



**MAIL** für BS2000/OSD und z/OS

**Benutzerhandbuch**

**Ausgabe September 2016 (Mail V4.33)**

BS2000/OSD ist ein eingetragenes Warenzeichen der Fujitsu Technology Solutions GmbH

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwendung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadensersatz.

Im Laufe der Entwicklung des Programms können Leistungsmerkmale ohne vorhergehende Ankündigung hinzugefügt bzw. geändert werden oder entfallen.

Copyright © **OPG** Online-Programmierung GmbH, 1998 - 2016  
Sendlinger Str. 28, 80331 München,  
Tel. 089/267831, Fax 089/2609929, E-Mail [info@opg.de](mailto:info@opg.de), <http://www.opg.de>  
Alle Rechte vorbehalten.

## Änderungsprotokoll MAIL

Im folgenden sind die Erweiterungen des Programms MAIL zusammengestellt:

- Sept. 2016 Neue Anweisung `PRIORITY` (S. 58): Über die neue Anweisung `PRIORITY=HIGH/NORMAL/LOW` kann die Wichtigkeit einer Mail angegeben werden. Dies bewirkt, dass beim Empfänger die Mail entsprechend der Wichtigkeit gekennzeichnet wird.
- Dez.2015 Neuer Parameter `POP3USER=` (S. 77): Falls der in der `<USER>`-Sektion angegebene Name nicht dem POP3-Anmeldenamen entspricht, so kann über den neuen Parameter `POP3USER=pop3user` ein anderer Anmeldenamen angegeben werden.
- Oktober 2010 Neue Unterprogramm-Schnittstelle `OPGMAILS` (S. 83): Aufruf der Funktion "MAILS" über den ASTI-Service `OPGMAILCLNT` (S. 17). Die neue Schnittstelle hat den Vorteil, daß nur ein kleiner Verbindungsmodul dazu gebunden wird, der mit dem ASTI-Service kommuniziert. Probleme mit verschiedenen Socket- und CRTE-Versionen werden dadurch umgangen. Für die Schnittstelle werden die gleichen Parameter benutzt wie für die Schnittstelle `MAILUP`.
- Nov. 2009 Beim Empfangen von Attachments im BS2000-Format (`MODE=BS2`) und `OPGCRPYT`-Format (`COMP=Y`) wird das ursprüngliche CCS auch für die Empfangsdatei gesetzt.
- Januar 2009 Ab Version 4.0 wird die Socket-Schnittstelle Version 2.0 (Subsystem `SOC6`) verwendet. Die Funktionen `OPGCrypt` wird auch über einen ASTI-Service (S. 17) zur Verfügung gestellt. Die Kommunikation mit dem Service erfolgt über das neue SDF-Kommando `OPGCRYPT-FILE` (S. 96)
- Sept. 2008 Neue Variable `!ERROR` für die Folgeverarbeitung `DEFAULT-PROCESSING` (S. 72) und `SUBJECT-PROCESSING` (S. 78).
- Mai 2008 Neues Kommando `WARNINGON` (S. 39) zur Definition eines Prozeßschalters bei Ausgabe von Warn-Meldungen.
- Sept. 2007 Neue Option `NCTRL` zum Operanden `MODE` der Anweisung `PDF-LAYOUT` (S. 54): Die Auswertung der Steuerzeichen `X'11'` bis `X'1F'` im Text kann damit ausgeschaltet werden.
- August 2007 Die Anwendungen `MAILS`, `MAILR` einschl. PDF-Generierung stehen nun auch für das IBM-Betriebssystem `z/OS` zur Verfügung.
- Juli 2007 Neue Anweisung `SMTPPORT` (S. 60), falls ein vom Standard-Port 25 abweichender Port benutzt werden soll.  
Neue Anweisung `POP3PORT` (S. 63), falls ein vom Standard-Port 110 abweichender Port benutzt werden soll.  
Die Anweisung `DECRYPTKEY` (S. 71) kann nun pro Absender-Angabe bzw. für eine Gruppe von Absendern angegeben werden.  
Für das Default-Processing (S. 72) und das Subject-Processing (S. 78) wurden die Parameter `ATTACHFN1` bis `ATTACHFN64` und `ATTACHSFN1` bis `ATTACHSFN64` eingeführt. Damit werden sowohl die Dateinamen des Sendesystems als auch die neuen Dateinamen auf dem Empfangssystem an die Prozedur weitergegeben.  
Neue Anweisung `AUTH` (S. 71) für die verschlüsselte Übertragung von User und Passwort beim Empfangen.

# Änderungsprotokoll

---

- Neue Parameter `AUTH` zur Anweisung `SERVER` (S. 59) für die verschlüsselte Übertragung von User und Passwort beim Senden.
- März 2007 Bei der PDF-Generierung kann neben den Hintergrundbildern auch ein Wasserzeichen angegeben und die automatische Seitenanpassung eingestellt werden. Die Anweisung `PDF-LAYOUT` (S. 54) wurde entsprechend erweitert.
- Dezember 2006 Bei der PDF-Generierung werden auch Vorschübe auf Kanal 2-11 berücksichtigt. Die Option `MODE` zur Anweisung `PDF-LAYOUT` (S. 54) wurde um die Angaben zum Kanalsvorschub erweitert.
- Juli 2006 Die Anzahl der maximal möglichen Attachments (S. 21) wurde auf 1.024 erweitert.
- Die max. Satzlänge der Datensätze mit den Mail-Informationen zwischen den Anweisungen `MSGBEG` (S. 28) und `MSGEND` bzw. der Datei `MSGFILE` (S. 29) wurde auf 32.767 erweitert.
- Juni 2006 Anweisung `OPEN-ZIP` (S. 31): Mit der Option `CRYPTKEY` kann die erzeugte ZIP-Datei mit `OPGCRYPT` verschlüsselt werden.
- Mai 2006 Anweisungen `ATTACH` (S. 21) und `MSGFILE` (S. 29): Es können nun auch Bibliothekselemente mit der Syntax `bibliothek(typ/element[/version])` und ZIP-Elemente (`*zip:zip-file(element)`) angegeben werden, z.B. `ATTACH LIB.SRC(S/*) ->` alle Typ S Elemente.
- März 2006 Anweisung `SERVER` (S. 46): Falls zum Empfangen und Senden verschiedene Server verwendet werden sollen, können im Abschnitt `<RECEIVE>` ein oder mehrere Server (S. 64) eingetragen werden, die nur für das Empfangen von Mails benutzt werden sollen. Sind im Abschnitt `<RECEIVE>` keine Anweisungen `SERVER` enthalten, gelten zum Empfangen die gleichen Server, wie im Abschnitt `<GLOBAL>`.
- Falls mehrere Server eingetragen sind, wird beim Senden nur der erste verfügbare Server verwendet. Dadurch ist es möglich, alternative Mailserver anzugeben, die nur benutzt werden, wenn die anderen vorher definierten Server nicht ansprechbar sind. Beim Empfangen von Mails werden alle Server abgefragt.
- Anweisung `SERVER` beim Empfangen: Falls in der Parameterdatei im Abschnitt `<GLOBAL>` mehrere `SERVER`-Anweisungen enthalten waren, so wurde bisher immer nur der zuletzt angegebene Server benutzt. Ab Version 3.66 werden standardmäßig alle angegebenen Server abgefragt. Falls dies unerwünscht ist, so kann über eine `SERVER` Anweisung im Abschnitt `<RECEIVE>` dieses Verhalten abgeändert werden (hier dann den vorher letzten `SERVER` eintragen).
- Januar 2006 Die Option `CODE` zur Anweisung `PDF-LAYOUT` (S. 54) wurde um die Variante `DK` (Dänische Sonderzeichen) erweitert.
- Januar 2005 Die Anweisungen `ATTACH-APPLICATION` (S. 61) und `RECORD-LENGTH` (S. 63) können nun auch im Abschnitt `USER` `username` angegeben werden. Dadurch ist es möglich, Spezialfälle über ein bestimmtes Postfach zu verarbeiten. Außerdem wurden bei diesen beiden Anweisungen zusätzliche Parameter vorgesehen: Satzlänge und Format "F" (Daten mit fester Satzlänge).
- Die Anweisung `PDF-LAYOUT` (S. 54) kann auch über `SYSDTA` eingegeben werden. Zusätzliche Funktionen für die PDF-Generierung (Option `MODE`): Es können auch druckaufbereitete Dateien mit ASA-Steuerzeichen verarbeitet werden und es ist die Auswahl eines Spaltenbereichs, in dem die Druckdaten enthalten sind, möglich.

# Änderungsprotokoll

---

- Juli 2004 Bei der Anweisung PDF-LAYOUT (S. 54) kann beim Parameter PAGE neben den vordefinierten Seitengrößen A4, A4Q, A5 und A5Q auch die Größe in Millimeter angegeben werden.
- Juni 2004 Die Auswahlmöglichkeiten von zulässigen Attachments in der Anweisung ATTACHNAME (S. 65) wurde um die zusätzliche Angabe einer User-ID und der Variante "\$\*" als Platzhalter für alle User-ID's erweitert.
- April 2004 Beliebige viele Dateien können in ein ZIP-Archiv aufgenommen werden und dann als ein Attachment versandt werden. Dazu wurden die Anweisungen OPEN-ZIP (S. 31) und CLOSE-ZIP (S. 25) eingeführt. Für ein Mail können auch mehrere ZIP-Archive erzeugt werden.
- Oktober 2003 Die Option MODE=STD bei der Anweisung ATTACH (S. 21) bzw. ATTACH ohne MODE-Angabe bewirkt, daß PAM-Dateien sowie PDF- und OPGCRYPT-Dateien binär versandt werden. Bisher wurden in diesem Fall die Daten immer im Textmodus versandt.
- September 2003 Bei der Erzeugung von PDF-Dateien können nun auch Hintergrundbilder (Wasserzeichen, Formulare, Logos) eingeblendet werden. Die Hintergrundbilder können mit dem PDF-Layout (S. 54) in der Parameterdatei definiert werden.
- August 2003 Einführung einer Logging-Datei (S. 20) zur Protokollierung aller gesendeten Mails. Die Logging-Datei wird von einem Servertask geschrieben. Sobald der Servertask gestartet ist (siehe Prozedur J/MAILLOG in der MAILLIB), werden alle gesendeten Mails automatisch protokolliert. In der Parameterdatei ist keine Ergänzung notwendig.
- Mit der Anweisung ABORT=TO|CC|FILE|RESTRICT (S. 47) im Abschnitt <SEND> kann gesteuert werden, ob bei bestimmten Fehlern das Senden abgebrochen oder nur eine Warnung ausgegeben werden soll.
- Das Unterprogramme MAILUP (S. 83) liefert den Rückkehrcode 4 für Warnungen und 8 für Fehler, die zum Abbruch führen, zurück. Mit der Anweisung ABORT kann festgelegt werden, ob bestimmte Fehlersituationen als Warnung oder als Fehler behandelt werden sollen. Außerdem kann beim Aufruf ein Bereich übergeben werden, in dem die Fehlermeldungen zurückgegeben werden.
- Juni 2003 Neue Anweisungen CONTENT-DESCRIPTION=Y|N (S. 52) und EMPTY-SUBJECT=Y|N (S. 53) im Abschnitt <SEND> für Sonderfälle.
- Januar 2003 In der Parameterdatei kann die Anweisung SERVER (S. 46) mehrmals vorkommen. Die Server werden in der angegebenen Reihenfolge geprüft, wobei der erste gültige Server benutzt wird. Dadurch ist es möglich, alternative Mailserver anzugeben, die nur benutzt werden, wenn die anderen vorher definierten Server nicht ansprechbar sind.
- Dezember 2002 Optimierung der automatischen Anpassung der Seitenbreite bei der Erzeugung von PDF-Dateien mit MODE=PDF (S. 21).
- November 2002 Die Komprimierung von verschlüsselten Attachments (siehe Option CRYPTKEY zur Anweisung ATTACH (S. 21)) wurde optimiert: Die Dateigröße von verschlüsselten Attachments wurde um 30 % reduziert.

- Oktober 2002 Erzeugen von PDF-Dateien und versenden als Attachment:  
Die Optionen (z.B. Schriftart, Schriftgröße, Seitengröße, Zeilenanzahl) für die Konvertierung können in der Parameterdatei für verschiedene Layouts definiert werden. Bei der ATTACH-Anweisung wird nur der Name des Layouts angegeben.  
- Parameter `MODE=PDF` . . . zur Anweisung ATTACH (S. 21).  
- Anweisung `PDF-LAYOUT` im Abschnitt `<SEND>` (S. 54).
- August 2002 Mit der neuen Option `OPGC` für die Anweisung `TARGETNAME` (S. 60) kann bestimmt werden, ob bei verschlüsselten bzw. komprimierten Dateien die Endung `".OPGC"` an den Dateinamen angehängt wird.  
  
Neue Anweisung `UNCOMPRESS=Y|N` (S. 80): Damit kann bestimmt werden, ob beim Empfang eines komprimierten Attachments automatisch entkomprimiert werden soll.  
  
Die `FNAM`-Datei (S. 75) wurde um das Format des Attachments (`COMP`, `CRYPT`) vor und ev. nach einem automatischen Entschlüsseln ergänzt.
- Juni 2002 Neue Anweisung `TRTAB-ADDR=Y|N` (S. 47): Damit kann gesteuert werden, ob die Mail-Adressen mit der durch die `TRTAB`-Anweisungen geänderten Translatetabelle oder mit der Standard-Translate-Tabelle von EBCDIC nach ISO-8859 und umgekehrt übersetzt werden sollen.
- Mai 2002 Wie bei der Anweisung `MSGFILE` kann nun auch bei der Anweisung ATTACH (S. 21) der Datentyp mit der Option `TYPE=content-type` bestimmt werden. Dies ist notwendig, weil manche MAIL-Clients z.B. nur dann HTML-Text anzeigen, wenn als Content-Type `text/html` angegeben ist.
- März 2002 Verschlüsseln und Entschlüsseln von Attachments:  
- Anweisung `CRYPTKEY` (S. 27) im Abschnitt `<SEND>` oder über `SYSDTA`  
- Anweisung `DECRYPTKEY` (S. 71) im Abschnitt `<USER-username>`  
- Parameter `CRYPTKEY` (S. 21) zur Anweisung ATTACH  
- Weitere Hinweise zum Verschlüsselungs-Verfahren siehe Anlage 2 (S. 95).  
  
Komprimierung von Attachments: Option `COMP` zur Anweisung ATTACH (S. 21).
- November 2001 Neues Format `BIN` (S. 21) für Attachments beim Versenden von E-Mail's.
- Oktober 2001 Mit der neuen Anweisung `FNAMFILE` (S. 75) im Abschnitt `<USER>` kann der Original-Dateiname von Attachments in der Datei `mail-id.FNAM` gespeichert werden.  
  
In Abhängigkeit der Anweisung `DELETE=N` (S. 74) im Abschnitt `<USER>` können die E-Mail's auf dem Server gespeichert bleiben (bisher werden sie nach dem Abholen gelöscht). Siehe auch neue Anweisung `INDEXFILE` (S. 62).  
  
Mit der Anweisung `CYCLE-MSG` (S. 62) im Abschnitt `<RECEIVE>` können die Aktivitäten von MAILR auf `SYSOUT` protokolliert werden.  
  
Mit der Anweisung `ATTACH-APPLICATION` (S. 61) im Abschnitt `<RECEIVE>` kann das Format einer Attachment-Datei im BS2000 bestimmt werden, die mit dem Content-Type "application/...." empfangen wird.  
  
