weiterleitet.

Die folgende Abbildung veranschaulicht die Position des Mail-Senders im Zusammenwirken mit den anderen Mail-Services im BS2000/OSD.

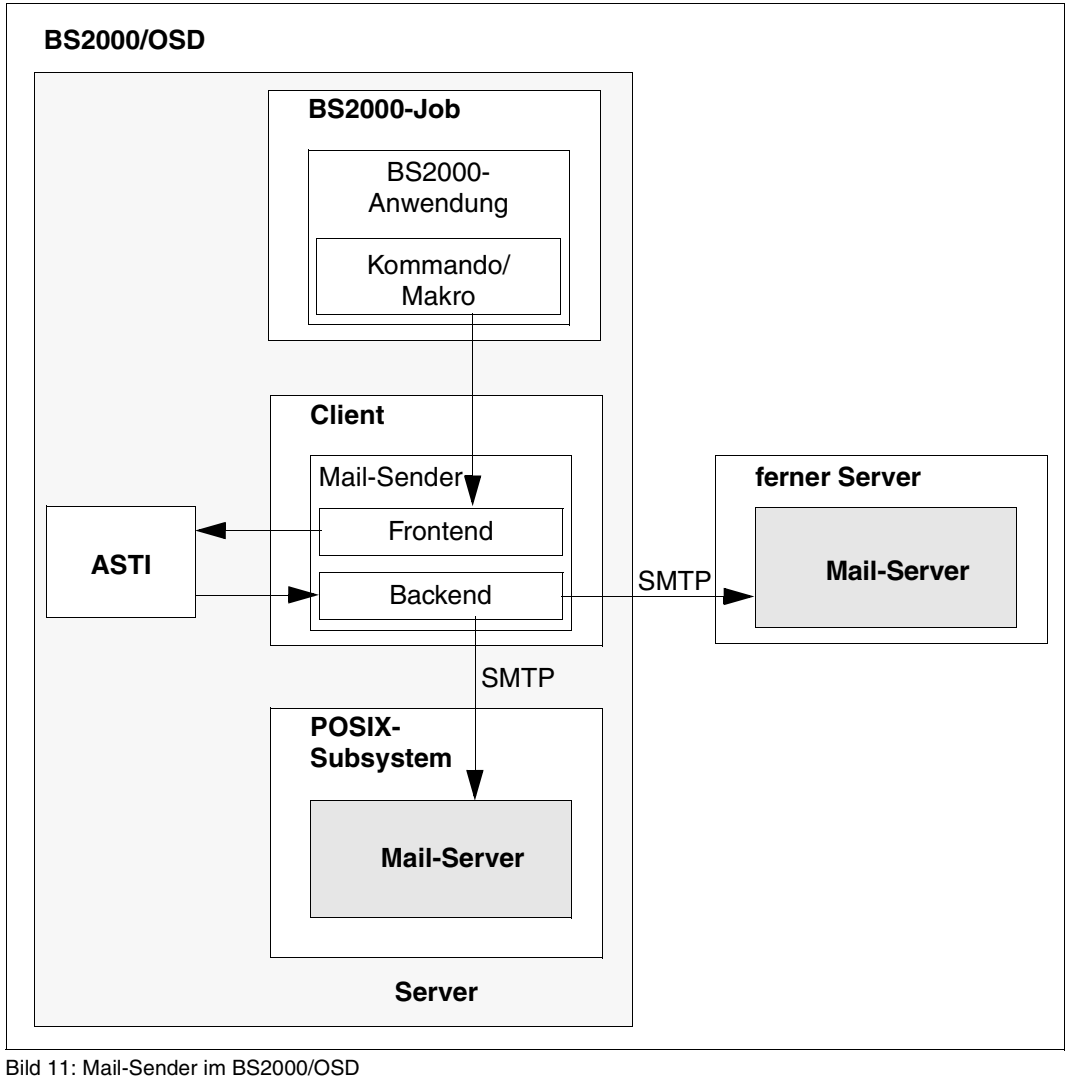


Bild 11: Mail-Sender im BS2000/OSD

Das vorliegende Kapitel informiert über folgende Themen:

- Konfigurationsdatei für das Mail-Sender Frontend (Benutzer-Option-Datei)
- SDF-Schnittstelle des Mail-Sender Frontend
- Makro-Schnittstelle des Mail-Sender Frontend

9.1 Konfigurationsdatei für das Mail-Sender Frontend (Benutzer-Option-Datei)

Um die Definition von Parametern zu erleichtern, die sich selten verändern, unterstützt die Frontend-Schnittstelle des Mail-Senders eine Benutzer-Option-Datei.
(Standardmäßig: *SYSDAT.MAIL.033.USER.OPT*)

Die Options müssen folgenden Regeln entsprechen:

- Jede Option beginnt in einer neuen Zeile.
- Wenn die Argumente einer Option länger sind als eine Zeile, müssen Zeilen, die weitergeführt werden sollen, mit dem Zeichen „\“ enden.
- Zeilen, die mit „#“ beginnen, werden ignoriert.
- Die Option-Namen sind nicht abhängig von der Groß-/Kleinschreibung.
- Option-Werte sind nicht abhängig von der Groß-/Kleinschreibung, wenn das nicht ausdrücklich erwähnt ist.

Im Folgenden werden die Options beschrieben.

fromAddress

Legt die Mail-Adresse fest, die als Absender-Adresse verwendet werden soll, sofern diese nicht anderweitig definiert ist. Wenn die Absender-Adresse weder in der Option-Datei noch in den entsprechenden Operanden der SDF- oder Makro-Schnittstelle definiert ist, akzeptiert der Mail-Sender die Mail nicht.

Kann ein Mail-Server eine Mail nicht weiterleiten, dann sendet er eine so genannte Bounce Mail an die Absender-Adresse. Es ist daher wichtig, dass eine gültige Absender-Adresse definiert ist.

fromAddress
`<sender-mail-addresse>`

<sender-mail-addresse>
Mail-Adresse, die als Absender-Adresse verwendet werden soll.

fromDisplayName

Mit dieser Option wird eine Zeichenkette spezifiziert, die im Mail-Header der eigentlichen Absenderadresse vorangestellt wird, wenn nicht schon anderweitig eine Absender-Adresse für den Mail-Header angegeben wurde (z.B. per fake_from-Spezifikation an der Unterprogrammschnittstelle).

Die angegebene Zeichenkette wird ohne XCHS-Konvertierung und ohne QP-/Base64-Enkodierung verwendet, d.h. sie darf nach einer Standard-EBCDIC-ISO8859-Konvertierung nur aus 7-Bit-ASCII-Zeichen bestehen. Sollen dennoch nicht-ASCII-Zeichen angezeigt werden, so muss der Anwender die QP- oder Base64-Enkodierung gemäß RFC 2047 selbst vornehmen.

fromDisplayName
`<display name 1..63>`

<display name 1..63>
Zeichenkette, die der Absender-Adresse vorangestellt wird.

fromDisplayNameWithHostName

Diese Option gibt an, ob gegebenenfalls die mit der Option from DisplayName spezifizierte Zeichenkette um den in eckige Klammern gesetzten Hostnamen ergänzt werden soll.

Wenn aufgrund der fromDisplayName-Option die Absenderadresse um einen Display-Namen ergänzt wird, dann wird beim Parameterwert YES an diesen Display-Namen der in eckige Klammern gesetzte Hostname angehängt.

Dies ist insbesondere dann nützlich, wenn von mehreren Systemen aus eine auf einem Shared-Pubset abgelegte Benutzer-Option-Datei verwendet wird.

fromDisplayNameWithHostName
NO | **YES**

NO Der in eckige Klammern gesetzte Hostname wird nicht angehängt.
Voreinstellung.

YES Der in eckige Klammern gesetzte Hostname wird angehängt.

sign

Legt fest, ob die Mail mit S/MIME signiert wird.

sign
NO | **YES**

NO Die Mail wird nicht mit S/MIME signiert.
NO ist Voreinstellung.

YES Die Mail wird mit S/MIME signiert.

encrypt

Legt fest, ob die Mail mit S/MIME verschlüsselt wird.

encrypt
NO | **YES**

NO Die Mail wird nicht mit S/MIME verschlüsselt.
NO ist Voreinstellung.

YES Die Mail wird mit S/MIME verschlüsselt.

privateKeyFile

Die in dieser Option angegebene Datei enthält einen privaten Schlüssel im PEM-Format, der zum Unterzeichnen von Mails mit S/MIME verwendet werden kann. Wenn das Zertifikat und der private Schlüssel in der selben Datei enthalten sind, müssen Sie diese Option nicht definieren. Der Schlüssel darf nicht mit einer Passphrase geschützt werden, da keine Möglichkeit zur Eingabe einer Passphrase besteht.

privateKeyFile
<dateiname 1..54> | ***NONE**

<dateiname 1..54>
Datei, die den privaten Schlüssel im PEM-Format enthält.

***NONE**
Es wird keine eigene Datei für den privaten Schlüssel verwendet.
*NONE ist Voreinstellung.

signerCertificateFile

Die in dieser Option angegebene Datei enthält ein X.509-Zertifikat im PEM-Format, das zur Signierung von Mails mit S/MIME verwendet werden kann. Falls erforderlich, kann die Datei zusätzlich den privaten Schlüssel enthalten. Üblicherweise werden Zertifikat und Schlüssel jedoch in verschiedenen Dateien gespeichert. In diesem Fall spezifizieren Sie den Schlüssel in der Option *privateKeyFile*. Das Zertifikat muss zum tatsächlich verwendeten privaten Schlüssel passen. Darauf ist insbesondere zu achten, wenn das Zertifikat per SDF-Kommando und der Schlüssel per Benutzer-Option definiert wird oder umgekehrt.

signerCertificateFile
<dateiname 1..54> | ***NONE**

<dateiname 1..54>
Datei, die das X.509-Zertifikat im PEM-Format enthält.

***NONE**
Es wird keine eigene Datei mit X.509-Zertifikaten verwendet.
*NONE ist Voreinstellung.

addSignerCertificatesFile

Die in dieser Option angegebene Datei enthält zusätzliche X.509-Zertifikate im PEM-Format. Diese Zertifikate sind u.U. erforderlich, um eine lückenlose Kette der Zertifikate vom Unterzeichnerzertifikat bis zum Root-CA-Zertifikat bereit zu stellen.

addSignerCertificatesFile
<dateiname 1..54> | ***NONE**

<dateiname 1..54>
Datei, die zusätzliche X.509-Zertifikate im PEM-Format enthält.

***NONE**
Es wird keine eigene Datei mit X.509-Zertifikaten verwendet.
*NONE ist Voreinstellung.

recipientCertificatesFile

Die in dieser Option angegebene Datei enthält X.509-Zertifikate im PEM-Format, die zur S/MIME-Verschlüsselung von Mails verwendet werden können. Die Zertifikate müssen eine Mail-Adresse als Bestandteil des Subject DN (Distinguished Name) oder als Bestandteil des X.509v3 Subject Alternative Name enthalten. Während der Verschlüsselung der Mail wird jedes Zertifikat daraufhin überprüft, ob es eine Mail-Adresse enthält, die mit einer Mail-Adresse aus der Liste der Mail-Empfänger übereinstimmt. Bei Übereinstimmung wird das Zertifikat zur Verschlüsselung verwendet.

Wird die *recipientCertificatesFile*-Option mehrmals verwendet, dann wird nach den dort angegebenen Dateien in der Reihenfolge der korrespondierenden Options gesucht.

recipientCertificatesFile
<dateiname 1..54> | ***NONE**

<dateiname 1..54>
Datei, die X.509-Zertifikate im PEM-Format enthält.

***NONE**
Es wird keine eigene Datei mit X.509-Zertifikaten verwendet.
*NONE ist Voreinstellung.

certificateRevocationListFile

Die Datei, die durch diese Option definiert wird, enthält Widerruflisten (Certificate Revocation List, CRL) für die mit *recipientCertificatesFiles* spezifizierten X.509-Zertifikate.

certificateRevocationListFile
<dateiname 1..54> | ***NONE**

<dateiname 1..54>
Datei, die Widerruflisten für X.509-Zertifikate enthält.

***NONE**
Es wird keine Datei für Widerruflisten verwendet.
*NONE ist Voreinstellung.

CACertificatesFile

Mit der Option *CACertificatesFile* wird eine Datei spezifiziert, die die für die Überprüfung der Mail-Empfänger-Zertifikate (mithilfe der Widerruflisten) erforderlichen CA-Zertifikate im PEM-Format enthält. Die einzelnen PEM-Zertifikate sind sequenziell in der Datei angeordnet.

Zum Hinzufügen oder Löschen von Zertifikaten können Sie die Datei mit einem beliebigen Text-Editor bearbeiten. Die einzelnen Zertifikate sind wie folgt in der Datei notiert:

```
-----BEGIN CERTIFICATE-----
< CA-Zertifikat in Base64-Codierung >
-----END CERTIFICATE-----
```

Text außerhalb dieser Sequenzen wird vom Mail-Client ignoriert und kann deshalb zum Kennzeichnen der Zertifikate verwendet werden, die wegen der ASN.1/Base64-Codierung in nicht lesbarer Form vorliegen.

CACertificatesFile
<dateiname 1..54> | ***NONE**

<dateiname 1..54>
: Name der Datei, die die für die Überprüfung der Mail-Empfänger-Zertifikate erforderlichen Zertifikate im PEM-Format enthält.

***NONE**
: Es wird keine Datei mit CA-Zertifikaten angegeben.
*NONE ist Voreinstellung.

cipher

Gibt an, welcher symmetrische Verschlüsselungsalgorithmus verwendet werden soll, falls die Mail verschlüsselt wird.

cipher
3DES | **DES** | **RC2-40** | **RC2-64** | **RC2-128** | **AES-128** | **AES-192** | **AES-256**

3DES Voreinstellung

logFile

Benennt die Datei, in die das benutzerspezifische Logging erfolgen soll.

Wenn die Datei zeitweilig nicht beschreibbar ist, weil z.B. von einer anderen Task aus ein SHOW-FILE-Kommando auf sie abgesetzt wurde, dann gehen während dieser Zeit generierte Logging-Einträge verloren. Die Voreinstellung für den Dateinamen ist SYSDAT.MAIL.033.USER.MAILLOG.

logFile
`<file name 1..54>`

<file name 1..54>
Datei, in die das benutzerspezifische Logging erfolgen soll.

logItems

Steuert, was in die benutzerspezifische Logging-Datei geschrieben wird.

Die Option kann mehrfach angegeben werden, um eine Liste der zu loggenden Daten zu definieren.

Bei Order-Einträgen wird nach der (versuchten) Übergabe eines Mail-Sendeauftrags an ASTI ein Eintrag in die Logging-Datei geschrieben. Der Eintrag enthält einen Zeitstempel, die ASTI-Order-Id (im Erfolgsfall), die Empfänger-Adresse(n) und den Subject-Header der Mail. Im Fehlerfall wird eine entsprechende Fehlermeldung hinterlegt.

Bei Result-Einträgen wird, nachdem der Mailauftragsstatus erfolgreich abgeholt wurde, ein Zeitstempel, die ASTI-Order-Id und die Resultat-Information geschrieben. Letztere zeigt an, ob die Mail von der Backend-Task erfolgreich an einen Mail-Server übergeben werden konnte, wenn ja, dann enthält die Meldung vom Mail-Server typischerweise einen Teil der Message-Id, die u.U. für die weitere Mail-Verfolgung von Nutzen ist. Schlägt die Übergabe an einen Mail-Server fehl, dann wird dies mit einem erläuternden Text im Resultat angezeigt. Schlug das Abholen des Mailauftragsstatus fehl, wird dies durch eine entsprechende Meldung in der Logging-Datei dokumentiert.

logItems
None | **Order** | **Result**

None Löscht die Liste der zu loggenden Daten.
Die Voreinstellung ist None, d.h. es wird mit einer leeren Liste begonnen.

Order Die zugehörigen Loggingeinträge werden in die Liste aufgenommen.

Result Die zugehörigen Loggingeinträge werden in die Liste aufgenommen.

9.2 SDF-Kommando-Schnittstelle des Mail-Sender Frontend

Mit den folgenden SDF-Kommandos können Sie Mail-Sende-Aufträge an das Subsystem ASTI senden und diese Aufträge verwalten:

- SEND-MAIL
- REQUEST-MAIL-ORDER-RESULT
- DELETE-MAIL-ORDER
- SHOW-MAIL-ORDER-STATUS

SEND-MAIL - Mail senden

Domäne:
UTILITIES

Erforderliche Berechtigung:
STD-PROCESSING

Mit dem Kommando SEND-MAIL können Sie einen Auftrag zum Versenden von Mails erstellen und an das Subsystem ASTI übermitteln. ASTI reicht den Mail-Sende-Auftrag weiter an die Service-Task des Mail-Sender Backend, die den Auftrag ausführt und einen Status Code zurücksendet. ASTI beschränkt die Größe der übermittelten Aufträge auf 32 KB, abzüglich einiger Bytes, die für Verwaltungszwecke reserviert sind.

Wenn ASTI einen SEND-MAIL-Aufruf wegen Überschreitung der maximalen Länge zurückweist, verfahren Sie wie folgt:

- Versuchen Sie, die Mail in mehrere kleinere Mails mit kleinerer Empfängerliste und/oder weniger Anhängen aufzuteilen.
- Speichern Sie die Mail-Texte und Anhänge in Dateien, anstatt die entsprechenden Inhalte direkt im Kommando anzugeben, und spezifizieren Sie im Kommando nur die zugehörigen Dateinamen.

SEND-MAIL

FROM=*USER-OPTION / <c-string 1..255 with-lower-case>

,TO=*NONE / <c-string 1..1800 with-lower-case>

,CC=*NONE / <c-string 1..1800 with-lower-case>

,BCC=*NONE / <c-string 1..1800 with-lower-case>

,REPLY-TO=*NONE / <c-string 1..255 with-lower-case>

,SUBJECT=*NONE / <c-string 1..1800 with-lower-case>

,ADDITIONAL-HEADER=*NONE / list-poss(10): **[*HEADER]**(…)

[*HEADER](…)
NAME=<c-string 1..63 with-lower-case>
,**BODY**=<c-string 1..255 with-lower-case>

,HEADER-CONVERSION=*STD / ***BY-CODED-CHAR-SET**(…)

***BY-CODED-CHAR-SET**(…)
SOURCE=<name 1..8>
,**DESTINATION**=<name 1..8>
,**CHARSET-NAME**=<c-string 1..63 with-lower-case>

```
,MESSAGE=*STD / *PARAMETERS(…)
    *PARAMETERS(…)
        TEXT=<c-string 1..1800 with-lower-case> / *FILE(…)
            *FILE(…)
                FILE-NAME=<filename 1..54 without-gen>
        CONVERSION=*STD / *NO / *BY-CODED-CHAR-SET(…)
            *BY-CODED-CHAR-SET(…)
                SOURCE=<name 1..8>
                ,DESTINATION=<name 1..8>
                ,CHARSET-NAME=<c-string 1..63 with-lower-case>
        ENCODING=*BIT-7 / *BIT-8 / *QP / *BASE64
        CONTENT-TYPE=*STD / <c-string 1..255 with lower-case>
        CONTENT-DISPOSITION=*INLINE / *ATTACHMENT
,ATTACHMENT=*NO / list-poss(20): [*PARAMETERS](…)
    [*PARAMETERS](…)
        TEXT=<c-string 1..1800 with-lower-case> / *FILE(…)
            *FILE(…)
                FILE-NAME=<filename 1..54 without-gen>
        CONVERSION=*STD / *NO / *BY-CODED-CHAR-SET(…)
            *BY-CODED-CHAR-SET(…)
                SOURCE=<name 1..8>
                ,DESTINATION=<name 1..8>
                ,CHARSET-NAME=<c-string 1..63 with-lower-case>
        ENCODING=*BIT-7 / *BIT-8 / *QP / *BASE64
        CONTENT-TYPE=*STD / <c-string 1..255 with-lower-case>
        CONTENT-DISPOSITION=*ATTACHMENT / *INLINE
```

```
,SECURITY=*NO / *SMIME(…)
    *SMIME(…)
        SIGNING=*USER-OPTION / *NO / *YES(…)
            *YES(…)
                CERTIFICATE-FILE=*USER-OPTION / <filename 1..54 without-gen>
                ,KEY-FILE=*USER-OPTION / <filename 1..54 without-gen>
                ,ADDITIONAL-CERT-FILE=*USER-OPTION / *NONE / <filename 1..54 without-gen>
        ENCRYPTING=*USER-OPTION / *NO / *YES(…)
            *YES(…)
                CERTIFICATE-FILE=*USER-OPTION / <filename 1..54 without-gen>
                ,CIPHER=*USER-OPTION / *DES3 / *DES / *RC2-40 / *RC2-64 / *RC2-128 /
                        *AES-128 / *AES-192 / *AES-256
                ,CRL-FILE=*USER-OPTION / *NONE / <filename 1..54 without-gen>
                ,CA-CERTIFICATES-FILE=*USER-OPTION / *NONE / <filename 1..54 without-gen>
,USER-OPTION-FILE=*STD / <filename 1..54 without-gen>
,WAIT-FOR-RESULT=*NO(…) / *YES
        *NO(…)
            RESULT=*DISCARD / *BY-REQUEST-CMD
```

Beschreibung der Operanden

FROM=*USER-OPTION / <c-string 1..255 with-lower-case>
Mail-Adresse des Mail-Absenders. An diese Adresse werden Informationen über Probleme bei der Übertragung der Mail gesendet („Bounce Mails“). Der Wert des Operanden wird auch in das Header-Feld *FROM* der Mail eingetragen. Die Adresse geben Sie im Format „<lokaler Teil>@<domain>“ an.

FROM=*USER-OPTION
Die Adresse des Mail-Absenders wird aus der Benutzer-Option-Datei übernommen. Falls die Adresse dort nicht definiert ist und die SYSSSI-Option senderSuffix nicht verwendet wird, wird das Kommando mit Fehlercode abgewiesen.

FROM=<c-string 1..255 with-lower-case>
Die Adresse des Absenders geben Sie im Format „<lokaler Teil>@<domain>“ an.

TO=*NONE / <c-string 1..1800 with-lower-case>
Angabe der Mail-Empfänger-Adresse(n).

TO=*NONE
Sie müssen die Mail-Empfänger über andere Operanden definieren. Das Header-Feld *TO* wird nicht generiert.

TO=<c-string 1..1800 with-lower-case>
Mail-Adresse des Empfängers oder eine Liste der Mail-Adressen der Empfänger. Die Adressen müssen in der Liste durch Kommata getrennt werden. Der Wert des Operanden wird auch in das Header-Feld *TO* der Mail eingetragen. Die Adressen geben Sie im Format „<lokaler Teil>@<domain>“ an.

CC=*NONE / <c-string 1..1800 with-lower-case>
Angabe der CC (Carbon Copy) Mail-Empfänger-Adresse(n).

CC=*NONE
Sie müssen die Mail-Empfänger über andere Operanden definieren. Das Header-Feld *CC* wird nicht generiert.

CC=<c-string 1..1800 with-lower-case>
CC Mail-Adresse des Empfängers oder Liste der CC Mail-Adressen der Empfänger. Die CC-Adressen müssen in der Liste durch Kommata getrennt werden. Der Wert des Operanden wird auch in das Header-Feld *CC* der Mail eingetragen. Die Adressen geben Sie im Format „<lokaler Teil>@<domain>“ an.

BCC=*NONE / <c-string 1..1800 with-lower-case>
Angabe der BCC (Blind Carbon Copy) Mail-Empfänger-Adresse(n).

BCC=*NONE
Sie müssen die Mail-Empfänger über andere Operanden definieren.

BCC=<c-string 1..1800 with-lower-case>
BCC Mail-Adresse des Empfängers oder Liste der BCC Mail-Adressen der Empfänger. Die BCC-Adressen müssen in der Liste durch Kommata getrennt werden. Der Wert des Operanden wird in kein Header-Feld der Mail eingetragen, so dass die Empfänger nicht erkennen, dass die Mail zusätzlich an diese Adressen gesendet wurde. Die Adressen geben Sie im Format „<lokaler Teil>@<domain>“ an.

REPLY-TO=*NONE / <c-string 1..255 with-lower-case>
Gegebenfalls die Mail-Adresse des Mail-Absenders, zu der die Empfänger Antworten senden sollen.

REPLY-TO=*NONE
Es wird kein Header-Feld *REPLY-TO* angefügt. Der Empfänger muss für eine Antwort die Adresse aus einem anderen Header-Feld übernehmen, z.B. aus dem Header-Feld *FROM*.

