FUJITSU Software BS2000 interNet Services

Version 3.4A
*3 Mai 2016

Readme-Datei

1 Einleitung

Diese Readme-Datei enthält Änderungen und Erweiterungen zu interNet Services V3.4, die nach Herausgabe der Handbücher implementiert wurden.

*1 Änderungen gegenüber dem ersten Freigabestand im Mai 2014 sind mit *1 gekennzeichnet.
*2 Änderungen gegenüber dem zweiten Freigabestand im April 2015 sind mit *2 gekennzeichnet.
*3 Änderungen gegenüber dem dritten Freigabestand im November 2015 sind mit *3 gekennzeichnet.

1.1 Betroffene Handbücher

Die hier beschriebenen Änderungen betreffen folgende Handbücher:

[1] interNet Services V3.4A
Administratorhandbuch
Bestellnummer U41095-J-Z125-5
Ausgabe Dezember 2010

[2] interNet Services V3.4A
Benutzerhandbuch
Bestellnummer U41096-J-Z125-5
Ausgabe Dezember 2010

2 Software-Erweiterungen

2.1 Neue Funktionalität mit interNet Services V3.4A10

Unterstützung der TLS-Protokolle TLSv1.1 und TLSv1.2

Die TLS/SSL-Unterstützung bei den Services FTP, TELNET, Mail-Sender und Mail-Reader wird um die Protokolle TLSv1.1 und TLSv1.2 erweitert.

Unterstützung des Last Byte Pointer im FTP

Zusätzlich zur standardmäßigen Markierung des genauen Endes einer PAM-Datei mittels des Strings “C-DATEIENDE“ wird die Methode namens Last Byte Pointer (kurz: LBP) unterstützt. LBP verwendet Informationen aus dem Dateikatalogeintrag und die Datei wird dabei selbst nicht modifiziert. Die Unterstützung des LBP muss explizit über das Kommando setfile bzw. quote site SFIL eingeschaltet werden.

Unterstützung der High Availability bei FTP und TELNET

Neu erstellte Start-Dateien für FTP und TELNET über das SDF-Kommando SET-FTP-TELNET-PARAMERTERS (standardmäßig SYSENT.TCP-IP-AP.nnn.FTPD bzw. .TELNETD) haben keine Abhängigkeit mehr zum HSI und sind somit auf allen Business Servern einsetzbar.

2.2 Änderung im Administrationshandbuch [1]

2.2.1 Neue Funktionalität mit TCP-IP-AP V5.2A10

*2 **Kapitel 4.3 Konfiguration von FTP via Option-Datei**

*2 Erweiterung/Änderung bei Option -tlsProtocol (Seite 89):

*2 OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-Protokoll in den Versionen 1, 1.1 und 1.2. Mit der Option *-tlsProtocol* können einige dieser Protokolle selektiv aktiviert werden.

-tlsProtocol
[**+** | **-**] {**SSLv2** | **SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **ALL**} …

*2 …

*2 **SSLv3**
*2 SSL-Protokoll der Version 3

*2 **[i]** Das SSL-Protokoll in der Version 3 weist einige sicherheitstechnische Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

*2 **TLSv1.1**
*2 TLS-Protokoll der Version 1.1

*2 **TLSv1.2**
*2 TLS-Protokoll der Version 1.2

*2 Erweiterung/Änderung bei Option -tlsCipherSuite (Seite 91):

*2 Zusätzliche bzw. erweiterte Einträge in der „Zulässige Chiffre-Mnemonics“-Liste:

*2 kEDH, kDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymen Suiten.

*2 kEECDH, kECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymen Suiten.

*2 EECDH, ECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, ohne anonyme Suiten.

*2 AECDH
*2 Anonyme Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung.

*2 ECDH
*2 Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymem, kurzlebigem und fixiertem ECDH.

*2 aECDSA
*2 Verschlüsselungs-Suiten, die ECDSA-Authentifizierung verwenden, d.h. die Zertifikate beinhalten ECDSA-Schlüssel.

TLSv1.2, TLSv1.1, TLSv1, SSLv3, SSLv2
TLSv1.2-, TLSv1.1-, TLSv1-, SSLv3- oder SSLv2-Verschlüsselungs-Suiten. Anmerkung: Es gibt keine TLSv1.1-spezifischen Verschlüsselungs-Suiten.

AES128, AES256, AES
Verschlüsselungs-Suiten, die 128-Bit AES, 256-Bit AES oder eins von beiden verwenden.

AESGCM
Verschlüsselungs-Suiten, die AES im „Galois Counter Mode (GCM)“ verwenden. Diese Verschlüsselungs-Suiten werden nur durch TLSv1.2 unterstützt.

CAMELLIA128, CAMELLIA256, CAMELLIA
Verschlüsselungs-Suiten, die 128-Bit CAMELLIA, 256-Bit CAMELLIA oder eins von beiden verwenden.

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

[i] Da praktikable Angriffe auf SHA1 immer näher rücken, sollten so schnell wie möglich auf Verschlüsselungs-Suiten gewechselt werden, die z.B. die Hash-Funktionen SHA256 bzw. SHA384 verwenden. Dies impliziert aber in der Regel auch den Wechsel auf TLS-Protokollversion 1.2.