Mit der Anweisung `RECORD-LENGTH` (S. 63) im Abschnitt `<RECEIVE>` kann die maximale Satzlänge von Textdateien festgelegt werden.
- Mai 2001 Neue Unterprogramm-Schnittstelle zum Empfangen von E-Mail's für das Programm MAILR (S. 86)

- April 2001 Mail-Texte können auch im HTML-Format versandt werden. Damit ist es möglich, die Gestaltungsmöglichkeiten des HTML-Standards, wie z.B. Schriftgröße, Schriftart, Farbe usw., zu nutzen. Dazu muß der Datentyp `text/html` oder `<font ...>` bei den Anweisungen `MSGFILE` (S. 29) und `MSGBEG` (S. 28) angegeben werden.
- Mit der neuen Anweisung `HTMLFILE` (S. 76) können auch Daten mit dem Content-Type `text/html` vom Mail-Receiver gespeichert werden. Bisher wurden diese Daten ignoriert.
- März 2001 Die Daten der E-Mails werden bisher von MAILR für jedes einzelne Mail in eine eigene Datei gespeichert. Nun können die Daten aller empfangenen E-Mails in eine Datei, getrennt durch Header-Sätze, gespeichert werden. Die Anweisungen `TEXTFILE` (S. 80), `MAILFILE` (S. 76) und `ORIGINAL` (S. 77) wurden entsprechend erweitert.
- Mit der Anweisung `EMPTY-RECORD` (S. 74) im Abschnitt `<USER-username>` kann die Behandlung von Leersätzen beim Empfangen von E-Mails gesteuert werden.
- Durch Verwendung einer IMON-Registrierung (S. 15) kann eine eindeutige Parameterdatei und Nachladebibliothek festgelegt werden. Dadurch kann die Verwendung von nicht genehmigten Parameter-Dateien ausgeschlossen werden.
- Über 2 Jobvariable kann das Programm MAILR (Receiver) gesteuert werden. Mit der `REQUEST-JV` (S. 63) kann das Programm angehalten, aktiviert oder beendet werden. In der `STATUS-JV` (S. 64) wird der Status `ACTIVE`, `HOLD` oder `TERM` zurückgegeben. Die Namen der JV können in der Parameterdatei festgelegt werden.
- Januar 2001 In der Parameterdatei wurde der neue Abschnitt `<RESTRICT $user-id>` (S. 64) eingeführt. In diesem Abschnitt, können für jede BS2000-User-ID Beschränkungen bezüglich der Absender-Angaben (`FROM`-Anweisung), der Empfänger-Angaben (`TO`-Anweisung) und der zulässigen Namen und Größe für Attachments definiert werden.
- August 2000 Neues Kommando `RR` (Return to Recipient, siehe S. 37) zur Angabe einer E-Mail-Adresse, an die eine Bestätigung versandt werden soll, wenn die Nachricht vom Empfänger geöffnet wurde.
- Die Uhrzeit im Feld "Date" der gesendeten E-Mails wurde um die Zeitzone ergänzt. Ohne diese Angabe wird die Zeit vom Mailserver teilweise als UTC (Universal Time Coordinate oder Greenwich-Zeit) interpretiert und in die lokale Uhrzeit umgerechnet bzw. als fehlerhaft abgewiesen.
- Juli 2000 Neues Kommando `REPLY` (S. 37) zur Angabe einer E-Mail-Adresse für die Antwort.
- Februar 2000 Die globale Parameterdatei (S. 44) kann auch über ein BS2000 File Kommando zugewiesen werden, falls die Unterprogramm-Schnittstelle verwendet wird..
- Die CFS-Prozedur `EMAIL` (S. 19), die bisher nur für die User-Actioncodes `%EMT` und `%EMA` verwendet wurde, kann nun auch mit dem CFS-Kommando `DO` aufgerufen werden. Durch die Eingabe von `"ON&"` oder `"ONX"` im Feld Attachment können alle mit X ausgewählten Dateien bzw. alle Dateien einer Dateienliste als Attachment zu einem E-Mail versandt werden.
- Januar 2000 Neben Standard-SAM-Dateien können nun auch alle anderen Dateiformate, wie ISAM, feste Satzlänge, PAM- und PLAM-Dateien versandt und empfangen werden. Bestimmte Formate, wie z.B. PLAM sind in der Regel nur zum Empfang durch MAILR im BS2000 geeignet. Werden die Dateien vom Programm MAILR empfangen, so wird immer das Originalformat (`FCBTYPE`, `BLKSIZE`, `RECSIZE`, `RECFORM`) wieder hergestellt. Das Dateiformat kann mit der Option `MODE` der Anweisung `ATTACH` (S. 21) beeinflusst werden.
- Neue Optionen `*SAME`, `MODE` und `ISAM` zur Anweisung `ATTACH` (S. 21).

Neue Anweisung `ISAMCODE` zur globalen Einstellung der Option für die Behandlung von ISAM-Dateien.

Neue Anweisung `TARGETNAME` zur globalen Einstellung für die Bildung des Dateinamens auf dem Zielsystem.

Dez. 1999      Version 2.0

Neues Programm MAILR zum Empfangen von E-Mails. Sie können in Abhängigkeit des Benutzernamens und des Inhalts von "Subject" eine Folgeverarbeitung definieren. Einzelheiten finden Sie unter Kapitel 5 Programm MAILR (S. 40) und Kapitel 6 Parameterdatei (S. 44). Für dieses Programm muß eine eigene Lizenz erworben werden. Ohne Lizenz läuft das Programm einen Monat als Testversion.

Die Translate-Tabellen wurden in die Parameterdatei übernommen (Abschnitte `<TRTAB_SEND>` und `<TRTAB_RECEIVE>`). Als Standardtabellen werden nun die 8-Bit-Tabellen EBCDIC.DF.04.1 und ISO 8859-1 verwendet. Dadurch ist es möglich, auch nicht abdruckbare Zeichen zu versenden und zu empfangen. Bisher wurden die Code-Tabellen EBCDIC.DF.03 und US-ASCII verwendet. Nicht abdruckbare Zeichen und Zeichen > 128 wurden beim Versenden in das Zeichen "Leerstelle" übersetzt. Der alte Modus kann durch die Anweisung `TRTAB` bzw. durch Löschen des Abschnitts `<TRTAB_SEND>` eingestellt werden.

Die Parameterdatei wurde in Abschnitte gegliedert. Der Abschnittsname ist in spitze Klammern ( `<abschnittsname>` ) eingeschlossen. Für alle Anweisungen über `SYSDATA` und in der Parameterdatei wird die einheitliche Syntax `anweisung=param` verwendet. Die alte Parameterdatei ohne Abschnitte und mit der Syntax `anweisung param` wird vom Programm MAILS weiterhin richtig verarbeitet.

Neues Kommando `ATTACHSIZE` (S. 48) zur Begrenzung der Größe der zu versendenden Dateien.

Neues Kommando `ATTACHNAME` (S. 48) zur Einschränkung der zu versendenden Dateien.

Sept. 1999    Neues Kommando `LOCK` (S. 54) zum Sperren von bestimmten Kommandos durch den Administrator.

Neues Kommando `ERRORON` (S. 28) zur Definition eines Benutzerschalters im Fehlerfall.

März 1999    Einführung der globalen und privaten Parameterdatei `MAIL.PAR` (S. 44).

Mai 1999     Neues Kommando `DOMAINADD` (S. 28) zur automatischen Ergänzung der Internet-Adressen.

User-Actioncode `%EMT` (S. 15) und `%EMA` zum Aufruf von MAIL für eine Datei in der CFS-Dateienliste.

Juli 1999    Neue Unterprogramm-Schnittstelle `MAILUP` (S. 83).



**Inhaltsverzeichnis**

<b>1</b>	<b>ALLGEMEINE VEREINBARUNGEN .....</b>	<b>12</b>
<b>2</b>	<b>KURZBESCHREIBUNG.....</b>	<b>13</b>
	Senden von E-Mails .....	13
	Empfangen von E-Mails .....	13
	Systemvoraussetzungen .....	13
<b>3</b>	<b>INSTALLATION / AUSGELIEFERTE DATEIEN.....</b>	<b>14</b>
<b>4</b>	<b>PROGRAMM MAIL (SENDEN VON E-MAILS).....</b>	<b>18</b>
	Kurzbeschreibung .....	18
	Musterprozedur DO.MAIL .....	18
	CFS-Prozedur EMAIL .....	19
	Protokollierung der versandten Mails .....	20
	Kommandos .....	21
	ATTACH .....	21
	BCC .....	25
	CC .....	25
	CLOSE-ZIP .....	25
	CODEFILE .....	26
	CRYPTKEY .....	27
	DOMAIN .....	27
	DOMAINADD .....	28
	END .....	28
	ERRORON .....	28
	FROM .....	28
	MSGBEG .....	28
	MSGEND .....	29
	MSGFILE .....	29
	MSGLOG .....	31
	OPEN-ZIP .....	31
	PDF-LAYOUT .....	33
	PRIORITY .....	37
	REPLY .....	37
	RR .....	37
	SEND .....	37
	SERVER .....	37
	SUBJECT .....	38
	TO .....	38
	TRTAB .....	38
	WARNINGON .....	39
<b>5</b>	<b>PROGRAMM MAILR (RECEIVE: EMPFANGEN VON E-MAILS).....</b>	<b>40</b>
	Kurzbeschreibung .....	40
	Mail-ID .....	40
	Erzeugte Dateien .....	40
	Dateiformate der Attachments .....	41
	Folgeberarbeitung .....	41
	Programmaufruf .....	42
	Programmbeendigung .....	43

<b>6</b>	<b>PARAMETERDATEI MAIL.PAR</b>	<b>44</b>
	Allgemeines	44
	*                   Kommentarzeile	45
	Abschnitt <GLOBAL>	45
	DOMAIN            Domäne	45
	ERRORON           Fehlerschalter definieren	45
	LICENSE           Lizenzangaben	46
	MSGLOG            Logging einschalten	46
	SERVER            Mail-Server	46
	TRTAB-ADDR        TRTAB für Mail-Adressen verwenden	47
	WARNINGON        Schalter für Warnungen definieren	47
	Abschnitt <SEND>	47
	ABORT            Unvollständige Mails abbrechen	47
	ATTACHSIZE        Dateigröße begrenzen	48
	ATTACHNAME        zulässige Dateien	48
	CODEFILE          Datei mit User-Codes für PDF-Generierung	51
	CONTENT-DESCRIPTION    Dateiformat	52
	CRYPTKEY          Paßwort für Verschlüsselung	52
	DOMAINADD        Domain-Name ergänzen	53
	EMPTY-SUBJECT     Leere Betreffzeile	53
	FROM             Absender	53
	ISAMMODE         Behandlung von Isam-Dateien	53
	LOCK             Sperrungen von Kommandos	54
	PDF-LAYOUT        Optionen für PDF-Generierung	54
	PRIORITY         Priorität	58
	REPLY            Antwort-Adresse	59
	RR                Return to Recipient	59
	SERVER            Mail-Server	59
	SMTPPORT         Port zum Senden	60
	SYSDTA-Anweisungen    Programm MAIL	60
	SUBJECT          Betrifft-Angabe	60
	TARGETNAME        Dateiname im Zielsystem	60
	Abschnitt <RECEIVE>	61
	ATTACH-APPLICATIOON    Format Attachment	61
	CYCLE             Zeitintervall für den Mailbox-Check	62
	CYCLE-MSG          Protokollierung Mailbox-Check	62
	INDEXFILE         Dateiname für Index-File	62
	POP3PORT         Port zum Empfangen	63
	RECORD-LENGTH        Format Attachment	63
	REQUEST-JV         JV für Nachrichten an MAILR	63
	SERVER            Mail-Server	64
	STATUS-JV         JV für Status von MAILR	64
	Abschnitt <RESTRICT \$user-id>	64
	ATTACHNAME        zulässige Dateien	65
	ATTACHSIZE        zulässige Dateigröße	67
	FROM             zulässige FROM-Adressen	68
	TO                zulässige Mail-Empfänger	68
	Abschnitt <USER username>	69
	ATTACHMENT        Attachment-Dateien erstellen	69
	ATTACH-APPLICATIOON    Format Attachment	70
	AUTH             Verschlüsselung von User und Passwort	71
	DECRYPTKEY         Paßwort für Entschlüsselung	71
	DEFAULT-PROCESSING    Standard-Folgeverarbeitung	72
	DELETE            E-Mail auf Server löschen	74
	EMPTY-RECORD       Behandlung von Leersätzen	74
	FNAMFILE         Dateinamen der Attachments	75

HTMLFILE	Datei mit Mail-Nachricht im HTML-Format erstellen.....	76
MAILFILE	Datei mit Mail-Nachricht erstellen.....	76
ORIGINAL	Datei mit Original-Nachricht erstellen.....	77
PASS	Passwort für den User.....	77
POP3USER	POP3 Benutzer für Anmeldung.....	77
PREFIX	Prefix für die Dateinamen.....	77
RECORD-LENGTH	Format Attachment.....	78
SUBJECT-PROCESSING	Subject-Folgeverarbeitung.....	78
TEXTFILE	Datei mit Mail-Text erstellen.....	80
UNCOMPRESS	Automatisch entkomprimieren.....	80
Abschnitt <TRTAB-RECEIVE>..... 81		
Translate-Anweisung	ANSI → EBCDIC.....	81
Abschnitt <TRTAB-SEND>..... 81		
Translate-Anweisung	EBCDIC → ANSI.....	81
<b>7</b>	<b>UNTERPROGRAMM-SCHNITTSTELLE MAILUP.....</b>	<b>83</b>
	MAILUP (Senden).....	83
	MAILUPR (Empfangen).....	86
<b>A</b>	<b>ANLAGE1: CODE-UMWANDLUNG ZWISCHEN EBCDIC UND ASCII.....</b>	<b>87</b>
	Code-Umwandlung.....	87
	7-bit-Code.....	87
	Erweiterte Zeichensätze.....	87
	Aktivierung der 8-bit-Umgebung.....	88
	Translate-Tabelle EBCDIC → ASCII.....	89
	Translate-Tabelle ASCII → EBCDIC.....	90
	Code-Tabelle EBCDIC.DF.03 (CCSN: EDF03IRV).....	91
	Code-Tabelle EBCDIC.DF.04.1 (CCSN: EDF041).....	92
	Code-Tabelle EBCDIC.DF.04-DRV (CCSN: EDF04DR).....	93
	Code-Tabelle ASCII ISO-8859-1.....	94
<b>A2</b>	<b>ANLAGE 2: VERSCHLÜSSELUNGS DER ATTACHMENTS.....</b>	<b>95</b>
	Verschlüsseln im BS2000.....	95
	Verschlüsseln beim Versenden vom PC.....	95
	Entschlüsseln im BS2000.....	96
	Entschlüsseln beim Empfang im Windows.....	96
	SDF-Kommando OPGCRYPT-FILE.....	96
	Optionen Programm OPGCRYPT.....	98
	Algorithmus für die Verschlüsselung.....	104
<b>S</b>	<b>STICHWORTVERZEICHNIS.....</b>	<b>105</b>

## 1 Allgemeine Vereinbarungen

Im vorliegenden Buch werden bestimmte Zeichen (sogenannte Metazeichen) zur Darstellung der verschiedenen Eingabemöglichkeiten verwendet.

Formale Darstellung	Erläuterung	Beispiel
XXXX YYYY	Worte in Großbuchstaben bezeichnen Kommandos, die in dieser Form eingegeben werden müssen. Die Eingabe kann aber in Kleinbuchstaben erfolgen.	FROM <a href="mailto:test@opg.de">test@opg.de</a>  Einzugeben ist: from <a href="mailto:test@opg.de">test@opg.de</a>
<i>kursiv</i>	kursive Kleinbuchstaben bezeichnen Variablen, die bei der Eingabe durch aktuelle Werte ersetzt werden müssen, d.h. ihr Inhalt kann von Fall zu Fall verschieden sein. Die Eingabe erfolgt in der Regel in Kleinbuchstaben, bei Bedarf, z.B. bei Dateinamen, Suchbegriffen in Groß- und Kleinbuchstaben.	ATTACH <i>bs2file</i> '  Einzugeben ist: attach dateil
[ ..... ]	Eckige Klammern schließen Wahleingaben ein. Diese können angegeben oder weggelassen werden. Steht bei Wahleingaben ein Komma innerhalb der - Klammer, so wird es nur bei Verwendung dieser Eingabe verlangt. Runde Klammern (..) müssen stets so eingegeben werden wie beschrieben.	ATTACH <i>bs2file</i> [ <i>file</i> ]  Einzugeben ist: attach file oder attach file, <i>filepc</i>
<u>AAAAAA</u>	Die Unterstreichung hebt den Standardwert (Voreinstellung) hervor. Dies ist der Wert, der eingesetzt wird, wenn Sie keine Angabe gemacht haben.	
	Der senkrechte Strich trennt alternativ zu verwendende Eingaben.	

## 2 Kurzbeschreibung

### Programm MAIL (Senden)

Mit dem Programm MAIL können Sie aus dem BS2000/OSD E-Mails an beliebige Internet/Intranet-Benutzer versenden. Der Anwendungsschwerpunkt liegt beim Versenden der E-Mails über BS2000-Prozeduren. Sie können z.B. schwere Fehler in zeitkritischen Batchläufen über E-Mail an alle Personen melden, die sofort informiert werden müssen oder Druckdaten sofort über MAIL versenden statt auszu-drucken und per Post weiterzuleiten.

Der Ablauf wird durch Kommandos gesteuert, die über SYSDTA eingelesen werden. Der Inhalt der Nachricht wird wahlweise über SYSDTA oder in einer Datei bereitgestellt. Es können beliebig viele Empfänger, CC-Adressen, BCC-Adressen und Attachments angegeben werden. Bestimmte Kommandos können alternativ auch in einer zusätzlichen Parameterdatei bereitgestellt werden, die vorallem die Angaben enthält, die für alle Benutzer gleich sind.

Durch Einsatz der CFS-Musterprozedur EMAIL können mit den Action-Codes %EMA bzw. %EMT Dateien auch von der CFS-Dateienliste aus versandt werden. Siehe hierzu die Ausführungen im nachfolgenden Kapitel Installation Buchstabe g.

### Programm MAILR (Empfangen)

Mit dem Programm MAILR können Sie im BS2000/OSD E-Mails von einem im Netz erreichbaren MAIL-Server empfangen.