REPLY-TO=<c-string 1..255 with-lower-case>
Mail-Adresse des Mail-Absenders, an die die Empfänger Antworten senden sollen. Der Wert des Operanden wird in das Header-Feld *REPLY-TO* eingetragen. Die Adresse geben Sie im Format „<lokaler Teil>@<domain>“ an.

SUBJECT=*NONE / <c-string 1..255 with-lower-case>
Mail-Betreff

SUBJECT=*NONE
Kein Mail-Betreff. Es wird kein *SUBJECT* Header-Feld angefügt. Es wird empfohlen, den Betreff anzugeben, um dem Empfänger den Umgang mit den Mails zu erleichtern.

SUBJECT=<c-string 1..255 with-lower-case>
Mail-Betreff. Der Wert des Operanden wird in das *SUBJECT* Header-Feld der Mail eingetragen.

ADDITIONAL-HEADER=*NONE / list-poss(10): *HEADER(…)
Definition zusätzlicher Header-Felder für die Mail.

ADDITIONAL-HEADER=*NONE
Keine zusätzlichen Header-Felder.

ADDITIONAL-HEADER=list-poss(10): *HEADER(…)
In einer Liste können Sie bis zu 10 zusätzliche Header-Felder definieren. Name und Body werden in den entsprechenden Operanden festgelegt.

NAME=<c-string 1..63 with-lower-case>
Name des Header-Feldes.

BODY=<c-string 1..255 with-lower-case>
Body des Header-Feldes.

HEADER-CONVERSION=*STD / *BY-CODED-CHAR-SET(…)
legt fest, welche Zeichensatz-Konvertierung auf die Header-Felder angewendet wird.

HEADER-CONVERSION=*STD
Die Zeichensatz-Konvertierung von XHCS CCSN EDF03IRV zu CCSN ISO646 wird angewendet. Danach wird angenommen, dass die Header-Felder ausschließlich ASCII Zeichen enthalten, so dass keine Codierung gemäß RFC 2047 notwendig ist.

Der Anwender kann auch Zeichensätze verwenden, die nicht durch XHCS unterstützt werden. Voraussetzung: Der Anwender liefert Header, die nach Ausführung der oben beschriebenen Zeichensatz-Konvertierung konform sind mit einer Codierung gemäß RFC 2047.

HEADER-CONVERSION=*BY-CODED-CHAR-SET(…)
Die Header-Felder werden als Daten behandelt, auf die üblicherweise eine Zeichensatz-Konvertierung von einer EBCDIC-Variante in eine ISO-8859-Variante angewendet werden muss. Die Konvertierung wird mithilfe von XHCS ausgeführt. Nähere Informationen zu XHCS finden Sie im Handbuch „XHCS (BS2000/OSD)“.

SOURCE=<name 1..8>
Ausgangs-XHCS CCSN der Benutzerdaten.

DESTINATION=<name 1..8>
Ziel-XHCS CCSN der Benutzerdaten.

CHARSET-NAME=<c-string 1..63 with-lower-case>
Name des verwendeten Zeichensatzes in einer Codierung gemäß RFC 2047.

MESSAGE=*STD / *PARAMETERS(...)
definiert die zu sendende Nachricht.

MESSAGE=*STD
Eine leere Nachricht wird gesendet. Diese kann Anhänge enthalten, die im Operanden ATTACHMENT definiert sind.

MESSAGE=*PARAMETERS(...)
Es wird eine Nachricht mit den im Folgenden erläuterten Parametern erstellt:

TEXT=<c-string 1..1800 with-lower-case> / *FILE(...)
legt den Text der Nachricht fest, der an den/die Empfänger gesendet werden soll.

TEXT=<c-string 1..1800 with-lower-case>
Text der Nachricht. Einen expliziten Zeilenumbruch erzwingen Sie durch Einfügen der Zeichenfolge „\n“ an entsprechender Stelle. Wenn diese Zeichenfolge Bestandteil des Textes sein soll, muss sie als „\\n“ eingegeben werden. Weitere Umwandlungen werden nicht durchgeführt.

TEXT=*FILE(...)
Der Text der Nachricht muss aus einer Datei ausgelesen werden.

FILE-NAME=<filename 1..54 without-gen>
Name der Datei, in der die Nachricht enthalten ist. Die Datei muss unverändert bleiben, bis der Mail-Versand durch das Backend abgeschlossen ist.

CONVERSION=*STD / *NO / *BY-CODED-CHAR-SET(...)
legt fest, ob die Benutzerdaten aus Text oder aus binären Daten bestehen. Bestehen die Benutzerdaten aus Text, dann wird außerdem festgelegt, welcher Zeichensatz verwendet wird und welche EBCDIC-*x* zu ISO-*y* Konvertierung durchzuführen ist.

CONVERSION=*STD
Es wird eine Zeichensatz-Konvertierung von XHCS CCSN EDF03IRV nach CCSN ISO646 durchgeführt. Für das Header-Feld *Content-Type* wird „text/plain; charset=us-ascii“ festgelegt. Das Header-Feld *Content-Transfer-Encoding* wird auf den Wert „7bit“ eingestellt.

CONVERSION=*NO
Die Benutzerdaten werden als binäre Daten behandelt, d.h. es findet z.B. keine Zeichensatz-Konvertierung statt. Hat der Operand ENCODING den (Standard-)Wert *BIT-7, so wird dieser Wert durch *BASE64 ersetzt.

CONVERSION=*BY-CODED-CHAR-SET(...)
Die Benutzerdaten werden als Text behandelt, für den üblicherweise eine Zeichensatz-Konvertierung von einer EBCDIC-Variante in eine ISO-8859-Variante durchgeführt werden muss. Die Konvertierung wird mithilfe von XHCS ausgeführt. Nähere Informationen

zu XHCS finden Sie im Handbuch „XHCS (BS2000/OSD)“. Wenn XHCS in der Version 2.0 oder höher verfügbar ist, dann können auch die Unicode-bezogenen CCSNs UTF8, UTF16, UNICODE und UTFE verwendet werden.

SOURCE=<name 1..8>
Ausgangs-XHCS CCSN der Benutzerdaten.

DESTINATION=<name 1..8>
Ziel-XHCS CCSN der Benutzerdaten.

CHARSET-NAME=<c-string 1..63 with-lower-case>
Wenn der Inhalt der Nachricht aus Text besteht (siehe Operand CONTENT-TYPE), wird der Wert des Operanden CHARSET-NAME als Zeichensatz-Parameter im Header-Feld *Content-Type* der Nachricht eingefügt. Andernfalls wird der Operand ignoriert.

ENCODING=*BIT-7 / *BIT-8 / *QP / *BASE64
legt fest, wie die Benutzerdaten für die Übertragung codiert werden.

ENCODING=*BIT-7
Nach der Zeichensatz-Konvertierung sollten alle Zeichen als 7-Bit-ASCII-Zeichen darstellbar sein. Falls dies nicht der Fall ist, können manche Zeichen beim Mail-Transport verändert werden. Im Header-Feld *Content-Transfer-Encoding* wird der Wert „7bit“ eingetragen. Wenn Ihre Mail Zeilen länger als 998 Zeichen (ohne das abschließende CR LF) hat, verwenden Sie bitte ENCODING=*QP oder ENCODING=*BASE64, da andernfalls Ihre Mail beim Transport höchstwahrscheinlich verstümmelt oder anderweitig verändert wird.

ENCODING=*BIT-8
Im Header-Feld *Content-Transfer-Encoding* wird der Wert „8bit“ eingetragen.

Wenn Sie sich nicht sicher sind, dass alle Mail-Server in der Transportkette "8-bit-clean" sind, sollten Sie ENCODING=*QP oder ENCODING=*BASE64 verwenden. Die gleiche Empfehlung gilt, wenn Ihre Mail Zeilen länger als 998 Zeichen hat (ohne das abschließende CR LF), da andernfalls Ihre Mail beim Transport höchstwahrscheinlich verstümmelt oder anderweitig verändert wird.

ENCODING=*QP
Nach der Zeichensatz-Konvertierung werden alle Zeichen, die nicht direkt als 7-Bit-Zeichen darstellbar sind, gemäß dem „Quoted-Printable“-Algorithmus codiert (siehe RFC 2045). Dies resultiert in Zeilen von 7-Bit-Zeichen mit nicht mehr als 80 Zeichen. Diese Codierung ist vor allem dann hilfreich, wenn nur wenige Zeichen konvertiert werden müssen. In diesem Fall bleibt der überwiegende Teil der Daten für den Menschen lesbar. Im Header-Feld *Content-Transfer-Encoding* wird der Wert „quoted-printable“ eingetragen.

ENCODING=*BASE64
Nach der Zeichensatzkonvertierung werden die Daten gemäß dem„Base64“-Algorithmus codiert (siehe RFC 2045). Dies resultiert in Zeilen von 7-Bit-Zeichen mit nicht mehr als 80 Zeichen. Diese Codierung ist vor allem bei binären Daten sinnvoll, wenn nur wenige Zeichen direkt als 7-Bit-ASCII Zeichen darstellbar sind. In diesem Fall würde die Codierung mit „Quoted-Printable“ gegenüber „Base64“ zu weit umfangreicheren Daten führen, ohne für den Menschen lesbar zu sein. Die Codierung mit „Base64“ vergrößert die Datenmenge um ein Drittel.
Im Header-Feld *Content-Transfer-Encoding* wird der Wert „base64“ eingetragen.

CONTENT-TYPE=*STD / <c-string 1..255 with-lower-case>
Der Wert dieses Operanden wird in das Header-Feld *Content-Type* der Nachricht eingetragen. Dieses Feld zeigt dem empfangenden Mail-Client den Medien-Typ der übermittelten Nachricht an (z.B. reiner Text, Textverarbeitungsprogramm-Datei, Bild-Datei, Movie-Datei).

Beispiel

Für von Menschen lesbaren Text geben Sie „text/plain“ an, für Bilder im JPEG-Format geben Sie „image/jpeg“ an.

CONTENT-TYPE=*STD
Das Header-Feld *Content-Type* wird auf den Wert „text/plain“ und den Wert des Operanden CHARSET-NAME gesetzt.

CONTENT-TYPE=<c-string 1..255 with-lower-case>
Im Header-Feld *Content-Type* wird der Wert dieses Operanden eingetragen. Verwenden Sie einen Wert, der Ihren Daten entspricht (siehe z.B. RFC 2046).

CONTENT-DISPOSITION=*INLINE / *ATTACHMENT
Der Wert dieses Operanden wird in das Header-Feld *Content-Disposition* der Nachricht eingetragen (siehe RFC 2183). Dies zeigt dem empfangenden Mail-Client an, ob dieser Teil der Mail automatisch angezeigt werden soll, oder nur auf Anforderung des Benutzers, der die Mail liest.

CONTENT-DISPOSITION=*INLINE
Das Header-Feld *Content-Disposition* der Nachricht wird auf „inline“ gesetzt. Dies zeigt dem empfangenden Mail-Client an, dass dieser Mail-Body beim Anzeigen der Meldung automatisch angezeigt werden soll.

CONTENT-DISPOSITION=*ATTACHMENT
Das Header-Feld *Content-Disposition* der Nachricht wird auf „attachment“ gesetzt. Dies zeigt dem empfangenden Mail-Client an, dass dieser Teil des Meldungs-Body nur auf Anforderung des Benutzers angezeigt werden soll.

ATTACHMENT=*NO / *PARAMETERS(...)
Erzeugt einen oder mehrere Anhänge, die zu einer Nachricht hinzugefügt werden.

ATTACHMENT=*NO
Es werden keine Anhänge erzeugt.

ATTACHMENT=*PARAMETERS(...)
Ein Anhang mit den nachfolgend beschriebenden Parametern wird erzeugt.

TEXT=<c-string 1..1800 with-lower-case> / *FILE(...)
legt die Benutzerdaten fest, die an den/die Empfänger gesendet werden.

TEXT=<c-string 1..1800 with-lower-case>
Text der Nachricht. Einen expliziten Zeilenumbruch erzwingen Sie durch Einfügen der Zeichenfolge „\n“ an entsprechender Stelle. Wenn diese Zeichenfolge Bestandteil des Textes sein soll, muss sie als „\\n“ eingegeben werden. Weitere Konvertierungen werden nicht durchgeführt.

TEXT=*FILE(...)
Die Benutzerdaten müssen aus einer Datei ausgelesen werden.

FILE-NAME=<filename 1..54 without-gen>
Der Name der Datei, die die Benutzerdaten enthält. Die Datei muss unverändert bleiben, bis der Mail-Versand durch das Backend abgeschlossen ist.

CONVERSION=*STD / *NO / *BY-CODED-CHAR-SET(...)
Legt fest, ob die Benutzerdaten aus Text oder aus binären Daten bestehen. Bestehen die Benutzerdaten aus Text, dann wird außerdem festgelegt, welcher Zeichensatz verwendet wird und welche EBCDIC-*x* zu ISO-*y* Konversion durchzuführen ist.

CONVERSION=*STD
Eine Zeichensatz-Konvertierung von XHCS CCSN EDF03IRV nach CCSN ISO646 wird durchgeführt. Im Header-Feld *Content-Type* wird der Wert „text/plain; charset=us-ascii“ eingestellt. Im Header-Feld *Content-Transfer-Encoding* wird der Wert „7bit“ eingestellt.

CONVERSION=*NO
Die Benutzerdaten werden als binäre Daten behandelt, d.h. es findet z.B. keine Zeichensatz-Konvertierung statt. Hat der Operand ENCODING den (Standard-) Wert *BIT-7, dann wird dieser Wert durch *BASE64 ersetzt.

CONVERSION=*BY-CODED-CHAR-SET(...)
Die Benutzerdaten werden als Text behandelt, für den üblicherweise eine Zeichensatz-Konvertierung von einer EBCDIC-Variante in eine ISO-8859-Variante durchgeführt werden muss. Die Konvertierung wird mithilfe von XHCS ausgeführt. Nähere Informationen zu XHCS finden Sie im Handbuch „XHCS (BS2000/OSD)“. Wenn XHCS in der Version 2.0 oder höher verfügbar ist, dann können auch die Unicode-bezogenen CCSNs UTF8, UTF16, UNICODE und UTFE verwendet werden.

SOURCE=<name 1..8>
Ausgangs-XHCS CCSN der Benutzerdaten.

DESTINATION=<name 1..8>
Ziel-XHCS CCSN der Benutzerdaten.

CHARSET-NAME=<c-string 1..63 with-lower-case>
Wenn der Inhalt der Nachricht aus Text besteht (siehe Operand CONTENT-TYPE), wird der Wert des Operanden als Zeichensatz-Parameter im Header-Feld *Content-Type* eingefügt. Andernfalls wird der Operand ignoriert.

ENCODING=*BIT-7 / *BIT-8 / *QP / *BASE64
legt fest, wie die Benutzerdaten für die Übertragung codiert werden.

ENCODING=*BIT-7
Nach der Zeichensatz-Konvertierung sollten alle Zeichen als 7-Bit-ASCII-Zeichen darstellbar sein. Falls dies nicht der Fall ist, können manche Zeichen beim Mail-Transport verändert werden. Im Header-Feld *Content-Transfer-Encoding* wird der Wert „7bit“ eingetragen. Wenn Ihre Mail Zeilen länger als 998 Zeichen (ohne das abschließende CR LF) hat, verwenden Sie bitte ENCODING=*QP oder ENCODING=*BASE64, da andernfalls Ihre Mail beim Transport höchstwahrscheinlich verstümmelt oder anderweitig verändert wird.

ENCODING=*BIT-8
Im
Header-Feld *Content-Transfer-Encoding* wird der Wert „8bit“ eingetragen.

Wenn Sie sich nicht sicher sind, dass alle Mail-Server in der Transportkette "8-bit-clean" sind, sollten Sie ENCODING=*QP oder ENCODING=*BASE64 verwenden. Die gleiche Empfehlung gilt, wenn Ihre Mail Zeilen länger als 998 Zeichen hat (ohne das abschließende CR LF), da andernfalls Ihre Mail beim Transport höchstwahrscheinlich verstümmelt oder anderweitig verändert wird.

ENCODING=*QP
Nach der Zeichensatz-Konvertierung werden alle Zeichen, die nicht direkt als 7-Bit-Zeichen darstellbar sind, gemäß dem „Quoted-Printable“-Algorithmus codiert (siehe RFC 2045). Dies resultiert in Zeilen von 7-Bit-Zeichen mit nicht mehr als 80 Zeichen. Diese Codierung ist vor allem dann hilfreich, wenn nur wenige Zeichen konvertiert werden müssen. In diesem Fall bleibt der überwiegende Teil der Daten für den Menschen lesbar. Im Header-Feld *Content-Transfer-Encoding* wird der Wert „quoted-printable“ eingetragen.

ENCODING=*BASE64
Nach der Zeichensatzkonvertierung werden die Daten gemäß dem„Base64“-Algorithmus codiert (siehe RFC 2045). Dies resultiert in Zeilen von 7-Bit-Zeichen mit nicht mehr als 80 Zeichen. Diese Codierung ist vor allem bei binären Daten sinnvoll, wenn nur wenige Zeichen direkt als 7-Bit-ASCII Zeichen darstellbar sind. In diesem Fall würde die Codierung mit „Quoted-Printable“ gegenüber „Base64“ zu weit umfangreicheren Daten

führen, ohne für den Menschen lesbar zu sein. Die Codierung mit „Base64“ vergrößert die Datenmenge um ein Drittel.
Im Header-Feld *Content-Transfer-Encoding* wird der Wert „base64“ eingetragen.

CONTENT-TYPE=*STD / <c-string 1..255 with-lower-case>
Der Wert dieses Operanden wird in das Header-Feld *Content-Type* der Nachricht eingetragen. Dieses Feld zeigt dem empfangenden Mail-Client den Medien-Typ der übermittelten Nachricht an (z.B. reiner Text, Textverarbeitungsprogramm-Datei, Bild-Datei, Movie-Datei).

Beispiel

Für von Menschen lesbaren Text geben Sie „text/plain“ an, für Bilder im JPEG-Format geben Sie „image/jpeg“ an.

CONTENT-TYPE=*STD
Das Header-Feld *Content-Type* wird auf den Wert „text/plain“ und den Wert des Operanden CHARSET-NAME gesetzt.

CONTENT-TYPE=<c-string 1..255 with-lower-case>
Im Header-Feld *Content-Type* wird der Wert dieses Operanden eingetragen. Verwenden Sie einen Wert, der Ihren Daten entspricht.

CONTENT-DISPOSITION=*ATTACHMENT / *INLINE /
Der Wert dieses Operanden wird in das Header-Feld *Content-Disposition* der Nachricht eingetragen (siehe RFC 2183). Es zeigt dem empfangenden Mail-Client an, ob dieser Teil der Mail automatisch angezeigt werden soll oder nur auf Anforderung des Benutzers, der die Mail liest.

CONTENT-DISPOSITION=*ATTACHMENT
Das Header-Feld *Content-Disposition* der Meldung wird auf „attachment“ gesetzt. Dies zeigt dem Mail-Client des Empfängers an, dass dieser Teil des Meldungs-Body nur auf Anforderung des Benutzers angezeigt werden soll.

CONTENT-DISPOSITION=*INLINE
Das Header-Feld *Content-Disposition* der Nachricht wird auf „inline“ gesetzt. Dies zeigt dem empfangenden Mail-Client an, dass dieser Mail-Body beim Anzeigen der Meldung automatisch angezeigt werden soll.

SECURITY=*NO / *SMIME(…)
legt fest, ob die Mail verschlüsselt und/oder signiert werden soll.

SECURITY=*NO
Die Mail wird weder verschlüsselt noch signiert.

SECURITY=*SMIME(…)
Die Mail wird mit S/MIME verschlüsselt und/oder signiert.

SIGNING=*USER-OPTION / *NO / *YES(…)
legt fest, ob die Mail signiert werden soll.

SIGNING=*USER-OPTION
Die Mail wird abhängig von der Einstellung in der Benutzer-Option-Datei signiert.

SIGNING=*NO
Die Mail wird nicht signiert.

SIGNING=*YES(…)
Die Mail wird signiert.

CERTIFICATE-FILE=*USER-OPTION / <filename 1..54 without-gen>
gibt die Datei an, in der das X.509-Zertifikat gespeichert ist, das für die Signierung verwendet wird.

CERTIFICATE-FILE=*USER-OPTION
Die Datei mit dem zu verwendenden X.509-Zertifikat muss in der Benutzer-Option-Datei spezifiziert werden.

CERTIFICATE-FILE=<filename 1..54 without-gen>
Name der Datei, die das zu verwendende X.509-Zertifikat enthält. Das Zertifikat muss im PEM-Format gespeichert sein.

KEY-FILE=*USER-OPTION / <filename 1..54 without-gen>
gibt die Datei an, die den privaten Schlüssel enthält, der zu dem im Parameter CERTIFICATE-FILE spezifizierten X.509-Zertifikat gehört.

KEY-FILE=*USER-OPTION
Die Datei mit dem zu verwendenden privaten Schlüssel muss in der Benutzer-Option-Datei spezifiziert werden.

KEY-FILE=<filename 1..54 without-gen>
Name der Datei, die den zu verwendenden privaten Schlüssel enthält. Der private Schlüssel muss im PEM-Format gespeichert sein.

ADDITIONAL-CERT-FILE=*USER-OPTION / *NONE / <filename 1..54 without-gen>
spezifiziert eine Datei mit zusätzlichen X.509-Zertifikaten, die zum Signieren verwendet werden können. Diese Zertifikate unterstützen den Empfänger beim Verifizieren der Signatur, wenn das im CERTIFICATE-FILE-Operanden spezifizierte Zertifikat nicht von einer Root-CA (Certificate Authorithy), sondern von einer Intermediate CA herausgegeben wurde.

ADDITIONAL-CERT-FILE=*USER-OPTION
Falls eine Datei mit zusätzlichen X.509-Zertifikaten benötigt wird, muss sie in der Benutzer-Option-Datei spezifiziert werden.

ADDITIONAL-CERT-FILE=*NONE
Es werden keine zusätzlichen Zertifikate verwendet.

ADDITIONAL-CERT-FILE=<filename 1..54 without-gen>
Name der Datei, die die zu verwendenden zusätzlichen Zertifikate enthält. Die Zertifikate müssen im PEM-Format gespeichert sein.

ENCRYPTING=*USER-OPTION / *NO / *YES(…)
legt fest, ob die Mail verschlüsselt gesendet wird.

ENCRYPTING=*USER-OPTION
Die Verschlüsselung der Mail hängt von der Einstellung in der Benutzer-Option-Datei ab.

ENCRYPTING=*NO
Die Mail wird nicht verschlüsselt.

ENCRYPTING=*YES(…)
Die Mail wird verschlüsselt.

CERTIFICATE-FILE=*USER-OPTION / <filename 1..54 without-gen>
gibt die Datei an, in der die X.509-Zertifikate gespeichert sind, die für die Verschlüsselung verwendet werden.

CERTIFICATE-FILE=*USER-OPTION
Die Datei mit den zu verwendenden X.509-Zertifikaten muss in der Benutzer-Option-Datei spezifiziert werden.

CERTIFICATE-FILE=<filename 1..54 without-gen>
Name der Datei, die die zu verwendenden X.509-Zertifikate enthält. Die Zertifikate müssen im PEM-Format gespeichert sein.

CIPHER=*USER-OPTION / *DES3 / *DES / *RC2-40 / *RC2-64 / *RC2-128 / *AES-128 / *AES-192 / *AES-256
spezifiziert den zu verwendenden Verschlüsselungsalgorithmus. Der ausgewählte Verschlüsselungsalgorithmus muss von allen Mail-Empfängern unterstützt werden. Andernfalls können einige der Empfänger die Mail nicht entschlüsseln.

CIPHER=*USER-OPTION
Die gewünschte Cipher muss in der Benutzer-Option-Datei angegeben werden.

CIPHER=*DES3
Verschlüsselung mit Triple-DES. Die effektive Länge beträgt 112 Bit.

CIPHER=*DES
Verschlüsselung mit DES.
Dieses Verfahren sollten Sie nur wählen, wenn keine besseren Alternativen zur Verfügung stehen, da die Schlüssellänge (56 Bit) in diesem Verfahren inzwischen als zu kurz bewertet wird.

CIPHER=*RC2-40
Verschlüsselung mit RC2-40.
Dieses Verfahren sollten Sie nur dann wählen, wenn keine besseren Alternativen zur Verfügung stehen, da die Schlüssellänge (40 Bit) in diesem Verfahren inzwischen als bei weitem zu kurz bewertet wird.

CIPHER=*RC2-64
Verschlüsselung mit RC2-64.
Dieses Verfahren sollten Sie nur wählen, wenn keine besseren Alternativen zur Verfügung stehen, da die Schlüssellänge (64 Bit) in diesem Verfahren inzwischen als zu kurz bewertet wird.