SHA256, SHA384
Verschlüsselungs-Suiten, die die SHA256- bzw. SHA384-Hash-Funktion für die MAC-(Message Authentication Code)-Berechnung verwenden. Bei Verschlüsselungs-Suiten, die AESGCM und damit AEAD (Authenticated Encryption with Associated Data) als MAC-Methode verwenden, hat das SHA256 bzw. SHA384 im Namen eine andere Bedeutung,

Die Tabelle der verfügbaren Verschlüsselungs-Suiten auf Seite 93 wird um folgende Einträge erweitert:

Name	Version	Schlüssel-Austausch	Authentifizierung	Verschlüsselung	MAC/ Digest
ECDHE-ECDSA-AES256-GCM-SHA384	TLSv1.2	ECDH	ECDSA	AESGCM(256)	AEAD
ECDHE-ECDSA-AES128-GCM-SHA256	TLSv1.2	ECDH	ECDSA	AESGCM(128)	AEAD
ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	ECDH	RSA	AESGCM(256)	AEAD
ECDHE-RSA-AES128-GCM-SHA256	TLSv1.2	ECDH	RSA	AESGCM(128)	AEAD
AES256-GCM-SHA384	TLSv1.2	RSA	RSA	AESGCM(256)	AEAD
AES128-GCM-SHA256	TLSv1.2	RSA	RSA	AESGCM(128)	AEAD
DHE-RSA-AES256-GCM-SHA384	TLSv1.2	DH	RSA	AESGCM(256)	AEAD

*2

DHE-RSA-AES128-GCM-SHA256	TLSv1.2	DH	RSA	AESGCM(128)	AEAD
DHE-DSS-AES356-GCM-SHA384	TLSv1.2	DH	DSS	AESGCM(256)	AEAD
DHE-DSS-AES128-GCM-SHA256	TLSv1.2	DH	DSS	AESGCM(128)	AEAD
ADH-AES256-GCM-SHA384	TLSv1.2	DH	keine	AESGCM(256)	AEAD
ADH-AES128-GCM-SHA256	TLSv1.2	DH	keine	AESGCM(128)	AEAD
ECDHE-ECDSA-AES256-SHA384	TLSv1.2	ECDH	ECDSA	AES(256)	SHA384
ECDHE-ECDSA-AES128-SHA256	TLSv1.2	ECDH	ECDSA	AES(128)	SHA256
ECDHE-RSA-AES256-SHA384	TLSv1.2	ECDH	RSA	AES(256)	SHA384
ECDHE-RSA-AES128-SHA256	TLSv1.2	ECDH	RSA	AES(128)	SHA256
DHE-RSA-AES256-SHA256	TLSv1.2	DH	RSA	AES(256)	SHA256
DHE-RSA-AES128-SHA256	TLSv1.2	DH	RSA	AES(128)	SHA256
DHE-DSS-AES256-SHA256	TLSv1.2	DH	DSS	AES(256)	SHA256
DHE-DSS-AES128-SHA256	TLSv1.2	DH	DSS	AES(128)	SHA256
AES256-SHA256	TLSv1.2	RSA	RSA	AES(256)	SHA256
AES128-SHA256	TLSv1.2	RSA	RSA	AES(128)	SHA256
ADH-AES256-SHA256	TLSv1.2	DH	keine	AES(256)	SHA256
ADH-AES128-SHA256	TLSv1.2	DH	keine	AES(128)	SHA256
ECDHE-ECDSA-AES256-SHA	SSLv3	ECDH	ECDSA	AES(256)	SHA1
ECDHE-ECDSA-AES128-SHA	SSLv3	ECDH	ECDSA	AES(128)	SHA1
ECDHE-ECDSA-DES-CBC3-SHA	SSLv3	ECDH	ECDSA	3DES(168)	SHA1
ECDHE-ECDSA-RC4-SHA	SSLv3	ECDH	ECDSA	RC4(128)	SHA1
ECDHE-ECDSA-NULL-SHA	SSLv3	ECDH	ECDSA	keine	SHA1
ECDHE-RSA-AES256-SHA	SSLv3	ECDH	RSA	AES(256)	SHA1
ECDHE-RSA-AES128-SHA	SSLv3	ECDH	RSA	AES(128)	SHA1
ECDHE-RSA-DES-CBC3-SHA	SSLv3	ECDH	RSA	3DES(168)	SHA1

ECDHE-RSA-RC4-SHA	SSLv3	ECDH	RSA	RC4(128)	SHA1
ECDHE-RSA-NULL-SHA	SSLv3	ECDH	RSA	keine	SHA1
AECDH-AES256-SHA	SSLv3	ECDH	keine	AES(256)	SHA1
AECDH-AES128-SHA	SSLv3	ECDH	keine	AES(128)	SHA1
AECDH-DES-CBC3-SHA	SSLv3	ECDH	keine	3DES(168)	SHA1
AECDH-RC4-SHA	SSLv3	ECDH	keine	RC4(128)	SHA1
AECDH-NULL-SHA	SSLv3	ECDH	keine	keine	SHA1
DHE-RSA-CAMELLIA256-SHA	SSLv3	DH	RSA	Camellia(256)	SHA1
DHE-RSA-CAMELLIA128-SHA	SSLv3	DH	RSA	Camellia(128)	SHA1
DHE-DSS-CAMELLIA256-SHA	SSLv3	DH	DSS	Camellia(256)	SHA1
DHE-DSS-CAMELLIA128-SHA	SSLv3	DH	DSS	Camellia(128)	SHA1
CAMELLIA256-SHA	SSLv3	RSA	RSA	Camellia(256)	SHA1
CAMELLIA128-SHA	SSLv3	RSA	RSA	Camellia(128)	SHA1
ADH-CAMELLIA256-SHA	SSLv3	DH	keine	Camellia(256)	SHA1
ADH-CAMELLIA128-SHA	SSLv3	DH	keine	Camellia(128)	SHA1

Kapitel 5.3 Konfiguration von TELNET via Option-Datei

Erweiterung/Änderung bei Option -Z Protocol (Seite 185):

OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-Protokoll in den Versionen 1, 1.1 und 1.2. Mit der Option *-Z Protocol* können einige dieser Protokolle selektiv aktiviert werden.