Nach dem Empfang eines E-Mails kann eine Folgeverarbeitung in Form eines beliebigen BS2000-Kommandos / MVS-Batch-Jobs angestoßen werden. Die Art der Folgeverarbeitung kann für jeden Benutzer und zusätzlich in Abhängigkeit vom Inhalt der SUBJECT-Angabe gesteuert werden. In der Regel wird die Folgeverarbeitung aus einem Kommando ENTER-PROC bestehen. Wichtige Informationen können über Parameter an die Prozedur weitergegeben werden, wie z.B. der Benutzername, die Dateinamen, der Inhalt von SUBJECT, die FROM-Adresse usw.

### Systemvoraussetzungen

- a) BS2000-Betriebssystem ab Version OSD1 und Subsystem SOCKETS ab Version 1.1 bzw. z/OS ab Version 1.4
- b) Das Subsystem SOCKETS muß im BS2000 geladen sein.

### 3 Installation

#### BS2000

- a) über E-Mail oder CD erhalten Sie ein selbstextrahierendes Archiv `MAIL.EXE`, das dieses Manual `MAIL.PDF` und eine verschlüsselte PLAM-Bibliothek `MAILLIB.EXP` mit folgenden Dateien enthält:

<code>C/MAIL</code>	Programm <code>MAIL</code>
<code>C/MAILR</code>	Programm <code>MAILR</code>
<code>L/MAILUP</code>	Modul <code>MAILUP</code> zum Nachladen von <code>MAILS</code> als Unterprogramm bzw. zum Laden über das SDF-Kommando <code>START-PROGRAM</code>
<code>L/MAILUPR</code>	Modul <code>MAILUP</code> zum Nachladen von <code>MAILR</code> als Unterprogramm bzw. zum Laden über das SDF-Kommando <code>START-PROGRAM</code>
<code>J/DO.MAIL</code>	Prozedur zum Starten des Programms
<code>J/EMAIL.SUCC</code>	Musterprozedur für die Folgeverarbeitung
<code>J/INSTALL.OPG-MAIL.IMON</code>	IMON-Installations-Prozedur
<code>S/EMAIL</code>	CFS-Prozedur für Actioncode <code>%EMA</code> und <code>%EMT</code> und zum Versenden von mehreren Attachments aus einer CFS-Dateienliste zu einem E-Mail.
<code>S/CFS.USERACT</code>	Musterdatei <code>CFS.USERACT</code>
<code>S/CALLMAIL.SRC</code>	Muster Assemblersource Aufruf Modul <code>MAILUP</code>
<code>S/MAIL.PAR</code>	globale Parameter und Lizenzinformationen
<code>S/SYSSII.OPG-MAIL.vers</code>	IMON-Installations-Datei

- b) Nach dem Entpacken die Datei `MAILLIB.EXP` mit FT im Textformat ins BS2000 übertragen

- c) Dekodieren der verschlüsselten Datei `MAILLIB.EXP`:  
CFS-Kommando `FILIMP MAILLIB.EXP, MAILLIB`

oder

BS2000-Kommando `/EXEC (FILIMP,$user-id.CFSLIB)`: über `SYSDTA` werden die Namen der Ein- und Ausgabedatei angefordert. Geben Sie hier ein:  
`MAILLIB.EXP, MAILLIB`

- d) Selektieren des Elements `S/MAIL.PAR` aus der Bibliothek `MAILLIB` als BS2000-Datei `MAIL.PAR`. Sie müssen mindestens folgende Anweisungen im Abschnitt `<GLOBAL>` anpassen:

In der Parameterdatei `MAIL.PAR` müssen Sie mindestens folgende Anweisungen im Abschnitt `<GLOBAL>` anpassen:

`SERVER:` BCAM-Name des Mailservers bzw. des Relays.

`DOMAIN:` Internet-Domain, z.B. `opg.de`

Eine ausführliche Beschreibung aller Anweisungen siehe S. 44.

- e) Anpassungen zum Senden von E-Mails (für Programm `MAIL`):

In der ausgelieferten Musterprozedur `DO.MAIL` müssen im Prozedurkopf die folgenden Parameter angepaßt werden:

`PROGRAMM:` `$user-id.MAIL` oder `lib=$user-id.MAILLIB, elem=MAIL`

- f) Anpassungen zum Empfangen von E-Mails (Für Programm `MAILR`):

- Neuen Benutzer auf dem Mailserver für `MAILR` einrichten.

- In der Parameterdatei `MAIL.PAR` einen neuen Abschnitt `<USER username>` aufnehmen. Musteranweisungen finden Sie als Bemerkungszeilen in der ausgelieferten Parameterdatei.

g) Anpassungen CFS zum Senden von E-Mails (Installation der Actioncodes `%EMT` und `%EMA`):

- Element `MAILLIB(S/EMAIL)` in die PLAM-Bibliothek `CFS.USERLIB.STD` (globale Userlib in der UserID, in der CFS installiert ist) oder `CFS.USERLIB` (private Userlib) übertragen (Variable Action `ONXADD CFS.USERLIB`) und ggf. Standard-Einstellungen (insbesondere Parameter `MAILLIB` und `FHSLIB`) anpassen. Diese CFS-Prozedur fordert die notwendigen Parameter über eine FHS-Maske an und ruft `MAILUP` als Unterprogramm auf.
- Die Daten aus dem Element `MAILLIB(S/CFS.USERACT)` in die bestehende Datei `CFS.USERACT` übernehmen bzw. das ganze Element als Datei selektieren, falls die Datei `CFS.USERACT` noch nicht vorhanden ist. Soll der Actioncode für alle Benutzer zur Verfügung stehen, muß die Datei `CFS.USERACT` in der UserID stehen, in der CFS installiert ist. Der Prozeduraufruf in der Datei `CFS.USERACT` muß noch an Ihre Umgebung angepaßt werden (insbesondere der Parameter "FR").

Danach können Sie mit den Actioncodes `%EMA` (Datei als Attachment versenden) bzw. `%EMT` (Datei als Text versenden) Dateien der CFS-Dateienliste als E-Mail versenden.

h) IMON-Registrierung und Erzeugen des SDF-Kommandos `OPGCRYPT-FILE`

Falls Sie aus Sicherheitsgründen die Verwendung einer geschützten globalen Parameterdatei erzwingen wollen, können Sie mit der IMON-Prozedur `INSTALL.OPG-MAIL.IMON` den vollständigen Dateinamen der Parameterdatei und der `MAILLIB` als logische Namen "Logical-ID" definieren. Die Programme `MAIL` und `MAILR` ermitteln aus den logischen Namen `MAILPAR` den vollständigen Dateinamen und verwenden dann immer diese Parameterdatei. Dadurch können Sie verhindern, daß die Sicherheitseinstellungen in der globalen Parameterdatei umgangen werden, indem die `MAILLIB` auf einer anderen Kennung installiert wird und folglich als globale Parameterdatei die `MAIL.PAR` von dieser Kennung verwendet würde.

Folgende Logische Namen (Logical-ID) werden unter der Installation-Unit `OPG-MAIL` registriert:

`MAILLIB`: enthält den vollständigen Dateinamen der `MAILLI`

`MAILPAR`: enthält den vollständigen Dateinamen der `MAIL.PAR`

Sie können diese logische Namen auch in Prozeduren verwenden, z.B.

```
/maillib = installation-path(logical-id='maillib',install-unit='opg-mail',default-path-name='$tsos.maillib')
/exec lib=&maillib, elem=mail
/start-prog *module (&maillib,mailup,prog-mode=*any,run-mode=*advanced)
```

Die Programme `MAIL` und `MAILR` laden LLM-Module nach. Zur Ermittlung des Namens der Nachladebibliothek wird der logische Namen `MAILLIB` benutzt, falls die Programme als BS2000-Dateien gespeichert sind. Dadurch ist es möglich, daß sich die Programm-Dateien und die `MAILLIB` in verschiedenen User-ID's befinden.

Der Installationspfad kann auch über Assembler-Makros ermittelt werden. Einzelheiten siehe Benutzerhandbuch IMON und Beispiel in dem Musterprogramm `S/CALLMAIL.SRC`.

i) Die Prozeduren `J/SYSSPR.*` für die SDF-Kommandos ggf. anpassen.

## z/OS

- a) über E-Mail oder CD erhalten Sie ein selbstextrahierendes Archiv `MAIL-ZOS.EXE`, das folgende Dateien enthält:

<code>FTP.SEND-TO-ZOS.BAT</code>	DOS-Prozedur mit FTP zum Senden der Daten an den Host
<code>OPGMAIL.MAIL.PAR</code>	Parameterdatei
<code>OPGMAIL.LOAD.XMIT</code>	PO-Bibliothek mit den Lademodulen <code>MAILS</code> und <code>MAILR</code>
<code>OPGMAIL.JCL.XMIT</code>	PO-Bibliothek mit Muster-Jobs
<code>OPGMAIL.CLIST.XMIT</code>	PO-Bibliothek mit REXX-Prozeduren <code>MAILS</code> und <code>MAILTO</code>
<code>OPGMAIL.PANELS.XMIT</code>	PO-Bibliothek mit Panel für REXX <code>MAILTO</code>

- b) Nach dem Entpacken mit der DOS-Prozedur `FTP.SEND.BAT` die Daten zum z/OS übertragen. Die Prozedur erzeugt zusätzlich eine JCL-Datei `OPGMAIL.RECEIVE.JCL` zur Wiederherstellung der `TRANSMIT`-Dateien.

Aufruf: `FTP.SEND.BAT host user password [HLQ] [lokales Verz.]`

*host* Name des Hosts  
*user* Login-Name fuer FTP-Anmeldung  
*password* Passwort fuer FTP-Anmeldung

wahlweise:

*HLQ* High level qualifier (Standard = `OPGMAIL`)

*lokales Verz.* Lokales Verzeichnis fuer FTP (Standard = `c:\programme\opg\mail`)

- c) Die Prozedur `OPGMAIL.RECEIVE.JCL` starten und damit die `XMIT` - Dateien zu `PO`-Bibliotheken konvertieren.
- d) Muster-Parameterdatei `OPGMAIL.MAIL.PAR` kopieren in `MAIL.PAR`. In der Parameterdatei `MAIL.PAR` müssen Sie mindestens folgende Anweisungen im Abschnitt `<GLOBAL>` anpassen:

`SERVER:` Name des Mailservers.  
`DOMAIN:` Internet-Domain, z.B. `opg.de`

Eine ausführliche Beschreibung aller Anweisungen siehe S. 44.

- e) Anpassungen zum Senden von E-Mails (für Programm `MAIL`):
- In der ausgelieferten Musterprozedur `OPGMAIL.JCL (MAILTO)` mind. folgende Zeilen anpassen  
SET-Anweisungen für Parameter `HLQ`
  - Die ausgelieferten REXX-Prozedur `OPGMAIL.CLIST (MAILTO)` und `OPGMAIL.CLIST (MAILTOM)` in eine andere `CLIST`-Library mit `CLIST`-Verkettung kopieren oder `OPGMAIL.CLIST` in die Verkettung aufnehmen und in der REXX `MAILTO` die Variable `MAILPATH` anpassen.
- f) Anpassungen zum Empfangen von E-Mails (Für Programm `MAILR`):
- Neuen Benutzer auf dem Mailserver für `MAILR` einrichten.
  - In der Parameterdatei `MAIL.PAR` einen neuen Abschnitt `<USER username>` aufnehmen bzw. Musteranweisungen für `<USER OPGTEST>` in der ausgelieferten Parameterdatei anpassen.



- In der ausgelieferten Musterprozeduren `OPGMAIL.JCL(MAILRCV)` und `OPGPGMAIL.JCL(FOLGET)` mind. folgende Zeilen anpassen:  
SET-Anweisungen für Parameter HLQ

### ASTI-Services:

Das Kommando `OPGCRYPT-FILE` wendet sich für die Ausführung der Funktionen an einen ASTI-Service. Dazu wird in den entsprechenden SDF-Prozeduren (`SYSSPR.*`) das Kommando `SEND-ORDER` aufgerufen, mit dem der Auftrag erteilt wird und die Parameter übergeben werden.

Da bei der Ausführung kein Programm geladen wird, können diese Kommandos auch in Programmen verwendet werden, z.B in:

der Kommandozeile des CFS,  
in CFS-Prozeduren,  
in CFS-User-Actioncodes,  
in Assembler-Programmen (Makro CMD),  
in C-Programmen (Funktion system).

Die Unterprogramm-Schnittstelle `OPGMAILS` verwendet den ASTI-Service `OPGMAILCLNT`. Der Vorteil liegt darin, daß nur der Verbindungsmodul `OPGMAILS` dazu gebunden werden muß. Probleme mit unterschiedlichen Socket- und CRTE-Versionen können nicht auftreten.

Der ASTI-Service muß vorher, in der Regel beim System-Start, gestartet werden. Der Service für `OPGCRYPT-FILE` kann mit dem Kommando `START-OPG-SERVICE` gestartet und mit dem Kommando `STOP-OPG-SERVICE` beendet werden. Für die Installation dieser SDF-Kommandos wird die Bibliothek `OPGCOM.LIB` benötigt. Beschreibung siehe S. 96 und Manual `OPGCOM`.

Der Service für `OPGMAILS` kann mit der Prozedur `MAILLIB(J/START.SERVICE)` gestartet und mit der Prozedur `MAILLIB(J/STOP.SERVICE)` beendet werden.

Das Subsystem ASTI ist immer vorhanden, weil es auch vom System benutzt wird.

### 4 Programm MAIL (Senden)

#### Kurzbeschreibung

Mit dem Programm MAIL können Sie aus dem BS2000/OSD bzw. z/OS E-Mails an beliebige Internet/Intranet-Benutzer versenden. Der Anwendungsschwerpunkt liegt beim Versenden der E-Mails über Prozeduren. Sie können z.B. schwere Fehler in zeitkritischen Batchläufen über E-Mail an alle Personen melden, die sofort informiert werden müssen oder Druckdaten sofort über MAIL versenden statt auszu-drucken und per Post weiterzuleiten.

Der Ablauf wird durch Kommandos gesteuert, die über SYSDTA eingelesen werden. Der Inhalt der Nachricht wird wahlweise über SYSDTA/SYSIN oder in einer Datei bereitgestellt. Es können beliebig viele Empfänger, CC-Adressen, BCC-Adressen und Attachments angegeben werden. Bestimmte Kommandos können alternativ auch in einer zusätzlichen Parameterdatei bereitgestellt werden, die vorallem die Angaben enthält, die für alle Benutzer gleich sind.

Durch Einsatz der Muster-Prozedur EMAIL können im BS2000 mit den Action-Codes %EMA bzw. %EMT Dateien auch von der CFS-Dateienliste aus versandt werden. Siehe hierzu die Ausführungen im nachfolgenden Kapitel 1, Installation Buchstabe g.

#### Prozedur DO.MAIL

In der Prozedur `DO.MAIL` können alle notwendigen Parameter für das Programm MAIL über Prozedurvariablen modifiziert werden. Im Prozedurkopf müssen die Mustereinträge der folgenden Parameter angepaßt werden:

LIB                    \$user-id.MAILLIB

Beim Aufruf der Prozedur müssen dann noch folgende Prozedur-Parameter versorgt werden:

- TO:                    Mailadresse des Empfängers (mehrere Empfänger durch Komma getrennt).
- SUBJECT:            "Betrifft" oder Bezeichnung des E-Mails
- MSG:                   Mail-Text  
                          und/oder
- MSGFILE:            Datei, die den Mail-Text enthält
- wahlweise:
- FROM:                 Absenderadresse (Std = [userid.host@domain](#))
- ATTACH:             Datei(en), die angehängt werden soll(en) (mehrere Dateien durch Semikolon getrennt)
- CC                    Empfänger für Kopien (Carbon Copy)
- BCC                  Verborgene Empfänger für Kopien (Blind Carbon Copy)
- TRTAB                abweichende Translateanweisungen – alter Modus - für Codeumw. EBCDIC --> ASCII

### CFS-Prozedur EMAIL

Mit der CFS-Prozedur `EMAIL` werden alle notwendigen Parameter für das Programm `MAIL` über eine FHS-Maske angefordert, anschließend wird das Programm `MAIL` über die Unterprogramm-Schnittstelle `UPMAIL` aufgerufen. Im Prozedurkopf müssen ev. die folgende Variablenzuweisungen angepaßt werden:

```
*&MAILLIB=' $TSOS.MAILLIB '
```

```
*&FHSLIB=' $TSOS.MFHSROUT '
```

Nach Aufruf der Prozedur mit dem CFS-Kommando `DO EMAIL` wird folgende Maske ausgegeben:

```

C800PG1
Datei Bearbeiten Terminal KIP Zusätze Einstellungen Anzeige Diagnose Hilfe
03.02.2000 17:53:16 HOST: C800PG1 TSN: 80IG USER-ID: TT08
E - M A I L
To:
From:
Subject:
Cc:
Bcc:
Attached:
Msgfile:
Message:
error: no recipient ! Return: K1
LTG TAST
```

Im Feld "Attached" sind folgende Eingaben möglich:

- Dateinamen Namen der Dateien, die als Attachment versandt werden sollen, wie bei der Anweisung `ATTACH (S. 21)`.
- ONX Nach dem Absenden der Maske werden Sie aufgefordert, die zu versendenden Dateien in der Dateienliste mit "X" zu markieren.
- ON& Alle Dateien der Dateienliste werden als Attachment versandt.