CIPHER=*RC2-128
Verschlüsselung mit RC2-128. Die Schlüssellänge beträgt 128 Bit.

CIPHER=*AES-128
Verschlüsselung mit AES-128.
Die Schlüssellänge beträgt 128 Bit. AES ist ein Nachfolger von DES/3DES und noch relativ neu. Daher wird AES eventuell noch nicht von allen Mail-Programmen unterstützt.

CIPHER=*AES-192
Verschlüsselung mit AES-192.
Die Schlüssellänge beträgt 192 Bit. AES ist ein Nachfolger von DES/3DES und noch relativ neu. Daher wird AES eventuell noch nicht von allen Mail-Programmen unterstützt.

CIPHER=*AES-256
Verschlüsselung mit AES-256.
Die Schlüssellänge beträgt 256 Bit. AES ist ein Nachfolger von DES/3DES und noch relativ neu. Daher wird AES eventuell noch nicht von allen Mail-Programmen unterstützt.

CRL-FILE=*USER-OPTION / *NONE / <filename 1..54 without-gen>
spezifiziert die Datei, die die CRL (Certificate Revocation List) enthält. Die CRL wird verwendet, um die Gültigkeit von Empfänger-Zertifikaten zu überprüfen.

CRL-FILE=*USER-OPTION
Die Datei mit der zu verwendenden CRL ist gegebenenfalls in der Benutzer-Option-Datei spezifiziert. Wenn in der Benutzer-Option-Datei *NONE angegeben ist (Voreinstellung), werden die Empfänger-Zertifikate nicht auf Gültigkeit überprüft.

CRL-FILE=*NONE
Es wird keine CRL verwendet, d.h. die Empfänger-Zertifikate werden nicht auf Gültigkeit überprüft.

CRL-FILE=<filename 1..54 without-gen>
Datei, die die zu verwendende CRL enthält.

CA-CERTIFICATES-FILE=*USER-OPTION / *NONE /
<filename 1..54 without-gen>
spezifiziert die Datei, die die für die Überprüfung der Gültigkeit der Empfänger-Zertifikate (siehe Operand CRL-FILE) erforderlichen Zertifikate im PEM-Format enthält.

CA-CERTIFICATES-FILE=*USER-OPTION
Die Datei mit den zu verwendenden CA-Zertifikaten ist gegebenenfalls in der Benutzer-Option-Datei spezifiziert.

CA-CERTIFICATES-FILE=*NONE
Es wird keine Datei mit CA-Zertifikaten verwendet.

CA-CERTIFICATES-FILE=<filename 1..54 without-gen>
Datei, die die Zertifikate enthält.

USER-OPTION-FILE=*STD / <filename 1..54 without-gen>
spezifiziert eine Benutzer-Option-Datei, die Standard-Werte für verschiedene Operanden enthält. Eine detaillierte Beschreibung der Benutzer-Option-Datei finden Sie in Abschnitt „Konfigurationsdatei für das Mail-Sender Frontend (Benutzer-Option-Datei)“ auf Seite 403.

USER-OPTION-FILE=*STD
Die Standard-Benutzer-Options werden aus der Datei ermittelt, die in der Option-Datei *SYSSSI* mit der Option *defaultOptionFileName* spezifiziert ist (siehe „interNet Services (BS2000/OSD), Administratorhandbuch“). Die Voreinstellung für diese Option ist *SYSDAT.MAIL.033.USER.OPT*.

USER-OPTION-FILE=<filename 1..54 without-gen>
Als Benutzer-Option-Datei wird die hier angegebene Datei verwendet.

WAIT-FOR-RESULT=*NO(...) / *YES
Der Operand gibt an, ob das Kommando direkt nach dem Abschicken des Mail-Sende-Auftrags an den Mail-Server beendet wird, oder ob das Kommando die Ausführung des Auftrags abwarten soll.

WAIT-FOR-RESULT=*NO(...)
Das Kommando wartet nicht, bis der Transfer-Auftrag an den Mail-Server abgeschlossen ist.

RESULT=*DISCARD
Der Ausführungs-Status wird nicht gesichert. Er kann daher später nicht mit dem Kommando REQUEST-MAIL-ORDER-RESULT (siehe Seite 430) abgefragt werden.

RESULT=*BY-REQUEST-CMD
Den Ausführungs-Status können und sollten Sie später mit dem Kommando REQUEST-MAIL-ORDER-RESULT (siehe Seite 430) abfragen. Wenn der Status nicht abgefragt wird, belegen die Status-Informationen auf unbegrenzte Zeit Speicherplatz im Subsystem ASTI. Diese Status-Informationen speichert ASTI in SYS.*-Dateien unter der Kennung des Mail-Senders.

WAIT-FOR-RESULT=*YES
Das Kommando wartet auf den Abschluss des Mail-Sende-Auftrags und informiert, ob der Transfer zum (ersten) Mail-Server erfolgreich war.

Return Codes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kein Fehler.
	64	CMD0216	Der Benutzer hat nicht die notwendige Berechtigung für das Kommando.
	32	CMD0220	Interner Fehler
	32	CMD2009	Fehler während der Erstellung der S-Variable.
	64	YML0120	Subsystem ASTI ist nicht verfügbar.
	64	YML0142	Benutzer-Option-Datei ist nicht vorhanden oder nicht lesbar.
	64	YML0144	Als Anhang spezifizierte Datei ist nicht vorhanden oder nicht lesbar.
	64	YML0146	Mail-Sende-Auftrag ist zu umfangreich.
	128	YML0148	Maximale Anzahl von Aufträgen erreicht.
	64	YML0171	SMTP Protokoll Fehler.
	64	YML0172	Mail-Service nicht aktiviert.
	64	YML0174	Keine Absender-Adresse angegeben.
	32	YML0176	Unerwarteter ASTI-Fehler
	64	YML0203	S/MIME Datei ist fehlerhaft oder wurde nicht gefunden
	128	YML0214	Ressourcen sind erschöpft.

SC1/2=Stubcode 1/2 in Dezimal-Darstellung

Ausgabe-Daten

Wenn ein Mail-Sende-Auftrag erfolgreich an ASTI übermittelt wurde, wird eine YML0160-Meldung ausgegeben, die die Auftrags-ID als Insert enthält. Auf Anforderung wird die Auftrags-ID auch in einer S-Variable mit der Komponente ORDER-ID abgelegt.

Wenn das SEND-MAIL-Kommando synchron verwendet wird (WAIT-FOR-RESULT=*YES), dann wird zusätzlich eine YML0170-Meldung (oder im Fehlerfall eine andere passende Meldung) ausgegeben. Wenn angefordert, dann werden die Komponenten RETURN-CODE und RETURN-MSG einer OPS-Variablen mit entsprechenden Werten versorgt (siehe auch die Beschreibung der „Ausgabe-Daten“ auf Seite 432 beim Kommando REQUEST-MAIL-ORDER-RESULT).

Beispiele:

```
/EXECUTE-CMD CMD=(SEND-MAIL -
/                 TO='Heinrich.Schuetz@dresden.example', -
/                 SUBJECT='Opus ultimum', -
/                 WAIT=*NO(RESULT=*BY-REQUEST-CMD)), -
/            STRUCTURE-OUTPUT=OUT
%  YML0160 MAIL-SENDE-AUFTRAG UEBERGEBEN MIT AUFTRAGS-ID
'077AF4BF00000046'

/SHOW-VARIABLE OUT
OUT(*LIST).ORDER-ID = 077AF4BF00000046
OUT(*LIST).RETURN-CODE = Ok
OUT(*LIST).RETURN-MSG =

/EXECUTE-CMD CMD=(SEND-MAIL -
/                 TO='Heinrich.Schuetz@dresden.example', -
/                 SUBJECT='Opus ultimum', -
/                 WAIT=*YES), -
/            STRUCTURE-OUTPUT=OUT
%  YML0160 MAIL-SENDE-AUFTRAG UEBERGEBEN MIT AUFTRAGS-ID
'077AF4BF0000004F'
%  YML0170 MAIL-SENDE-AUFTRAG DURCHGEFUEHRT; MAIL-SERVER-
RESULTAT: '250 OK: QUEUED AS 5720B6EC20'

/SHOW-VARIABLE OUT
OUT(*LIST).ORDER-ID = 077AF4BF0000004F
OUT(*LIST).RETURN-CODE = Ok
OUT(*LIST).RETURN-MSG = 250 Ok: queued as 5720B6EC20
```

REQUEST-MAIL-ORDER-RESULT - Mail-Resultat abholen

Domäne:
UTILITIES

Erforderliche Berechtigung:
STD-PROCESSING
TSOS

Mit dem Kommando REQUEST-MAIL-ORDER-RESULT können Sie zu Mail-Sende-Aufträgen den zugehörigen Ausführungs-Status abfragen, den die Backend-Task nach Bearbeitung des Auftrags zurückliefert. Die Auftragsdaten werden dabei aus den internen ASTI-Tabellen gelöscht. Dabei wird auch die zugehörige SYS-Datei unter der Kennung des Mail-Senders entfernt.

Zu Aufträgen, die noch auf ihre Bearbeitung warten, liefert das Kommando REQUEST-MAIL-ORDER-RESULT keine Informationen. In diesem Fall können Sie sich mithilfe des Kommandos SHOW-MAIL-ORDER-STATUS über den Auftrag informieren (siehe Seite 435). Das Kommando REQUEST-MAIL-ORDER-RESULT unterstützt eine strukturierte Ausgabe in S-Variablen (siehe Handbuch „Kommandos, Band 6, S-Variable“).

REQUEST-MAIL-ORDER-RESULT
ORDER-ID=*__ANY__ / <x-text 1..16> ,**WAIT-FOR-RESULT**=*__NO__ / ***YES** ,**USER-OPTION-FILE**=__*STD__ / <filename 1..54 without-gen>

Beschreibung der Operanden

ORDER-ID=*__ANY__ / <x-text 1..16>
spezifiziert den Auftrag, dessen Ausführungs-Status abgefragt wird.

ORDER-ID=*__ANY__
Der Ausführungsstatus eines, vom Kommando-Aufrufer abgesendeten, abgeschlossenen Auftrags wird abgefragt. Liegen mehrere abgeschlossene Aufträge vor, dann ist nicht definiert, von welchem dieser Aufträge der Status abgefragt wird.

ORDER-ID=<x-text 1..16>
ASTI-Auftrags-ID des Auftrags, dessen Ausführungs-Status abgefragt wird.

WAIT-FOR-RESULT=*__NO__ / *YES
legt fest, ob das Kommando wartet, bis die Bearbeitung des Auftrags abgeschlossen ist.

WAIT-FOR-RESULT=*NO
Das Kommando wartet nicht das Ende eines noch nicht abgeschlossenen Auftrags ab. Den Ausführungs-Status des Auftrags können Sie zu einem späteren Zeitpunkt mit einem erneuten REQUEST-MAIL-ORDER-RESULT-Aufruf abfragen.

WAIT-FOR-RESULT=*YES
Das Kommando wartet, bis der Auftrag abgeschlossen ist.
Wenn ORDER-ID=*ANY angegeben ist, wird das Warten intern durch eine periodische Abfrage mit einem Intervall von 60 Sekunden realisiert, so dass durchschnittlich 30 Sekunden nach Auftragsende das Warten abgeschlossen ist.

USER-OPTION-FILE=*STD / <filename 1..54 without-gen>
spezifiziert eine Benutzer-Option-Datei, die Standard-Werte für verschiedene Operanden enthält. Eine detaillierte Beschreibung der Benutzer-Option-Datei finden Sie in Abschnitt „Konfigurationsdatei für das Mail-Sender Frontend (Benutzer-Option-Datei)“ auf Seite 403.

USER-OPTION-FILE=*STD
Die Standard-Benutzer-Options werden aus der Datei ermittelt, die in der Option-Datei *SYSSSI* mit der Option *defaultOptionFileName* spezifiziert ist (siehe „interNet Services (BS2000/OSD), Administratorhandbuch“). Die Voreinstellung für diese Option ist *SYSDAT.MAIL.033.USER.OPT*.

USER-OPTION-FILE=<filename 1..54 without-gen>
Als Benutzer-Option-Datei wird die hier angegebene Datei verwendet.

Return Codes

(SC2) SC1	Maincode	Bedeutung
0	CMD0001	Kein Fehler.
32	CMD0220	Interner Fehler.
32	CMD2009	Fehler während der Erstellung der S-Variable.
64	YML0120	Subsystem ASTI nicht verfügbar.
32	YML0176	Unerwarteter ASTI-Fehler.
64	YML0210	Auftrag wurde nicht gefunden.
128	YML0214	Ressourcen sind erschöpft.
64	YML0215	Bei Auftragstellung wurde kein Ergebnis angefordert.
64	YML0216	Auftrag wurde durch fremde Task ausgeführt.
128	YML0222	Auftrag nicht abgeschlossen.

SC1/2=Subcode 1/2 in Dezimal-Darstellung

Beispiele

```
/REQUEST-MAIL-ORDER-RESULT ORDER-ID=02BC49BB0000000D
%  ORDER-ID:              02BC49BB0000000D
%  RETURN CODE:           Ok
%  RETURN MESSAGE:        250 Ok: queued as 0E7026E860
```

```
/EXECUTE-CMD CMD=(REQUEST-MAIL-ORDER-RESULT ORDER-ID= 02BC49BB00000017),-
/            STRUCTURE-OUTPUT=OUT
%  ORDER-ID:              02BC49BB00000017
%  RETURN CODE:           Ok
%  RETURN MESSAGE:        250 Ok: queued as 2B9AE6E860


/SHOW-VARIABLE OUT
OUT(*LIST).ORDER-ID =      02BC49BB00000017
OUT(*LIST).RETURN-CODE =   Ok
OUT(*LIST).RETURN-MSG =    250 Ok: queued as 2B9AE6E860
```

Ausgabe-Daten

Wenn ein Auftrag beendet ist, gibt das Kommando drei Zeilen mit Informationen zum Ausführungs-Status aus.

- Die erste Zeile (ORDER-ID) gibt die ID des Auftrags an, für den die Daten ausgegeben werden. Dies ist vor allem relevant, wenn ORDER-ID=*ANY spezifiziert wurde.
- Die zweite Zeile (RETURN CODE) gibt an, ob ein Fehler aufgetreten ist, und wenn ja, welcher. Mögliche Werte sind:

  OK
  : Mail erfolgreich gesendet.

  Error during SMTP protocol
  : Der Mail-Server hat einen Fehler an die Backend-Task gemeldet.

  Error during S/MIME operation
  : Während der S/MIME-Verarbeitung ist ein Fehler aufgetreten, z.B. Zertifikatsprobleme.

  Internal error
  : Bei allen anderen Fehlern.

- Die dritte Zeile (RETURN MESSAGE) enthält folgende Informationen:
  - Im Fehlerfall:
    Zusätzliche textuelle Informationen über den Fehler.
  - Bei einem erfolgreich beendeten Auftrag:
    Abschluss-Meldung des Mail-Servers. Diese Meldung enthält üblicherweise die (partielle) Message-ID, die der Mail vom Server zugewiesen wurde. Tritt im weiteren Mail-Transfer ein Fehler auf, dann kann der Administrator mithilfe dieser ID den Transfer der Mail über die Kette der Mail-Server zurückverfolgen lassen.

Die Daten können auch an eine S-Variable übergeben werden, die entsprechend der Ausgabe des Kommandos strukturiert ist.

DELETE-MAIL-ORDER - Mail löschen

Domäne:
UTILITIES

Erforderliche Berechtigung:
STD-PROCESSING
TSOS

Mit dem Kommando DELETE-MAIL-ORDER können Sie bislang unverarbeitete Mail-Sende-Aufträge löschen, die an das Subsystem ASTI übergeben wurden. Für Aufträge, deren Bearbeitung bereits abgeschlossen ist, löscht das Kommando DELETE-MAIL-ORDER in den ASTI-internen Tabellen die Informationen zum zugehörigen Ausführungs-Status. Auch die zugehörigen SYS-Dateien unter der Kennung des Mail-Senders werden gelöscht.

DELETE-MAIL-ORDER
ORDER-ID=*__**ALL**__ / <x-text 1..16>
,**SENDER-USERID**=*__**OWN**__ / ***ANY** / <name 1..8>

Beschreibung der Operanden

ORDER-ID=*__ALL__ / <x-text 1..16>
gibt an, welche Aufträge gelöscht werden sollen.

ORDER-ID=*__ALL__
Alle Aufträge der unter SENDER-USERID angegebenen Benutzerkennung werden gelöscht. Hierbei kann es vorkommen, dass auch neue Mails gelöscht werden, die parallel zur Ausführung von DELETE-MAIL-ORDER von einer anderen Task derselben Benutzerkennung gesendet werden.

Falls DELETE-MAIL-ORDER einen Fehler zurückliefert, konnten eventuell nicht alle Aufträge gelöscht werden. Wiederholen Sie in diesem Fall das Kommando, um die restlichen Aufträge zu löschen.

ORDER-ID=<x-text 1..16>
Der Auftrag mit der angegebenen ASTI-Auftrags-ID wird gelöscht, sofern der angegebene Benutzer der Eigentümer des Auftrags ist. Nur Benutzer mit TSOS-Berechtigung dürfen Aufträge mit einer fremden Benutzerkennung löschen.

SENDER-USERID=*__OWN__ / *ANY / <name 1..8>
spezifiziert die Benutzer, deren Aufträge gelöscht werden sollen.

SENDER-USERID=*__OWN__
Nur die Aufträge des Kommando-Aufrufers werden gelöscht.

SENDER-USERID=*ANY
Für Benutzer ohne TSOS-Berechtigung entspricht diese Angabe der Angabe von *OWN. Für Benutzer mit TSOS-Berechtigung bedeutet diese Angabe, dass die Aufträge aller Benutzer gelöscht werden.

SENDER-USERID=<name 1..8>
Benutzerkennung des Eigentümers der zu löschenden Aufträge. Nur Benutzer mit TSOS-Privilegien dürfen Aufträge einer fremden Benutzerkennung löschen.

Return codes

(SC2) SC1	Maincode	Bedeutung
0	CMD0001	Kein Fehler.
64	CMD0216	Der Benutzer hat nicht die notwendige Berechtigung für das Kommando.
32	CMD0220	Interner Fehler.
64	YML0120	Subsystem ASTI ist nicht verfügbar.
32	YML0176	Unerwarteter ASTI-Fehler.
64	YML0210	Auftrag nicht gefunden.
64	YML0216	Auftrag durch fremde Task ausgeführt.
64	YML0220	Der Benutzer ist nicht berechtigt, diesen Auftrag zu löschen.
64	YML0221	Der Benutzer ist nicht berechtigt, Aufträge anderer Benutzer zu löschen.

SC1/2=Subcode 1/2 in Dezimal-Darstellung

SHOW-MAIL-ORDER-STATUS - Informationen zur Mail abfragen

Domäne:
UTILITIES

Erforderliche Berechtigung:
STD-PROCESSING
TSOS

Mit dem Kommando SHOW-MAIL-ORDER-STATUS können Sie den Status der abgeschickten Mail-Sende-Aufträge abfragen.

Für alle aktuellen Aufträge zeigt SHOW-MAIL-ORDER-STATUS an, ob diese

– auf die Ausführung warten,
– aktuell ausgeführt werden oder
– bereits ausgeführt wurden.

Das Kommando SHOW-MAIL-ORDER-STATUS unterstützt die strukturierte Ausgabe in S-Variablen (siehe Handbuch „Kommandos, Band 6, S-Variable“).

SHOW-MAIL-ORDER-STATUS
ORDER-ID=*__ALL__ / list-poss(30): <x-text 1..16>
,**SENDER-USERID**=*__OWN__ / ***ANY** / <name 1..8>
,**INFORMATION**=*__SUMMARY__ / ***ALL**

Beschreibung der Operanden

ORDER-ID=*ALL / list-poss(30): <x-text 1..16>
legt fest, welche Aufträge ausgewählt werden.

ORDER-ID=*ALL
Alle Aufträge der mit dem Operanden SENDER-USERID spezifizierten Benutzer werden ausgewählt.

ORDER-ID=list-poss(30): <x-text 1..16>
Hier können Sie bis zu 30 Aufträge angeben, die ausgewählt werden sollen. Nur Benutzer mit TSOS-Berechtigung dürfen Aufträge mit einer fremden Benutzerkennung angeben.

SENDER-USERID=*OWN / *ANY / <name 1..8>
spezifiziert die Benutzer, deren Aufträge ausgewählt werden sollen.

SENDER-USERID=*OWN
Nur die Aufträge des Kommando-Aufrufers werden ausgewählt.

SENDER-USERID=*ANY
Für Benutzer ohne TSOS-Berechtigung entspricht diese Angabe der Angabe von *OWN. Für Benutzer mit TSOS-Berechtigung bedeutet diese Angabe, dass die Aufträge aller Benutzer ausgewählt werden.

SENDER-USERID=<name 1..8>
Benutzerkennung, deren Aufträge ausgewählt werden sollen. Nur Benutzer mit TSOS-Berechtigung dürfen Aufträge einer fremden Benutzerkennung angeben.

INFORMATION=*SUMMARY / *ALL
legt fest, welche Informationen zu den Aufträgen ausgegeben werden.

INFORMATION=*SUMMARY
Es werden nur die jeweiligen Summen der Auftragskategorien ausgegeben.

INFORMATION=*ALL
Es werden alle verfügbaren Informationen zu den ausgewählten Aufträgen ausgegeben.

Return codes

(SC2) SC1	Maincode	Bedeutung
0	CMD0001	Kein Fehler.
64	CMD0216	Der Benutzer hat nicht die notwendige Berechtigung für das Kommando.
32	CMD0220	Interner Fehler.
32	CMD2009	Fehler während der Erstellung der Ausgabe-Variable.
64	YML0120	Subsystem ASTI ist nicht verfügbar.
32	YML0176	Unerwarteter ASTI-Fehler.
64	YML0210	Auftrag nicht gefunden.
128	YML0214	Resourcen-Auslastung.
64	YML0216	Auftrag durch fremde Anwendung ausgeführt.
64	YML0218	Auftrag ist kein Mail-Sende-Auftrag.

SC1/2=Subcode 1/2 in Dezimal-Darstellung

Ausgabe-Daten

- Beim Operanden INFORMATION=*SUMMARY werden die Mail-Sende-Aufträge anhand der Operanden ORDER-ID und SENDER-USERID ausgewählt und in Kategorien zusammengefasst. Die Summe der Aufträge wird ausgegeben. Es werden fünf Kategorien unterschieden:

WAITING	Auftrag wartet auf die Durchführung durch die Backend-Task.
DEFERRED	Bei der Ausführung durch die Backend-Task ist ein zeitlich begrenzter Fehler aufgetreten (z.B. wenn der Mail-Server heruntergefahren ist). Nach einiger Zeit wird erneut versucht den Auftrag auszuführen.
ACTIVE	Auftrag wird derzeit durch die Backend-Task ausgeführt. Die Anwendung unterstützt keine parallele Verarbeitung. Daher kann die Zahl der Aufträge in dieser Kategorie nur 0 oder 1 sein, außer wenn ein Fehler auftritt.
SENT	Der Auftrag wurde erfolgreich ausgeführt.
FAILED	Bei der Verarbeitung des Auftrags ist ein Fehler aufgetreten.

  Die Daten können in einer S-Variablen abgelegt werden, die die Namen der Kategorien als Komponenten enthält.

- Bei INFORMATION=*ALL werden für jeden ausgewählten Auftrag mehrere Zeilen ausgegeben.

Die folgende Tabelle stellt den Labeln der SYSOUT-Zeilen die entsprechenden Komponenten-Namen der S-Variablen gegenüber. Falls ein bestimmter Wert nicht vorhanden ist, wird die zugehörige Zeile bzw. Variablen-Komponente unterdrückt.