-Z Protocol
=[**+** | **-**] {**SSLv2** | **SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **ALL**} …

…

SSLv3
SSL-Protokoll der Version 3

[i] Das SSL-Protokoll in der Version 3 weist einige sicherheitstechnische Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

TLSv1.1
TLS-Protokoll der Version 1.1

TLSv1.2
TLS-Protokoll der Version 1.2

Erweiterung/Änderung bei Option -Z CipherSuite (Seite 179):

Zusätzliche bzw. erweiterte Einträge in der „Zulässige Chiffre-Mnemonics“-Liste:

kEDH, kDHE
Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymen Suiten.

*2 kEECDH, kECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymen Suiten.
*2
*2 EECDH, ECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, ohne anonyme Suiten.
*2
*2 AECDH
*2 Anonyme Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung.
*2
*2 ECDH
*2 Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymem, kurzlebigem und fixiertem ECDH.
*2
*2 aECDSA
*2 Verschlüsselungs-Suiten, die ECDSA-Authentifizierung verwenden, d.h. die Zertifikate beinhalten ECDSA-Schlüssel.
*2
*2 TLSv1.2, TLSv1.1, TLSv1, SSLv3, SSLv2
*2 TLSv1.2-, TLSv1.1-, TLSv1-, SSLv3- oder SSLv2-Verschlüsselungs-Suiten. Anmerkung: Es gibt keine TLSv1.1-spezifischen Verschlüsselungs-Suiten.
*2
*2 AES128, AES256, AES
*2 Verschlüsselungs-Suiten, die 128-Bit AES, 256-Bit AES oder eins von beiden verwenden.
*2
*2 AESGCM
*2 Verschlüsselungs-Suiten, die AES im „Galois Counter Mode (GCM)“ verwenden. Diese Verschlüsselungs-Suiten werden nur durch TLSv1.2 unterstützt.
*2
*2 CAMELLIA128, CAMELLIA256, CAMELLIA
*2 Verschlüsselungs-Suiten, die 128-Bit CAMELLIA, 256-Bit CAMELLIA oder eins von beiden verwenden.
*2
*2 SHA1, SHA
*2 Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.
*2
*2 **[i]** Da praktikable Angriffe auf SHA1 immer näher rücken, sollten so schnell wie möglich auf Verschlüsselungs-Suiten gewechselt werden, die z.B. die Hash-Funktionen SHA256 bzw. SHA384 verwenden. Dies impliziert aber in der Regel auch den Wechsel auf TLS-Protokollversion 1.2.
*2
*2 SHA256, SHA384
*2 Verschlüsselungs-Suiten, die die SHA256- bzw. SHA384-Hash-Funktion für die MAC-(Message Authentication Code)-Berechnung verwenden. Bei Verschlüsselungs-Suiten, die AESGCM und damit AEAD (Authenticated Encryption with Associated Data) als MAC-Methode verwenden, hat das SHA256 bzw. SHA384 im Namen eine andere Bedeutung,
*2
*2 Für die Erweiterung der Tabelle der verfügbaren Verschlüsselungs-Suiten auf Seite 181 siehe die entsprechende Tabelle in Abschnitt 2.2.1 dieser Readme.
*2

*2
*2 ### 2.2.2 Neue Funktionalität mit MAIL V3.3A08
*2
*2 **Kapitel 11.2.2 Konfigurationsdatei für das Mail-Sender Backend**
*2
*2 Erweiterung/Änderung bei Option tlsProtocol (Seite 388):
*2
*2 OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-
*2 Protokoll in den Versionen 1, 1.1 und 1.2. Mit der Option *tlsProtocol* können einige
*2 dieser Protokolle selektiv aktiviert werden.
*2
*2
*2

tlsProtocol
[**+** | **-**] {**SSLv2** | **SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **ALL**} …

*2
*2 …
*2
*2 **SSLv2**
*2 SSL-Protokoll der Version 2
*2
*2
*2 **[i]** Das SSL-Protokoll in der Version 2 weist einige sicherheitstechnische
*2 Mängel auf und sollte nach Möglichkeit nicht verwendet werden.
*2
*2 **SSLv3**
*2 SSL-Protokoll der Version 3
*2
*2
*2 **[i]** Das SSL-Protokoll in der Version 3 weist einige sicherheitstechnische
*2 Mängel auf und sollte nach Möglichkeit nicht verwendet werden.
*2
*2 **TLSv1.1**
*2 TLS-Protokoll der Version 1.1
*2
*2 **TLSv1.2**
*2 TLS-Protokoll der Version 1.2
*2
*2 Erweiterung/Änderung bei Option tlsCipherSuite (Seite 390):
*2
*2 Zusätzliche bzw. erweiterte Einträge in der „Zulässige Chiffre-Mnemonics“-Liste:
*2
*2 kEDH, kDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüssel-
*2 vereinbarung, einschließlich anonymen Suiten.
*2
*2 kEECDH, kECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hell-
*2 man-Schlüsselvereinbarung, einschließlich anonymen Suiten.
*2
*2 EECDH, ECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hell-
*2 man-Schlüsselvereinbarung, ohne anonyme Suiten.
*2
*2 AECDH
*2 Anonyme Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüssel-
*2 vereinbarung.
*2
*2 ECDH
*2 Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinba-
*2 rung, einschließlich anonymem, kurzlebigem und fixiertem ECDH.
*2
*2 aECDSA

Verschlüsselungs-Suiten, die ECDSA-Authentifizierung verwenden, d.h. die Zertifikate beinhalten ECDSA-Schlüssel.