Hinweis:

Falls die LMS-Bibliothek `MAILLIB` nicht in der gleichen User-ID steht wie die LMS-Bibliothek `CFSLIB`, muß die globale Parameterdatei `MAIL.PAR` per File-Anweisung mit dem Linknamen `MAILPARG` zugewiesen (z.B. `/FILE $MAIL.MAI.PAR, LINK=MAILPARG`) oder der Name der `MAILLIB` mit `IMON` registriert werden (S. 15). Der Name der `MAILLIB` kann mit Hilfe des logischen Namens `MAILLIB` mit den entsprechenden `IMON`-Makros ermittelt werden (siehe Musterprogramm `S/CALLMAIL.SRC` in der `MAILLIB`).

### Protokollierung der versandten Mails (nur im BS2000)

Wahlweise können alle versandten Mails in einer Logging-Datei protokolliert werden. Die Logging-Datei wird von einem Servertask geschrieben, damit bei gleichzeitigem Mailversand von mehreren Tasks keine Probleme entstehen. Das Programm MAILS, das die Mails versendet, gibt die Daten für die Protokollierung über Intertaskkommunikation an den Servertask weiter. Der Servertask erstellt für jedes Mail einen Datensatz in der Logging-Datei.

Die Logging-Datei wird von dem Programm MAILLOG erstellt. Zum Starten kann die Prozedur J/MAILLOG in der MAILLIB verwendet werden. Über die Prozedurvariablen können der Dateinamen der Logging-Datei und der Name der Ladebibliothek für das Programm MAILLOG festgelegt werden. Sobald der Servertask gestartet ist, werden alle gesendeten Mails automatisch protokolliert. In der Parameterdatei ist keine Ergänzung notwendig.

### Kommandos

Die Kommandos werden über SYSDTA bzw. SYSIN eingelesen. Bestimmte Kommandos können auch in der Parameterdatei MAIL.PAR als Standard definiert werden und können dann in der SYSDTA-Datei entfallen. Dabei gilt folgende Reihenfolge der Verarbeitung:

1. Globale Parameterdatei MAIL.PAR
2. Private Parameterdatei MAIL.PAR
3. SYSDTA-Datei

Ist ein Kommando in mehreren Dateien vorhanden, so gilt jeweils das zuletzt verarbeitete Kommando, d.h. die Kommandos aus der SYSDTA-Datei überschreiben die Kommandos der beiden Parameterdateien. Die private Parameterdatei überschreibt die Kommandos aus der globalen Parameterdatei. Mit dem Kommando LOCK in der globalen Parameterdatei ist es möglich, bestimmte Kommandos in der privaten Parameterdatei und in der SYSDTA-Datei zu sperren. Die Kommandos können in beliebiger Reihenfolge angegeben werden und können klein oder groß geschrieben sein.

Die Kommandos sind wie folgt aufgebaut:

KOMMANDONAME [=parameter]

Aus Kompatibilitätsgründen kann auch noch das alte Format "KOMMANDONAME [parameter]" benutzt werden.

### Datei anhängen

**ATTACH=sendfile** [,targetfile | \*SAME] [,MODE=STD | TEXT | BS2 | TEXT-BIN | BIN | PDF | PDF-layout ] [,ISAM=Y|N] [,CRYPTKEY=Y | *cryptkey* ] [,COMP=Y] [,TYPE=content-type] [;sendfile.....]

Datei / Dateien, die mit dem E-Mail versandt werden sollen. Es können mehrere Dateien, durch Semikolon getrennt, angegeben werden. Diese Anweisung kann mehrmals angegeben werden. Max. sind 1.024 Attachments zulässig.

Nur im BS2000:

Es ist auch möglich, aus mehreren Dateien und / oder Bibliothekselementen ein temporäres ZIP-Archiv zu erzeugen und als ein Attachment zu versenden. Siehe hierzu die Anweisungen OPEN-ZIP (S. 31) und CLOSE-ZIP (S. 25).

Zur Erzeugung eines ZIP-Archivs sind folgende Anweisungen notwendig:

```
OPEN-ZIP zipfile
ATTACH .....
ev. weitere ATTACH-Anweisungen
CLOSE-ZIP
```

In diesem Fall sind folgende Parameter nicht zulässig:

```
MODE=BS2
ISAM=N
CRYPTKEY=Y | kryptkey
COMP=Y
TYPE=content-type
```

<i>sendfile</i>	<i>bs2file</i>   <i>lib (typ/elem[/vers])</i>   *ZIP:zip-archiv (elem)   <i>mvs-file</i>
<i>bs2file</i>	Voll- oder teilqualifizierter Dateiname, ggf. mit Cat-ID und User-ID in der Syntax des BS2000-Kommandos <code>FSTAT</code> .
<i>lib (typ/elem[/vers])</i> oder <i>lib (typ, elem[, vers])</i>	Name einer PLAM-Bibliothek, Elementtyp, Elementnamen und ggf. Version.
*ZIP:zip-archiv (elem)	Name eines ZIP-Archivs und der Elementname.  Es können auch die Jokerzeichen "*" und "/" für die Wildcard-Syntax verwendet werden. Das Zeichen * ersetzt eine beliebige, auch leere Zeichenfolge, das Zeichen "/" ersetzt genau ein Zeichen. Für Bibliothekselemente ist als Jokerzeichen für ein Zeichen das Zeichen "?" (statt "/") zu verwenden.  Endet der String mit einem Punkt (teilqualifizierter Dateiname), z.B. "ASSEMB.", werden alle Dateien ausgewählt, deren Name mit diesem String beginnt und noch mindestens ein Zeichen danach enthält.  Beginnt der Dateiname mit dem Zeichen "#", so werden alle temporären Dateien ausgewählt, die den weiteren Auswahlkriterien entsprechen.  Beispiele:
<code>DAT.</code>	alle Dateinamen, die mit <code>DAT.</code> beginnen
<code>LIB1 (*/*)</code>	alle Elemente der Plam-Bibliothek LIB1
<code>LIB1 (S/*)</code>	alle Elemente der Plam-Bibliothek LIB1 mit Typ S
<code>LIB1 (S/ELEM?)</code>	alle Elemente der Plam-Bibliothek LIB1 mit Typ S und Namen ELEM? (?=beliebiges Zeichen)
<code>LIB1 (S/ASS.*)</code>	alle Elemente der Plam-Bibliothek LIB1 mit Typ S und die mit "ASS." beginnen
<code>*ZIP:ZIP1 (*)</code>	alle Elemente des ZIP-Archivs ZIP1
<code>*ZIP:ZIP1 (TEST*)</code>	alle Elemente des ZIP-Archivs ZIP1, die mit "TEST" beginnen.
<code>LIB1 (S/*/100)</code>	alle Elemente der Plam-Bibliothek LIB1 mit Typ S und Version 100
<code>DAT*TEST</code>	alle Dateinamen, die mit <code>DAT</code> beginnen und <code>TEST</code> enden
<code>DAT//TEST*.SRC</code>	alle Dateinamen, die mit <code>DAT</code> beginnen, in der 6.-9. Stelle <code>TEST</code> enthalten und mit <code>.SRC</code> enden, z.B. <code>DAT01TESTX.SRC</code> .
<code>:TEST:\$MAIL.*</code>	alle Dateien der User-ID <code>TEST</code>
<code>#TEST*</code>	alle temporären Dateien, die mit <code>TEST</code> beginnen
<i>mvs-file</i>	<i>data-set</i>   <i>pds(member)</i>   'hlq.data-set'   'hlq.pds(member)'
<i>data-set</i>	Dateiname auf dem fernen Rechner ohne <code>HLQ</code> (high level qualifier). Vom Host-System wird der Dataset-Name automatisch um den <code>HLQ</code> ergänzt.

<i>pds(member)</i>	Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended) ohne HLQ. Vom Host-System wird der Member-Name automatisch um den HLQ ergänzt.
' <i>hlq.data-set</i> '	Vollqualifizierter Dateiname auf dem fernen Rechner. Aus diesem String wird für den Filetransfer der Dateiname ' <i>hlq.data-set</i> ' erzeugt.
' <i>hlq.pds(member)</i> '	Vollqualifiziertes Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended). Aus diesem String wird für den Filetransfer der Dateiname ' <i>hlq.pds(member)</i> ' erzeugt.
<i>targetfile</i>	Dateinamen auf dem Zielrechner. Wird dieser Name nicht angegeben, erhält die Datei auf dem Zielrechner den Namen <i>attn.txt</i> bzw. <i>attn.txt.opgc</i> (bei verschlüsselten Dateien) oder den Standardnamen, der mit der Anweisung <code>TARGETNAME</code> (S. 60) definiert wurde.
*SAME	Die Datei erhält auf dem Zielrechner den gleichen Namen wie im BS2000. Handelt es sich um ein Bibliothekselement, wird nur der Elementname verwendet. Dadurch kann es vorkommen, daß der Namen nicht eindeutig ist, z.B. bei gleichen Elementnamen mit unterschiedlichen Elementtyp oder gleichen Elementnamen mit unterschiedlichen Versionen.
MODE=STD	Standardmodus. Das Format der Daten wird automatisch bestimmt. PAM-Dateien, sowie PDF- und OPGCRYPT-Dateien werden mit MODE=BIN versandt. PDF- und OPGCRYPT-Dateien werden anhand des Inhalts der Daten, unabhängig vom Dateinamen, erkannt.  Alle anderen Dateien werden mit MODE=TEXT versandt. Diese Einstellung wird auch bei fehlender MODE-Angabe verwendet.
MODE=TEXT	Textmodus. Die Datei wird satz-stukturiert versandt, d.h. nach jedem logischen Satz bzw. beim PAM-Dateien nach jedem PAM-Block wird ein Satzendezeichen (X'0D0A') eingefügt. Die Daten werden von EBCDIC nach ASCII übersetzt. Bei PAM-Dateien ist in der Regel MODE=TEXT nicht sinnvoll.

Die Dateiattributen werden in dem MAIL-Header `Content-Description` mit der MAIL-Nachricht übermittelt. Wird eine Datei an einen anderen BS2000-Rechner versandt und vom Programm MAILR empfangen, wird die Datei mit den gleichen Dateiattributen wieder hergestellt. Allerdings ist dabei zu beachten, daß die Datei satz-strukturiert in ASCII versandt wird. Soweit in den Daten die Zeichenfolge `X'0D0A'` vorkommt oder die Send- und Empfangs-Systeme unterschiedliche Translate-Tabellen für die Umwandlung von EBDIC nach ASCII und umgekehrt verwenden, können auf dem Empfangssystem abweichende Daten ankommen. Durch die Angabe `MODE=BS2` werden diese Probleme umgangen. Wird die Nachricht von einem anderen MAIL-Client empfangen, werden diese Angaben ignoriert.

**MODE=TEXT-BIN** Die Datei wird satz-strukturiert versandt, jedoch nicht von EBCDIC auf ASCII umcodiert.

**MODE=BIN** Die Datei wird ohne Satzstruktur in EBCDIC versandt (wie Format BIN im FTP).

**MODE=BS2** Mit dieser Option kann jede Datei, unabhängig vom FCBTYPE und den sonstigen Dateiattributen auf einem anderen BS2000-Empfangssystem vom Mail-Client MAILR mit den Original-Dateiattributen erstellt werden. Die Datei wird als PAM-Datei im EBCDIC-Code (verschlüsselt mit BASE64) versandt. Die Dateiattributen werden zusätzlich in den Daten gespeichert. Dadurch ist es möglich, daß die Datei auf dem Empfangsrechner mit den Original-Dateiattributen wieder hergestellt werden kann, auch wenn sie zuerst an eine andere Plattform und von da aus erst zum BS2000-Rechner gesandt wird.

**MODE=PDF | PDF-layout** Aus der angegebenen Datei wird eine PDF-Datei erzeugt. Die Optionen für die Aufbereitung der Daten werden aus der Parameteranweisung `PDF-LAYOUT` mit dem entsprechenden Layout-Namen entnommen. Wird nur `MODE=PDF` angegeben, werden die Optionen aus der Parameteranweisung `PDF-LAYOUT=DEFAULT.....` entnommen. Beispiel: Bei `MODE=PDF-ASS` werden die Optionen der Parameteranweisung `PDF-LAYOUT=ASS, ...` verwendet.

Die PDF-Datei kann aus einer Datei mit und ohne Drucksteuerzeichen erzeugt werden. Neben den Drucksteuerzeichen im 1. Byte eines Datensatzes werden auch noch Steuerzeichen in den Daten ausgewertet, wie z.B. fett, kursiv, gesperrt usw. Weitere Einzelheiten siehe Anweisung `PDF-LAYOUT` als `SYSDTA`-Anweisung oder im Abschnitt `<SEND>`. (S. 54).

Die erzeugte PDF-Datei erhält den Suffix `.pdf`, damit im Windows automatisch das Programm Acrobat Reader zum Anzeigen verwendet wird.

**ISAM=Y|N** Behandlung des ISAM-Schlüssels bei `MODE=STD | TEXT-BIN`.

**Y** Der Isam-Schlüssel bleibt im Datensatz erhalten. Soweit die Datei direkt vom Programm MAILR im BS2000 empfangen wird, wird automatisch wieder eine ISAM-Datei mit den Original-Dateiattributen erstellt.

**N** Der Isam-Schlüssel wird aus dem Datensatz entfernt. Der FCBTYPE für die Empfangsdatei wird auf SAM geändert. Soweit die Datei direkt vom Programm MAILR im BS2000 empfangen wird, wird automatisch eine SAM-Datei erstellt. Die übrigen Dateiattributen bleiben erhalten.



Fehlt diese Option so gilt der Standardwert, der in der Parameterdatei mit der Anweisung `ISAMMODE` (S. 53) definiert ist.

<code>CRYPTKEY=Y cryptkey</code>	Attachment verschlüsseln:
<code>Y</code>	Die Datei wird mit dem Passwort aus einer vorhergehenden Anweisungen <code>CRYPTKEY</code> (S. 27) bzw. aus der <code>CRYPTKEY</code> -Anweisung der Parameterdatei verschlüsselt.
<code>cryptkey</code>	Passwort für die Verschlüsselung.
<code>COMP=Y</code>	Die Datei wird mit dem <code>OPGCRYPT</code> -Algorithmus ohne Verschlüsselung komprimiert. Zum Entkomprimieren kann das Programm <code>OPGCRYPT</code> bzw. <code>OPGCRYPTW</code> (für Windows und BS2000) benutzt werden.
<code>TYPE=content-type</code>	Format der Datei: Hier kann ein beliebiges Format eingegeben werden, das vom Mail-Client des Empfängers als Content-Type interpretiert werden kann. Natürlich müssen die nachfolgenden Daten auch diesem Format entsprechen. Das gebräuchlichste und relativ einfache Format ist <code>text/html</code> (es ist auch die Abkürzung <code>html</code> zulässig). Siehe hierzu auch das Beispiel zum Kommando <code>MSGBEG</code> (S. 28).

### zusätzliche Empfänger (verborgen)

**BCC=recipient[,recipient,...]** Mail-Adresse(n) für weitere Empfänger, die Kopien des E-Mails erhalten sollen. Die Empfänger-Liste wird nicht an die Mail-Adressaten weitergegeben. Mehrere E-Mail-Adressen werden durch Komma getrennt. Diese Anweisung kann mehrmals angegeben werden.

*recipient* Mail-Adresse des Empfängers, z.B. [Name@test.de](mailto:Name@test.de)

### zusätzliche Empfänger

**CC=recipient[,recipient,...]** Mail-Adresse(n) für weitere Empfänger, die Kopien des E-Mails erhalten sollen. Die Empfänger-Liste ist für jeden E-Mail-Adressat sichtbar. Mehrere E-Mail-Adressen werden durch Komma getrennt. Diese Anweisung kann mehrmals angegeben werden.

*recipient* Mail-Adresse des Empfängers, z.B. [Name@test.de](mailto:Name@test.de)

### ZIP-Archiv schließen und als Attachment versenden (nur BS2000)

**CLOSE-ZIP** Ein mit `OPEN-ZIP` (S. 31) erzeugtes ZIP-Archiv wird geschlossen und als Attachment versandt. Folgt nach der Anweisung `OPEN-ZIP` kein `CLOSE-ZIP` gilt das ZIP-Archiv bis zur nächsten Anweisung `OPEN-ZIP` oder bis zur Anweisung `SEND` bzw. `END`.

## Datei mit User-Codes

**CODEFILE=** *file* | *lib(typ,elem)*

Die User-Codes können zur Zeit nur für die PDF-Generierung in der Anweisung `PDF-LAYOUT` (S. 52), Parameter `CODE` benutzt werden. Für die Verschlüsselung des Mail-Inhalts und der sonstigen Attachments gilt die Translate-Tabelle `TRTAB-SEND` (S. 78).