SYSOUT	**S-Variable**	**Bedeutung**
ORDER-ID	ORDER-ID	Auftrags-ID
STATE	STA	Auftragsstatus (Waiting, Deferred, Active, Sent, Failed)
SEND TIME	SEND-TIME	Sendezeitpunkt der Mail
SENDER	SENDER	Benutzerkennung des Absenders der Mail
RETURN CODE	RETURN-CODE	Return Code der Backend-Task bei STATUS=Sent oder Failed
RETURN MESSAGE	RETURN-MSG	Auftragsende-Meldung des Mail-Servers oder Fehler-Meldung in Textform bei STATUS=Sent oder Failed
FROM	FROM	Wert des SEND-MAIL-Operanden FROM
TO	TO	Wert des SEND-MAIL-Operanden TO
CC	CC	Wert des SEND-MAIL-Operanden CC
BCC	BCC	Wert des SEND-MAIL-Operanden BCC

Beispiele

```
/EXECUTE-CMD CMD=(SHOW-MAIL-ORDER-STATUS INFORMATION=*SUMMARY),-
/            STRUCTURE-OUTPUT=OUT
%  # ORDERS
%    WAITING:     1
%    DEFERRED:    0
%    ACTIVE:      0
%    SENT:        1
%    FAILED:      0
%    TOTAL:       2

/SHOW-VARIABLE OUT
OUT(*LIST).WAITING = 1
OUT(*LIST).DEFERRED = 0
OUT(*LIST).ACTIVE = 0
OUT(*LIST).SENT = 1
OUT(*LIST).FAILED = 0

/EXECUTE-CMD CMD=(SHOW-MAIL-ORDER-STATUS ORDER=*ALL,INFORMATION=*ALL),-
/            STRUCTURE-OUTPUT=OUT
%  ORDER_ID:            01E7E48600000025
%  STATUS:              Waiting
%  SUBMISSION TIME:     2010-02-10 17:23:18
%  SUBMITTER:           CLAUDIO
%  FROM:                Claudio.Monteverdi@mantova.example
%  TO:                  Heinrich.Schuetz@dresden.example
%  CC:                  John.Bull@london.example
%  BCC:                 William.Byrd@london.example
%  ---
%  ORDER_ID:            01E7E48600000013
%  STATUS:              Sent
%  SUBMISSION TIME: 2010-02-10 11:05:54
%  SUBMITTER:           HEINRICH
%  RETURN CODE:         Ok
%  RETURN MESSAGE:      250 Ok: queued as BDB6F6EA2F
%  FROM:                Heinrich.Schuetz@dresden.example
%  TO:                  Claudio.Monteverdi@mantova.example

/SHOW-VARIABLE OUT
OUT(*LIST).ORDER-ID = 01E7E48600000025
OUT(*LIST).STA = Waiting
OUT(*LIST).SEND-TIME = 2010-02-10 17:23:18
OUT(*LIST).SENDER = CLAUDIO
OUT(*LIST).FROM = Claudio.Monteverdi@mantova.example
OUT(*LIST).TO = Heinrich.Schuetz@dresden.example
```

```
OUT(*LIST).CC = John.Bull@london.example
OUT(*LIST).BCC = William.Byrd@london.example
OUT(*LIST).ORDER-ID = 01E7E48600000013
OUT(*LIST).STA = Sent
OUT(*LIST).SEND-TIME = 2010-02-10 11:05:54
OUT(*LIST).SENDER = HEINRICH
OUT(*LIST).RETURN-CODE = Ok
OUT(*LIST).RETURN-MSG = 250 Ok: queued as BDB6F6EA2F
OUT(*LIST).FROM = Heinrich.Schuetz@dresden.example
OUT(*LIST).TO = Claudio.Monteverdi@mantova.example
```

9.3 Unterprogramm-Schnittstelle des Mail-Sender Frontend

Die Unterprogramm-Schnittstelle des Mail-Sender Frontend unterstützt die Programmiersprachen Assembler und C.

Dieser Abschnitt informiert über folgende Themen:

- Assembler-Makro-Schnittstelle
- Funktionsaufrufe in C

9.3.1 Assembler-Makro-Schnittstelle

Die Assembler-Unterprogramm-Schnittstelle des Mail-Sender Frontend wird durch die Assembler-Makro-Schnittstelle des DSSM-Subsystems MAILCLNT implementiert.

9.3.1.1 Eigenschaften

Die folgende Übersicht fasst die Eigenschaften der Makro-Schnittstelle zusammen, die für alle Makroaufrufe gelten:

Typ der Schnittstelle:	CALL
Linkage:	ISL-Linkage
SVC-Nummer	SVC 20 (dezimal)
Funktionsbereich:	TU (**T**ask **U**nprivileged) / TPR (**T**ask **Pr**ivileged)
Makro-Typ:	S
unterstützte MF-Formate:	MF-Format 3: MF = {C | D | E | L | M}
Assembler	`[ PREFIX = Y ]` `[ MACID = MLS ]`

9.3.1.2 Makroaufrufe (Überblick)

Folgende Makroaufrufe stehen zur Verfügung:

Makroaufruf	**Funktion**
YMLSML	Mail senden
YMLCML	Resultat abholen
YMLDML	Mail löschen
YMLGML	Information zur Mail abfragen

9.3.1.3 Beschreibungsformat für die Makroaufrufe

Die Makroaufrufe sind nach einem einheitlichen Schema beschrieben:

Makroname - Kurzbeschreibung der Funktionalität

Beschreibung der Funktionalität.

Entry-Namen oder SVC-Nummer(n)

Beschreibung der Entry-Namen und SVC-Nummern.

Makroauffruf-Format und Operandenbeschreibung

Makroname
Operanden ... . . .

Beschreibung der Operanden

Returncode

SRC2	SRC1	MRC	MRC-Name	Bedeutung
00	00	0000	Identifier	Bedeutung des Returncodes
. . .	. . .	. . .	. . .	...

SRC1/2=Sub Returncode 1/2 in Sedezimal-Darstellung;
MRC=Main Returncode in Sedezimal-Darstellung

Makroaufruf-Parameter

Beschreibung der Datenstruktur(en) für die Makroaufruf-Parameter.

Mail-Parameter

Beschreibung der Datenstruktur(en) für die Mail-Parameter.

9.3.1.4 Beschreibung der Makroaufrufe

Im Folgenden sind die einzelnen Assembler-Makroaufrufe beschrieben.

YMLSML - Mail senden

Mit diesem Makro können Sie Mails versenden.

Entry-Namen oder SVC-Nummer(n)

SVC 20 (dezimal)

UNIT=940, FUNCTION=20, VERSION=1

Makroaufruf-Format und Operandenbeschreibung

YMLSML
FL= ***TU** / ***TPR**
,**VERSION**=**1** / **2**
,**XPAND**=**PARAM** / **GENERAL** / **ADD_HEADER** / **DATA_SPEC** / **CHARSET** / **ENCODING** / **CONT_DISP** / **MSG_ATT**
,**MAILP**=<var: pointer>
,**MAILPL**=<integer: 1..32767> / <var: int:4>
,**OPTFILE**= ***NONE** / <var: char:54>
,**ENCRYPT**= ***NO** / ***YES** / ***OPTFILE**
,**SIGN**= ***NO** / ***YES** / ***OPTFILE**
,**SECPROT**= ***SMIME**
,**CIPHER**= ***3DES** / ***DES** / ***RC2-40** / ***RC2-64** / ***RC2-128** / ***AES-128** / ***AES-192** / ***AES-256** / ***OPTFILE**
,**WAIT**= ***NO-DISCARD** / ***NO** / ***YES**
,**WAITTIM=*UNLIM /** <integer: 1..65535> /<var: int:4>

FL=
Funktionsbereich

***TU**
SVC-Schnittstelle wird generiert.

***TPR**
CALL-Schnittstelle wird generiert.

VERSION=
Wählt die Schnittstellenversion aus.

1
Es wird die alte Schnittstellenversion ausgewählt.

2
Es wird die neue Schnittstellenversion ausgewählt, die den Operanden WAITTIM und zusätzliche Returncodes anbietet.

XPAND=
Diese Parameter steuern die Expansion der Datenstrukturen, die die Parameterliste des Makros und das Layout der Mail-Parameter beschreiben. Diese Datenstrukturen werden durch den Mail-Parameter MAILP referenziert. Eine Beschreibung der Datenstrukturen finden Sie ab Seite 447.

PARAM
Makro-Parameterliste wird generiert.

GENERAL
Daten-Layout für die allgemeinen Mail-Parameter wird generiert.

ADD_HEADER
Daten-Layout für die zusätzlichen Header-Zeilen wird generiert.

DATA_SPEC
Daten-Layout für die Spezifikation der Mail-Daten wird generiert.

CHARSET
Daten-Layout für die Spezifikation des Zeichensatzes wird generiert.

ENCODING
Daten-Layout für die Spezifikation der Kodierung wird generiert.

CONT_DISP
Daten-Layout für die Spezifikation der inhaltlichen Gliederung wird generiert.

MSG_ATT
Daten-Layout für die Klammerung von Nachrichten/Anhängen wird generiert.

MAILP=
Parameterbereich zur Beschreibung der Mail. Dieser Parameter ist obligatorisch.

IDENTIFIER
Variable, in der die Adresse des Parameterbereichs gespeichert ist, oder Register, das die Adresse des Parameterbereichs enthält.

MAILPL=
Länge des Parameterbereichs zur Mail-Beschreibung. Dieser Parameter ist obligatorisch.

INTEGER 1, 32767
Länge des Parameterbereichs.

IDENTIFIER
Variable, in der die Länge des Parameterbereichs gespeichert ist, oder Register, das die Länge des Parameterbereichs enthält.

OPTFILE=
Name der Benutzer-Option-Datei (siehe Seite 403). Die Options dieser Datei können durch einige Makro-Parameter überschrieben oder ergänzt werden.

***NONE**
Keine Datei definiert.

IDENTIFIER
Variable, in der der Name der Option-Datei gespeichert ist, oder Register, das die Adresse des Namens der Option-Datei enthält.

ENCRYPT=
Legt fest, ob der Anwender die zu sendende Nachicht verschlüsseln möchte.

***NO**
Mail wird nicht verschlüsselt.

***YES**
Mail wird verschlüsselt.

***OPTFILE**
Die Angabe aus der Benutzer-Option-Datei (siehe Seite 403) wird übernommen.

SIGN=
Dieser Parameter legt fest, ob der Aufrufer die zu sendende Nachicht signieren möchte.

***NO**
Mail wird nicht signiert.

***YES**
Mail wird signiert.

***OPTFILE**
Die Angabe aus der Benutzer-Option-Datei (siehe Seite 403) wird übernommen.

SECPORT=
Dieser Parameter legt das Verfahren zur Verschlüsselung/Signierung fest.
Derzeit wird nur das Verfahren S/MIME unterstützt.

***SMIME**
S/MIME wird verwendet.

CIPHER=
Symmetrischer Verschlüsselungs-Code.

***3DES**
Triple DES

***DES**
DES

***RC2-40**
RC2 mit 40 Bit Schlüssellänge.

***RC2-64**
RC2 mit 64 Bit Schlüssellänge.

***RC2-128**
RC2 mit 128 Bit Schlüssellänge.

***AES-128**
AES mit 128 Bit Schlüssellänge.

***AES-192**
AES mit 192 Bit Schlüssellänge.

***AES-256**
AES mit 256 Bit Schlüssellänge.

***OPTFILE**
Die Angabe aus der Benutzer-Option-Datei (siehe Seite 403) wird übernommen.

WAIT=
Legt fest, ob der Aufrufer die vollständige Erledigung des Mail-Sende-Auftrags durch das Mail-Sender-Backend abwarten will.

***NO-DISCARD**
Nicht warten und ASTI anweisen, den Ergebnis-Status des Backend der Mail-Sendung zu verwerfen.

***NO**
Nicht warten aber ASTI anweisen, den Ergebnis-Status des Backend der Mail-Sendung zu speichern.

***YES**
Warten bis das Mail-Sender-Backend die vollständige Erledigung des Sende-Auftrags oder einen Fehler beim Versenden der Mail meldet.

WAITTIM=
Mit diesem Operanden kann die Wartezeit bei WAIT=*YES begrenzt werden. Der Operand ist nur mit VERSION=2 verfügbar. Wenn die maximale Wartezeit abgelaufen ist, wird der Aufruf mit einem entsprechenden Return-Code beendet.

***UNLIM**
Unbegrenzte Wartezeit. Mit diesem Operandenwert ist das Verhalten wie bisher.

INTEGER 1,65535
Maximale Wartezeit in Sekunden.

IDENTIFIER
Variable, in der die Wartezeit gespeichert ist oder Register, das die Wartezeit enthält (jeweils in Sekunden).

Returncode

SRC2	SRC1	MRC	MRC-Name	Bedeutung
00	00	0000	YMLSSUCC	Es wurde kein Fehler festgestellt.
00	01	0001	YMLSPARE	Parameterfehler.
00	20	0002	YMLSINTE	Interner Fehler.
00	40	0003	YMLSMSYN	Syntax-Fehler im Mail-Parameter.
00	40	0004	YMLSOFNA	Option-Datei nicht verfügbar.
00	40	0005	YMLSMFNA	Nachricht oder Anhang nicht verfügbar.
00	40	0006	YMLSSFNA	Datei für SMIME nicht verfügbar.
00	40	0007	YMLSPTBG	Mail-Parameter zu groß.
00	40	0008	YMLSBACK	Backend-Fehler.
00	80	0009	YMLSMORD	Maximale Anzahl an Aufträgen überschritten.
00	80	0010	YMLSTIME	Maximale Wartezeit erreicht.
00	80	000A	YMLSSNAV	Mail-Client-Service nicht verfügbar.
00	40	000B	YMLSAINV	Mail-Parameter-Adresse ungültig.
00	40	000C	YMLSRSRC	Ressourcen sind erschöpft.
00	40	000D	YMLSANAV	Subsystem ASTI nicht verfügbar.
00	40	000E	YMLSNFRA	Keine FROM-Adresse angegeben.
00	20	000F	YMLSASTI	Unerwarteter ASTI-Fehler.

SRC1/2=Sub Returncode 1/2 in Sedezimal-Darstellung;
MRC=Main Returncode in Sedezimal-Darstellung

Makroaufruf-Parameter

Die Datenstruktur für die Makroaufruf-Parameter von YMLSML ist wie folgt aufgebaut:

VERSION=1

Distanz	Identifier	Wert	Bedeutung
	YMLSPARL		Parameter Area
000	YMLSHDR		Function-Header
008	YMLSIND		Eingabeparameter
008	YMLSMPAR		Adresse des Mail-Parameterbereichs
00C	YMLSMPRL		Länge des Mail-Parameterbereichs
010	YMLSWAIT		Auf Ende des Mail-Sende-Auftrags warten?
	YMLSWYES	1	Ja
	YMLSWNO	2	Nein; ASTI verwirft Ausführungsstatus des Auftrags.
	YMLSWRES	3	Nein; ASTI verwirft Ausführungsstatus des Auftrags nicht.
011	YMLSSPRO		Protokoll für Verschlüsselung und Signierung
	YMLSSMIM	1	S/MIME
012	YMLSENC		Verschlüsselung?
	YMLSGYES	1	Ja
	YMLSGNO	2	Nein
	YMLSGOPT	3	Wert aus Benutzer-Option-Datei
013	YMLSSIGN		Signierung?
	YMLSGYES	1	Ja
	YMLSGNO	2	Nein
	YMLSGOPT	3	Wert aus Benutzer-Option-Datei
014	YMLSCPHR		Verschlüsselungsalgorithmus:
	YMLSRC20	1	RC2-40
	YMLSRC24	2	RC2-64
	YMLSRC28	3	RC2-128
	YMLSDES	4	DES
	YMLS3DES	5	3DES
	YMLSAES8	6	AES-128
	YMLSAES2	7	AES-192
	YMLSAES6	8	AES-256
	YMLSFOPT	127	Wert aus Benutzer-Option-Datei
015	YMLSOPTF		Name der Benutzer-Option-Datei

Distanz	Identifier	Wert	Bedeutung
04B	YMLSRSV1		Reservierter Bereich
04C	YMLSOUTD		Ausgabeparameter
04C	YMLSOID		Auftrags-ID zur Referenzierung des Mail-Sende-Auftrags, falls der Aufrufer nicht auf die vollständige Übertragung wartet.
05C	YMLSRETC		Return-Code vom Back-End
	YMLSBOK	00	Ok
	YMLSBPER	01	Parameter-Fehler
	YMLSBRSC	02	Resourcen sind erschöpft
	YMLSBSMT	03	SMTP-Fehler
	YMLSBSMI	04	S/MIME-Fehler
	YMLSBINT	0A	Interner Fehler
060	YMLSRETM		Enthält ggf. entsprechende Fehlermeldungen in Textformat, z.B. SMTP-Fehler-Meldungen des SMTP-Servers.
100	YMLSARET		Wenn der YMLSML-Aufruf mit dem Return-Code YMLSASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.

VERSION=2:

Distanz	Identifier	Wert	Bedeutung
	YMLSPARL		Parameter Area
000	YMLSHDR		Function-Header
008	YMLSIND		Eingabeparameter
008	YMLSMPAR		Adresse des Mail-Parameterbereichs
00C	YMLSMPRL		Länge des Mail-Parameterbereichs
010	YMLSWTTM		Maximale Wartezeit
014	YMLSWAIT		Auf Ende des Mail-Sende-Auftrags warten?
	YMLSWYES	1	Ja
	YMLSWNO	2	Nein; ASTI verwirft Ausführungsstatus des Auftrags.
	YMLSWRES	3	Nein; ASTI verwirft Ausführungsstatus des Auftrags nicht.
015	YMLSSPRO		Protokoll für Verschlüsselung und Signierung
	YMLSSMIM	1	S/MIME
016	YMLSENC		Verschlüsselung?
	YMLSGYES	1	Ja
	YMLSGNO	2	Nein

Distanz	Identifier	Wert	Bedeutung
	YMLSGOPT	3	Wert aus Benutzer-Option-Datei
017	YMLSSIGN		Signierung?
	YMLSGYES	1	Ja
	YMLSGNO	2	Nein
	YMLSGOPT	3	Wert aus Benutzer-Option-Datei
018	YMLSCPHR		Verschlüsselungsalgorithmus:
	YMLSRC20	1	RC2-40
	YMLSRC24	2	RC2-64
	YMLSRC28	3	RC2-128
	YMLSDES	4	DES
	YMLS3DES	5	3DES
	YMLSAES8	6	AES-128
	YMLSAES2	7	AES-192
	YMLSAES6	8	AES-256
	YMLSFOPT	127	Wert aus Benutzer-Option-Datei
019	YMLSOPTF		Name der Benutzer-Option-Datei
04F	YMLSRSV1		Reservierter Bereich
050	YMLSOUTD		Ausgabeparameter
050	YMLSOID		Auftrags-ID zur Referenzierung des Mail-Sende-Auftrags, falls der Aufrufer nicht auf die vollständige Übertragung wartet.
060	YMLSRETC		Return-Code vom Back-End
	YMLSBOK	00	Ok
	YMLSBPER	01	Parameter-Fehler
	YMLSBRSC	02	Resourcen sind erschöpft
	YMLSBSMT	03	Allgemeiner SMTP-Fehler
	YMLSBSMI	04	S/MIME-Fehler
	YMLSBSMM	05	Fehler bei SMTP MAIL-Kommando
	YMLSBSMR	06	Fehler bei SMTP RCPT-Kommando
	YMLSBSMD	07	Fehler bei SMTP DATA-Kommando
	YMLSBFAC	08	Fehler bei Zugriff auf Benutzer-Option-Datei
	YMLSBMTL	09	Mail zu groß
	YMLSBINT	0A	Interner Fehler
064	YMLSRETM		Enthält ggf. ergänzende Fehlermeldungen in Textformat, z.B. SMTP-Fehler-Meldungen des SMTP-Servers.

Distanz	Identifier	Wert	Bedeutung
104	YMLSARET		Wenn der YMLSML-Aufruf mit dem Return-Code YMLSASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.
108	YMLSMID		Meldungs-ID
10F	YMLSRSV2		Reservierter Bereich

Ergänzende Erläuterungen zu den Ausgabeparametern:

YMLSRETC

Einige Fehler-Situationen, bei denen in VERSION=1 ein Sammel-Return-Code geliefert wird, werden bei VERSION=2 durch spezifische Return-Codes abgedeckt, um schneller die eigentliche Fehler-Ursache feststellen zu können. So ist der Return-Code YMLSBMM ein starker Hinweis darauf, dass ein Fehler in der Absender-Adresse vorliegt. Entsprechend weist der Return-Code YMLSBMR auf einen Fehler in mindestens einer der Empfänger-Adressen hin. Ein YMLSBMD kann z.B. auftreten, wenn der SMTP-Server die Verletzung bestimmter Regeln erst nach dem DATA-Kommando prüft, die Ursache der Regel-Verletzung kann mit den angegebenen Adressen zusammenhängen. In jedem Fall sollte das Feld YMLSRETM weitere (Text-)Information zur Fehlerursache geben.

YMLSRETM

Wenn der verwendete SMTP-Server den RFC 2034 unterstützt, dann enthält dieses Feld eine maschineninterpretierbare Fehleranzeige, die i.A. spezifischer ist als die SMTP-bezogenen Return-Codes des YMLSRETC-Feldes.

YMLSMID

Dieses Feld enthält einen YML-Meldungsschlüssel für eine Meldung, die den aufgetretenen Fehler beschreibt. Das ist normaler weise die gleiche Meldung, die ein vergleichbarer SEND-MAIL-Kommandoaufruf liefern würde.

Auflistung der Expansion der Datenstruktur für die Makroaufruf-Parameter

Expansion:

XPAND= PARAM

```
                                YMLSML MF=D,XPAND=PARAM
                     1          MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                     1                DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSPARL
000000               2 YMLSPARL DSECT ,
                     1 *   Parameter area
                     1 YMLSHDR  FHDR  MF=(C,YMLS),EQUATES=NO                          FHDR
                     1 *   main return codes
            00000000 1 YMLSSUCC EQU   0                        No error detected
            00000001 1 YMLSPARE EQU   1                        Parameter error
            00000002 1 YMLSINTE EQU   2                        Internal error
            00000003 1 YMLSMSYN EQU   3                        Syntax error
            00000004 1 YMLSOFNA EQU   4                        Option file not
                                                               accessible
            00000005 1 YMLSMFNA EQU   5                        Message or attachment
                                                               file
                     1 *                                       not accessible
            00000006 1 YMLSSFNA EQU   6                        SMIME related file not
                     1 *                                       accessible
            00000007 1 YMLSPTBG EQU   7                        Mail parameter too big
            00000008 1 YMLSBACK EQU   8                        Back-end error
            00000009 1 YMLSMORD EQU   9                        Max number of orders
                                                               exceeded
            0000000A 1 YMLSSNAV EQU   10                       Mailclient Service not
                     1 *                                       available
            0000000B 1 YMLSAINV EQU   11                       Mailpar address invalid
            0000000C 1 YMLSRSRC EQU   12                       Resource saturation
            0000000D 1 YMLSANAV EQU   13                       Subsystem ASTI not
                                                               available
            0000000E 1 YMLSNFRA EQU   14                       No FROM address specified
            0000000F 1 YMLSASTI EQU   15                       Unexpected ASTI error
                     1 *
                     1 *
000008               1 YMLSIND  DS    0XL68                   Input parameters
000008               1 YMLSMPAR DS    A                       Mail parameter area
00000C               1 YMLSMPRL DS    F                       Mail parameter area
                                                               length
000010               1 YMLSWAIT DS    FL1                     wait
                     1 *   Wait operand values
            00000001 1 YMLSWYES EQU   1                        YES
            00000002 1 YMLSWNO  EQU   2                        NO-DISCARD
            00000003 1 YMLSWRES EQU   3                        NO
                     1 *
000011               1 YMLSSPRO DS    FL1                     Protocol for encryption
                                                               and
                     1 *                                       signing
                     1 *   Mail encryption/signing protocol
            00000001 1 YMLSSMIM EQU   1                        SMIME
```

```
                    1 *
000012              1 YMLSENC  DS    FL1                       Encryption
                    1 *   General yes or no selection
          00000001  1 YMLSGYES EQU   1                         YES
          00000002  1 YMLSGNO  EQU   2                         NO
          00000003  1 YMLSGOPT EQU   3                         OPTFILE
                    1 *
000013              1 YMLSSIGN DS    FL1                       Signing
000014              1 YMLSCPHR DS    FL1                       Cipher
                    1 *   Cipher operand values
          00000001  1 YMLSRC20 EQU   1                         RC2-40
          00000002  1 YMLSRC24 EQU   2                         RC2-64
          00000003  1 YMLSRC28 EQU   3                         RC2-128
          00000004  1 YMLSDES  EQU   4                         DES
          00000005  1 YMLS3DES EQU   5                         3DES
          00000006  1 YMLSAES8 EQU   6                         AES-128
          00000007  1 YMLSAES2 EQU   7                         AES-192
          00000008  1 YMLSAES6 EQU   8                         AES-256
          0000007F  1 YMLSFOPT EQU   127                       OPTFILE
                    1 *
000015              1 YMLSOPTF DS    CL54                      Option file
00004B              1 YMLSRSV1 DS    CL1                       Reserved
                    1 *
                    1 *
00004C              1 YMLSOUTD DS    0XL184                    Output parameters
00004C              1 YMLSOID  DS    CL16                      Order Id
00005C              1 YMLSRETC DS    F                         Return code
                    1 *   rc
          00000000  1 YMLSBOK  EQU   0                         OK
          00000001  1 YMLSBPER EQU   1                         Parameter error
          00000002  1 YMLSBRSC EQU   2                         Resource saturation
          00000003  1 YMLSBSMT EQU   3                         SMTP error
          00000004  1 YMLSBSMI EQU   4                         SMIME error
          0000000A  1 YMLSBINT EQU   10                        Internal error
                    1 *
000060              1 YMLSRETM DS    CL160                     Return message
000100              1 YMLSARET DS    F                         Return code from ASTI
                    1 *
          00000104  1 YMLS#    EQU   *-YMLSHDR
```

Mail-Parameter

Eine Mail besteht aus vielen, zum Teil optionalen Bestandteilen, deren Länge meist variabel ist. Deshalb werden für die Spezifikation der Mail-Parameter geordnete Datenstrukturen verwendet, die auf Tupeln der Form (typ, länge, wert) basieren.