TLSv1.2, TLSv1.1, TLSv1, SSLv3, SSLv2
TLSv1.2-, TLSv1.1-, TLSv1-, SSLv3- oder SSLv2-Verschlüsselungs-Suiten. Anmerkung: Es gibt keine TLSv1.1-spezifischen Verschlüsselungs-Suiten.

AES128, AES256, AES
Verschlüsselungs-Suiten, die 128-Bit AES, 256-Bit AES oder eins von beiden verwenden.

AESGCM
Verschlüsselungs-Suiten, die AES im „Galois Counter Mode (GCM)“ verwenden. Diese Verschlüsselungs-Suiten werden nur durch TLSv1.2 unterstützt.

CAMELLIA128, CAMELLIA256, CAMELLIA
Verschlüsselungs-Suiten, die 128-Bit CAMELLIA, 256-Bit CAMELLIA oder eins von beiden verwenden.

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

[i] Da praktikable Angriffe auf SHA1 immer näher rücken, sollten so schnell wie möglich auf Verschlüsselungs-Suiten gewechselt werden, die z.B. die Hash-Funktionen SHA256 bzw. SHA384 verwenden. Dies impliziert aber in der Regel auch den Wechsel auf TLS-Protokollversion 1.2.

SHA256, SHA384
Verschlüsselungs-Suiten, die die SHA256- bzw. SHA384-Hash-Funktion für die MAC-(Message Authentication Code)-Berechnung verwenden. Bei Verschlüsselungs-Suiten, die AESGCM und damit AEAD (Authenticated Encryption with Associated Data) als MAC-Methode verwenden, hat das SHA256 bzw. SHA384 im Namen eine andere Bedeutung,

Für die Erweiterung der Tabelle der verfügbaren Verschlüsselungs-Suiten auf Seite 392 siehe die entsprechende Tabelle in Abschnitt 2.2.1 dieser Readme.

Im Folgenden wird eine neue Konfigurationsoption beschrieben:

smtpReadMaxWaitTime

Die Option *smtpReadMaxWaitTime* legt fest, wie lange das Mail-Sender Backend ggf. auf eine Antwort des SMTP-Servers warten soll. Wird ein Mail-Sende-Auftrag wegen zu langer Wartezeit abgebrochen, dann wird er wie bei Fehlermeldungen des SMTP-Servers, die auf ein temporär vorhandenes Problem hinweisen, nach einer gewissen Zeit (siehe Optionen *smtpRetryTimeBase* und *smtpRetryTimeMaxExp*) wiederholt.
Da die mit dem Mail-Sender Backend kommunizierenden Kommandos MODIFY-MAIL-SERVICE-PARAMETER, SHOW-MAIL-SERVICE-PARAMETER und STOP-MAIL-SERVICE einerseits und die Kommunikation des Backend mit dem SMTP-Server anderseits miteinander serialisiert werden, ist es für die möglichst prompte Abarbeitung dieser Kommandos wünschenswert, einen Wartezustand des Backend z.B. wegen einer Verklemmung aufgrund von SMTP-Server-Problemen zeitlich möglichst zu begrenzen. Anderseits sollte diese Begrenzung auch nicht zu drastisch ausfallen, da sonst z.B. ein überlasteter SMTP-Server durch Transfer-

Abbrüche und -Wiederholungen noch stärker belastet wird. Zeiten im einstelligen Minutenbereich sollten i.A. einen guten Kompromiss darstellen.

smtpReadMaxWaitTime
<zeit>[s | m | h | d]

<zeit>
Max. Wartezeit
Voreinstellung: 5m

Ohne Angabe einer Maßeinheit gilt die angegebene <zeit> als Minuten. Mit Angabe einer Maßeinheit (s für Sekunde, m für Minute, h für Stunde, d für Tag) muss diese unmittelbar hinter <zeit> stehen, d.h. ohne Leerzeichen. Wird 0 angegeben, dann wird die Wartezeit nicht begrenzt.

2.2.3 Neue Funktionalität mit MAIL V3.3A06

Kapitel 11.2.2 Konfigurationsdatei für das Mail-Sender Backend

Im Folgenden werden zwei neue Konfigurationsoptionen beschrieben:

smtpRetryTimeBase

Die Option *smtpRetryTimeBase* legt die Zeitbasis fest, welche zur Ermittlung der Zeit verwendet wird, nach der bei einem fehlgeschlagenen Mailversand ein erneuter Mailversand versucht wird. Für Details siehe Option *smtpRetryTimeMaxExp*.

smtpRetryTimeBase
<wert>[s | m | h | d]

<wert>
Zeitbasis
Voreinstellung: 15m

Ohne Angabe einer Maßeinheit gilt der angegebene <wert> als Minuten. Mit Angabe einer Maßeinheit (s für Sekunde, m für Minute, h für Stunde, d für Tag) muss diese unmittelbar hinter <wert> stehen, d.h. ohne Leerzeichen.

smtpRetryTimeMaxExp

Die Option *smtpRetryTimeMaxExp* begrenzt die Erhöhung der Wartezeit zwischen zwei Wiederholungen von Mailversandversuchen. Normalerweise verdoppelt sich die Wartezeit mit jedem fehlgeschlagenen Versandversuch, um bei länger andauernden Problemen den CPU-Verbrauch durch die Versandversuche zu begrenzen. Nach *smtpRetryTimeMaxExp* Verdopplungen bleibt die Wartezeit auf dem dann erreichten Wert.

smtpRetryTimeMaxExp
<wert>

<wert>
Voreinstellung: 6

Bei Fehlern während des Verbindungsaufbaus zum SMTP-Mailserver wird als Wartezeit bis zu einem erneuten Zustellversuch konstant die doppelte *smtpRetryTimeBase* verwendet.