Als Standard werden EBCDIC-Zeichen mit der Translate-Tabelle EDF041 to ISO8859-1 (S. 85) in "ASCII-Zeichen" umgewandelt. Intern sind drei verschiedene Varianten vordefiniert:

Hexa-Wert EBCDIC	FB	4F	FD	BB	BC	BD	FF	6A
INT: Internationale Zeichen:	{		}	[	\	]	~	^
DE: Deutsche Umlaute:	ä	ö	ü	Ä	Ö	Ü	ß	
DK Dänische Sonderzeichen:	æ	ø	å	Æ	Ø	Å	ü	Û

In der Codefile können zusätzlich bis zu 30 Code-Varianten definiert werden.

*file*

Dateiname der Codefile.

*lib(typ,elem)*

Bibliotheksname, Typ und Elementname.

In der Codefile können Translate-Tabellen von weiteren Code-Varianten definiert werden. Dabei wird als Grundlage immer die Translate-Tabelle EDF041 to ISO8859-1 (S. 85) verwendet. Nur davon abweichende Zeichen müssen in der Codefile angegeben werden. Aufgrund der Angaben in der Codefile wird eine Translate-Tabelle EBCDIC → ASCII und eine Translate-Tabelle ASCII → EBCDIC erzeugt.

Die Codefile ist wie folgt aufgebaut:

```
<TRTAB-codename>
```

```
eeaa [remark]
```

```
*remark
```

*codename* Name des Codes. Dieser Name ist in den entsprechenden Anweisungen anzugeben.

*eeaa* Translate-Anweisung. Die Reihenfolge der Translate-Anweisungen ist beliebig.

*ee* EBCDIC-Zeichen hexadezimal

*aa* ASCII-Zeichen hexadezimal

*remark* Bemerkung

### Beispiel Codefile:

```
* Translate-Tabelle fuer deutsche Umlaute und
* Internationale Zeichen lt. EDF04DR
<TRTAB-EDF04DR>
BBC4 [ = AE GROSS
FBE4 { = AE KLEIN
BCD6 \ = OE GROSS
4FF6 | = OE KLEIN
BDDC ] = UE GROSS
FDFC } = UE KLEIN
FFDF ~ = SS
C07B ù = {
D07D ¡ = }
E05C Û = \
CC5B ö = [
DC5C ü = ]
A17E ¯ = ~
437C ä = |
```

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Passwort für Verschlüsselung

**CRYPTKEY**=*password*

Definition eines Passwortes für die Verschlüsselung von Attachments. Dieses Passwort wird verwendet, wenn in einer nachfolgenden Anweisung ATTACH die Option "CRYPTKEY=Y" angegeben ist. Die Anweisung muß vor der entsprechenden ATTACH-Anweisung stehen.

*password*

Passwort für die Verschlüsselung. Das Passwort kann max. 56 Byte lang sein.

Weitere Hinweise zum Verschlüsselungs-Verfahren siehe Anlage 2 (S. 95).

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Internet-Domäne

**DOMAIN**=*domainname*

Internet-Domäne, z.B. opg.de

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <GLOBAL> angegeben werden.

### Domäne ergänzen

**DOMAINADD** =[Y|N]

Automatisches Ergänzen der Mail-Adressen um die Domäne.

Y Die Mail-Adressen FROM, TO, CC und BCC werden um den Domain-Namen einschl. Zeichen "@" ergänzt, falls die Adresse kein Zeichen "@" enthält. Es ist darauf zu achten, daß diese Anweisung vor den Anweisungen FROM, TO und BCC in der SYSDTA-Datei steht.

N Die Mail-Adressen werden nicht automatisch ergänzt.

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Daten senden

**END** oder **SEND**

Die Nachricht wird gesendet. Danach wird das Programm beendet.

### Fehlerschalter

**ERRORON**=*switch*[,RESET]

Falls während der Verarbeitung ein Fehler auftritt, wird der Prozeßschalter *switch* auf ON gesetzt. Fehlt dieses Kommando, wird der Prozeßschalter 31 benutzt. Siehe auch Anweisung WARNINGON (S. 39) für Fehlermeldungen.

*switch*

Prozeßschalter 0 – 31

RESET

Vor Beginn der Verarbeitung wird der angegebene Prozeßschalter gelöscht (SETSW *n*=OFF).

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <GLOBAL> angegeben werden.

### Eigene Mail-Adresse

**FROM**=*fromaddr*

Eigene Mailadresse, fehlt dieses Kommando, wird die Absender-Angabe aus der User-ID, dem Host-Namen und der eigenen Internet-Domäne gebildet, z.B. [USR1.HOST1@opg.de](mailto:USR1.HOST1@opg.de)

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Beginn Nachricht

**MSGBEG** [TYPE=*content-type* | *font* ]

Beginn E-Mail-Text. Nach diesem Kommando folgen beliebig viele Textzeilen mit der Nachricht. Die Textzeilen werden mit dem Kommando MSGEND abgeschlossen. Eine Textzeile kann bis zu 32.767 Bytes lang sein. In einer Textzeile können auch Zeilenendezeichen (X'0A') enthalten sein.

*content-type* Format der Nachricht: Hier kann ein beliebiges Format eingegeben werden, das vom Mail-Client des Empfängers als Content-Type interpretiert werden kann. Natürlich müssen die nachfolgenden Daten auch diesem Format entsprechen. Das gebräuchlichste und relativ einfache Format ist `text/html` (es ist auch die Abkürzung `html` zulässig).

*font* HTML Font-Angabe in der Form `<font ...>`  
Mit dieser Variante der Type-Angabe können einem nachfolgenden unformatierten Text Attribute für die Schriftart (Zeichensatz, Größe Farbe) zugewiesen werden. Der nachfolgende Text (bis MSGEND) wird um die notwendigen HTML-Anweisungen ergänzt.

Beispiel:

```
msgbeg type=html
<html><pre>
<font size="5" face="Arial" color="#FF0000">
<b>Zeile1 Arial fett, rot, große Schrift </b></font>
<font size="3" face="Courir" color="#000000">
<u>Zeile2 Courir unterstrichen, schwarz, normale Schrift
</u></font>
</pre></html>
msgend
```

```
msgbeg type=<font face="Arial" size="5" color="red">
Fehler!!! Job abgebrochen.
msgend
```

### Ende der Nachricht

### MSGEND

Ende E-Mail-Text. Der Mail-Text wird mit der Anweisung `MSGBEG` eingeleitet. Dazwischen folgen beliebig viele Textzeilen mit der Nachricht.

### Datei anhängen

**MSGFILE**=*msgfile* [*;msgfile...*] [*,TYPE=content-type | font* ]

Datei / Dateien, deren Inhalt als Nachricht versandt werden sollen. Es können mehrere Dateien, durch Semikolon getrennt, angegeben werden. Diese Anweisung kann mehrmals angegeben werden. Ein Datensatz der MSGFILE kann bis zu 32.767 Bytes lang sein.

*msgfile* *bs2file* | *lib* (*typ/elem*[*/vers*]) | \*ZIP:*zip-archiv* (*elem*) | *mvs-file*

*bs2file* Voll- oder teilqualifizierter Dateiname, ggf. mit Cat-ID und User-ID in der Syntax des BS2000-Kommandos `FSTAT`.

*lib* (*typ/elem*[*/vers*]) Name einer PLAM-Bibliothek, Elementtyp, Elementnamen und ggf. Version.

oder

*lib* (*typ, elem*[*, vers*])

\*ZIP:*zip-archiv* (*elem*) Name eines ZIP-Archivs und der Elementname.

Es können auch die Jokerzeichen "\*" und "/" für die Wildcard-Syntax verwendet werden. Das Zeichen \* ersetzt eine beliebige, auch leere Zeichenfolge, das Zeichen "/" ersetzt genau ein Zeichen. Für Bibliothekselemente ist als Jokerzeichen für ein Zeichen das Zeichen "?" (statt "/") zu verwenden.

Endet der String mit einem Punkt (teilqualifizierter Dateiname), z.B. "ASSEMB.", werden alle Dateien ausgewählt, deren Name mit diesem String beginnt und noch mindestens ein Zeichen danach enthält.

Beginnt der Dateiname mit dem Zeichen "#", so werden alle temporären Dateien ausgewählt, die den weiteren Auswahlkriterien entsprechen.

Beispiele:

DAT.	alle Dateinamen, die mit DAT. beginnen
LIB1 (*/*)	alle Elemente der Plam-Bibliothek LIB1
LIB1 (S/*)	alle Elemente der Plam-Bibliothek LIB1 mit Typ S
LIB1 (S/ELEM?)	alle Elemente der Plam-Bibliothek LIB1 mit Typ S und Namen ELEM? (?=beliebiges Zeichen)
LIB1 (S/ASS.*)	alle Elemente der Plam-Bibliothek LIB1 mit Typ S und die mit "ASS." beginnen
*ZIP:ZIP1 (*)	alle Elemente des ZIP-Archivs ZIP1
*ZIP:ZIP1 (TEST*)	alle Elemente des ZIP-Archivs ZIP1, die mit "TEST" beginnen.
LIB1 (S*/100)	alle Elemente der Plam-Bibliothek LIB1 mit Typ S und Version 100
DAT*TEST	alle Dateinamen, die mit DAT beginnen und TEST enden
DAT//TEST*.SRC	alle Dateinamen, die mit DAT beginnen, in der 6.-9. Stelle TEST enthalten und mit .SRC enden, z.B. DAT01TESTX.SRC.
:TEST:\$MAIL.*	alle Dateien der User-ID TEST
#TEST*	alle temporären Dateien, die mit TEST beginnen

*mvs-file*

*data-set | pds(member) | 'hlq.data-set' | 'hlq.pds(member)'*

*data-set*

Dateiname auf dem fernen Rechner ohne HLQ (high level qualifier). Vom Host-System wird der Dataset-Name automatisch um den HLQ ergänzt.

*pds(member)*

Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended) ohne HLQ. Vom Host-System wird der Member-Name automatisch um den HLQ ergänzt.

*'hlq.data-set'*

Vollqualifizierter Dateiname auf dem fernen Rechner. Aus diesem String wird für den Filetransfer der Dateiname *'hlq.data-set'* erzeugt.

*'hlq.pds(member)'*

Vollqualifiziertes Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended). Aus diesem String wird für den Filetransfer der Dateiname *'hlq.pds(member)'* erzeugt.

*content-type* Format der Nachricht: Hier kann ein beliebiges Format eingegeben werden, das vom Mail-Client des Empfängers als Content-Type interpretiert werden kann. Natürlich müssen die nachfolgenden Daten auch diesem Format entsprechen. Das gebräuchlichste und relativ einfache Format ist `text/html` (es ist auch die Abkürzung `html` zulässig). Siehe hierzu auch das Beispiel zum Kommando `msgbeg` (S. 28).

*font* HTML Font-Angabe in der Form `<font ...>`  
Mit dieser Variante der Type-Angabe können dem unformatierten Text dieser Datei Attribute für die Schriftart (Zeichensatz, Größe Farbe) zugewiesen werden. Die Daten werden um die notwendigen HTML-Anweisungen ergänzt.

Beispiele:

```
msgfile datei1,type=text/html
msgfile datei2,type=html
msgfile datei3,type=<font face="Arial" size="5" color="red">
msgfile datei4,type=<font face="Courir" size="3" color="black">
```

### Umfang der Protokollierung

**MSGLOG=N** | Y | L Umfang der Meldungen, die auf SYSOUT / SYSPRINT ausgegeben werden.

N Es werden nur Fehlermeldungen auf SYSOUT /SYSPRINT ausgegeben.

Y Außer den Fehlermeldungen werden verschiedene Aktionen protokolliert, wie z.B. die Verarbeitung der Parameterdateien und Attachments. Diese Option gilt als Standard bei fehlender MSGLOG-Anweisung.

L Für Diagnosezwecke werden zusätzlich alle Nachrichten, die mit dem Mail-server ausgetauscht werden, auf SYSOUT ausgegeben.

Dieses Kommando kann auch in der Parameterdatei im Abschnitt `<GLOBAL>` angegeben werden.

### ZIP-Archiv öffnen und folgende Attachments aufnehmen (nur BS2000)

**OPEN-ZIP=** [*archiv-name*] [ ,FORMAT=C | B ] [ ,CRYPTKEY=Y | *cryptkey* ]

Es wird ein temporäres ZIP-Archiv erzeugt, in das alle Dateien aufgenommen werden, die in den nachfolgenden Anweisung `ATTACH` (S. 21) vorkommen. Die `OPEN-ZIP`-Anweisung gilt bis `CLOSE-ZIP` (S. 25), bis zur nächsten Anweisung `OPEN-ZIP` oder bis zur Anweisung `END` bzw. `SEND`. Es können auch mehrere ZIP-Archive für ein Mail erzeugt werden.

*archiv-name* Name des ZIP-Archivs auf dem Zielsystem. Fehlt der Name, erhält das ZIP-Archiv auf dem Zielsystem den Namen `ATTn.ZIP`.

FORMAT = C	Compatible: Es wird ein ZIP-Archiv im kompatiblen Format erzeugt. Das ZIP-Archiv kann sowohl im Windows als auch im BS2000 verarbeitet werden. SAM- und ISAM-Dateien werden dabei von EBCDIC nach ISO8859 konvertiert und statt der Satzlänge wird ein Satz-Endezeichen erzeugt. Insbesondere bei nicht abdruckbaren Zeichen und Umlauten können hierbei Probleme auftreten. Ist in der ATTACH-Anweisung "MODE=BIN" angegeben, entfällt die Konvertierung von EBCDIC nach ISO8859. Die Angabe "MODE=BS2" ist im kompatiblen Format nicht zulässig.
FORMAT = B	<p>BS2000: Es wird ein ZIP-Archiv im BS2-Format erzeugt. Das ZIP-Archiv kann zwar auch unter Windows geöffnet werden, aber die Daten sind in EBCDIC kodiert und vor jedem Satz steht bei ISAM- und SAM-Dateien ein 4 Byte langes Satzlängenfeld. PAM-Dateien enthalten nur die Daten, d.h. daß z.B. mit PDFGEN erzeugte oder mit FTP übertragene PDF-Dateien auch aus einem Archiv im BS2000-FORMAT selektiert und im richtigen Format wieder hergestellt werden können. Mit dem Programm EDT für Windows können auch Dateien in EBCDIC und 4 Byte langen Satzlängenfeld angezeigt und verarbeitet werden.</p> <p>Alle Dateitypen können im BS2000 mit dem ursprünglichen Inhalt wieder hergestellt werden. Einzige Ausnahme: Bei ISAM-Dateien mit DUPKEY=YES wird beim Extrahieren nur der letzte Satz mit gleichen Schlüsselns geschrieben.</p>
CRYPTKEY=Y   <i>cryptkey</i>	Attachment verschlüsseln:
Y	Die Datei wird mit dem Passwort aus einer vorhergehenden Anweisungen CRYPTKEY (S. 27) bzw. aus der CRYPTKEY-Anweisung der Parameterdatei verschlüsselt.
<i>cryptkey</i>	Passwort für die Verschlüsselung.
	Hinweis:
	In einer ATTACH-Anweisung für Dateien, die in ein ZIP-Archiv aufgenommen werden sollen, sind folgende Parameter nicht zulässig:
	MODE=BS2 (nicht zulässig im kompatiblen Modus, bei Format = B nicht notwendig)
	ISAM=N
	CRYPTKEY=Y   <i>kryptkey</i>
	COMP=Y
	TYPE= <i>content-type</i>
	Beispiel:
	<pre>OPEN-ZIP TEST1.ZIP, FORMAT=B ATTACH DATEI1 ATTACH DATEI2 CLOSE-ZIP OPEN-ZIP TEST2.ZIP, FORMAT=C, CRYPTKEY=Y ATTACH DATEI* ATTACH LIB(S, *), MODE=PDF CLOSE-ZIP</pre>



### Optionen für PDF-Generierung

**PDF-LAYOUT**={ *layout* | DEFAULT } , *font* , *fontsize* , *page* , *maxlines* , *mode* , *code* [ ,*margin-left*] [ ,*margin-top*] [ ,*background*] [ ,AUTOSIZE= V | H | YES | NO ]

*layout* Name des Layouts. Dieser Name kann in der ATTACH-Anweisung im Parameter `MODE=PDF-layout` angegeben werden. Der Name `DEFAULT` bezeichnet das Standard-Layout, das zum Tragen kommt, wenn in der ATTACH-Anweisung (S. 21) `MODE=PDF` angegeben wird.

*font* Zeichensatz: Zulässig sind `COURIER` (feste Schrift), `HELVETICA` (Proportionalsschrift) oder `TIMES` (Proportionalsschrift). Die Namen der Zeichensätze können bis auf 1 Byte abgekürzt werden.

*fontsize* Schriftgröße: 1-72  
*fontwidth/fontheight* verschiedene Schriftgrößen für die Zeichenbreite und Zeilenhöhe.