Der Typ der einzelnen Parameter ist ein 2 Byte langes Integer-Feld. Das Feld enthält einen der Werte, die in der nachfolgenden Aufzählung aufgelistet sind.

Wertname	Tag #	Bedeutung
YMLSFROM	1	Envelope Sender-Adresse (wird auch in die FROM Header-Zeile gesetzt, wenn YMLSFFRM nicht angegeben ist).
YMLSFFRM	2	Absender-Adresse in der FROM Header-Zeile.
YMLSTO	3	Liste von Envelope Empfänger-Adressen, mit Komma getrennt (wird auch in die TO Header-Zeile gesetzt, wenn YMLSFTO nicht angegeben ist).
YMLSFTO	4	Liste von Empfänger-Adressen, mit Komma getrennt, wird in die To Header-Zeile gesetzt.
YMLSCC	5	Liste von Envelope Empfänger-Adressen, mit Komma getrennt (wird auch in die CC Header-Zeile gesetzt, wenn YMLSFCC nicht angegeben ist).
YMLSFCC	6	Liste von Empfänger-Adressen, mit Komma getrennt, wird in die Cc Header-Zeile gesetzt.
YMLSBCC	7	Liste von Envelope Empfänger-Adressen, mit Komma getrennt.
YMLSRPLT	8	Adresse in der REPLY-TO Header-Zeile.
YMLSSBJT	9	Text für die SUBJECT Header-Zeile.
YMLSAHDR	10	Zusätzliche Header-Zeilen, in denen der Aufrufer den Feld-Namen selbst bestimmen kann.
YMLSMBEG	11	Anfang der Definition des Nachrichten-Textes
YMLSMEND	12	Ende der Definition des Nachrichten-Textes
YMLSABEG	13	Anfang der Anhang-Definition
YMLSAEND	14	Ende der Anhang-Definition
YMLSDSPC	15	Daten-Definition für Nachricht/Anhang
YMLSCHST	16	Zeichensatz-Definition für Header/Nachricht/Anhang
YMLSTRCD	17	Transfer-Codierung von Nachricht/Anhang
YMLSCTTT	18	Inhaltstyp von Nachricht/Anhang
YMLSTCTD	19	Präsentationshinweis für Nachricht/Anhang
YMLSKEYF	21	Key-Datei (siehe Benutzer-Option-Datei „privateKeyFile“ auf Seite 406)
YMLSSGNF	22	Datei, die ein X.509-Zertifikat zum Signieren von Mails mit S/MIME enthält (siehe Benutzer-Option „signerCertificateFile“ auf Seite 406).

Wertname	Tag #	Bedeutung
YMLSASCF	23	Datei, die zusätzliche X.509-Zertifikate zum Signieren von Mails mit S/MIME enthält (siehe Benutzer-Option „addSignerCertificatesFile“ auf Seite 407).
YMLSRCTF	24	Datei, die X.509-Zertifikate zur Verschlüsselung von Mails mit S/MIME enthält (siehe Benutzer-Option „recipientCertificatesFile“ auf Seite 407).
YMLSCRLF	25	Datei, die Widerruf-Listen (CRL) für X.509-Zertifikate enthält (siehe Benutzer-Option „certificateRevocationListFile“ auf Seite 408).

Hinweise zur Definition des Eingabebereichs für die Mail-Parameter

– Wenn einer der Parameter *YMLSKEYF*, *YMLSSGNF*, *YMLSASCF* oder *YMLSCRLF* spezifiziert ist, wird die entsprechende Option der Benutzer-Option-Datei ignoriert. Durch Angabe des Wertes *NONE wird diese Option deaktiviert. Dies ist z.B. der Fall, wenn ein X.509-Zertifikat zur Signierung von Mails mit S/MIME verwendet (*YMLSSGNF*) wird, das sich von dem in der Option-Datei spezifizierten X.509-Zertifikat unterscheidet, und für das im Gegensatz zum X.509-Zertifikat in der Option-Datei keine zusätzlichen X.509-Zertifikate notwendig sind (YMLSASCF:*NONE).

– Wenn der Parameter *YMLSRCTF* angegeben ist, wird die dort spezifizierte Datei in der Suchreihenfolge vor die Dateien gestellt, die in *recipientCertificatesFile*-Options angegeben sind.

– Die Implementierung des Subsystems ASTI beschränkt die Größe der Mail-Parameter. Ein exakter Wert lässt sich hier nicht angeben. Einen Anhaltspunkt liefert ein Wert von ungefähr 32 KB, abzüglich einiger Bytes, die für zusätzliche Parameter und interne Verwaltungsdaten reserviert sind. Vermutlich wird diese Grenze nur dann erreicht, wenn der Text der Mail oder Daten in den Anhängen direkt in den Parameterbereich der Mail eingefügt werden. Tritt dieser Fehler auf, dann schreiben Sie den Mail-Text in eine (temporäre) Datei und geben Sie im Parameterbereich statt dessen nur den Dateinamen an.

– Die Daten mit variabler Länge beginnen direkt nach dem zugehörigen Header. Die Header müssen auf volle Wortlänge ausgerichtet sein. Daher müssen Sie Füll-Bytes einfügen, wenn der variable Teil eine Anzahl Bytes enthält, die kein Vielfaches von 4 ist.

– Der Feldname des Mail-Headers muss in der Datenstruktur für zusätzliche Header ohne den darauffolgenden Doppelpunkt eingegeben werden.

– Sofern nicht anders vermerkt, ist die Reihenfolge der Datenstrukturen ohne Bedeutung.

– Die Definition des Mail-Textes oder eines Anhangs muss mit zwei *YMLSATT* -Datenstrukturen geklammert sein, bei denen das Tag-Feld mit *YMLSMBEG*/*YMLSMEND* (Mail) bzw. *YMLSABEG*/*YMLSAEND* versehen ist.

  Die Definition des Mail-Textes oder eines Anhangs kann Datenstrukturen mit den folgenden Tags enthalten:

  ```
  YMLSDSPC, YMLSCHST, YMLSTRCD, YMLSTCTD
  ```

  Liegt eine *YMLSCSET* -Datenstruktur (mit YMLSCHST-Tag) außerhalb einer *YMLSATT* -Datenstruktur, dann wird sie auf die nachfolgenden Header-Zeilen (siehe RFC 2047) angewendet (gegebenenfalls bis zum Auftreten einer weiteren YMLSCSET-Datenstruktur).

– Alle spezifizierten Dateien müssen unverändert bleiben, bis die Mail gesendet ist.

Auflistung der Expansion der Datenstrukturen für die Mail-Parameter

Expansion:

XPAND= GENERAL, ADD_HEADER, DATA_SPEC, CHARSET, ENCODING, CONT_DISP, MSG_ATT

```
                              YMLSML MF=D,XPAND=GENERAL
                    1          MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                    1                DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSGNRL
000000              2 YMLSGNRL DSECT ,
                    1 *   Input parameters
000000              1 YMLSGTAG DS    H                        tag of general parameter
                    1 *   Tag value(s) for struct _general
        00000001    1 YMLSFROM EQU   1                        from
        00000002    1 YMLSFFRM EQU   2                        fake_from
        00000003    1 YMLSTO   EQU   3                        to
        00000004    1 YMLSFTO  EQU   4                        fake_to
        00000005    1 YMLSCC   EQU   5                        cc
        00000006    1 YMLSFCC  EQU   6                        fake_cc
        00000007    1 YMLSBCC  EQU   7                        envelope recipient mail
                    1 *                                      addresses
        00000008    1 YMLSRPLT EQU   8                        reply_to
        00000009    1 YMLSSBJT EQU   9                        subject
        00000012    1 YMLSCTTT EQU   18                       content_type
        00000014    1 YMLSOPFI EQU   20                       option file
        00000015    1 YMLSKEYF EQU   21                       key file
        00000016    1 YMLSSGNF EQU   22                       signer certificate file
        00000017    1 YMLSASCF EQU   23                       additional signer
                    1 *                                      certificates file
        00000018    1 YMLSRCTF EQU   24                       recipient certificate
                                                             file
        00000019    1 YMLSCRLF EQU   25                       CRL file
                    1 *
000002              1 YMLSGRS1 DS    XL2                      reserved
000004              1 YMLSGLEN DS    F                        general parameter length
        00000008    1 YMLSGNRL# EQU   *-YMLSGTAG


                              YMLSML MF=D,XPAND=ADD_HEADER
                    1          MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                    1                DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSHEAD
000000              2 YMLSHEAD DSECT ,
                    1 *   Input parameters
000000              1 YMLSHTAG DS    H                        tag of add. header
                    1 *   Tag value(s) for struct _add_header
        0000000A    1 YMLSAHDR EQU   10                       add. header
                    1 *
000002              1 YMLSHRS1 DS    XL2                      reserved
000004              1 YMLSHLNN DS    F                        add. header name length
```

```
000008             1 YMLSHLNB DS    F                          add. header body length
        0000000C   1 YMLSHEAD# EQU   *-YMLSHTAG


                             YMLSML MF=D,XPAND=DATA_SPEC
                   1         MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                   1               DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSDTSP
000000             2 YMLSDTSP DSECT ,
                   1 *   Data specification
000000             1 YMLSDTAG DS    H                          tag of add. header
                   1 *   Tag value(s) for struct _data_spec
        0000000F   1 YMLSDSPC EQU   15                         dataspec
                   1 *
000002             1 YMLSDTYP DS    FL1                        reserved
                   1 *   Type of data specification
        00000001   1 YMLSFILE EQU   1                          file
        00000002   1 YMLSDATA EQU   2                          data
                   1 *
000003             1 YMLSDRS1 DS    XL1                        reserved
000004             1 YMLSDLEN DS    F                          length of data. spec.
        00000008   1 YMLSDTSP# EQU   *-YMLSDTAG


                             YMLSML MF=D,XPAND=CHARSET
                   1         MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                   1               DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSCSET
000000             2 YMLSCSET DSECT ,
                   1 *   Data specification
000000             1 YMLSCTAG DS    H                          tag of add. header
                   1 *   Tag value(s) for struct _charset
        00000010   1 YMLSCHST EQU   16                         character set
                   1 *
000002             1 YMLSCBNR DS    FL1                        binary data
                   1 *   Binary data selection
        00000001   1 YMLSBYES EQU   1                          YES
        00000002   1 YMLSBNO  EQU   2                          NO
                   1 *
000003             1 YMLSCRS1 DS    XL1                        reserved
000004             1 YMLSCLEN DS    F                          length
000008             1 YMLSCSCX DS    CL8                        src_charset_XHCS
000010             1 YMLSCDCX DS    CL8                        dest_charset_XHCS
        00000018   1 YMLSCSET# EQU   *-YMLSCTAG


                             YMLSML MF=D,XPAND=ENCODING
                   1         MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                   1               DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSENCD
000000             2 YMLSENCD DSECT ,
                   1 *   Encoding
000000             1 YMLSETAG DS    H                          tag of encoding
                    1 *   Tag value(s) for struct _encoding
        00000011   1 YMLSTRCD EQU   17                         transfer encoding
```

```
                    1 *
000002              1 YMLSEMCH DS    FL1                        encoding mechanism
                    1 *   Type of encoding mechanism
          00000001  1 YMLSE7BT EQU   1                          7 bit
          00000002  1 YMLSE8BT EQU   2                          8 bit
          00000003  1 YMLSEBIN EQU   3                          binary
          00000004  1 YMLSEQP  EQU   4                          quoted printable
          00000005  1 YMLSEB64 EQU   5                          base64
                    1 *
000003              1 YMLSERS1 DS    XL1                        reserved
          00000004  1 YMLSENCD# EQU   *-YMLSETAG


                              YMLSML MF=D,XPAND=CONT_DISP
                    1         MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                    1               DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSDISP
000000              2 YMLSDISP DSECT ,
                    1 *   Content disposition
000000              1 YMLSITAG DS    H                         tag of content
                                                               disposition
                    1 *   Tag value(s) for struct _cont_disp
          00000013  1 YMLSTCTD EQU   19                         content disposition
                    1 *
000002              1 YMLSISPT DS    FL1                       content disposition
                    1 *   Disposition type
          00000001  1 YMLSINLN EQU   1                          inline
          00000002  1 YMLSIATT EQU   2                          attachment
                    1 *
000003              1 YMLSIRS1 DS    XL1                       reserved
          00000004  1 YMLSDISP# EQU   *-YMLSITAG


                              YMLSML MF=D,XPAND=MSG_ATT
                    1         MFTST MF=D,PREFIX=Y,MACID=MLS,ALIGN=F,
                    1               DMACID=MLS,SUPPORT=(E,D,C,M,L),DNAME=MLSATT
000000              2 YMLSATT  DSECT ,
                    1 *   Bracketing message or attachment specification
000000              1 YMLSATAG DS    H                         tag of msg | att
                    1 *   Tag value(s) for struct _msg_att
          0000000B  1 YMLSMBEG EQU   11                         msg_begin
          0000000C  1 YMLSMEND EQU   12                         msg_end
          0000000D  1 YMLSABEG EQU   13                         att_begin
          0000000E  1 YMLSAEND EQU   14                         att_end
                    1 *
000002              1 YMLSARS1 DS    XL2                       reserved
          00000004  1 YMLSATT# EQU   *-YMLSATAG
```

YMLCML - Mail-Resultat abholen

Mit diesem Makro können Sie überprüfen, ob ein Mail-Auftrag abgeschlossen wurde. Wenn der Auftrag ausgeführt wurde, wird der Ausführungs-Status an den Aufrufer übermittelt und aus den ASTI-Warteschlangen entfernt.

Entry-Namen oder SVC-Nummer(n)

SVC 20 (dezimal)

UNIT=940, FUNCTION=21, VERSION=1

Makroauffruf-Format und Operandenbeschreibung

YMLCML
FL= ***TU** / ***TPR** ,**VERSION**=**1** / **2** **,ORDER**= ***ANY** / ***SINGLE** ,**ORDERID**= <var: char:16> ,**WAIT**= ***NO** / ***YES** ,**WAITTIM=*UNLIM /** <integer: 1..65535> /**<**var: int:4> ,**OPTFILE=*NONE** / <var: char:54>

FL=
Funktionsbereich

***TU**
SVC-Schnittstelle wird generiert.

***TPR**
CALL-Schnittstelle wird generiert.

VERSION=
Wählt die Schnittstellenversion aus.

1
Es wird die alte Schnittstellenversion ausgewählt.

2
Es wird die neue Schnittstellenversion ausgewählt, die die Operanden WAITTIM und OPTFILE und zusätzliche Returncodes anbietet.

ORDER=
Legt fest, von welchen Mail-Sende-Aufträgen das Ergebnis angefordert werden soll.

***ANY**
Der Ausführungsstatus eines vom Makro-Aufrufer abgesendeten, abgeschlossenen Auftrags wird abgefragt. Liegen mehrere abgeschlossene Aufträge vor, dann ist nicht definiert, von welchem dieser Aufträge der Status abgefragt wird.

***SINGLE**
Das Ergebnis des durch den Parameter ORDERID identifizierten Mail-Sende-Auftrags wird abgefragt.

ORDERID=
Falls ORDER=*SINGLE angegeben ist, legt dieser Parameter die ID des zu überprüfenden Mail-Sende-Auftrags fest.

IDENTIFIER
Variable, in der die Auftrags-ID gespeichert ist, oder Register, das die Adresse einer Variablen enthält, in der die Auftrags-ID gespeichert ist.

WAIT=
Legt fest, ob der Aufrufer abwarten will, bis die Bearbeitung des Mail-Sende-Auftrags durch das Mail-Sender-Backend vollständig abgeschlossen ist.

***NO**
Nicht warten.

***YES**
Warten bis das Mail-Sender-Backend die vollständige Erledigung des Sende-Auftrags oder einen Fehler beim Versenden der Mail meldet.

WAITTIM=
Mit diesem Operanden kann die Wartezeit bei WAIT=*YES begrenzt werden. Der Operand ist nur mit VERSION=2 verfügbar. Wenn die maximale Wartezeit abgelaufen ist, wird der Aufruf mit einem entsprechenden Return-Code beendet.

***UNLIM**
Unbegrenzte Wartezeit. Mit diesem Operandenwert ist das Verhalten wie bisher.

INTEGER 1,65535
Maximale Wartezeit in Sekunden.

IDENTIFIER
Variable, in der die Wartezeit gespeichert ist oder Register, das die Wartezeit enthält (jeweils in Sekunden).

OPTFILE=
Mit diesem Operanden kann eine Benutzer-Option-Datei angegeben werden. Der Operand ist nur mit VERSION=2 verfügbar. Die Benutzer-Option-Datei kann durch Einstellungen beim benutzerspezifischen Logging Auswirkungen auf das Verhalten des Kommandos haben (siehe logItems-Option).

***NONE**
Keine Datei definiert.

IDENTIFIER
Variable, in der der Name der Option-Datei gespeichert ist, oder Register, das die Adresse des Namens der Option-Datei enthält.

Returncode

SRC2	SRC1	MRC	MRC-Name	Bedeutung
00	00	0000	YMLCSUCC	Mail ohne Fehler abgeschlossen.
00	01	0001	YMLCPARE	Parameterfehler.
00	20	0002	YMLCINTE	Interner Fehler.
00	40	0003	YMLCONTF	Auftrag nicht gefunden.
00	40	0004	YMLCFTSK	Auftrag durch fremde Task abgeschickt.
00	40	0005	YMLCANAV	Subsystem ASTI nicht verfügbar.
00	40	0006	YMLCNORR	Kein Ergebnis angefordert.
00	40	0007	YMLCONCM	Auftrag nicht abgeschlossen.
00	40	0008	YMLCASTI	Unerwarteter ASTI-Fehler.
00	40	0009	YMLCTIME	Maximale Wartezeit erreicht.
00	80	000A	YMLCOFNA	Fehler beim Zugriff auf Benutzer-Option-Datei.
00	80	000B	YMLCRSRC	Resourcen sind erschöpft.
00	40	000C	YMLCSNAV	Service MAILCLNT nicht verfügbar.

SRC1/2=Sub Returncode 1/2 in Sedezimal-Darstellung;
MRC=Main Returncode in Sedezimal-Darstellung

Makroaufruf-Parameter

Die Datenstruktur für die Makroaufruf-Parameter von YMLCML ist wie folgt aufgebaut:

VERSION=1:

Distanz	Identifier	Wert	Bedeutung
	YMLCPARL		Parameter Area
000	YMLCHDR		Function Header
008	YMLCIND		Eingabeparameter
008	YMLCOIDI		ID des Mail-Sende-Auftrags im Fall YMLCOSNG.
018	YMLCORDS		Von welchem Mail-Sende-Auftrag soll das Ergebnis abgefragt werden?
	YMLCOANY	1	Von beliebigen Aufträgen des Aufrufers.
	YMLCOSNG	2	Von dem Auftrag mit der ID YMLCOIDI.
019	YMLCWAIT		Auf Ende des Mail-Sende-Auftrags warten?
	YMLCWYES	1	Ja
	YMLCWNO	2	Nein
01A	YMLCRSV1		Reservierter Bereich
01C	YMLCOUTD		Ausgabeparameter
01C	YMLCOOID		Auftrags-ID des ausgewählten Auftrags, falls beliebiger Auftrag (YMLCOANY).
02C	YMLCRETC		Return-Code vom Back-End
	YMLCBOK	00	Ok
	YMLCBPER	01	Parameter-Fehler
	YMLCBRSC	02	Resourcen sind erschöpft
	YMLCBSMT	03	SMTP-Fehler
	YMLCBSMI	04	S/MIME-Fehler
	YMLCBINT	0A	Interner Fehler
030	YMLCRETM		Enthält ggf. entsprechende Fehlermeldungen in Textformat, z.B. SMTP-Fehler-Meldungen des SMTP-Servers.
0D0	YMLCARET		Wenn der YMLCML-Aufruf mit dem Return-Code YML-CASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.

VERSION=2:

Distanz	Identifier	Wert	Bedeutung
	YMLCPARL		Parameter Area
000	YMLCHDR		Function Header
008	YMLCIND		Eingabeparameter
008	YMLCOIDI		ID des Mail-Sende-Auftrags im Fall YMLCOSNG.
018	YMLCWTTM		Maximale Wartezeit
01C	YMLCORDS		Von welchem Mail-Sende-Auftrag soll das Ergebnis abgefragt werden?
	YMLCOANY	1	Von beliebigen Aufträgen des Aufrufers.
	YMLCOSNG	2	Von dem Auftrag mit der ID YMLCOIDI.
01D	YMLCWAIT		Auf Ende des Mail-Sende-Auftrags warten?
	YMLCWYES	1	Ja
	YMLCWNO	2	Nein
01E	YMLCOPTF		Name der Benutzer-Option-Datei
054	YMLCOUTD		Ausgabeparameter
054	YMLCOID		Auftrags-ID des ausgewählten Auftrags, falls beliebiger Auftrag (YMLCOANY).
064	YMLCRETC		Return-Code vom Back-End
	YMLCBOK	00	Ok
	YMLCBPER	01	Parameter-Fehler
	YMLCBRSC	02	Resourcen sind erschöpft
	YMLCBSMT	03	Allgemeiner SMTP-Fehler
	YMLCBSMI	04	S/MIME-Fehler
	YMLCBSMM	05	Fehler bei SMTP MAIL-Kommando
	YMLCBSMR	06	Fehler bei SMTP RCPT-Kommando
	YMLCBSMD	07	Fehler bei SMTP DATA-Kommando
	YMLCBFAC	08	Fehler bei Zugriff auf Benutzer-Option-Datei
	YMLCBMTL	09	Mail zu groß
	YMLCBINT	0A	Interner Fehler
068	YMLCRETM		Enthält ggf. ergänzende Fehlermeldungen in Textformat, z.B. SMTP-Fehler-Meldungen des SMTP-Servers.
108	YMLCARET		Wenn der YMLCML-Aufruf mit dem Return-Code YMLCASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.
10C	YMLCMID		Meldungs-ID

Distanz	Identifier	Wert	Bedeutung
113	YMLCRSV1		Reservierter Bereich

Ergänzende Erläuterungen zu den Ausgabeparametern:

YMLCRETC

Einige Fehler-Situationen, bei denen in VERSION=1 ein Sammel-Return-Code geliefert wird, werden bei VERSION=2 durch spezifische Return-Codes abgedeckt, um schneller die eigentliche Fehler-Ursache feststellen zu können. So ist der Return-Code YMLCBMM ein starker Hinweis darauf, dass ein Fehler in der Absender-Adresse vorliegt. Entsprechend weist der Return-Code YMLCBMR auf einen Fehler in mindestens einer der Empfängeradressen hin. Ein YMLCBMD kann z.B. auftreten, wenn der SMTP-Server die Verletzung bestimmter Regeln erst nach dem DATA-Kommando prüft, die Ursache der Regel-Verletzung kann mit den angegebenen Adressen zusammenhängen. In jedem Fall sollte das Feld YMLCRETM weitere (Text-) Information zur Fehlerursache geben.

YMLCRETM

Wenn der verwendete SMTP-Server den RFC 2034 unterstützt, dann enthält dieses Feld eine maschineninterpretierbare Fehleranzeige, die i.A. spezifischer ist als die SMTP-bezogenen Return-Codes des YMLCRETC-Feldes.

YMLCMID

Dieses Feld enthält einen YML-Meldungsschlüssel für eine Meldung, die den aufgetretenen Fehler beschreibt. Das ist normalerweise die gleiche Meldung, die ein vergleichbarer REQUEST-MAIL-ORDER-RESULT-Kommandoaufruf liefern würde.