Tritt der Fehler erst später im SMTP-Dialog auf, so dass es sich möglicherweise nicht um ein vergleichsweise schnell bemerktes, allgemeines Server-Problem handelt, sondern um ein mailspezifisches, welches oft erst nach einiger Zeit bemerkt wird, dann wird die Wartezeit zwischen zwei Versandversuchen (beginnend bei *smtpRetryTimeBase*) mit jedem Versuch verdoppelt, bis *smtpRetryTimeMaxExp* Verdopplungen erreicht sind.

Voreinstellung der max. Wartezeit zwischen zwei Zustellversuchen:

max. Wartezeit = *smtpRetryTimeBase* mal 2 hoch *smtpRetryTimeMaxExp*

max. Wartezeit = 15m * 2^6 = 960m = 16h

Szenario 1: Mailserver nicht erreichbar
Erneute Zustellversuche nach 30 min = 2 * 15m

Szenario 2: Verbindungsaufbau zum Mailserver möglich; mailspezifischer Fehler
Erneuter Zustellversuch nach 15 min = 15m * 2^0
Erneuter Zustellversuch nach 30 min = 15m * 2^1
Erneuter Zustellversuch nach 1 h = 15m * 2^2
Erneuter Zustellversuch nach 2 h = 15m * 2^3
Erneuter Zustellversuch nach 4 h = 15m * 2^4
Erneuter Zustellversuch nach 8 h = 15m * 2^5
Alle weiteren Zustellversuche nach 16 h = 15m * 2^6
bis *maxQueueLifeTime* (Voreinstellung 5 Tage) erreicht ist.

Hinweis:
Tendenziell sollte bei der Reduzierung der *smtpRetryTimeBase* gleichzeitig der Wert für *smtpRetryTimeMaxExp* erhöht werden, ansonsten belasten die häufigen Wiederholungen der Zustellversuche die CPU.

2.2.4 Neue Funktionalität mit MAIL V3.3A02

Kapitel 11.2.2 Konfigurationsdatei für das Mail-Sender Backend

Im Folgenden wird die neue Konfigurationsoption beschrieben:

maxQueueLifeTime

Die Option *maxQueueLifeTime* legt die maximal Lebensdauer einer Mail fest, während der eine fehlgeschlagene Mailversand wiederholt wird.

maxQueueLifeTime
<lifetime>[s | m | h | d]

<lifetime>
Voreinstellung: 5d

Ohne Angabe einer Maßeinheit gilt der angegebene <lifetime> als Tage. Mit Angabe einer Maßeinheit (s für Sekunde, m für Minute, h für Stunde, d für Tag) muss diese unmittelbar hinter <lifetime> stehen, d.h. ohne Leerzeichen.

Hinweis:
Die Option *retryLimit* (Seite 386) ist mit der Einführung von *maxQueueLifeTime* wirkungslos.

2.2.5 Korrekturen

Kapitel 5.3.2 Options für den sicheren Einsatz von TELNET mithilfe von Authentifizierung und Verschlüsselung

Ergänzung:
Das Gleichheitszeichen muss ohne Leerzeichen auf den Options-Namen folgen und auch nach dem Gleichheitszeichen darf kein Leerzeichen sein.

Kapitel 5.3.3 Option -Z Unterstützung der START-TLS-Option

Korrektur zu -Z tls-required (Seite 171)

-Z tls-required
[={yes | no | optional}]

optional
START-TLS-Unterstützung wird optional eingeschaltet, d.h. nur auf Anforderung des Telnet-Client erfolgt die TLS-Absicherung.

*3 **Kapitel 4.3 Konfiguration von FTP via Option-Datei**
*3 **Kapitel 5.3 Konfiguration von TELNET via Option-Datei**
*3
*3 Da SSLv2 von der nun verwendeten Version der OpenSSL-Bibliothek aus Sicher-
*3 heitsgründen nicht mehr unterstützt wird, wird bei der Option -tlsProtocol bzw. -Z
*3 Protocol die Angabe SSLv2 faktisch ignoriert.
*3 Die Parameterlänge bei den Ciphersuiten, die DH (Diffie-Hellman) für den Schlüs-
*3 selaustausch verwenden, wurde von 1024 auf 2048 Bit erhöht.

2.3 Änderung im Benutzerhandbuch [2]

2.3.1 Neue Funktionalität in MAIL V3.3A08 und TCP-IP-AP V5.2A10

*2 **Kapitel 3.3 Überblick über SSL**
*2
*2 Ergänzung von unterstützten TLS-Versionen (Seite 41).
*2
*2 Mit Einführung der genannten MAIL- und TCP-IP-AP-Versionen wird die Version
*2 1.0.2d des OpenSSL-Toolkits unterstützt. Die unterstützten Protokollversionen
*2 sind SSLv2, SSLv3, TLSv1, TLSv1.1 und TLSv1.2.
*2
*2 **Kapitel 3.3.2 SSL und TLS**
*2
*2 Erweiterung des Warn-Hinweises (Seite 42):
*2
*2 Das SSL-Protokoll in den Versionen 2 und 3 weist einige sicherheitstechnische
*2 Mängel auf und sollte nach Möglichkeit nicht verwendet werden.
*2 Bestimmte sicherheitstechnische Mängel sind erst mit der TLS-Version 1.2 auf
*2 grundlegende Weise behoben, so dass dieser Version nach Möglichkeit der Vor-
*2 zug vor älteren TLS-Versionen gegeben werden sollte.