*page* A4 | A4Q | A5 | A5Q | *höhe* x *breite* in mm  
Seitengröße:  
A4 = DINA4, A4Q = DINA4 quer, A5 = DINA5, A5Q = DINA5 quer oder *höhe* x *breite* in mm, jeweils dreistellig, z.B. 310x210.  
Ist eine Zeile länger als die definierte Breite der Seite, wird die Seitenbreite automatisch angepaßt. Beim Ausdruck kann im Acrobat Reader angegeben werden, daß solche Seiten auf DINA4 bzw. DINA5 verkleinert werden sollen (mit der entsprechenden Anpassung der Schriftgröße).

*maxlines* Max. Anzahl von Zeilen pro Seite. Bei Dateien mit Drucksteuerzeichen kommt dieser Parameter nur zum Tragen, falls wegen der Schriftgröße die Zeilen nicht auf eine Seite passen.

*mode* ASA | SIE | NO [ :*col1-col2*: ] [ ;CHchannel=*line* [ ;CHchannel=*line*..... ] ] [ ;NCTRL ]

SIE druckaufbereitete Datei mit Siemens-Steuerzeichen. Das Steuerzeichen wird aus der ersten Spalte der *col*-Angabe entnommen.

ASA druckaufbereitete Datei mit ASA-Steuerzeichen. Das Steuerzeichen wird aus der ersten Spalte der *col*-Angabe entnommen.

NO Datei ohne Druck-Steuerzeichen. Jeweils nach dem Erreichen der max. Zeilenanzahl wird eine neue Seite erzeugt.

:*col1-col2*: Spaltenangabe. Für die PDF-Erzeugung werden nur Daten von Spalte *col1* bis *col2* benutzt. Bei *mode* SIE oder ASA muß das Druck-Steuerzeichen in *col1* stehen. Bei fehlender Spaltenangabe wird der ganze Datensatz benutzt.

*channel* Kanal 1 bis 11. Für die Vorschübe auf Kanal 1 - 11 kann für jeden Kanal eine Zeile definiert werden, auf die bei dem entsprechenden Steuerzeichen positioniert werden soll (SIE: vor dem Drucken = X'C1' bis X'CB', nach dem Drucken = X'81' bis X'8B', ASA: C'1' bis C'9', C'A' und C'B').

*line* Zeile 1- 999, auf die bei Kanalvorschub positioniert werden soll.

Beispiele:     *sie*;ch1=3;ch2=10;ch3=30  
              *sie*:1-50:;ch1=3;ch5=15  
              no:10-120:  
              asa;ch1=3;ch2=10;ch3=30

NCTRL     No Control format character: Die Steuerzeichen X'11' bis X'1F' werden nicht ausgewertet.

Außer den Vorschubsteuerzeichen in Spalte 1 werden in den Daten folgende Format-Steuerzeichen berücksichtigt, falls nicht NCTRL angegeben wurde:

X'11' = Schrift höher stellen

X'12' = Schrift tiefer stellen

X'13' = hochgestellt bzw. tiefgestellt ausschalten

X'14' = Sperrschrift ein

X'15' = Sperrschrift aus

X'16'### = Beginn Oktalzeichen, danach abdruckbar 3 Bytes Oktalzeichen

z.B.

200 = €-Zeichen

274 = ¼-Zeichen

275 = ½-Zeichen

251 Copyright ©

256 Registered ®

231 Trademark ™

X'17'+C'R' = Schrift Rot

X'17'+C'B' = Schrift Blau

X'17'+C'G' = Schrift Grün

X'17'+C'N' = Normalschrift (Schwarz)

X'1A###' Neue Fontgröße ##, z.B. X'1A08' = Fontgröße 8

X'1A00' Originalfontgröße wiederherstellen

X'1B' = Proportionalschrift Helvetica ein

X'1C' = Fettschrift ein

X'1D' = Kursivschrift ein

X'1E' = Fett-kursiv ein

X'1F' = Reset (alles auf Ausgangswert, auch implizit 13, 15, 17 und 1a00)

*code*

DE | DK | INT | *usercode*

Angaben für die Code-Umwandlung von 7-Bit-EBCDIC nach ASCII bezüglich der Umlaute und Sonderzeichen. Das Problem besteht darin, daß bestimmte internationale Sonderzeichen und deutsche Umlaute nicht bzw. dänische Sonderzeichen gleichzeitig dargestellt werden können. Dabei geht es um folgende Zeichen:

Dänische Sonderzeichen:   æ   ø   å   Æ   Ø   Å   ü   Ü

Deutsche Umlaute:       ä   ö   ü   Ä   Ö   Ü   ß

Hexa-Wert               FB   4F   FD   BB   BC   BD   FF   6A

Internationale Zeichen:   {   |   }   [   \   ]   ~   ^

DE Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN = EDF03IRV, deutsche Variante), aus den entsprechenden EBCDIC-Zeichen sollen Umlaute (ÄÖÜäöüß) erzeugt werden.

DK Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN = EDF03IRV, dänische Variante), aus den entsprechenden EBCDIC-Zeichen sollen dänische Sonderzeichen (ÆØÅæøå) erzeugt werden.

INT Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN= EDF03IRV internationale Variante) oder 8-Bit-EBCDIC (CCSN=EDF041), aus den entsprechenden EBCDIC-Zeichen sollen die Zeichen "{ | } [ \ ] ~" erzeugt werden. Beim 8-Bit-Code EDF041 können die Daten sowohl die Sonderzeichen als auch die Umlaute enthalten.

*usercode* Name eines User-Codes. Die von der Standard-Codierung abweichenden Translate-Anweisungen sind in einer Datei definiert, die mit der Anweisung `CODEFILE` (S. 49) zugewiesen wird.

*margin-top* Seitenrand vom oberen Rand der Seite in mm, Standard = 4.

*margin-left* Seitenrand vom linken Rand der Seite in mm, Standard = 4.

### Hintergrundbilder und Wasserzeichen

Hier können Hintergrundbilder und ein Wasserzeichen angegeben werden. Das Wasserzeichen wird über ev. angegebenen Hintergrundbilder gedruckt.

*background* `jpg-file[,x,y,w,h] [ ;jpg-file[,x,y,w,h,p]... [ ;WATERMARK:watermark].... ]`

*jpg-file* Name einer JPG-Datei mit dem Hintergrundbild. Folgende Formate sind zulässig:

<i>bs2-file</i>	Name einer BS2000-Datei.
<i>lib(typ,elem[, version])</i>	Bibliothek, Elementtyp und Elementname.
<i>/posix-file</i>	Name einer POSIX-Datei.

### Position des Bildes:

*x* Entfernung des linken oberen Ecks der Hintergrundbildes vom linken Seitenrand in mm. Bei fehlender Angabe gilt der Wert 0.

*y* Entfernung des linken oberen Ecks der Hintergrundbildes vom oberen Seitenrand in mm. Bei fehlender Angabe gilt der Wert 0.

### Breite und Höhe des Bildes:

Bei fehlender Angabe von Höhe und Breite werden die in den Headerangaben der JPG-Datei gespeicherte Werte benutzt. Das Bild wird dann in der Originalgröße eingefügt. Ist das Bild breiter oder höher wie die ganze Seite, wird es abgeschnitten.

Soll die Breite oder Höhe angepaßt werden, können entweder die beiden Werte oder nur einer der Werte angegeben werden. Wird nur ein Wert angegeben, so wird automatisch der andere Werte im gleichen Verhältnis angepaßt, damit das Bild nicht verzerrt wird. Soll das Bild verzerrt werden, müssen immer die Breite und die Höhe angegeben werden.

Ist zum Beispiel in der JPG-Datei ein Wert von 200 mm Breite und 100 mm Höhe gespeichert und als Breite wird 100 mm angegeben, so wird die Höhe automatisch auf 50 mm geändert. Das Gleiche gilt, wenn nur die Höhe angegeben wird.

*w* Width. Breite des Hintergrundbildes in mm.

*h* Height. Höhe des Hintergrundbildes in mm.

### Seiten, auf denen das Hintergrundbild plaziert werden soll.

<i>p</i>	<i>page-from</i> [ - <i>page-to</i> ] [ E[VEN]   O[DD] ]
<i>page-from</i>	Nr. der Seite bzw. erste Seite, auf der das Bild plaziert werden soll. \$ bedeutet alle Seiten.
<i>page-to</i>	Letzte Seite, auf der das Bild plaziert werden soll. Die Angabe \$ bedeutet, bis zum Ende der Datei.
EVEN	Das Bild soll nur auf den geraden Seiten plaziert werden.
ODD	Das Bild soll nur auf den ungeraden Seiten plaziert werden.

### Wasserzeichen

<i>watermark</i>	<FS= <i>fontsize</i> > <i>watermark-text</i>
<i>fontsize</i>	Schriftgröße für das Wasserzeichen: 1-256
<i>watermark-text</i>	Text für das Wasserzeichen, max. 32 Bytes

### Automatische Größenanpassung

AUTOSIZE= V | H | YES | NO

H	Horizontal. Falls eine Zeile länger ist als die Seitenbreite, wird die Seitenbreite für die aktuelle Seite entsprechen vergrößert. Die Anpassung gilt nur für die aktuelle Seite.
V	Vertikal. Falls noch Zeilen auf der aktuellen Seite gedruckt werden sollen, die über den Seitenrand hinausgehen, wird die Seitenlänge für die aktuelle Seite entsprechen vergrößert. Die Anpassung gilt nur für die aktuelle Seite.
YES	Falls notwendig, wird die Seitenbreite und die Seitenlänge angepaßt.
NO	Eine automatische Anpassung wird nicht durchgeführt. Falls aufgrund der Anzahl der Zeichen bzw. Zeilen und der eingestellten Font-Größe die Seitenränder überschritten werden, gehen die Zeichen verloren (die Daten sind zwar im Dokument enthalten, werden aber vom Acrobat Reader nicht angezeigt). Diese Option gilt als Standard, falls dieser Parameter nicht angegeben wird.

### Beispiele:

```
pdf-layout=default,courier,8,a4q,60,sie,de,autosize=yes
pdf-layout=default,courier,8,320x170,60,sie,de
pdf-layout=a4pde,courier,10,a4,50,sie,de,25,15
pdf-layout=a4de,courier,10,a4,50,no,de,25,15,t1.jpg,10,10,,,1;t2.jpg,10,10,,,
  2-$EVEN;t3.jpg,10,10,,,2-$ODD
pdf-layout=a4helv,helvetica,8,a4,50,sie,de
pdf-layout=a4helv,helvetica,8,a4,50,sie,de,,,t1.jpg;watermark:<fs=50>Test
pdf-layout=a4qp,courier,8,a4q,60,sie,de,25,15,t1.jpg,10,10;t2.jpg,200,10
```

### Hinweis:

Es ist darauf zu achten, daß die Anweisung vor der ATTACH-Anweisung steht, in der erstmals auf dieses Layout verwiesen wird. Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Priorität festlegen

**PRIORITY=HIGH|NORM|LOW**

Über Priority HIGH/LOW kann erreicht werden, dass die Mail beim Empfänger entsprechend markiert angezeigt wird.

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Antwort-Adresse

**REPLY=*recipient***

E-Mail-Adresse, die vom Mail-Client des Empfängers automatisch als Adresse für die Beantwortung des E-Mails vorgeschlagen wird.

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Quittung des Empfängers anfordern

**RR=*recipient***

(Return to Recipient) E-Mail-Adresse für die Quittung des Empfängers, daß das E-Mail empfangen wurde. Die Behandlung dieser Anforderung kann je nach den Einstellungen des Mail-Client unterschiedlich behandelt werden:

- a) Die Quittungsanforderung wird von der Software unterdrückt, d.h. es kommt keine Quittung zurück.
- b) Die Quittung wird automatisch zurückgeschickt.
- c) Das Versenden der Quittung wird vom Empfänger angefordert, wenn die Nachricht geöffnet wird.

Es ist also nicht sichergestellt, daß die Quittung wirklich erstellt wird.

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Mail senden

**SEND oder END**

Die Nachricht wird gesendet. Danach wird das Programm beendet.

### Servername

**SERVER=*servername***

BCAM-Name des Mailservers bzw. des Relays.

Die Anweisung kann mehrmals vorkommen. Die Server werden in der angegebenen Reihenfolge geprüft, wobei der erste gültige Server benutzt wird. Dadurch ist es möglich, alternative Mailserver anzugeben, die nur benutzt werden, wenn die anderen vorher definierten Server nicht ansprechbar sind. Ist in der privaten Parameterdatei oder in den SYSDTA-Datei ein gültiger Server angegeben, wird dieser benutzt, auch wenn vorher schon ein gültiger Server erkannt wurde.

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <GLOBAL> angegeben werden.

### Betreffzeile

**SUBJECT**=*string* Beliebiger String, der beim Empfänger unter "Subject" zur Identifizierung bzw. als Kurzbeschreibung angezeigt wird.

*string* Beliebiger Text  
Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

### Mail-Adresse

**TO**=*recipient*[,*recipient*,...] Mail-Adresse des/der Empfänger. Mehrere E-Mail-Adressen werden durch Komma getrennt. Diese Anweisung kann mehrmals angegeben werden.

*recipient* Mail-Adresse des Empfängers, z.B. [Name@test.de](mailto:Name@test.de)

### Translate-Tabelle

**TRTAB**=*trfile* Name einer Datei, in der die Translate-Anweisungen für eine abweichende Code-Umwandlung von EBCDIC nach ANSI stehen (Beispiel siehe Musterdatei S/MAIL.TR TAB).

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <SEND> angegeben werden.

Im Programm ist eine Standardtabelle enthalten. Die Translate-Anweisungen in dieser Datei überschreiben die Standardtabelle. Es müssen also nur die Zeichen angegeben werden, die abweichend behandelt werden sollen.

Dieses Kommando wird nur noch aus Kompatibilitätsgründen zum Programm MAIL, Version 1.2 unterstützt. Die Translate-Anweisungen sind ab der Version 2.0 in der Parameterdatei, Abschnitt <TRTAB-SEND> enthalten. Es ist aber zu beachten, daß beim Kommando TRTAB eine andere Standard-Translate-Tabelle verwendet wird, als in der Parameterdatei, Abschnitt <TRTAB-SEND>.

Format der Anweisungen:

*xxyy* [*remark*]

xx = EBCDIC-Zeichen

yy = ANSI-Zeichen

remark = wahlweise Bemerkung

Beispiel:

```
BBC4    UMLAUT GROSS A
BCD6    UMLAUT KLEIN A
```

### Prozeßschalter für Warnings

**WARNINGON=switch[,RESET]** Wird während der Verarbeitung eine Warn-Meldung ausgegeben, wird der Prozeßschalter *switch* auf ON gesetzt. Fehlt dieses Kommando, wird **kein** Schalter gesetzt. Siehe auch Anweisung `ERRORON` (S. 28) für Fehlermeldungen.

*switch*

Prozeßschalter 0 – 31

RESET

Vor Beginn der Verarbeitung wird der angegebene Prozeßschalter gelöscht (`SETSW n=OFF`).

Dieses Kommando kann auch in der Parameterdatei im Abschnitt <GLOBAL> angegeben werden.

### 5 Programm MAILR (Receive)

#### Kurzbeschreibung

Mit dem Programm MAILR können Sie im BS2000/OSD E-Mails von jedem im Netz erreichbaren MAIL-Server empfangen. Als Attachments können außer den üblichen Text- und Binärdateien auch Dateien im Format BS2 (S. 41) und OPGCRYPT (S. 41) empfangen werden.

Der Ablauf wird durch die Anweisungen in der Parameterdatei MAIL.PAR gesteuert. Der Inhalt der Nachricht wird aufgeteilt in die eigentliche Mitteilung und in die Daten der mitgelieferten Dateien (Attachments). Nach dem Empfang eines E-Mails kann eine Folgeverarbeitung in Form eines beliebigen BS2000-Kommandos angestoßen werden.

Es sind drei verschiedene Verarbeitungs-Modi möglich:

- a) einmal alle vorhandenen E-Mail's empfangen (CYCLE=0).
- b) alle *n* Sekunden alle vorhandenen E-Mail's empfangen (CYCLE > 0, siehe S. 62).
- c) Nochmaliges Aufbereiten der E-Mail-Daten aus der Originaldatei einer bereits abgeholten E-Mail.

#### MAIL-ID

Für die Dateinamen, die von MAILR erzeugt werden, wird eine einheitliche MAIL-ID verwendet, die wie folgt aufgebaut ist:

*prefix*.*yyyymmdd*.*hhmmss*.

*prefix* Zeichenfolge, die mit dem Kommando PREFIX in der Parameterdatei für jeden Benutzer definiert werden muß (S. 77).

*yyyymmdd* Datum des Tages, an dem die Nachricht vom Mail-Server abgeholt wird.

*hhmmss* Uhrzeit des Beginns der Verarbeitung der entsprechenden Mail-Nachricht im BS2000.

#### Erzeugte Dateien

Die einzelnen Teile der Mail-Nachricht sind auf dem Mail-Server in einer Datei gespeichert und sind nur durch SMTP-Anweisungen voneinander getrennt. Zur komfortablen Weiterverarbeitung werden in Abhängigkeit der Einstellungen in der Parameterdatei vom Programm MAILR folgende Dateien erstellt:

*mail-id*.*OMSG* Originalnachricht mit allen SMTP-Anweisungen, einschl. der Original-Satzende-Trennzeichen und der Daten für die Attachments. Diese Datei wird erstellt, wenn die Parameterdatei das Kommando ORIGINAL=Y enthält. Wahlweise können alle E-Mails in eine Datei, getrennt durch Header-Sätze, geschrieben werden (ORIGINAL=*file* (S. 77)).