Auflistung der Expansion der Datenstruktur für die Makroaufruf-Parameter

Expansion:

XPAND= PARAM

```
                        YMLCML MF=D
                   1          MFTST MF=D,PREFIX=Y,MACID=MLC,ALIGN=F,
                   1                DMACID=MLC,SUPPORT=(E,D,C,M,L),DNAME=MLCPARL
000000             2 YMLCPARL DSECT ,
                   1 *   rc
        00000000   1 YMLCBOK  EQU   0                       OK
        00000001   1 YMLCBPER EQU   1                       Parameter error
        00000002   1 YMLCBRSC EQU   2                       Resource saturation
        00000003   1 YMLCBSMT EQU   3                       SMTP error
        00000004   1 YMLCBSMI EQU   4                       SMIME error
        0000000A   1 YMLCBINT EQU   10                      Internal error
                   1 *
                   1 *   Parameter area
                   1 YMLCHDR  FHDR  MF=(C,YMLC),EQUATES=NO                              FHDR
                   1 *   main return codes
        00000000   1 YMLCSUCC EQU   0                       No error detected
        00000001   1 YMLCPARE EQU   1                       Parameter error
        00000002   1 YMLCINTE EQU   2                       Internal error
        00000003   1 YMLCONTF EQU   3                       Order not found
        00000004   1 YMLCFTSK EQU   4                       Order issued by foreign task
        00000005   1 YMLCANAV EQU   5                       Subsystem ASTI not available
        00000006   1 YMLCNORR EQU   6                       No result requested
        00000007   1 YMLCONCM EQU   7                       Order not completed
        00000008   1 YMLCASTI EQU   8                       Unexpected ASTI error
                   1 *
                   1 *
000008             1 YMLCIND  DS    0XL20                  Input parameters
000008             1 YMLCOIDI DS    CL16                   Order Id
000018             1 YMLCORDS DS    FL1                    Order specification
                   1 *   order
        00000001   1 YMLCOANY EQU   1                       ANY
        00000002   1 YMLCOSNG EQU   2                       SINGLE
                   1 *
000019             1 YMLCWAIT DS    FL1                    Wait
                   1 *   wait
        00000001   1 YMLCWYES EQU   1                       YES
        00000002   1 YMLCWNO  EQU   2                       NO
                   1 *
00001A             1 YMLCRSV1 DS    CL2                    Reserved
                   1 *
                   1 *
00001C             1 YMLCOUTD DS    0XL168                 Output parameters
00001C             1 YMLCRETC DS    F                      Return code
000020             1 YMLCRETM DS    CL160                  Return message
0000C0             1 YMLCARET DS    F                      Return code from ASTI
                   1 *
        000000C4   1 YMLC#    EQU   *-YMLCHDR
```

YMLDML - Mail löschen

Mit diesem Makro können Sie Mails löschen, die noch nicht vom Mail-Sender-Backend gesendet wurden.

Entry-Namen oder SVC-Nummer(n)

SVC 20 (dezimal)

UNIT=940, FUNCTION=22, VERSION=1

Makro-Auffrufformat und Operandenbeschreibung

YMLDML
FL= **<u>*TU</u>** / ***TPR**
,**VERSION**=**<u>1</u>** / **2**
,**ORDER**= **<u>*ALL</u>** / ***SINGLE**
,**ORDERID**= <var: char:16>
,**OWNER**= **<u>*OWN</u>** / ***ALL** / ***OTHER**
,**USERID**= <var: char:8>
,**OPTFILE=<u>*NONE</u>** / <var: char:54>

FL=
Funktionsbereich

<u>*TU</u>
SVC-Schnittstelle wird generiert.

***TPR**
CALL-Schnittstelle wird generiert.

VERSION=
Wählt die Schnittstellenversion aus.

<u>1</u>
Es wird die alte Schnittstellenversion ausgewählt.

2
Es wird die neue Schnittstellenversion ausgewählt, die den Operanden OPTFILE und einen zusätzlichen Returncode anbietet.

ORDER=
Legt fest welcher Mail-Auftrag gelöscht werden soll.

***ALL**
Alle noch nicht gesendeten Mails des Benutzers werden gelöscht.

***SINGLE**
Die im Parameter ORDERID spezifizierte Mail wird gelöscht.

ORDERID=
Wenn ORDER=*SINGLE angegeben ist, bestimmt dieser Parameter die ID des zu löschenden Mail-Auftrags.

IDENTIFIER
Variable, in der die Auftrags-ID gespeichert ist, oder Register, das die Adresse einer Variablen enthält, in der die Auftrags-ID gespeichert ist.

OWNER=
Bestimmt den Benutzer, dessen Mail-Aufträge gelöscht werden.

***OWN**
Nur Mail-Aufträge des Aufrufers löschen.

***ALL**
Mail-Aufträge aller Benutzer löschen. Aufrufer ohne TSOS-Berechtigung erhalten hier das gleiche Ergebnis wie bei *OWN.

***OTHER**
Mail-Aufträge des Benutzers löschen, der im Parameter USERID spezifiziert ist (nur für Aufrufer mit TSOS-Berechtigung erlaubt).

USERID
Bestimmt die Benutzerkennung, deren Mail-Aufträge gelöscht werden.

IDENTIFIER
Variable, in der die Benutzerkennung gespeichert ist, oder Register, das die Adresse einer Variablen enthält, in der die Benutzerkennung gespeichert ist.

OPTFILE=
Mit diesem Operanden kann eine Benutzer-Option-Datei angegeben werden. Der Operand ist nur mit VERSION=2 verfügbar. Im Augenblick gibt es keine Benutzer-Option, die das Verhalten von YMLDML beeinflussen würde. Da sich dies aber in Zukunft ändern kann, empfiehlt es sich, schon jetzt beim YMLDML-Aufruf den OPTFILE-Operanden mit dem gleichen Wert wie bei den YMLSML- und YMLCML-Aufrufen zu versorgen.

***NONE**
Keine Datei definiert.

IDENTIFIER
Variable, in der der Name der Option-Datei gespeichert ist, oder Register, das die Adresse des Namens der Option-Datei enthält.

Returncode

SRC2	SRC1	MRC	MRC-Name	Bedeutung
00	00	0000	YMLDSUCC	Es wurde kein Fehler festgestellt.
00	01	0001	YMLDPARE	Parameterfehler.
00	20	0002	YMLDINTE	Interner Fehler.
00	40	0003	YMLDONTF	Auftrag nicht gefunden.
00	40	0004	YMLDIPRV	Berechtigung nicht ausreichend.
00	40	0005	YMLDWROW	Auftrag gehört nicht dem spezifizierten Eigentümer.
00	40	0006	YMLDFTSK	Auftrag durch fremde Task abgeschickt.
00	40	0007	YMLDANAV	Subsystem ASTI nicht verfügbar.
00	20	0008	YMLDASTI	Unerwarteter ASTI-Fehler.
00	40	0009	YMLDSNAV	Service MAILCLNT nicht verfügbar.

SRC1/2=Sub Returncode 1/2 in Sedezimal-Darstellung;
MRC=Main Returncode in Sedezimal-Darstellung

Makroaufruf-Parameter

Die Datenstruktur für die Makroaufruf-Parameter von YMLDML ist wie folgt aufgebaut:

VERSION=1

Distanz	Identifier	Wert	Bedeutung
	YMLDPARL		Parameter Area
000	YMLDHDR		Function Header
008	YMLDIND		Eingabeparameter
008	YMLDOIDI		ID des zu löschenden Auftrags, falls YMLDORDS=YMLDOSNG
018	YMLDORDS		Spezifiziert, welche Aufträge gelöscht werden sollen.
	YMLDOALL	1	Alle noch nicht gesendeten Aufträge werden gelöscht.
	YMLDOSNG	2	Der unter YMLDOIDI spezifizierte Auftrag wird gelöscht.
019	YMLDOWNS		Spezifiziert den Benutzer, dessen Aufträge gelöscht werden sollen.
	YMLDWOWN	1	Es werden nur Mail-Aufträge des Aufrufers gelöscht.
	YMLDWALL	2	Alle noch nicht gesendeten Aufträge werden gelöscht.
	YMLDWOTH	3	Aufträge des unter YMLUSID spezifizierten Benutzers werden gelöscht.
01A	YMLDRSV1		Reservierter Bereich
01C	YMLDUSID		Benutzerkennung, deren Aufträge gelöscht werden sollen.
024	YMLDOUTD		Ausgabeparameter
024	YMLDARET		Wenn der YMLDML-Aufruf mit dem Return-Code YMLDASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.

VERSION=2:

Distanz	Identifier	Wert	Bedeutung
	YMLDPARL		Parameter Area
000	YMLDHDR		Function Header
008	YMLDIND		Eingabeparameter
008	YMLDOIDI		ID des zu löschenden Auftrags, falls YMLDORDS=YMLDOSNG
018	YMLDORDS		Spezifiziert, welche Aufträge gelöscht werden sollen.
	YMLDOALL	1	Alle noch nicht gesendeten Aufträge werden gelöscht.
	YMLDOSNG	2	Der unter YMLDOIDI spezifizierte Auftrag wird gelöscht.
019	YMLDOWNS		Spezifiziert den Benutzer, dessen Aufträge gelöscht werden sollen.

Distanz	Identifier	Wert	Bedeutung
	YMLDWOWN	1	Es werden nur Mail-Aufträge des Aufrufers gelöscht.
	YMLDWALL	2	Alle noch nicht gesendeten Aufträge werden gelöscht.
	YMLDWOTH	3	Aufträge des unter YMLUSID spezifizierten Benutzers werden gelöscht.
01A	YMLDUSID		Benutzerkennung, deren Aufträge gelöscht werden sollen.
022	YMLDOPTF		Benutzer-Option-Datei
058	YMLDOUTD		Ausgabeparameter
058	YMLDARET		Wenn der YMLDML-Aufruf mit dem Return-Code YMLDASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.
05C	YMLDMID		Meldungs-ID
063	YMLDRSV1		Reservierter Bereich

Ergänzende Erläuterungen zu den Ausgabeparametern:

YMLDMID

Dieses Feld enthält einen YML-Meldungsschlüssel für eine Meldung, die den aufgetretenen Fehler beschreibt. Das ist normalerweise die gleiche Meldung, die ein vergleichbarer DELETE-MAIL-ORDER-Kommandoaufruf liefern würde.

Auflistung der Expansion der Datenstruktur für die Makroaufruf-Parameter

Expansion:

XPAND= PARAM

```
                                        YMLSML MF=D,XPAND=PARA

                          YMLDML MF=D
                    1     MFTST MF=D,PREFIX=Y,MACID=MLD,ALIGN=F,
                    1           DMACID=MLD,SUPPORT=(E,D,C,M,L),DNAME=MLDPARL
000000              2 YMLDPARL DSECT ,
                    1 *   Parameter area
                    1 YMLDHDR  FHDR  MF=(C,YMLD),EQUATES=NO    FHDR
                    1 *   main return codes
        00000000    1 YMLDSUCC EQU   0                         No error detected
        00000001    1 YMLDPARE EQU   1                         Parameter error
        00000002    1 YMLDINTE EQU   2                         Internal error
        00000003    1 YMLDONTF EQU   3                         Order not found
        00000004    1 YMLDIPRV EQU   4                         Insufficient privileges
        00000005    1 YMLDWROW EQU   5                         Order not owned by specified
                    1 *                                        owner
        00000006    1 YMLDFTSK EQU   6                         Order issued by foreign task
        00000007    1 YMLDANAV EQU   7                         Subsystem ASTI not available
        00000008    1 YMLDASTI EQU   8                         Unexpected ASTI error
                    1 *
                    1 *
000008              1 YMLDIND  DS    0XL28                     Input parameters
000008              1 YMLDOIDI DS    CL16                      Order Id
000018              1 YMLDORDS DS    FL1                       Order specification
                    1 *   order
        00000001    1 YMLDOALL EQU   1                         ALL
        00000002    1 YMLDOSNG EQU   2                         SINGLE
                    1 *
000019              1 YMLDOWNS DS    FL1                       Owner specification
                    1 *   owner
        00000001    1 YMLDWOWN EQU   1                         OWN
        00000002    1 YMLDWALL EQU   2                         ALL
        00000003    1 YMLDWOTH EQU   3                         OTHER
                    1 *
00001A              1 YMLDRSV1 DS    CL2                       Reserved
00001C              1 YMLDUSID DS    CL8                       User id of owner
                    1 *
                    1 *
000024              1 YMLDOUTD DS    0XL4                      Output parameters
000024              1 YMLDARET DS    F                         Return code from ASTI
                    1 *
        00000028    1 YMLD#    EQU   *-YMLDHDR
```

YMLGML - Information zur Mail abfragen

Mit diesem Makro können Sie Informationen zu den Mail-Aufträgen abfragen.

Entry-Namen oder SVC-Nummer(n)

SVC 20 (dezimal)

UNIT=940, FUNCTION=23, VERSION=1

Makroaufrufformat und Operandenbeschreibung

YMLGML
FL= ***TU** / ***TPR** ,**VERSION**=**1** / **2** ,**XPAND**=**PARAM** / **OUTPAR** ,**ORDER**=***SUM** / ***ALL** ,**ORDERID**= <var: char:16> ,**OWNER**= ***OWN** / ***ALL** / ***OTHER** ,**USERID**= <var: char:8> ,**OUTPAR**= <var: pointer> ,**OUTPARL**=<integer: 1..32767> / <var: int:4> ,**OPTFILE=*NONE** / <var: char:54>

FL=
Funktionsbereich

***TU**
SVC-Schnittstelle wird generiert.

***TPR**
CALL-Schnittstelle wird generiert.

VERSION=
Wählt die Schnittstellenversion aus.

1
Es wird die alte Schnittstellenversion ausgewählt.

2
Es wird die neue Schnittstellenversion ausgewählt, die den Operanden OPTFILE und einen zusätzlichen Returncode anbietet.

XPAND=
Dieser Parameter steuert die Expansion der Datenstrukturen, die die Parameterliste des Makros und das Layout des Ausgabebereichs beschreiben.

PARAM
Parameterliste erstellen.

OUTPAR
Daten-Layout für den Ausgabebereich erstellen.

ORDER=
Legt fest, welche Informationen über die Mails in der Warteschlange ausgegeben werden.

***SUM**
Fragt die Anzahl der Mails in der Benutzer-Warteschlange ab.

***ALL**
Fragt die Auftrags-ID aller Mails in der Benutzer-Warteschlange ab.

***SINGLE**
Fragt die Parameter der Mail ab, die im Parameter ORDERID festgelegt wurde.

Es empfiehlt sich, den Makro zunächst mit ORDER=*ALL aufzurufen, um in einem ersten Schritt die IDs aller eigenen Mails (Mails des aufrufenden Benutzers) zu erhalten. Anschließend rufen Sie den Makro gesondert für jede Auftrags-ID mit ORDER=*SINGLE auf, um die Daten der Mail-Parameter abzurufen.

Der zweite Aufruf kann mit YMLGONTF fehlschlagen, wenn zwischen den beiden Aufrufen z.B. eine andere Task einen Aufruf YMCML/YMCMLC oder YMLDML/YMLDMCL abgesetzt hat.

Ist der Ausgabebereich zu klein für die Parameter-Daten der Mail, beendet sich der Aufruf mit YMLGOSML. Wiederholen Sie in diesem Fall den Aufruf mit einem größeren Ausgabebereich.

ORDERID=
Wenn ORDER=*SINGLE angegeben ist, legt dieser Parameter die ID des Mail-Auftrags fest, zu dem Informationen angefordert werden.

IDENTIFIER
Variable, in der die Auftrags-ID gespeichert ist, oder Register, das die Adresse einer Variablen enthält, in der die Auftrags-ID gespeichert ist.

OWNER=
Legt den Benutzer fest, über dessen Mail-Aufträge Informationen abgerufen werden sollen.

***OWN**
Nur Mail-Aufträge des aufrufenden Benutzers abfragen.

***ALL**
Mail-Aufträge aller Benutzer abfragen. Aufrufer ohne TSOS-Berechtigung erhalten hier das gleiche Ergebnis wie bei *OWN.

***OTHER**
Mail-Aufträge des Benutzers abfragen, der durch den Parameter USERID spezifiziert ist (nur für Aufrufer mit TSOS-Berechtigung erlaubt).

USERID
Wenn OWNER=*OTHER angegeben ist, legt dieser Parameter die Benutzerkennung des Eigentümers der Mail-Aufträge fest.

IDENTIFIER
Variable, in der die Benutzerkennung gespeichert ist, oder Register, das die Adresse einer Variablen enthält, in der die Benutzerkennung gespeichert ist.

OUTPAR=
Legt den Ausgabebereich fest. Dieser Parameter ist erforderlich, wenn ORDER=*ALL und ORDER=*SINGLE angegeben ist.
ORDER=*ALL:
Der Ausgabebereich enthält eine Liste der IDs der Mail-Sende-Aufträge.
ORDER=*SINGLE:
Der Ausgabebereich enthält den Mail-Parameter-Bereich der im Parameter ORDERID festgelegten Auftrags-ID.

IDENTIFIER
Variable, in der die Adresse des Ausgabebereichs gespeichert ist, oder Register, das die Adresse einer Variablen enthält, in der die Adresse des Ausgabebereichs gespeichert ist.

OUTPARL=
legt Länge des Ausgabebereichs fest.

INTEGER (1,32767)
Länge des Ausgabebereichs.

IDENTIFIER
Variable, in der die Länge des Ausgabebereichs gespeichert ist, oder Register, das die Länge des Ausgabebereichs enthält.

OPTFILE=
Mit diesem Operanden kann eine Benutzer-Option-Datei angegeben werden. Der Operand ist nur mit VERSION=2 verfügbar. Im Augenblick gibt es keine Benutzer-Option, die das Verhalten von YMLGML beeinflussen würde. Da sich dies aber in Zukunft ändern kann, empfiehlt es sich, schon jetzt beim YMLGML-Aufruf den OPTFILE-Operanden mit dem gleichen Wert wie bei den YMLSML- und YMLCML-Aufrufen zu versorgen.

***NONE**
Keine Datei definiert.

IDENTIFIER
Variable, in der der Name der Option-Datei gespeichert ist, oder Register, das die Adresse des Namens der Option-Datei enthält.

Returncode

SRC2	SRC1	MRC	MRC-Name	Bedeutung
00	00	0000	YMLGSUCC	Es wurde kein Fehler festgestellt.
00	01	0001	YMLGPARE	Parameterfehler.
00	20	0002	YMLGINTE	Interner Fehler.
00	40	0003	YMLGONTF	Auftrag nicht gefunden.
00	40	0004	YMLGOSML	Ausgabebereich zu klein.
00	40	0005	YMLGONTO	Auftrag gehört nicht dem Aufrufer.
00	40	0006	YMLGRSRC	Ressourcen sind erschöpft.
00	40	0007	YMLGFTSK	Auftrag durch fremde Task ausgeführt.
00	40	0008	YMLGANAV	Subsystem ASTI nicht verfügbar.
00	40	0009	YMLGANAV	Kein Auftrag zum Versenden von Mails.
00	20	000A	YMLGASTI	Unerwarteter ASTI-Fehler.
00	40	000B	YMLGSNAV	Service MAILCLNT nicht verfügbar.

SRC1/2=Sub Returncode 1/2 in Sedezimal-Darstellung; MRC=Main Returncode in Sedezimal-Darstellung

Makroaufruf-Parameter und Ausgabebereich

Der Makroaufruf YMLGML verwendet folgende Datenstrukturen:

– Datenstruktur für Eingabe- und Ausgabeparameter des Makroaufrufs YMLGML (XPAND=PARAM)
– Ausgabe-Datenstruktur für die von YMLGML gelieferten Informationen (XPAND=OUTPAR)

Die Datenstruktur für die Makroaufruf-Parameter von YMLGML ist wie folgt aufgebaut.

VERSION=1:

Distanz	Identifier	Wert	Bedeutung
	YMLGPARL		Parameterbereich
000	YMLGHDR		Function-Header
008	YMLGIND		Eingabeparameter
008	YMLGOIDI		ID des Auftrags, zu dem Informationen abgefragt werden sollen, falls YMLGORDS=YMLGOSNG
018	YMLGORDS		Spezifiziert, welche Informationen abgefragt werden sollen.
	YMLGOSUM	1	Fragt Anzahl der Mails in der Benutzer-Warteschlange ab.
	YMLGOALL	2	Fragt Auftrags-ID aller Mails in der Benutzerwarteschlange ab.
	YMLGOSNG	3	Fragt Parameter des in YMLGOIDI spezifizierten Auftrags ab.
019	YMLGOWNS		Spezifiziert den Benutzer, über dessen Aufträge Informationen abgefragt werden sollen.
	YMLGWOWN	1	Nur Aufträge des aufrufenden Benutzers werden abgefragt.
	YMLGWALL	2	Aufträge aller Benutzer werden abgefragt.
	YMLGWOTH	3	Aufträge des in YMLGUSID spezifizierten Benutzers werden abgefragt.
01A	YMLGRSV1		Reservierter Bereich
01C	YMLGUSID		Benutzerkennung, deren Aufträge gelöscht werden sollen.
024	YMLGOUT		Adresse des Ausgabebereichs
028	YMLGOUTL		Größe des Ausgabebereichs
02C	YMLGOUTD		Ausgabeparameter
02C	YMLGSUM		Anzahl der Mails in der Warteschlange
030	YMLGARET		Wenn der YMLGML-Aufruf mit dem Return-Code YMLGASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.

VERSION=2:

Distanz	Identifier	Wert	Bedeutung
	YMLGPARL		Parameterbereich
000	YMLGHDR		Function-Header
008	YMLGIND		Eingabeparameter
008	YMLGOIDI		ID des Auftrags, zu dem Informationen abgefragt werden sollen, falls YMLGORDS=YMLGOSNG
018	YMLGORDS		Spezifiziert, welche Informationen abgefragt werden sollen.
	YMLGOSUM	1	Fragt Anzahl der Mails in der Benutzer-Warteschlange ab.
	YMLGOALL	2	Fragt Auftrags-ID aller Mails in der Benutzerwarteschlange ab.
	YMLGOSNG	3	Fragt Parameter des in YMLGOIDI spezifizierten Auftrags ab.
019	YMLGOWNS		Spezifiziert den Benutzer, über dessen Aufträge Informationen abgefragt werden sollen.
	YMLGWOWN	1	Nur Aufträge des aufrufenden Benutzers werden abgefragt.
	YMLGWALL	2	Aufträge aller Benutzer werden abgefragt.
	YMLGWOTH	3	Aufträge des in YMLGUSID spezifizierten Benutzers werden abgefragt.
01A	YMLGRSV1		Reservierter Bereich
01C	YMLGUSID		Benutzerkennung, deren Aufträge gelöscht werden sollen.
024	YMLGOUT		Adresse des Ausgabebereichs
028	YMLGOUTL		Größe des Ausgabebereichs
02C	YMLGOPTF		Benutzer-Option-Datei
062	YMLGRSV2		Reservierter Bereich
064	YMLGOUTD		Ausgabeparameter
064	YMLGSUM		Anzahl der Mails in der Warteschlange
068	YMLGARET		Wenn der YMLGML-Aufruf mit dem Return-Code YMLGASTI zurückgewiesen wird, enthält dieses Feld den ASTI-Return-Code.
06C	YMLGMID		Meldungs-ID
073	YMLGRSV3		Reservierter Bereich

Ergänzende Erläuterungen zu den Ausgabeparametern:

YMLGMID

Dieses Feld enthält einen YML-Meldungsschlüssel für eine Meldung, die den aufgetretenen Fehler beschreibt. Das ist normalerweise die gleiche Meldung, die ein vergleichbarer SHOW-MAIL-ORDER-STATUS-Kommandoaufruf liefern würde.