*2 **Kapitel 4.1 FTP-Server im BS2000/OSD**

*2 Aufruf der FTP-Server-Funktionen durch den FTP-Client des Partnerrechners
*2 (Seite 68):

*2 *quote site sfil datend on|off|lbp*
*2 Ein-/Ausschalten des speziellen EOF-Markers (Default: eingeschaltet) oder Ver-
*2 wendung der neuen EOF-Markierungsmethode Last Byte Pointer (LBP).

*2 **Kapitel 4.7 Parametereinstellung mithilfe von Option-Dateien**

*2 Erweiterung/Änderung bei Option -tlsProtocol (Seite 96):

*2 OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-
*2 Protokoll in den Versionen 1, 1.1 und 1.2. Mit der Option *-tlsProtocol* können ei-
*2 nige dieser Protokolle selektiv aktiviert werden.

-tlsProtocol
[**+** | **-**] {**SSLv2** | **SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **ALL**} …

*2 …

*2 **SSLv3**
*2 SSL-Protokoll der Version 3

*2 **[i]** Das SSL-Protokoll in der Version 3 weist einige sicherheitstechnische
*2 Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

*2 **TLSv1.1**
*2 TLS-Protokoll der Version 1.1

*2 **TLSv1.2**
*2 TLS-Protokoll der Version 1.2

*2 Erweiterung/Änderung bei Option -tlsCipherSuite (Seite 98):

*2 Zusätzliche bzw. erweiterte Einträge in der „Zulässige Chiffre-Mnemonics“-Liste:

*2 kEDH, kDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüssel-
*2 vereinbarung, einschließlich anonymen Suiten.

*2 kEECDH, kECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hell-
*2 man-Schlüsselvereinbarung, einschließlich anonymen Suiten.

*2 EECDH, ECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hell-
*2 man-Schlüsselvereinbarung, ohne anonyme Suiten.

*2 AECDH
*2 Anonyme Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüssel-
*2 vereinbarung.

*2 ECDH
*2 Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinba-
*2 rung, einschließlich anonymem, kurzlebigem und fixiertem ECDH.

*2 aECDSA
*2 Verschlüsselungs-Suiten, die ECDSA-Authentifizierung verwenden, d.h. die Zertifikate beinhalten ECDSA-Schlüssel.
*2
*2 TLSv1.2, TLSv1.1, TLSv1, SSLv3, SSLv2
*2 TLSv1.2-, TLSv1.1-, TLSv1-, SSLv3- oder SSLv2-Verschlüsselungs-Suiten. Anmerkung: Es gibt keine TLSv1.1-spezifischen Verschlüsselungs-Suiten.
*2
*2 AES128, AES256, AES
*2 Verschlüsselungs-Suiten, die 128-Bit AES, 256-Bit AES oder eins von beiden verwenden.
*2
*2 AESGCM
*2 Verschlüsselungs-Suiten, die AES im „Galois Counter Mode (GCM)“ verwenden. Diese Verschlüsselungs-Suiten werden nur durch TLSv1.2 unterstützt.
*2
*2 CAMELLIA128, CAMELLIA256, CAMELLIA
*2 Verschlüsselungs-Suiten, die 128-Bit CAMELLIA, 256-Bit CAMELLIA oder eins von beiden verwenden.
*2
*2 SHA1, SHA
*2 Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.
*2
*2 **[i]** Da praktikable Angriffe auf SHA1 immer näher rücken, sollten so schnell wie möglich auf Verschlüsselungs-Suiten gewechselt werden, die z.B. die Hash-Funktionen SHA256 bzw. SHA384 verwenden. Dies impliziert aber in der Regel auch den Wechsel auf TLS-Protokollversion 1.2.
*2
*2 SHA256, SHA384
*2 Verschlüsselungs-Suiten, die die SHA256- bzw. SHA384-Hash-Funktion für die MAC-(Message Authentication Code)-Berechnung verwenden. Bei Verschlüsselungs-Suiten, die AESGCM und damit AEAD (Authenticated Encryption with Associated Data) als MAC-Methode verwenden, hat das SHA256 bzw. SHA384 im Namen eine andere Bedeutung,
*2
*2
*2 Für die Erweiterung der Tabelle der verfügbaren Verschlüsselungs-Suiten auf Seite 100 siehe die entsprechende Tabelle in Abschnitt 2.2.1 dieser Readme.
*2
*2
*2 **Kapitel 4.10 Kommandoübersicht (FTP-Client)**
*2
*2 **setfile - Datei-Marker ein-/ausschalten** (Seite 195):
*2
*2 Es gibt zwei Methoden, das genaue Ende einer PAM-Datei zu markieren. Die hergebrachte Methode verwendet hierfür einen speziellen String, der u.a. "C-DATEI-ENDE" enthält. Die neue Methode namens Last Byte Pointer (kurz: LBP) verwendet Informationen, die im Dateikatalogeintrag hinterlegt werden, die Datei selbst wird dabei nicht modifiziert. Soll die Datei von Programmen weiter verarbeitet werden, bei denen die Markierung mit dem speziellen String zu Problemen führt, dann muss das Anfügen eines Markers abgeschaltet oder alternativ die LBP-Methode verwendet werden.
*2
*2

setfile
[datend on | off | lbp] [pademptyrec on | off]

*2
*2 datend on | off | lbp

schaltet die Verwendung des Dateiende-Markers „C-DATEIENDE“ ein bzw. aus bzw. aktiviert die Verwendung der neuen Dateiende-Markierungsmethode LBP.