*mail-id*.*MAIL* MAIL-Nachricht einschl. aller SMTP-Anweisungen ohne Satzende-Trennzeichen und ohne Daten für die Attachments. Vor allem die Angaben FROM und SUBJECT sind unter anderem in dieser Datei enthalten. Es ist jedoch zu beachten, daß diese Angaben auch in der Folgeverarbeitung als Parameter zur Verfügung stehen. Diese Datei wird erstellt, wenn die Parameterdatei das Kommando MAILFILE=Y enthält. Wahlweise können alle E-Mails in eine Datei, getrennt durch Header-Sätze, geschrieben werden (MAILFILE=*file* (S. 76)).

*mail-id*.*TEXT* MAIL-Nachricht ohne SMTP-Anweisungen. Diese Datei enthält nur die eigentlichen Daten ohne Attachments. Diese Datei wird erstellt, wenn die Parameterdatei das Kommando TEXTFILE=Y enthält. Wahlweise können alle E-Mails in eine Datei, getrennt durch Header-Sätze, geschrieben werden (TEXTFILE=*file* (S. 80)).



- mail-id*.HTML MAIL-Nachricht ohne SMTP-Anweisungen im HTML-Format. Diese Datei enthält nur die eigentlichen Daten einschl. HTML-Anweisungen ohne Attachments. Diese Datei wird erstellt, wenn die Parameterdatei das Kommando `HTMLFILE=Y` enthält und die Daten HTML-Formatierungen enthalten (`Content-Type = text/html`). Es kann sein, daß E-Mails den gesamten Text im Format `text/plain` und im Format `text/html` enthalten. In diesem Fall wird die Datei *mail-id*.TEXT und *mail-id*.HTML erstellt. Wahlweise können alle E-Mails in eine Datei, getrennt durch Header-Sätze, geschrieben werden (`HTMLFILE=file` (S. 76)).
- mail-id*.FNAM Original-Dateinamen der Attachments. Da die Attachments im BS2000 den Namen *mail-id*.ATTN erhalten, geht der Originalnamen verloren. Falls dieser noch benötigt wird, kann der Original-Dateinamen in dieser Datei gespeichert werden. Diese Datei enthält für jedes Attachment einen Datensatz. In den ersten 54 Bytes steht der BS2000-Dateiname und ab dem Byte 55 steht der Original-Dateiname. Diese Datei wird erstellt, wenn die Parameterdatei das Kommando `FNAMFILE=Y` enthält. Wahlweise können alle Dateinamen-Sätze in eine Datei, getrennt durch Header-Sätze, geschrieben werden (`FNAMFILE=file` (S. 75)).
- mail-id*.ATTN Daten, die einem Attachment zugeordnet sind. Diese Datei wird erstellt, wenn die Parameterdatei das Kommando `ATTACHMENT=Y` enthält.

### Dateiformate der Attachments

#### Attachments im Format MODE=BS2

Mit MAILR können Sie BS2000-Dateien beliebigen Formats als Attachment empfangen (PLAM-Dateien, ISAM-Dateien usw.), soweit Sie von dem Programm MAIL vom BS2000 aus versandt wurden. Diese Nachrichten werden vom Programm MAIL mit einem besonderen SMTP-Header versehen, damit beim Empfangen die BS2000-Datei wieder mit den gleichen Dateiattributen erstellt werden kann. Durch Angabe der Option "MODE=BS2" beim Senden mit dem Programm MAIL wird erreicht, daß die Datei als PAM-Datei in EBCDIC-Code versandt wird und 1:1 auf dem Zielsystem wieder hergestellt werden kann.

#### Attachments im Format OPGCRYPT

Dateien, die mit dem Programm OPGCRYPT verschlüsselt bzw. komprimiert wurden, können von MAILR automatisch entschlüsselt werden. Soweit das Attachment nur komprimiert ist, ist keine weitere Angabe notwendig. Falls das Attachment verschlüsselt ist kann ein Key für die Entschlüsselung mit der Anweisung `DECRYPTKEY` (S. 71) im Abschnitt `<USER-username>` definiert werden. Falls eine Entschlüsselung nicht möglich ist, wird die Datei im verschlüsselten Zustand gespeichert und kann mit dem Programm OPGCRYPT entschlüsselt werden.

### Folgeverarbeitung

Nach dem Empfang eines E-Mails kann eine Folgeverarbeitung in Form eines beliebigen BS2000-Kommandos bzw. im z/OS eines Batch-Jobs angestoßen werden. Die Art der Folgeverarbeitung kann für jeden Benutzer und zusätzlich in Abhängigkeit vom Inhalt der SUBJECT-Angabe gesteuert werden. In der Regel wird die Folgeverarbeitung aus einem Kommando `ENTER-PROC` bestehen. Wichtige Informationen können über Parameter an die Prozedur weitergegeben werden, wie z.B. der Benutzername, die Dateinamen, der Inhalt von SUBJECT, die FROM-Adresse usw.

Die Folgeverarbeitung kann aus Sicherheitsgründen nur in der globalen Parameterdatei definiert werden.

### Programmaufruf

#### E-Mail's empfangen

##### a) BS2000

```
/EXEC LIB=MAILLIB,ELEM=MAILR
```

oder

```
/START-PROG *MODULE (SRCLIB.MAIL,MAILUPR,PROG-MODE=*ANY,RUN-MODE=*ADVANCED)
```

##### b) z/OS

```
//JOB LIB DD DSN=MAIL.LOAD,DISP=SHR
```

```
//MAILR EXEC PGM=MAILR
```

Nach dem Laden des Programms werden die aktuell vorhandenen E-Mails abgeholt. Fehlt die CYCLE-Anweisung in der Parameterdatei (Abschnitt <RECEIVE> ), wird das Programm danach beendet.

Ist eine CYCLE-Anweisung vorhanden, wird alle *n* Sekunden geprüft, ob neue E-Mails eingetroffen sind und ggf. verarbeitet.

Das Programm kann auch als Unterprogramm geladen werden, siehe S. 86.

#### E-Mail aufbereiten

##### a) BS2000

```
/FILE omsg-file, LINK=OMSGINPF
```

```
/EXEC LIB=MAILLIB,ELEM=MAILR
```

##### b) z/OS

```
//JOB LIB DD DSN=MAIL.LOAD,DISP=SHR
```

```
//OMSGINPF DD DSN=omsg-file,DISP=SHR
```

```
//MAILR EXEC PGM=MAILR
```

Aus der Originaldatei *mail-id.OMSG* eines bereits empfangenen E-Mail's werden nochmals die Ergebnisdateien *mail-id.MAIL*, *mail-id.TEXT*, *mail-id.HTML*, *mail-id.FNAM* und *mail-id.ATTN* (S. 40) erzeugt.

#### Unterbrechung der Verarbeitung

Das Programm kann durch Versorgung der REQUEST-JV mit dem Wert `HOLD` angehalten werden. Mit dem Wert `RESUME` wird die Verarbeitung wieder fortgesetzt. Der Name der JV muß mit der Anweisung REQUEST-JV (S. 63) in der Parameterdatei festgelegt werden. Der aktuelle Status des Programms (ACTIVE / HOLD / TERM) wird in der STATUS-JV (S. 64) angezeigt.

Beispiel: SETJV MAIL-REQUEST,'HOLD'

### Programmbeendigung im CYCLE-Modus unter BS2000

Das Programm kann im Dialogmodus mit der Taste K2 beendet werden.

Das Programm kann mit folgenden Aktionen beendet werden:

- a) Durch Versorgen der REQUEST-JV mit dem Wert `TERM`. Der Name der JV muß mit der Anweisung REQUEST-JV (S. 63) in der Parameterdatei festgelegt werden.
- b) mit dem Konsolkommando `/INTR R, tsn`
- c) mit dem Kommando `/CAN tsn`.
- d) Taste K2 (nur im Dialog-Modus).

Beispiel: `SETJV MAIL-REQUEST,'TERM'`

### Programmbeendigung im CYCLE-Modus unter z/OS

Das Programm kann mit folgenden Aktionen beendet werden:

- a) mit dem Konsolkommando `STOP (P jobname)`
- b) mit dem Konsolkommando `CANCEL (C jobname)`

## 6 Parameterdatei MAIL.PAR

### Allgemeines

Das Format dieser Anweisungen entspricht dem Format der SYSDTA-Anweisungen. Die Anweisungen der Parameterdatei werden vor dem Einlesen der Kommandos von SYSDTA verarbeitet. Wird das gleiche Kommando auch über SYSDTA eingelesen, überschreiben die SYSDTA-Angaben die Werte aus der Parameterdatei.

### Globale Parameterdatei MAIL.PAR

Die globale Parameterdatei wird als Datei `MAIL.PAR` von dem Verzeichnis (User-ID) gelesen, von dem das Programm `MAIL` geladen wurde.

Wird die Unterprogramm-Schnittstelle benutzt, kann über den Linknamen/DD-Namen `MAILPARG` eine beliebige Dateien als globale Parameterdatei zugewiesen werden. Wird die Unterprogramm-Schnittstelle benutzt, gilt als Ladeverzeichnis die User-ID des aufrufenden Programms.

Falls die IMON-Registrierung (S. 15) durchgeführt wurde, wird die globale Parameterdatei immer von der Datei gelesen, die mit dem logischen Namen `MAILPAR` verknüpft ist. Ist diese Datei nicht vorhanden, wird das Programm beendet.

### Private Parameterdatei

Zusätzlich gibt es noch die Möglichkeit eine private Parameterdatei zu benutzen. Die Kommandos der privaten Parameterdatei überschreiben die Werte aus der globalen Parmameterdatei. Z.B. könnten die `FROM-` und `SUBJECT-`Angaben aus der privaten Parameterdatei und die `SERVER-` und `DOMAIN-`Angaben aus der globalen Parameterdatei gelesen werden

Die private Parameterdatei wird zuerst unter dem Linknamen `MAILPAR` (mit beliebigem Dateinamen) und falls dieser nicht vorhanden ist, unter der eigenen User-ID als Datei `MAIL.PAR` gesucht. Wird die Unterprogramm-Schnittstelle benutzt, gilt als Ladeverzeichnis die User-ID des aufrufenden Programms.

### Reihenfolge der Verarbeitung

Die Angaben in der privaten Parameterdatei überschreiben die Werte aus der globalen Parameterdatei.

Bestimmte Kommandos können sowohl in den Parameterdateien als auch in der SYSDTA-Datei vorkommen. Die Angaben in der SYSDTA-Datei überschreiben in diesen Fällen die Werte aus den Parameterdateien.

Durch das Kommando `LOCK` in der globalen Parameterdatei können bestimmte Kommandos gesperrt werden, d.h. sie werden mit Fehlermeldung abgewiesen, falls sie in der privaten Parameterdatei oder in der SYSDTA-Datei vorkommen.

### Aufbau der Parameterdatei

Die Parameterdatei ist in verschiedene Abschnitte aufgeteilt. Der Abschnittsname ist in eckigen Klammern eingeschlossen (z.B. `<GLOBAL>` ). Folgende Abschnitte sind vorgesehen:

	<b>Globale Parameter für beide Programme MAIL und MAILR:</b>
<code>&lt;GLOBAL&gt;</code>	Globale Parameter für MAIL (Senden) und MAILR (Empfangen).
	<b>Parameter für das Programm MAIL (Senden):</b>
<code>&lt;SEND&gt;</code>	Allgemeine Parameter für das Programm MAIL.
<code>&lt;TRTAB-SEND&gt;</code>	Translate-Tabelle für die Übersetzung EBCDIC → ASCII

### Parameter für das Programm MAILR (Receive / Empfangen):

<RECEIVE>	Allgemeine Parameter für das Programm MAILR.
<RESTRICT <i>user-id</i> >	Einschränkungen pro BS2000-User-ID.
<USER <i>username</i> >	Parameter pro Benutzer (Mail-Account).
<TRTAB-RECEIVE>	Translate-Tabelle für die Übersetzung ASCII → EBCDIC

Aus Kompatibilitätsgründen ist auch eine Parameterdatei ohne Abschnitte zulässig, soweit nur Parameter der Abschnitte <GLOBAL> und <SEND> enthalten sind (wie bis MAIL Version 1.2).

### Kommentarzeilen

*\*string* In allen Abschnitten sind Kommentarzeilen zulässig. Sie enthalten in der ersten Spalte das Zeichen "\*", gefolgt von einer beliebigen Zeichenfolge.

### Abschnitt <GLOBAL>

Dieser Abschnitt enthält Anweisungen, die für das gesamte MAIL-System (Programme MAIL und MAILR) gelten.

### Internet-Domäne

**DOMAIN=***domainname* Internet-Domäne, z.B. `opg.de`  
Diese Anweisung kann auch als SYSDTA-Anweisung (S. 27) für das Programm MAIL angegeben werden.

### Fehlerschalter

**ERRORON=***switch*[**,RESET**] Falls während der Verarbeitung ein Fehler auftritt, wird der Prozeßschalter *switch* auf ON gesetzt. Fehlt dieses Kommando, wird der Prozeßschalter 31 benutzt. Siehe auch Anweisung `WARNINGON` (S. 37) für Fehlermeldungen.

*switch* Prozeßschalter 0 – 31

**RESET** Vor Beginn der Verarbeitung wird der angegebene Prozeßschalter gelöscht (`SETSW n=OFF`).

Diese Anweisung kann auch als SYSDTA-Anweisung (S. 28) für das Programm MAIL angegeben werden.

## Lizenznummer

**LICENSE**=*licensestring*

Lizenzinformationen.

*licensestring*

Der Lizenzeintrag besteht aus einer 8-stelligen Zeichenkombination. In der ausgelieferten Datei MAIL.PAR ist der Lizenzeintrag für die Testversion eingetragen. Mit diesem Lizenzeintrag läuft die Testversion 4 Wochen. Zusätzlich wird in die Nachricht und die Attachment-Dateien ein Lizenztext vorangestellt. Der Lizenzeintrag für die unbegrenzte Nutzung wird Ihnen mit der Auslieferung mitgeteilt.

## Umfang der Protokollierung

**MSGLOG**=N | Y | L

Umfang der Meldungen, die auf SYSOUT / SYSPRINT ausgegeben werden.

N

Es werden nur Fehlermeldungen auf SYSOUT /SYSPRINT ausgegeben.

Y

Außer den Fehlermeldungen werden verschiedene Aktionen protokolliert, wie z.B. die Verarbeitung der Parameterdateien und Attachments. Diese Option gilt als Standard bei fehlender MSGLOG-Anweisung.

L

Für Diagnosezwecke werden zusätzlich alle Nachrichten, die mit dem Mailserver ausgetauscht werden, auf SYSOUT ausgegeben.

Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

## Servername

**SERVER**=*servername* [,AUTH=LOGIN, *username*, *password*]

BCAM-Name des Mailservers bzw. des Relays.

Falls zum Empfangen und Senden verschiedene Server verwendet werden sollen, können im Abschnitt <RECEIVE> ein oder mehrere Server eingetragen werden, die nur für das Empfangen von Mails benutzt werden sollen. Sind im Abschnitt <RECEIVE> keine Anweisungen SERVER enthalten, gelten zum Empfangen die gleichen Server, wie im Abschnitt <GLOBAL>.

Falls mehrere Server eingetragen sind, gilt folgendes:

Senden: Die Server werden in der angegebenen Reihenfolge geprüft, wobei der erste gültige Server benutzt wird. Dadurch ist es möglich, alternative Mailserver anzugeben, die nur benutzt werden, wenn die anderen vorher definierten Server nicht ansprechbar sind. Ist in der privaten Parameterdatei oder in den SYSDTA-Datei ein gültiger Server angegeben, wird dieser benutzt, auch wenn vorher schon ein gültiger Server erkannt wurde.

Empfangen: Die Postfächer werden auf allen Servern abgefragt. Sind im Abschnitt <RECEIVE> Server definiert, werden nur diese Server abgefragt.

## Authentifizierung beim Senden von Mails

LOGIN

Username und Password werden mit "basx64" verschlüsselt. Andere Verschlüsselungsmethoden sind im Moment nicht vorgesehen.

*username*

Benutzername, falls vom Server eine Authentifizierung verlangt wird.

*password* Passwort, falls vom Server eine Authentifizierung verlangt wird.  
Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### TRTAB für MAIL-Adressen verwenden

**TRTAB-ADDR=N** | Y Im Abschnitt <TRTAB-RECEIVE> (S. 81) und <TRTAB-SEND> (S. 81) kann eine abweichende Translate-Tabelle für die Konvertierung von EBCDIC nach ISO-8859 und umgekehrt definiert werden. Dies kann bei Mail-Adressen (FROM, TO, CC, BCC, RR, REPLY) zu Problemen führen, wenn z.B. das Zeichen "@" undefiniert wurde.