Auflistung der Expansion der Datenstrukturen für Makroaufruf-Parameter und Ausgabebereich

Expansion:

XPAND= PARAM, OUTPAR

```
                                YMLGML MF=D,XPAND=PARAM
                    1          MFTST MF=D,PREFIX=Y,MACID=MLG,ALIGN=F,
                    1                DMACID=MLG,SUPPORT=(E,D,C,M,L),DNAME=MLGPARL
000000              2 YMLGPARL DSECT ,
                    1 *   Parameter area
                    1 YMLGHDR  FHDR  MF=(C,YMLG),EQUATES=NO                          FHDR
                    1 *   main return codes
           00000000 1 YMLGSUCC  EQU   0                       No error detected
           00000001 1 YMLGPARE  EQU   1                       Parameter error
           00000002 1 YMLGINTE  EQU   2                       Internal error
           00000003 1 YMLGONTF  EQU   3                       Order not found
           00000004 1 YMLGOSML  EQU   4                       Output area too small
           00000005 1 YMLGONTO  EQU   5                       Order not own
           00000006 1 YMLGRSRC  EQU   6                       Resource saturation
           00000007 1 YMLGFTSK  EQU   7                       Order issued by foreign
                                                              task
           00000008 1 YMLGANAV  EQU   8                       Subsystem ASTI not
                                                              available
           00000009 1 YMLGNMSO  EQU   9                       Not a mail send order
           0000000A 1 YMLGASTI  EQU   10                      Unexpected ASTI error
                    1 *
                    1 *
000008              1 YMLGIND   DS    OXL36                   Input parameters
000008              1 YMLGOIDI  DS    CL16                    Order Id
000018              1 YMLGORDS  DS    FL1                     Order specification
                    1 *   order
           00000001 1 YMLGOSUM  EQU   1                       sum
           00000002 1 YMLGOALL  EQU   2                       all
           00000003 1 YMLGOSNG  EQU   3                       single
                    1 *
000019              1 YMLGOWNS  DS    FL1                     Owner specification
                    1 *   owner
           00000001 1 YMLGWOWN  EQU   1                       OWN
           00000002 1 YMLGWALL  EQU   2                       ALL
           00000003 1 YMLGWOTH  EQU   3                       OTHER
                    1 *
00001A              1 YMLGRSV1  DS    CL2                     Reserved
00001C              1 YMLGUSID  DS    CL8                     User id of owner
000024              1 YMLGOUT   DS    A                       Output area
000028              1 YMLGOUTL  DS    F                       Output area length
                    1 *
                    1 *
00002C              1 YMLGOUTD  DS    OXL8                    Output parameters
00002C              1 YMLGSUM   DS    F                       Number of queued mails
```

```
000030             1 YMLGARET  DS    F                        Return code from ASTI
                   1 *
        00000034   1 YMLG#     EQU   *-YMLGHDR

                             YMLGML MF=D,XPAND=OUTPAR
                   1         MFTST MF=D,PREFIX=Y,MACID=MLG,ALIGN=F,
                   1               DMACID=MLG,SUPPORT=(E,D,C,M,L),DNAME=MLGOUTPAR
000000             2 YMLGOUTPAR DSECT ,
                   1 *   rc
        00000000   1 YMLGBOK   EQU   0                        OK
        00000001   1 YMLGBPER  EQU   1                        Parameter error
        00000002   1 YMLGBRSC  EQU   2                        Resource saturation
        00000003   1 YMLGBSMT  EQU   3                        SMTP error
        00000004   1 YMLGBSMI  EQU   4                        SMIME error
        0000000A   1 YMLGBINT  EQU   10                       Internal error
                   1 *
                   1 *   STRUCT Output parameters
000000             1 YMLGOUTP  DS    0XL192                   UNION Output parameters
                   1 *
000000             1 YMLGMDAT  DS    0XL192                   mail data
000000             1 YMLGSTAT  DS    F                        mail status
                   1 *   order status
        00000001   1 YMLGWAIT  EQU   1                        waiting
        00000002   1 YMLGDEFE  EQU   2                        deferred
        00000003   1 YMLGACTV  EQU   3                        active
        00000004   1 YMLGSENT  EQU   4                        send successful
        00000005   1 YMLGFAIL  EQU   5                        send failed
                   1 *
000004             1 YMLGTIME  DS    F                        submission time
000008             1 YMLGUSER  DS    CL8                      submitter
000010             1 YMLGSLCT  DS    F                        data selector
                   1 *   Data type selector
        00000001   1 YMLGSORD  EQU   1                        order data
        00000002   1 YMLGSDAT  EQU   2                        result data
                   1 *
000014             1 YMLGDATA  DS    0XL164                   UNION data
                   1 *
000014             1 YMLGRDAT  DS    0XL164                   result data
000014             1 YMLGRETC  DS    F                        return code
000018             1 YMLGRETM  DS    CL160                    return message
                   1 *
0000B8  00000014   1          ORG   YMLGDATA
                   1 *
000014             1 YMLGODAT  DS    0XL4                     order data
000014             1 YMLGCNTT  DS    F                        # sending tries
                   1 *
000018  000000B8   1          ORG   YMLGDATA+164
0000B8             1 YMLGMPLN  DS    F                        mail parameter length
0000BC             1 YMLGMP    DS    CL4                      mail parameters, real
                                                              size:
                   1 *                                       mail_par_len
                   1 *
```

```
0000C0  00000000   1          ORG   YMLGOUTP
                   1 *
000000             1 YMLGOIDS  DS    0XL20                  order Ids
000000             1 YMLGNORD  DS    F                      number of orders
000004             1 YMLGOID   DS    1CL16                  Array of order ids, real
                   1 *                                      array size: num_order
        00000001   1 YMLGOID#  EQU   1
                   1 *
000014  000000C0   1          ORG   YMLGOUTP+192
        000000C0   1 YMLGOUTPAR# EQU  *-YMLGSTAT
```

9.3.2 Funktionsaufrufe in C

Die C-Unterprogramm-Schnittstelle des Mail-Sender Frontend unterstützt die folgenden Funktionsaufrufe:

Funktionsaufruf	**Funktion**	**Zugehörige Include-Datei mit Funktions-deklaration und Datenstrukturen**
YMLSML()	Mail senden	YMLSML.H
YMLCML()	Mail-Resultat abholen	YMLCML.H
YMLDML()	Mail löschen	YMLDML.H
YMLGML()	Information zur Mail abfragen	YMLGML.H

C-Funktionsaufrufe und verwendete Include-Dateien

Die bei der Assembler-Makro-Schnittstelle (ab Seite 440) beschriebene Erweiterung der Unterprogrammschnittstelle um den Parameter VERSION sowie weitere Parameter und Return-Codes wirkt sich ebenfalls auf die C-Include-Dateien aus. Die Auswahl zwischen der alten und der neuen Schnittstellen-Version findet durch die Definition eines Präprozessor-Symbols vor dem Inkludieren der jeweiligen Include-Datei statt.

Wenn Sie die VERSION=1-Variante der Include-Dateien verwenden, geben Sie Folgendes an:

```
#define _YMLCML_H_VERSION_1
#include "YMLCML.H"

#define _YMLDML_H_VERSION_1
#include "YMLDML.H"

#define _YMLGML_H_VERSION_1
#include "YMLGML.H"

#define _YMLSML_H_VERSION_1
#include "YMLSML.H"
```

Wenn Sie die VERSION=2-Variante der Include-Dateien verwenden, geben Sie Folgendes an:

```
#define _YMLCML_H_VERSION_2
#include "YMLCML.H"

#define _YMLDML_H_VERSION_2
#include "YMLDML.H"

#define _YMLGML_H_VERSION_2
#include "YMLGML.H"

#define _YMLSML_H_VERSION_2
#include "YMLSML.H"
```

Für die Versorgung der Function-Header mit Unit-, Function- und Version-Werten werden in den Include-Dateien entsprechende Präprozessorsymbole definiert, so z.B. für YMLSML die Symbole YMLSML_UNIT, YMLSML_FUNCTION und YMLSML_VERSION. Letzteres ist abhängig von der ausgewählten Schnittstellen-Version.

Die im Folgenden aufgelisteten C-Include-Dateien deklarieren die Funktionen sowie die Datenstrukturen, die von ihnen verwendet werden.

9.3.2.1 Include-Datei YMLSML.H zu YMLSML() - Mail senden

Nachfolgend ist die Include-Datei YMLSML.H aufgelistet. Ein Beispielprogramm, das die Verwendung des Funktionsaufrufs YMLSML() zeigt, finden Sie auf Seite 496.

```
#ifndef _YMLSML_H
#define _YMLSML_H

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

 TITLE             (/ Submit send mail order /)
 NAME              YMLSML.H
 DOMAIN            MAIL
 LANGUAGE          C
 COPYRIGHT         (C) Fujitsu Technology Solutions GmbH 2010
                       ALL RIGHTS RESERVED
 COMPILATION-SCOPE USER
 INTERFACE-TYPE    CALL,
                      ORDER
 RUN-CONTEXT       TU
 PURPOSE           (/ Submit an order to send mail
                   /)

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

/* Wait operand values                                          */
/* ENUM wait_s                                                  */
#define YMLSML_yes 1                    /* YES                  */
#define YMLSML_no 2                     /* NO-DISCARD           */
#define YMLSML_res 3                    /* NO                   */

/* Mail encryption/signing protocol                             */
/* ENUM sec_prot_s                                              */
#define YMLSML_smime 1                  /* SMIME                */

/* General yes or no selection                                  */
/* ENUM yesno_s                                                 */
```

```
#define YMLSML_yes 1                    /* YES                          */
#define YMLSML_no 2                     /* NO                           */
#define YMLSML_optfile 3                /* OPTFILE                      */

/* Cipher operand values                                               */
/* ENUM cipher_s                                                       */
#define YMLSML_rc2_40 1                 /* RC2-40                       */
#define YMLSML_rc2_64 2                 /* RC2-64                       */
#define YMLSML_rc2_128 3                /* RC2-128                      */
#define YMLSML_des 4                    /* DES                          */
#define YMLSML_des3 5                   /* 3DES                         */
#define YMLSML_aes_128 6                /* AES-128                      */
#define YMLSML_aes_192 7                /* AES-192                      */
#define YMLSML_aes_256 8                /* AES-256                      */
#define YMLSML_cipher_from_optfile 127  /* OPTFILE                      */

/* main return codes                                                   */
/* mret_code                                                           */
#define YMLSML_successful 0             /* No error detected            */
#define YMLSML_parameter_error 1        /* Parameter error              */
#define YMLSML_int_error 2              /* Internal error               */
#define YMLSML_syntax_error 3           /* Syntax error                 */
#define YMLSML_opt_file_error 4         /* Option file not accessible   */
#define YMLSML_msgatt_file_error 5      /* Message or attachment file   */
                                        /* not accessible               */
#define YMLSML_smime_file_error 6       /* SMIME related file not       */
                                        /* accessible                   */
#define YMLSML_param_too_big 7          /* Mail parameter too big       */
#define YMLSML_backend_error 8          /* Back-end error               */
#define YMLSML_max_order 9              /* Max number of orders         */
                                        /* exceeded                     */
#define YMLSML_serv_not_avail 10        /* Mailclient Service not       */
                                        /* available                    */
#define YMLSML_addr_invalid 11          /* Mailpar address invalid      */
#define YMLSML_resource_sat 12          /* Resource saturation          */
#define YMLSML_asti_not_avail 13        /* Subsystem ASTI not           */
                                        /* available                    */
#define YMLSML_no_from_addr 14          /* No FROM address specified    */
#define YMLSML_asti_error 15            /* Unexpected ASTI error        */

/* Parameter area                                                      */
struct YMLSML_pl_mdl {

        /* FHDR                                                        */
        struct ESMFHDR hdr;

        /* Input parameters                                            */
        struct {
                void* mail_par;         /* Mail parameter area          */
                unsigned long mail_par_len;
                                        /* Mail parameter area length   */
                  unsigned char wait;     /* wait                       */
```

```
                unsigned char sec_prot; /* Protocol for encryption    */
                                        /* and signing                */
                unsigned char encrypt;  /* Encryption                 */
                unsigned char sign;     /* Signing                    */
                unsigned char cipher;   /* Cipher                     */
                char optfile[54];       /* Option file                */
                char reserved1[1];      /* Reserved                   */
        } in_data;

        /* Output parameters                                          */
        struct {
                char order_id[16];      /* Order id                   */
                char error_message[160];
                                        /* Error message              */
                unsigned long asti_rc;  /* Return code from ASTI      */
        } out_data;
};

/*  Entry for YMLSML  */
#ifdef __SNI_HOST_BS2000
#ifdef __cplusplus
extern "C" void _SVC(int, void*);
inline void YMLSML(struct YMLSML_pl_mdl& param)
{_SVC(20, &param);}
#else
void _SVC(int, void*);
#define YMLSML(p) _SVC(20, &p)
#endif
#endif

/* Tag value(s) for struct _general                                  */
/* ENUM tag_s                                                        */
#define YMLSML_tag_from 1               /* from                      */
#define YMLSML_tag_fake_from 2          /* fake_from                 */
#define YMLSML_tag_to 3                 /* to                        */
#define YMLSML_tag_fake_to 4            /* fake_to                   */
#define YMLSML_tag_cc 5                 /* cc                        */
#define YMLSML_tag_fake_cc 6            /* fake_cc                   */
#define YMLSML_tag_bcc 7                /* envelope recipient mail   */
                                        /* addresses                 */
#define YMLSML_tag_replyto 8            /* reply_to                  */
#define YMLSML_tag_subject 9            /* subject                   */
#define YMLSML_tag_cont_type 18         /* content_type              */
#define YMLSML_tag_optfile 20           /* option file               */
#define YMLSML_tag_keyfile 21           /* key file                  */
#define YMLSML_tag_signfile 22          /* signer certificate file   */
#define YMLSML_tag_scertfile 23         /* additional signer         */
                                        /* certificates file         */
#define YMLSML_tag_rcertfile 24         /* recipient certificate file */
#define YMLSML_tag_crlfile 25           /* CRL file                  */

/* Input parameters                                                  */
```

```
struct YMLSML_general {
        unsigned short tag;             /* tag of general parameter   */
        char reserved[2];               /* reserved                   */
        unsigned long len;              /* general parameter length   */
};

/* Tag value(s) for struct _add_header                                */
/* ENUM tag_s                                                         */
#define YMLSML_tag_add_header 10        /* add. header                */

/* Input parameters                                                   */
struct YMLSML_add_header {
        unsigned short tag;             /* tag of add. header         */
        char reserved[2];               /* reserved                   */
        unsigned long len_field_name;   /* add. header name length    */
        unsigned long len_field_body;   /* add. header body length    */
};

/* Tag value(s) for struct _data_spec                                 */
/* ENUM tag_s                                                         */
#define YMLSML_tag_data_spec 15         /* dataspec                   */

/* Type of data specification                                         */
/* ENUM type_s                                                        */
#define YMLSML_file 1                   /* file                       */
#define YMLSML_data 2                   /* data                       */

/* Data specification                                                 */
struct YMLSML_data_spec {
        unsigned short tag;             /* tag of add. header         */
        unsigned char type;             /* reserved                   */
        char reserved[1];               /* reserved                   */
        unsigned long len;              /* length of data. spec.      */
};

/* Tag value(s) for struct _charset                                   */
/* ENUM tag_s                                                         */
#define YMLSML_tag_charset 16           /* character set              */

/* Binary data selection                                              */
/* ENUM bnry_s                                                        */
#define YMLSML_yes 1                    /* YES                        */
#define YMLSML_no 2                     /* NO                         */
/* Data specification                                                 */
struct YMLSML_charset {
        unsigned short tag;             /* tag of add. header         */
        unsigned char binary;           /* binary data                */
        char reserved[1];               /* reserved                   */
        unsigned long len;              /* length                     */
        char src_charset_XHCS[8];       /* src_charset_XHCS           */
        char dest_charset_XHCS[8];      /* dest_charset_XHCS          */
};
```

```
/* Tag value(s) for struct _encoding                                    */
/* ENUM tag_s                                                           */
#define YMLSML_tag_encoding 17          /* transfer encoding            */

/* Type of encoding mechanism                                           */
/* ENUM mech_s                                                          */
#define YMLSML_7bit 1                   /* 7 bit                        */
#define YMLSML_8bit 2                   /* 8 bit                        */
#define YMLSML_binary 3                 /* binary                       */
#define YMLSML_qp 4                     /* quoted printable             */
#define YMLSML_b64 5                    /* base64                       */

/* Encoding                                                             */
struct YMLSML_encoding {
        unsigned short tag;             /* tag of encoding              */
        unsigned char mechanism;        /* encoding mechanism           */
        char reserved[1];               /* reserved                     */
};

/* Tag value(s) for struct _cont_disp                                   */
/* ENUM tag_s                                                           */
#define YMLSML_tag_cont_disp 19         /* content disposition          */

/* Disposition type                                                     */
/* ENUM disptype_s                                                      */
#define YMLSML_inline 1                 /* inline                       */
#define YMLSML_attachment 2             /* attachment                   */

/* Content disposition                                                  */
struct YMLSML_cont_disp {
        unsigned short tag;             /* tag of content disposition */
        unsigned char disp_type;        /* content disposition          */
        char reserved[1];               /* reserved                     */
};

/* Tag value(s) for struct _msg_att                                     */
/* ENUM tag_s                                                           */
#define YMLSML_tag_msg_begin 11         /* msg_begin                    */
#define YMLSML_tag_msg_end 12           /* msg_end                      */
#define YMLSML_tag_att_begin 13         /* att_begin                    */
#define YMLSML_tag_att_end 14           /* att_end                      */

/* Bracketing message or attachment specification                       */
struct YMLSML_msg_att {
        unsigned short tag;             /* tag of msg | att             */
        char reserved[2];               /* reserved                     */
};

#endif          /* _YMLSML_H */
```

Hinweise zur Definition des Eingabebereichs für die Mail-Parameter

– Die Implementierung des Subsystems schränkt die Größe der Mail-Parameter ein. Ein exakter Wert lässt sich hier nicht angeben. Einen Anhaltspunkt liefert ein Wert von ungefähr 32 KB, abzüglich einiger Bytes, die für zusätzliche Parameter und interne Verwaltungsdaten reserviert sind. Vermutlich wird diese Grenze nur dann erreicht, wenn der Text der Mail oder Daten in den Anhängen direkt in den Parameterbereich der Mail eingefügt werden. Tritt dieser Fehler auf, dann schreiben Sie den Mail-Text in eine (temporäre) Datei und geben Sie im Parameterbereich statt dessen nur den Dateinamen an.

– Die Daten mit variabler Länge beginnen direkt nach dem zugehörigen Header. Die Header müssen auf Wortlänge ausgerichtet sein. Daher müssen Sie Füll-Bytes einfügen, wenn die Länge des variablen Teils kein Vielfaches von 4 ist.

– Der Feldname des Mail-Header muss in der Datenstruktur für zusätzliche Header ohne den folgenden Doppelpunkt eingegeben werden.

– Die Reihenfolge der Datenstrukturen ist irrelevant, falls nicht anders vermerkt.

– Die Definition des Mail-Texts oder eines Anhangs muss mit zwei `YMLSML_msg_att`-Datenstrukturen geklammert sein, wobei

  ```
  tag = YMLSML_tag_msg_begin / YMLSML_tag_msg_end
  ```

  oder

  ```
  tag = YMLSML_tag_att_begin / YMLSML_tag_att_end
  ```

  ist.

  Die Definition des Mail-Textes oder eines Anhangs kann Datenstrukturen mit den folgenden Tags enthalten:

  ```
  YMLSML_tag_{data_spec, charset, encoding, cont_type}
  ```

  Liegt eine Datenstruktur `YMLSML_charset` außerhalb einer Datenstruktur `YMLSML_tag_msg_att`, dann wird sie auf die Header-Zeilen (siehe RFC 2047) und auf Meldungen oder Anhänge ohne eigene Zeichensatz-Spezifikation angewendet.

– Dateien müssen unverändert bleiben, bis die Mail gesendet ist.

9.3.2.2 Include-Datei YMLCML.H zu YMLCML() - Mail-Resultat abholen

Nachfolgend ist die Include-Datei YMLCML.H aufgelistet. Ein Beispielprogramm, das die Verwendung des Funktionsaufrufs YMLCML() zeigt, finden Sie auf Seite 496.

```
#ifndef _YMLCML_H
#define _YMLCML_H

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

 TITLE             (/ Check send mail order /)
 NAME              YMLCML.H
 DOMAIN            MAIL
 LANGUAGE          C
 COPYRIGHT         (C) Fujitsu Technology Solutions GmbH 2010
                       ALL RIGHTS RESERVED
 COMPILATION-SCOPE USER
 INTERFACE-TYPE    CALL
 RUN-CONTEXT       TU
 PURPOSE           (/ Check, whether a submitted send mail order has completed
                   /)

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

/* order                                                        */
/* ENUM order_s                                                 */
#define YMLCML_any 1                  /* ANY                    */
#define YMLCML_single 2               /* SINGLE                 */

/* wait                                                         */
/* ENUM wait_s                                                  */
#define YMLCML_yes 1                  /* YES                    */
#define YMLCML_no 2                   /* NO                     */

/* rc                                                           */
/* ENUM rc_s                                                    */
#define YMLCML_be_ok 0                /* OK                     */
#define YMLCML_be_param_error 1       /* Parameter error        */
#define YMLCML_be_resource_sat 2      /* Resource saturation    */
#define YMLCML_be_smtp_error 3        /* SMTP error             */
#define YMLCML_be_smime_error 4       /* SMIME error            */
#define YMLCML_be_int_error 10        /* Internal error         */

/* main return codes                                            */
/* mret_code                                                    */
#define YMLCML_successful 0           /* No error detected      */
#define YMLCML_parameter_error 1      /* Parameter error        */
```

```
#define YMLCML_int_error 2              /* Internal error             */
#define YMLCML_order_not_found 3        /* Order not found            */
#define YMLCML_foreign_task 4           /* Order issued by foreign    */
                                        /* task                       */
#define YMLCML_asti_not_avail 5         /* Subsystem ASTI not         */
                                        /* available                  */
#define YMLCML_no_result_req 6          /* No result requested        */
#define YMLCML_order_not_compl 7        /* Order not completed        */
#define YMLCML_asti_error 8             /* Unexpected ASTI error      */

/* Parameter area                                                     */
struct YMLCML_pl_mdl {

        /* FHDR                                                       */
        struct ESMFHDR hdr;

        /* Input parameters                                           */
        struct {
                char order_id[16];      /* Order Id                   */
                unsigned char order;    /* Order specification        */
                unsigned char wait;     /* Wait                       */
                char reserved1[2];      /* Reserved                   */
        } in_data;

        /* Output parameters                                          */
        struct {
                unsigned long rc;       /* Return code                */
                char ret_msg[160];      /* Return message             */
                unsigned long asti_rc;  /* Return code from ASTI      */
        } out_data;
};

/*  Entry for YMLCML  */
#ifdef __SNI_HOST_BS2000
#ifdef __cplusplus
extern "C" void _SVC(int, void*);
inline void YMLCML(struct YMLCML_pl_mdl& param)
{_SVC(20, &param);}
#else
void _SVC(int, void*);
#define YMLCML(p) _SVC(20, &p)
#endif
#endif
#endif          /* _YMLCML_H */
```

9.3.2.3 Include-Datei YMLDML.H zu YMLDML() - Mail löschen

Nachfolgend ist die Include-Datei YMLDML.H aufgelistet. Ein Beispielprogramm, das die Verwendung des Funktionsaufrufs YMLDML() zeigt, finden Sie auf Seite 499.

```
#ifndef _YMLDML_H
#define _YMLDML_H

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

 TITLE             (/ Delete send mail order /)
 NAME              YMLDML.H
 DOMAIN            MAIL
 LANGUAGE          C
 COPYRIGHT         (C) Fujitsu Technology Solutions GmbH 2010
                       ALL RIGHTS RESERVED
 COMPILATION-SCOPE USER
 INTERFACE-TYPE    CALL
 RUN-CONTEXT       TU
 PURPOSE           (/ Delete send mail order which is yet uncompleted
                   /)

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

/* order                                                        */
/* ENUM order_s                                                 */
#define YMLDML_all 1                  /* ALL                    */
#define YMLDML_single 2               /* SINGLE                 */

/* owner                                                        */
/* ENUM owner_s                                                 */
#define YMLDML_own 1                  /* OWN                    */
#define YMLDML_all_users 2            /* ALL                    */
#define YMLDML_other 3                /* OTHER                  */

/* main return codes                                            */
/* mret_code                                                    */
#define YMLDML_successful 0           /* No error detected      */
#define YMLDML_parameter_error 1      /* Parameter error        */
#define YMLDML_int_error 2            /* Internal error         */
#define YMLDML_order_not_found 3      /* Order not found        */
#define YMLDML_insuff_priv 4          /* Insufficient privileges*/
#define YMLDML_wrong_owner 5          /* Order not owned by     */
                                      /* specified owner        */
#define YMLDML_foreign_task 6         /* Order issued by foreign*/
                                      /* task                   */
```

```
#define YMLDML_asti_not_avail 7         /* Subsystem ASTI not         */
                                        /* available                  */
#define YMLDML_asti_error 8             /* Unexpected ASTI error      */
/* Parameter area                                                     */
struct YMLDML_pl_mdl {

        /* FHDR                                                       */
        struct ESMFHDR hdr;

        /* Input parameters                                           */
        struct {
                char order_id[16];      /* Order Id                   */
                unsigned char order;    /* Order specification        */
                unsigned char owner;    /* Owner specification        */
                char reserved1[2];      /* Reserved                   */
                char user_id[8];        /* User id of owner           */
        } in_data;

        /* Output parameters                                          */
        struct {
                unsigned long asti_rc;  /* Return code from ASTI      */
        } out_data;
};

/*  Entry for YMLDML  */
#ifdef __SNI_HOST_BS2000
#ifdef __cplusplus
extern "C" void _SVC(int, void*);
inline void YMLDML(struct YMLDML_pl_mdl& param)
{_SVC(20, &param);}
#else
void _SVC(int, void*);
#define YMLDML(p) _SVC(20, &p)
#endif
#endif

#endif          /* _YMLDML_H */
```