Kapitel 6.1.3.3 START-TLS-Option

Erweiterung/Änderung bei Option -Z Protocol (Seite 306):

OpenSSL unterstützt das SSL-Protokoll in den Versionen 2 und 3 sowie das TLS-Protokoll in den Versionen 1, 1.1 und 1.2. Mit der Option *-Z Protocol* können einige dieser Protokolle selektiv aktiviert werden.

-Z Protocol
=[**+** | **-**] {**SSLv2** | **SSLv3** | **TLSv1** | **TLSv1.1** | **TLSv1.2** | **ALL**} …

…

SSLv3
SSL-Protokoll der Version 3

[i] Das SSL-Protokoll in der Version 3 weist einige sicherheitstechnische Mängel auf und sollte nach Möglichkeit nicht verwendet werden.

TLSv1.1
TLS-Protokoll der Version 1.1

TLSv1.2
TLS-Protokoll der Version 1.2

Erweiterung/Änderung bei Option -Z CipherSuite (Seite 301):

Zusätzliche bzw. erweiterte Einträge in der „Zulässige Chiffre-Mnemonics“-Liste:

kEDH, kDHE
Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymen Suiten.

kEECDH, kECDHE
Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymen Suiten.

EECDH, ECDHE
Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, ohne anonyme Suiten.

AECDH
Anonyme Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung.

ECDH
Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinbarung, einschließlich anonymem, kurzlebigem und fixiertem ECDH.

aECDSA
Verschlüsselungs-Suiten, die ECDSA-Authentifizierung verwenden, d.h. die Zertifikate beinhalten ECDSA-Schlüssel.

*2 TLSv1.2, TLSv1.1, TLSv1, SSLv3, SSLv2
*2 TLSv1.2-, TLSv1.1-, TLSv1-, SSLv3- oder SSLv2-Verschlüsselungs-Suiten. An-
*2 merkung: Es gibt keine TLSv1.1-spezifischen Verschlüsselungs-Suiten.
*2
*2 AES128, AES256, AES
*2 Verschlüsselungs-Suiten, die 128-Bit AES, 256-Bit AES oder eins von beiden
*2 verwenden.
*2
*2 AESGCM
*2 Verschlüsselungs-Suiten, die AES im „Galois Counter Mode (GCM)“ verwen-
*2 den. Diese Verschlüsselungs-Suiten werden nur durch TLSv1.2 unterstützt.
*2
*2 CAMELLIA128, CAMELLIA256, CAMELLIA
*2 Verschlüsselungs-Suiten, die 128-Bit CAMELLIA, 256-Bit CAMELLIA oder eins
*2 von beiden verwenden.
*2
*2 SHA1, SHA
*2 Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.
*2
*2 **[i]** Da praktikable Angriffe auf SHA1 immer näher rücken, sollten so schnell
*2 wie möglich auf Verschlüsselungs-Suiten gewechselt werden, die z.B.
*2 die Hash-Funktionen SHA256 bzw. SHA384 verwenden. Dies impliziert
*2 aber in der Regel auch den Wechsel auf TLS-Protokollversion 1.2.
*2
*2 SHA256, SHA384
*2 Verschlüsselungs-Suiten, die die SHA256- bzw. SHA384-Hash-Funktion für die
*2 MAC-(Message Authentication Code)-Berechnung verwenden. Bei Verschlüs-
*2 selungs-Suiten, die AESGCM und damit AEAD (Authenticated Encryption with
*2 Associated Data) als MAC-Methode verwenden, hat das SHA256 bzw. SHA384
*2 im Namen eine andere Bedeutung,
*2
*2
*2 Für die Erweiterung der Tabelle der verfügbaren Verschlüsselungs-Suiten auf Sei-
*2 te 303 siehe die entsprechende Tabelle in Abschnitt 2.2.1 dieser Readme.
*2
*2
*2 **Kapitel 8.2.3 POP3/IMAP-Server: Parameterbereich SERVER**
*2
*2 Erweiterung/Änderung bei Option PROTOCOL (Seite 383):
*2
*2 **PROTOCOL=<protocol spec>**
*2 Sie können die verwendeten Protokolle einschränken. Grundsätzlich werden SSL
*2 Version 2 und 3 und TLS Version1, 1.1 und 1.2 unterstützt.
*2 Erlaubt sind die Angaben SSLv2, SSLv3, TLSv1, TLSv1.1, TLSv1.2 und ALL.
*2
*2 Erweiterung/Änderung bei Option CIPHER_SUITE (Seite 385):
*2
*2 Zusätzliche bzw. erweiterte Einträge in der „Zulässige Chiffre-Mnemonics“-Liste:
*2
*2 kEDH, kDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Diffie-Hellman-Schlüssel-
*2 vereinbarung, einschließlich anonymen Suiten.
*2
*2 kEECDH, kECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hell-
*2 man-Schlüsselvereinbarung, einschließlich anonymen Suiten.
*2
*2 EECDH, ECDHE
*2 Verschlüsselungs-Suiten mit kurzlebiger (ephemeral) Elliptic Curve Diffie-Hell-
*2 man-Schlüsselvereinbarung, ohne anonyme Suiten.