9.3.2.4 Include-Datei YMLGML.H zu YMLGML() - Informationen zur Mail abfragen

Nachfolgend ist die Include-Datei YMLGML.H aufgelistet. Ein Beispielprogramm, das die Verwendung des Funktionsaufrufs YMLDML() zeigt, finden Sie auf Seite 500

```
#ifndef _YMLGML_H
#define _YMLGML_H

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

 TITLE             (/ Get info about send mail orders /)
 NAME              YMLGML.H
 DOMAIN            MAIL
 LANGUAGE          C
 COPYRIGHT         (C) Fujitsu Technology Solutions GmbH 2010
                       ALL RIGHTS RESERVED
 COMPILATION-SCOPE USER
 INTERFACE-TYPE    CALL
 RUN-CONTEXT       TU
 PURPOSE           (/ Get info about submitted send mail order(s)
                   /)

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

/* order                                                        */
/* ENUM order_s                                                 */
#define YMLGML_sum 1                  /* sum                    */
#define YMLGML_all 2                  /* all                    */
#define YMLGML_single 3               /* single                 */

/* owner                                                        */
/* ENUM owner_s                                                 */
#define YMLGML_own 1                  /* OWN                    */
#define YMLGML_all_users 2            /* ALL                    */
#define YMLGML_other 3                /* OTHER                  */

/* main return codes                                            */
/* mret_code                                                    */
#define YMLGML_successful 0           /* No error detected      */
#define YMLGML_parameter_error 1      /* Parameter error        */
#define YMLGML_int_error 2            /* Internal error         */
#define YMLGML_order_not_found 3      /* Order not found        */
#define YMLGML_out_too_small 4        /* Output area too small  */
#define YMLGML_order_not_own 5        /* Order not own          */
#define YMLGML_resource_sat 6         /* Resource saturation    */
#define YMLGML_foreign_task 7         /* Order issued by foreign*/
                                      /* task                   */
```

```
#define YMLGML_asti_not_avail 8         /* Subsystem ASTI not         */
                                        /* available                  */
#define YMLGML_not_mail_send_order 9    /* Not a mail send order      */
#define YMLGML_asti_error 10            /* Unexpected ASTI error      */

/* Parameter area                                                     */
struct YMLGML_pl_mdl {

        /* FHDR                                                       */
        struct ESMFHDR hdr;

        /* Input parameters                                           */
        struct {
                char order_id[16];      /* Order Id                   */
                unsigned char order;    /* Order specification        */
                unsigned char owner;    /* Owner specification        */
                char reserved1[2];      /* Reserved                   */
                char user_id[8];        /* User id of owner           */
                void* out;              /* Output area                */
                unsigned long outl;     /* Output area length         */
        } in_data;

        /* Output parameters                                          */
        struct {
                unsigned long sum;      /* Number of queued mails     */
                unsigned long asti_rc;  /* Return code from ASTI      */
        } out_data;
};

/*  Entry for YMLGML  */
#ifdef __SNI_HOST_BS2000
#ifdef __cplusplus
extern "C" void _SVC(int, void*);
inline void YMLGML(struct YMLGML_pl_mdl& param)
{_SVC(20, &param);}
#else
void _SVC(int, void*);
#define YMLGML(p) _SVC(20, &p)
#endif
#endif

/* order status                                                       */
/* ENUM status_s                                                      */
#define YMLGML_waiting 1                /* waiting                    */
#define YMLGML_deferred 2               /* deferred                   */
#define YMLGML_active 3                 /* active                     */
#define YMLGML_sent_ok 4                /* send successful            */
#define YMLGML_sent_fail 5              /* send failed                */

/* Data type selector                                                 */
/* ENUM slctr_s                                                       */
#define YMLGML_sel_order 1              /* order data                 */
```

```
#define YMLGML_sel_result 2             /* result data                */

/* rc                                                                 */
/* ENUM rc_s                                                          */
#define YMLGML_be_ok 0                  /* OK                         */
#define YMLGML_be_param_error 1         /* Parameter error            */
#define YMLGML_be_resource_sat 2        /* Resource saturation        */
#define YMLGML_be_smtp_error 3          /* SMTP error                 */
#define YMLGML_be_smime_error 4         /* SMIME error                */
#define YMLGML_be_int_error 10          /* Internal error             */

/* Output parameters                                                  */
struct YMLGML_order {
        unsigned long num_order;        /* number of orders           */
        char order_id[1][16];           /* Array of order ids, real   */
                                        /* array size: num_order      */
};

/* order specific data                                                */
struct YMLGML_order_data {
        unsigned long cnt_tries;        /* # sending tries            */
};

/* result specific data                                               */
struct YMLGML_result_data {
        unsigned long rc;               /* return code                */
        char ret_msg[160];              /* return message             */
};

/* mail parameter                                                     */
struct YMLGML_mail_data {
        unsigned long status;           /* mail status                */
        unsigned long time;             /* submission time            */
        char user[8];                   /* submitter                  */
        unsigned long slctr;            /* data selector              */

        /* UNION data                                                 */
        union /* _data */ {
                struct YMLGML_result_data result_data;
                                        /* result data                */
                struct YMLGML_order_data order_data;
                                        /* order data                 */
        } _data;
        unsigned long mail_par_len;     /* mail parameter length      */
        char mail_par[4];               /* mail parameters, real      */
                                        /* size: mail_par_len         */
};
```

```
/* STRUCT Output parameters                                          */
struct YMLGMLoutpar {

        /* UNION Output parameters                                   */
        union /* _outp */ {
                  struct YMLGML_mail_data mail_data;
                                          /* mail data              */
                struct YMLGML_order order_ids;
                                          /* order Ids              */
        } _outp;
};

#endif          /* _YMLGML_H */
```

9.3.2.5 C-Beispielprogramme

Die nachfolgendend abgedruckten C-Beispielprogramme zeigen die Verwendung der C-Funktionsaufrufe YMLSML(), YMLCML(), YMLDML(), YMLDML().

Beispiel 1: Funktionsaufrufe YMLSML() und YMLCML()

```
#include <stdio.h>
#include <stdlib.h>
#include "FHDR.H"
#include "YMLSML.H"
#include "YMLCML.H"

#define ALIGNMENT 4
#define PAD(length) ((length + ALIGNMENT - 1) & ~(ALIGNMENT - 1))

main(int argc, char *argv[])
{
  struct YMLSML_pl_mdl sendParam;
  struct YMLCML_pl_mdl checkParam;
  enum {UNIT = 940, SEND_FUNCTION = 20, CHECK_FUNCTION = 21,
        VERSION = 1, parLen = 2048, orderIdLen = 16};
  char sender[] = "Claudio.Monteverdi@mantova.example";
  char recipient[] = "Heinrich.Schuetz@dresden.example";
  char subject[] = "Dies ist eine Testnachricht";
  char msgText[] = "Dies ist eine Testzeile.\n"
                   "Und noch eine weitere Zeile.\n";
  char srcCCSN[8 + 1] = "EDFO4DRV";
  char destCCSN[8 + 1] = "ISO88591";
  char charSetName[] = "ISO-8859-1";
  char *ptr;
  struct YMLSML_general *ptrGeneral;
  struct YMLSML_msg_att *ptrMsgAtt;
  struct YMLSML_data_spec *ptrDataSpec;
  struct YMLSML_charset *ptrCharset;
  struct YMLSML_encoding *ptrEncoding;
  char *mailPar;
  char orderId[orderIdLen];

  mailPar = ptr = malloc(parLen);
  /* [Do some error handling] */

  /* Supplying sender address */
  ptrGeneral = (struct YMLSML_general *) ptr;
  ptrGeneral->tag = YMLSML_tag_from;
  ptrGeneral->len = strlen(sender);
  ptr += sizeof(*ptrGeneral);
  memcpy(ptr, sender, ptrGeneral->len);
  ptr += PAD(ptrGeneral->len);
```

```
/* Supplying recipient address */
ptrGeneral = (struct YMLSML_general *) ptr;
ptrGeneral->tag = YMLSML_tag_to;
ptrGeneral->len = strlen(recipient);
ptr += sizeof(*ptrGeneral);
memcpy(ptr, recipient, ptrGeneral->len);
ptr += PAD(ptrGeneral->len);

/* Supplying subject line */
ptrGeneral = (struct YMLSML_general *) ptr;
ptrGeneral->tag = YMLSML_tag_subject;
ptrGeneral->len = strlen(subject);
ptr += sizeof(*ptrGeneral);
memcpy(ptr, subject, ptrGeneral->len);
ptr += PAD(ptrGeneral->len);

/* Supplying message body */
ptrMsgAtt = (struct YMLSML_msg_att *) ptr;
ptrMsgAtt->tag = YMLSML_tag_msg_begin;
ptr += sizeof(*ptrMsgAtt);

ptrDataSpec = (struct YMLSML_data_spec *) ptr;
ptrDataSpec->tag = YMLSML_tag_data_spec;
ptrDataSpec->type = YMLSML_data;
ptrDataSpec->len = strlen(msgText);
ptr += sizeof(*ptrDataSpec);
memcpy(ptr, msgText, ptrDataSpec->len);
ptr += PAD(ptrDataSpec->len);

ptrCharset = (struct YMLSML_charset *) ptr;
ptrCharset->tag = YMLSML_tag_charset;
ptrCharset->binary = YMLSML_no;
memcpy(ptrCharset->src_charset_XHCS, srcCCSN,
       sizeof(ptrCharset->src_charset_XHCS));
memcpy(ptrCharset->dest_charset_XHCS, destCCSN,
       sizeof(ptrCharset->dest_charset_XHCS));
ptrCharset->len = strlen(charSetName);
ptr += sizeof(*ptrCharset);
memcpy(ptr, charSetName, ptrCharset->len);
ptr += PAD(ptrCharset->len);

ptrEncoding = (struct YMLSML_encoding *) ptr;
ptrEncoding->tag = YMLSML_tag_encoding;
ptrEncoding->mechanism = YMLSML_qp;
ptr += sizeof(*ptrEncoding);

ptrMsgAtt = (struct YMLSML_msg_att *) ptr;
ptrMsgAtt->tag = YMLSML_tag_msg_end;
ptr += sizeof(*ptrMsgAtt);
```

```
memset(&sendParam, 0x00, sizeof(sendParam));
FHDR_SET_RC_NIL(sendParam.hdr);
FHDR_MOD_IFID(sendParam.hdr, UNIT, SEND_FUNCTION, VERSION);
sendParam.in_data.mail_par = mailPar;
sendParam.in_data.mail_par_len = ptr - mailPar;
sendParam.in_data.wait = YMLSML_res;
sendParam.in_data.sec_prot = YMLSML_smime;
sendParam.in_data.encrypt = YMLSML_no;
sendParam.in_data.sign = YMLSML_no;
sendParam.in_data.cipher = YMLSML_cipher_from_optfile;
memcpy(sendParam.in_data.optfile, "*NONE", 5);
YMLSML(sendParam);
if (sendParam.hdr.FHDR_RC_MAINCODE == YMLSML_successful) {
  printf("YMLSML called successfully\n");
  memcpy(orderId, sendParam.out_data.order_id, orderIdLen);
  printf("  Order id: %.16s\n", sendParam.out_data.order_id);
  if (sendParam.in_data.wait == YMLSML_yes) {
    printf("  Return code: %d\n", sendParam.out_data.rc);
    printf("  Return message: %.160s\n", sendParam.out_data.ret_msg);
    exit(0);
  }
}
else {
  printf("Error in call of YMLSML: %08X\n", sendParam.hdr.FHDR_RC_NBR);
  printf("  Order id: %.16s\n", sendParam.out_data.order_id);
  printf("  Return code: %d\n", sendParam.out_data.rc);
  printf("  Return message: %.160s\n", sendParam.out_data.ret_msg);
  if (sendParam.hdr.FHDR_RC_MAINCODE == YMLSML_asti_error)
    printf("  ASTI error: %08X\n", sendParam.out_data.asti_rc);
  exit(1);
}

memset(&checkParam, 0x00, sizeof(checkParam));
FHDR_SET_RC_NIL(checkParam.hdr);
FHDR_MOD_IFID(checkParam.hdr, UNIT, CHECK_FUNCTION, VERSION);
checkParam.in_data.wait = YMLCML_yes;
checkParam.in_data.order = YMLCML_single;
memcpy(checkParam.in_data.order_id, orderId, orderIdLen);
YMLCML(checkParam);
if (checkParam.hdr.FHDR_RC_MAINCODE == YMLCML_successful) {
  printf("YMLCML called successfully\n");
  printf("  Order id: %.16s\n", checkParam.out_data.order_id);
  printf("  Return code: %d\n", checkParam.out_data.rc);
  printf("  Return message: %.160s\n", checkParam.out_data.ret_msg);
}
else {
  printf("Error in call of YMLCML: %08X\n", checkParam.hdr.FHDR_RC_NBR);
  printf("  Order id: %.16s\n", checkParam.out_data.order_id);
  printf("  Return code: %d\n", checkParam.out_data.rc);
  printf("  Return message: %.160s\n", checkParam.out_data.ret_msg);
```

```
    if (checkParam.hdr.FHDR_RC_MAINCODE == YMLCML_asti_error)
      printf("  ASTI error: %08X\n", checkParam.out_data.asti_rc);
  }
}
```

Beispiel 2: Funktionsaufruf YMLDML()

```
#include <stdio.h>
#include <stdlib.h>
#include "FHDR.H"
#include "YMLDML.H"

main(int argc, char *argv[])
{
  int c;
  int deleteAll = 0;
  struct YMLDML_pl_mdl deleteParam;
  enum {UNIT = 940, DELETE_FUNCTION = 22,
        VERSION = 1, orderIdLen = 16};
  char orderId[orderIdLen + 1];

  while ((c = getopt(argc, argv, "AO:")) != EOF) {
    if (c == 'O') {
      strupper(optarg, NULL);
      strncpy(orderId, optarg, orderIdLen);
      orderId[orderIdLen] = 0x00;
    }
    else if (c == 'A')
      deleteAll = 1;
  }
  memset(&deleteParam, 0x00, sizeof(deleteParam));
  FHDR_SET_RC_NIL(deleteParam.hdr);
  FHDR_MOD_IFID(deleteParam.hdr, UNIT, DELETE_FUNCTION, VERSION);
  if (deleteAll)
    deleteParam.in_data.order = YMLDML_all;
  else {
    deleteParam.in_data.order = YMLDML_single;
    memcpy(deleteParam.in_data.order_id, orderId, orderIdLen);
  }
  deleteParam.in_data.owner = YMLDML_own;
  YMLDML(deleteParam);
  if (deleteParam.hdr.FHDR_RC_MAINCODE == YMLDML_successful) {
    printf("YMLDML called successfully\n");
  }
  else {
    printf("Error in call of YMLDML: %08X\n",
           deleteParam.hdr.FHDR_RC_NBR);
    if (deleteParam.hdr.FHDR_RC_MAINCODE == YMLDML_asti_error)
      printf("  ASTI error: %08X\n", deleteParam.out_data.asti_rc);
    exit(1);
  }
}
```

Beispiel 3: Funktionsaufruf YMLGML()

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "FHDR.H"
#include "YMLGML.H"

main(int argc, char *argv[])
{
  int c;
  int getInfo = 0;
  struct YMLGML_pl_mdl getInfoParam;
  enum {UNIT = 940, GETINFO_FUNCTION = 23, VERSION = 1,
        orderIdLen = 16, maxListSize = 30, mailParSize = 32768};
  char orderId[orderIdLen + 1];
  struct YMLGML_order *orderIdList;
  int orderIdListSize = sizeof(orderIdList->num_order) +
                        sizeof(orderIdList->order_id) * maxListSize;
  struct YMLGMLoutpar *mailPar;
  char *statusTxt[5] =
    {"Waiting", "Deferred", "Active", "Sent", "Failed"};
  char *rcTxt[11] =
    {"Ok", "Parameter error", "Resource saturation", "SMTP error",
     "S/MIME error", NULL, NULL, NULL, NULL, NULL, "Internal error"};
  int i;
  unsigned long time;

  while ((c = getopt(argc, argv, "AO:S")) != EOF) {
    getInfo = c;
    if (getInfo == 'O') {
      strupper(optarg, NULL);
      strncpy(orderId, optarg, orderIdLen);
      orderId[orderIdLen] = 0x00;
    }
  }
  memset(&getInfoParam, 0x00, sizeof(getInfoParam));
  FHDR_SET_RC_NIL(getInfoParam.hdr);
  FHDR_MOD_IFID(getInfoParam.hdr, UNIT, GETINFO_FUNCTION, VERSION);
  if (getInfo == 'A') {
    getInfoParam.in_data.order = YMLGML_all;
    if (!(orderIdList = malloc(orderIdListSize))) {
      printf("malloc failed\n");
      exit(1);
    }
    getInfoParam.in_data.out = orderIdList;
    getInfoParam.in_data.outl = orderIdListSize;
  }
  else if (getInfo == 'S') {
    getInfoParam.in_data.order = YMLGML_sum;
    getInfoParam.in_data.out = NULL;
    getInfoParam.in_data.outl = 0;
```

```
}
else {
  getInfoParam.in_data.order = YMLGML_single;
  memcpy(getInfoParam.in_data.order_id, orderId, orderIdLen);
  if (!(mailPar = malloc(mailParSize))) {
    printf("malloc failed\n");
    exit(1);
  }
  getInfoParam.in_data.out = mailPar;
  getInfoParam.in_data.outl = mailParSize;
}
getInfoParam.in_data.owner = YMLGML_own;
YMLGML(getInfoParam);
if (getInfoParam.hdr.FHDR_RC_MAINCODE == YMLGML_successful) {
  printf("YMLGML called successfully\n");
  if (getInfo == 'A' || getInfo == 'S')
    printf("  Number of mail orders: %d\n",
           getInfoParam.out_data.sum);
  if (getInfo == 'A') {
    for (i = 0; i < orderIdList->num_order; i++)
      printf("  OrderId[%d]: %.16s\n",
             i + 1, orderIdList->order_id[i]);
  }
  else if (getInfo == 'O') {
    printf("  Mail status: %s\n",
      statusTxt[mailPar->_outp.mail_data.status - 1]);
    printf("  Mail submitter: %.8s\n",
           mailPar->_outp.mail_data.user);
    time = mailPar->_outp.mail_data.time -
            (50 * 365 + 12) * 24 * 60 * 60;
    printf("  Mail send time: %s", ctime(&time));
    if (mailPar->_outp.mail_data.slctr == YMLGML_sel_result) {
      printf("  Mail result: %s\n",
             rcTxt[mailPar->_outp.mail_data._data.result_data.rc]);
      printf("  Mail result message: %s\n",
             mailPar->_outp.mail_data._data.result_data.ret_msg);
    }
    printf("  Mail parameter length: %d\n",
           mailPar->_outp.mail_data.mail_par_len);
    printf("  Mail parameter:");
    for (i = 0; i < mailPar->_outp.mail_data.mail_par_len; i++) {
      if (i % 32 == 0)
        printf("\n    ");
      else if (i % 4 == 0)
        printf(" ");
       printf("%02X", mailPar->_outp.mail_data.mail_par[i]);
    }
  }
}
```

```
  else {
    printf("Error in call of YMLGML: %08X\n",
           getInfoParam.hdr.FHDR_RC_NBR);
    if (getInfoParam.hdr.FHDR_RC_MAINCODE == YMLGML_asti_error)
      printf("  ASTI error: %08X\n", getInfoParam.out_data.asti_rc);
    exit(1);
  }
}
```

Literatur

Die Handbücher sind online unter *http://manuals.ts.fujitsu.com* zu finden oder in gedruckter Form gegen gesondertes Entgelt unter *http://manualshop.ts.fujitsu.com* zu bestellen.

openNet Server (BS2000/OSD)
BCAM
Benutzerhandbuch

interNet Services (BS2000/OSD)
Administratorhandbuch

openNet Server (BS2000/OSD)
IPv6 Einführung und Umstellhandbuch Stufe 1
Benutzerhandbuch

openNet Server (BS2000/OSD)
IPSec
Benutzerhandbuch

openCRYPT (BS2000/OSD)
Sicherheit mit Kryptographie
Benutzerhandbuch

openNet Server V2.0, interNet Services V2.0
SNMP-Management für openNet Server und interNet Services
Benutzerhandbuch

openFT für BS2000/OSD
Managed File Transfer in der offenen Welt
Benutzerhandbuch

openFT für BS2000/OSD
Meldungen
Benutzerhandbuch

CMX (BS2000)
Kommunikationsmethode im BS2000
Benutzerhandbuch

SNMP Management V5.0
SNMP Management für BS2000/OSD
Benutzerhandbuch

SNMP Management V6.0
SNMP Management für BS2000/OSD
Benutzerhandbuch

C-Bibliotheksfunktionen (BS2000/OSD)
für POSIX-Anwendungen
Referenzhandbuch

XHCS (BS2000/OSD)
8-bit-Code- und Unicode-Unterstützung im BS2000/OSD
Benutzerhandbuch

LMS (BS2000/OSD)
SDF-Format
Benutzerhandbuch

BS2000
Benutzerkommandos (ISP-Format)
Benutzerhandbuch

BS2000/OSD-BC
Kommandos
Benutzerhandbuch

IMON (BS2000/OSD)
Installationsmonitor
Benutzerhandbuch

BS2000/OSD-BC
Einführung in die Systembetreuung
Benutzerhandbuch

Zusätzliche Literatur

SSL and TLS
Designing and Building Secure Systems
von Eric Rescorla
ISBN 0-201-61598-3

Inhalt
Ausführliche Beschreibung von SSL und TLS und des Anwendungumfelds

Secrets & Lies
IT-Sicherheit in einer vernetzten Welt
von Bruce Schneier
ISBN 3-89864-113-9

Inhalt
Tour d'horizon durch die IT-Sicherheit

Die englische Originalausgabe "Secrets & Lies" ist unter der ISBN 0-471-25311-1 erhältlich.

Postfix
The Definitive Guide
von Kyle D. Dent
ISBN 0-596-00212-2 (Original-Ausgabe)
ISBN 3-89721-372-9 (Deutsche Übersetzung)

Inhalt
Relativ kompakte und aktuelle Einführung in Konfiguration und Nutzung des Postfix-Mail-Servers.

http://www.oreilly.de/catalog/postfix/index.html
http://www.oreilly.de/catalog/postfixger/index.html (Deutsche Übersetzung)

Postfix
von Richard Blum
ISBN 0-672-32114-9

Inhalt
Ausführliche Beschreibung von Konfiguration und Nutzung des Postfix-Mail-Servers. Die Ausgabe von Mai 2001 ist nicht mehr ganz aktuell, so dass leichte Unterschiede zur BS2000/OSD-Portierung vorhanden sein können, außerdem ist die Darstellung etwas Linux-zentriert. Dennoch ist es durch die in die Tiefe gehende Beschreibung ein beachtenswertes Buch.

Das Postfix-Buch
Sichere Mail-Server mit Linux

von Peter Heinlein
ISBN 3-937514-04-X

Inhalt
Eine Linux-zentrierte Postfix-Beschreibung, die relativ breit auch nicht-Postfix-spezifische Mail-Themen abhandelt (Rechtliche Aspekte, allgemeine Server-Sicherheit).

The Book of Postfix
State-ot-the-Art Message Transport
von Ralf Hildebrandt und Patrick Koetter
ISBN 1-59327-001-1

Inhalt
Breite Darstellung der Themen Mail-Transport-Beschränkungen und Mail-Filterung. Außerdem erhält das Buch von den hier genannten Büchern die detaillierteste Beschreibung der TLS-Nutzung. Theorie-orientierte Kapitel werden durch ausführliche praktische Fall-Beispiele ergänzt (meist anhand von Linux-Systemen).

http://www.postfix-book.com/index.html

RFCs

Umfassende Informationen zu den Request for Comments (RFCs) finden Sie auf der Home Page der Internet Engineering Task Force (IETF):

www.ietf.org

Stichwörter

B

P

T