*2 AECDH
*2 Anonyme Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüssel-
*2 vereinbarung.
*2
*2 ECDH
*2 Verschlüsselungs-Suiten mit Elliptic Curve Diffie-Hellman-Schlüsselvereinba-
*2 rung, einschließlich anonymem, kurzlebigem und fixiertem ECDH.
*2
*2 aECDSA
*2 Verschlüsselungs-Suiten, die ECDSA-Authentifizierung verwenden, d.h. die Zer-
*2 tifikate beinhalten ECDSA-Schlüssel.
*2
*2 TLSv1.2, TLSv1.1, TLSv1, SSLv3, SSLv2
*2 TLSv1.2-, TLSv1.1-, TLSv1-, SSLv3- oder SSLv2-Verschlüsselungs-Suiten. An-
*2 merkung: Es gibt keine TLSv1.1-spezifischen Verschlüsselungs-Suiten.
*2
*2 AES128, AES256, AES
*2 Verschlüsselungs-Suiten, die 128-Bit AES, 256-Bit AES oder eins von beiden
*2 verwenden.
*2
*2 AESGCM
*2 Verschlüsselungs-Suiten, die AES im „Galois Counter Mode (GCM)“ verwen-
*2 den. Diese Verschlüsselungs-Suiten werden nur durch TLSv1.2 unterstützt.
*2
*2 CAMELLIA128, CAMELLIA256, CAMELLIA
*2 Verschlüsselungs-Suiten, die 128-Bit CAMELLIA, 256-Bit CAMELLIA oder eins
*2 von beiden verwenden.
*2
*2 SHA1, SHA
*2 Verschlüsselungs-Suiten mit SHA1-Hash-Funktion.
*2
*2 **[i]** Da praktikable Angriffe auf SHA1 immer näher rücken, sollten so schnell
*2 wie möglich auf Verschlüsselungs-Suiten gewechselt werden, die z.B.
*2 die Hash-Funktionen SHA256 bzw. SHA384 verwenden. Dies impliziert
*2 aber in der Regel auch den Wechsel auf TLS-Protokollversion 1.2.
*2
*2 SHA256, SHA384
*2 Verschlüsselungs-Suiten, die die SHA256- bzw. SHA384-Hash-Funktion für die
*2 MAC-(Message Authentication Code)-Berechnung verwenden. Bei Verschlüs-
*2 selungs-Suiten, die AESGCM und damit AEAD (Authenticated Encryption with
*2 Associated Data) als MAC-Methode verwenden, hat das SHA256 bzw. SHA384
*2 im Namen eine andere Bedeutung,
*2
*2
*2 Für die Erweiterung der Tabelle der verfügbaren Verschlüsselungs-Suiten auf Sei-
*2 te 386 siehe die entsprechende Tabelle in Abschnitt 2.2.1 dieser Readme.
*2

2.3.2 Korrekturen

*3 **Kapitel 3.4.1 Prozedur MAKE.CERT - Test-Zertifikate und CSRs erzeugen**

*3 Korrektur zur Schlüssellänge (Seite 47 und 49).

*3 Die Prozedur MAKE.CERT generiert für RSA ein Schlüsselpaar mit 2048 Bit
*3 Schlüssellänge, bei DSA bleibt die Schlüssellänge bei 1024 Bit.

Kapitel 7.3.1 scp – sicheres Kopieren von Dateien zwischen Rechnern im Netz

Ergänzung zum Schalter `-X binary`:

Bei Transfers zu EBCDIC-Servern ist zusätzlich vorher oder nachher eine Transformation von EBCDIC zu ASCII (z.B. per Posix-Kommando iconv -f edf04 -t 8859) durchzuführen, um effektiv eine binäre Übertragung zu erhalten. Entsprechend ist bei Transfers von EBCDIC-Servern eine zusätzliche Transformation von ASCII nach EBCDIC durchzuführen (z.B. per Posix-Kommando iconv -f 8859 -t edf04).

*1 **Kapitel 8.1 MAIL-Reader starten/beenden**
*1
*1 **Mail-Reader beenden**
*1
*1 Für die beschriebenen /INTR Kommandos gelten folgende Bedingungen:
*1 - nur über die Konsolschnittstelle vom Systembediener zulässig
*1 - nur wirksam, wenn der Mail-Reader als Batch-Task ausgeführt wird
*1 - die TSN der Batch-Task angegeben wird
*1
*1 Der Benutzer beendet den als Batch-Task ausgeführten Mail-Reader mit:
*1
*1 /CANCEL-JOB JOB-IDENTIFICATION=*TSN(TSN=<tsn der Batch-Task>)

*1 **Kapitel 8.2 Konfigurationsdatei**
*1
*1 **Konfiguration des Mail-Readers via Konfigurationsdatei ändern**
*1
*1 Für das beschriebene /INTR Kommando im Batch-Betrieb gilt folgende Bedin-
*1 gung:
*1 - nur über die Konsolschnittstelle vom Systembediener zulässig

*2 **Kapitel 8.4.1 Aufbau einer Mail**
*2
*2 **Ergänzung zum Thema Zeichensatzkonvertierung:**
*2
*2 Der Mail-Reader konvertiert alle vom IMAP- bzw. POP3-Server erhaltenen Daten
*2 von ISO-8859-1 nach EDF041. Wenn ein MIME-Bestandteil der Mail als Content-
*2 Transfer-Encoding ‚base64' verwendet und der Content-Type ‚text' ist, dann wer-
*2 den die Daten nach der base64-Dekodierung wiederum von ISO-8859-1 nach
*2 EDF041 konvertiert. In allen anderen Fällen muss eine gegebenenfalls notwen-
*2 dige Zeichensatzkonvertierung von den Verarbeitungsprozeduren vorgenommen
*2 werden.