N Für die Konvertierung der Mail-Adressen wird immer die interne Standard-Tabelle verwendet. Diese Einstellung gilt als Standard.

Y Für die Konvertierung der Mail-Adressen wird die abweichende TRTAB-Tabelle verwendet.

### Prozeßschalter für Warnings

**WARNINGON=switch** [,RESET] Wird während der Verarbeitung eine Warn-Meldung ausgegeben, wird der Prozeßschalter *switch* auf ON gesetzt. Fehlt dieses Kommando, wird **kein** Schalter gesetzt. Siehe auch Anweisung `ERRORON` (S. 27) für Fehlermeldungen.

*switch* Prozeßschalter 0 – 31

RESET Vor Beginn der Verarbeitung wird der angegebene Prozeßschalter gelöscht (`SETSW n=OFF`).

Diese Anweisung kann auch als SYSDTA-Anweisung (S. 39) für das Programm MAIL angegeben werden.

### Abschnitt <SEND>

Dieser Abschnitt enthält allgemeine Anweisungen für das Programm MAIL (Senden).

### Versand bei unvollständigen Mails.

**ABORT=TO** | **CC** | **FILE** | **RESTRICT**

Mit dieser Anweisung kann bestimmt werden, ob in bestimmten Fehlersituationen das Mail unvollständig versandt oder der Vorgang abgebrochen werden soll. Der Standard ist so eingestellt, daß ein Mail auch unvollständig versandt wird, z.B. wenn einer von mehreren Empfänger ungültig ist oder ein Attachment nicht angehängt werden kann.

TO	Der Vorgang soll abgebrochen werden, wenn ein Empfänger aus den TO-Angaben ungültig ist.
CC	Der Vorgang soll abgebrochen werden, wenn ein Empfänger aus den CC- oder BCC-Angaben ungültig ist.
FILE	Der Vorgang soll abgebrochen werden, wenn bei der Verarbeitung von Dateien (MSG-File, Attachments, PDF-File) Fehler auftreten und dadurch ein unvollständiges Mail erzeugt würde.
RESTRICT	Der Vorgang soll abgebrochen werden, wenn auf Grund der Restriktionen, die im Abschnitt <RESTRICT> festgelegt sind, Empfänger abgewiesen werden oder der Zugriff auf eine Datei nicht erlaubt wird.

### Dateigröße beschränken

**ATTACHSIZE=nnn [M|K|H]** Mit dieser Anweisung kann die Größe von Dateien, die als Attachment oder MSG-File (Anweisungen `ATTACH` und `MSGFILE`) versandt werden dürfen, begrenzt werden.

*nnn* Anzahl der Bytes, Killobytes, Megabytes bzw. Halpages.

M Die Angabe erfolgt in Megabytes (1.048.576 Bytes).

K Die Angabe erfolgt in Killobytes (1024 Bytes).

H Die Angabe erfolgt in Halpages (2048 Bytes).

Die Anweisung kann vom Administrator mit der Anweisung `LOCK` (S. 54) gesperrt werden, so daß sie nur in der globalen Parameterdatei zulässig ist.

### zulässige Dateien

**ATTACHNAME=[-] *ausw*[[-] *ausw*.....]**

Mit dieser Anweisung kann festgelegt werden, welche Dateien mit der Anweisung `ATTACH` und `MSGFILE` versandt werden dürfen. Die Anweisung kann vom Administrator mit der Anweisung `LOCK` (S. 54) gesperrt werden, so daß sie nur in der globalen Parameterdatei zulässig ist. Sie kann beliebig oft in der Parameterdatei vorkommen. Fehlt diese Anweisung, werden alle Dateien zum Versenden zugelassen. Siehe auch Anweisung `ATTACHNAME` (S. 65) im Abschnitt <RESTRICTION>

**[ - ] *ausw*** **[ :*cat-id*: ] [ *user-id*. | \$\* . ] *file* | *prefix*\* | *\*suffix* | *\*string*\* | \***

- Ein Minuszeichen vor dem Auswahlbegriff bedeutet, daß eine zuvor definierte Auswahl von Dateien eingeschränkt werden soll. Alle Dateien der angegebenen User-ID bzw. der RESTRICT-User-ID, die den Auswahlbedingungen entsprechen, sind nicht zulässig. Die negative Auswahl gilt nur für die Dateinamen und nicht für die User-ID.



### Cat-ID und User-ID (gilt nur für BS2000):

<code>:cat-id:</code>	Cat-ID für die nachfolgende Auswahlbedingungen. Fehlt die Cat-ID, gelten die Auswahlbedingungen für alle Cat-ID's.
<code>user-id.</code>	User-ID für die nachfolgende Auswahlbedingungen. Fehlt die User-ID, gelten die Auswahlbedingungen für die aktuelle User-ID.
<code>\$*.</code>	Die Auswahlbedingungen gelten für alle User-ID's.
<code>file</code>	<code>bs2file</code>   <code>lib (typ/elem[/vers])</code>   <code>zip-archiv (elem)</code>   <code>mvs-file</code>

### BS2000-Dateien

<code>bs2file</code>	Voll- oder teilqualifizierter Dateiname, ggf. mit Cat-ID und User-ID in der Syntax des BS2000-Kommandos <code>FSTAT</code> .
<code>lib (typ/elem[/vers])</code>	Name einer PLAM-Bibliothek, Elementtyp, Elementnamen und ggf. Version.
<code>zip-archiv (elem)</code>	Name eines ZIP-Archivs und der Elementname. Es ist zu beachten, daß hier der Prefix " <code>*ZIP:</code> " nicht anzugeben ist.

### z/OS-Dateien

<code>mvs-file</code>	<code>data-set</code>   <code>pds(member)</code>   <code>'hlq.data-set'</code>   <code>'hlq.pds(member)'</code>
<code>data-set</code>	Dateiname auf dem fernen Rechner ohne <code>HLQ</code> (high level qualifier). Vom Host-System wird der Dataset-Name automatisch um den <code>HLQ</code> ergänzt.
<code>pds(member)</code>	Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended) ohne <code>HLQ</code> . Vom Host-System wird der Member-Name automatisch um den <code>HLQ</code> ergänzt.
<code>'hlq.data-set'</code>	Vollqualifizierter Dateiname auf dem fernen Rechner. Aus diesem String wird für den Filetransfer der Dateiname <code>'hlq.data-set'</code> erzeugt.
<code>'hlq.pds(member)'</code>	Vollqualifiziertes Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended). Aus diesem String wird für den Filetransfer der Dateiname <code>'hlq.pds(member)'</code> erzeugt.

### Auswahl von mehreren Dateien

<code>prefix*</code>	Teilqualifizierter Dateiengabe mit Angabe des Prefix, z.B. <code>DAT*</code> (alle Dateinamen, die mit "DAT" beginnen). Es ist zu beachten, daß bei Bibliothekselementen
<code>*suffix</code>	Teilqualifizierter Dateiname mit Angabe des Suffix, z.B. <code>*SRC</code> (alle Dateinamen, die mit "SRC" enden).
<code>*string*</code>	Teilqualifizierter Dateiengabe mit Angabe einer Zeichenfolge, die in der Mitte des Dateinamens vorkommt, z.B. <code>*DAT*</code> (alle Dateinamen, die an einer beliebigen Stelle "DAT" enthalten).
<code>*</code>	Alle Dateien einer User-ID.

### Hinweise:

Fehlt die Anweisung in einem Abschnitt, so gilt eine ev. vorhandene Anweisung `ATTACHNAME` des Abschnitts `<SEND>`.

Es können beliebig viele Anweisungen `ATTACHNAME` angegeben werden.

Die positiven Suchbegriffe aller `ATTACHNAME`-Anweisungen werden in der definierten Reihenfolge als Oder-Bedingungen verarbeitet. Die negativen Suchbedingungen stellen Einschränkungen einer positiven Auswahl dar. Die negativen Bedingungen müssen daher nach den positiven Bedingungen definiert werden.

### Beispiel:

<code>ATTACHNAME=DATEI1, *.TXT, *.LST</code>	DATEI1 und alle Dateien die mit .LST oder mit .TXT enden. Die Dateiauswahl bezieht sich auf die User-ID der RESTRICT-Anweisung
<code>ATTACHNAME&gt;--SRC*, -MAC*</code>	Alle Dateien der User-ID aus der RESTRICT-Anweisung, die durch die vorhergehende Anweisung ausgewählt wurden außer SRC* und MAC*
<code>ATTACHNAME=\$USR1.*, -\$USR1.SEC*</code>	alle Dateien der User-ID \$USR1, außer SEC*
<code>ATTACHNAME=:LST:\$*.*STATISTIK*</code>	Alle Dateien, die den String STATISTIK enthalten und sich in einer beliebigen User-ID mit der
	Cat-ID "LST" befinden.
<code>ATTACHNAME=\$*.LST.*</code>	Alle Dateien, die mit LST. beginnen in allen User-ID's
<code>ATTACHNAME=-\$TSOS.*</code>	Von der User-ID \$TSOS dürfen keine Dateien versandt werden, auch nicht die Dateien, die mit einer vorherigen Anweisung zugelassen wurden.
<code>ATTACHNAME=LIB1 (S/*</code>	Alle S-Elemente der PLAM-Bibliothek LIB1
<code>ATTACHNAME=LIB1 (S/SRC.*</code>	Alle S-Elemente der PLAM-Bibliothek LIB1, deren Namen mit SRC. beginnt.
<code>ATTACHNAME=ZIP1 (*</code>	Alle Elemente des ZIP-Archivs Bibliothek LIB1
<code>ATTACHNAME='TEST.DAT*'</code>	Alle Dateien, die mit TEST.DAT beginnen
<code>ATTACHNAME=LIB1 (TEST*</code>	Alle Elemente der Lib. LIB1, die mit TEST beginnen

ATTACHNAME='TEST.LIB (TEST\*'      Alle Elemente der Lib.'TEST.LIB',  
die mit TEST beginnen

## Datei mit User-Codes

**CODEFILE=** *file* | *lib(typ,elem)*

Die User-Codes können zur Zeit nur für die PDF-Generierung in der Anweisung PDF-LAYOUT (S. 54), Parameter CODE benutzt werden. Für die Verschlüsselung des Mail-Inhalts und der sonstigen Attachments gilt die Translate-Tabelle TRTAB-SEND (S. 81).

Als Standard werden EBCDIC-Zeichen mit der Translate-Tabelle EDF041 to ISO8859-1 (S. 89) in "ASCII-Zeichen" umgewandelt. Intern sind drei verschiedene Varianten vordefiniert:

Hexa-Wert EBCDIC	FB	4F	FD	BB	BC	BD	FF	6A
INT: Internationale Zeichen:	{		}	[	\	]	~	^
DE: Deutsche Umlaute:	ä	ö	ü	Ä	Ö	Ü	ß	
DK Dänische Sonderzeichen:	æ	ø	å	Æ	Ø	Å	ü	Û

In der Codefile können zusätzlich bis zu 30 Code-Varianten definiert werden.

*file*

Dateiname der Codefile.

*lib(typ,elem)*

Bibliotheksname, Typ und Elementname.

In der Codefile können Translate-Tabellen von weiteren Code-Varianten definiert werden. Dabei wird als Grundlage immer die Translate-Tabelle EDF041 to ISO8859-1 (S. 89) verwendet. Nur davon abweichende Zeichen müssen in der Codefile angegeben werden. Aufgrund der Angaben in der Codefile wird eine Translate-Tabelle EBCDIC → ASCII und eine Translate-Tabelle ASCII → EBCDIC erzeugt.

Die Codefile ist wie folgt aufgebaut:

```
<TRTAB-codename>  
eeaa [remark]  
*remark
```

*codename*      Name des Codes. Dieser Name ist in den entsprechenden Anweisungen anzugeben.

eeaa      Translate-Anweisung. Die Reihenfolge der Translate-Anweisungen ist beliebig.

ee      EBCDIC-Zeichen hexadezimal

aa      ASCII-Zeichen hexadezimal

*remark*      Bemerkung

### Beispiel Codefile:

```
* Translate-Tabelle fuer deutsche Umlaute und
* Internationale Zeichen lt. EDF04DR
<TRTAB-EDF04DR>
BBC4 [ = AE GROSS
FBE4 { = AE KLEIN
BCD6 \ = OE GROSS
4FF6 | = OE KLEIN
BDDC ] = UE GROSS
FDFC } = UE KLEIN
FFDF ~ = SS
C07B ù = {
D07D ¡ = }
E05C Û = \
CC5B ö = [
DC5C ü = ]
A17E ¯ = ~
437C ä = |
```

### Dateiformat im Header für Attachments

#### **CONTENT-DESCRIPTION=[Y|N]**

Im Header für die Attachments kann das Format der BS2000-Datei mitgegeben werden. Damit ist es möglich, daß das Format von einem Mail-Client ausgewertet werden kann, z.B. vom Mail-Receiver im BS2000. Manche Mail-Clients zeigen statt des Dateinamens das Dateiformat an. In diesen Fällen sollte Content-description ausgeschaltet werden.

Y Die Format-Beschreibung wird ausgegeben (Standard).

N Die Format-Beschreibung wird nicht ausgegeben.

### Passwort für Verschlüsselung

#### **CRYPTKEY=*password***

Definition eines Passwortes für die Verschlüsselung von Attachments. Dieses Passwort wird verwendet, wenn in einer nachfolgenden Anweisung ATTACH die Option "CRYPTKEY=Y" angegeben ist. Die Anweisung muß vor der entsprechenden ATTACH-Anweisung stehen.

#### *password*

Passwort für die Verschlüsselung. Das Passwort kann max. 56 Byte lang sein.

Weitere Hinweise zum Verschlüsselungs-Verfahren siehe Anlage 2 (S. 90).

Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### Domäne ergänzen

**DOMAINADD**=[Y|N]

Automatisches Ergänzen der Mail-Adressen um die Domäne.

Y Die Mail-Adressen `FROM`, `TO`, `CC` und `BCC` werden um den Domain-Namen einschl. Zeichen "@" ergänzt, falls die Adresse kein Zeichen "@" enthält.

N Die Mail-Adressen werden nicht automatisch ergänzt.

Diese Anweisung kann auch als `SYSDTA`-Anweisung für das Programm `MAIL` angegeben werden. Es ist darauf zu achten, daß diese Anweisung vor den Anweisungen `FROM`, `TO` und `BCC` in der `SYSDTA`-Datei steht.

### Leere Betreffzeile

**EMPTY-SUBJECT**=[Y|N]

Wahlweise kann eine leere Betreffzeile zugelassen werden.

Y Eine leere Betreffzeile ist zulässig.

N Ist die Betreffzeile leer, wird entweder der Text aus der Anweisung `SUBJECT` oder falls kein `SUBJECT` definiert ist, der Text "Message from &USERID" ausgegeben.

### Eigene Mail-Adresse

**FROM**=*fromaddr*

Eigene Mailadresse, fehlt dieses Kommando, wird die Absender-Angabe aus der User-ID, dem Host-Namen und der eigenen Internet-Domäne gebildet, z.B. [USR1.HOST1@opg.de](mailto:USR1.HOST1@opg.de)

Diese Anweisung kann auch als `SYSDTA`-Anweisung für das Programm `MAIL` angegeben werden.

### Behandlung von Isam-Dateien (gilt nur für BS2000)

**ISAMMODE**=[Y|N]

Standardeinstellung für die Option `ISAM=Y|N` der Anweisung `ATTACH` (S. 21).

Y Der Isam-Schlüssel bleibt im Datensatz erhalten. Soweit die Datei direkt vom Programm `MAILR` im BS2000 empfangen wird, wird automatisch wieder eine ISAM-Datei mit den Original-Dateiattributen erstellt.

N Der Isam-Schlüssel wird aus dem Datensatz entfernt. Der `FCBTYPE` für die Empfangsdatei wird auf `SAM` geändert. Soweit die Datei direkt vom Programm `MAILR` im BS2000 empfangen wird, wird automatisch eine `SAM`-Datei erstellt. Die übrigen Dateiattributen bleiben erhalten.

Erfolgt die Übertragung mit dem Modus `BS2`, so wird die Datei immer als `PAM`-Datei übertragen und auf dem Zielsystem wieder als `Isam`-Datei erstellt.

### Anweisungen sperren

**LOCK=cmd[,cmd....]**

Mit diesem Kommando kann der Administrator in der globalen Parameterdatei ein Kommando sperren. Nach Verarbeitung des LOCK-Kommandos ist das angegebene Kommando *cmd* nicht mehr zulässig und wird mit einer Fehlermeldung abgewiesen.

*cmd*

Zur Zeit können die Kommandos ATTACHSIZE, ATTACHNAME, FROM und DOMAIN gesperrt werden. Durch Sperren der Kommandos FROM und DOMAIN kann der Mißbrauch der FROM-Adresse verhindert werden. Das Sperren des DOMAIN-Kommandos ist notwendig, weil sonst durch Weglassen der Domäne in Verbindung mit dem Kommando DOMAINADD eine fremde Domäne angehängt werden könnte, wobei dies in der Regel vom MAIL-Server zurückgewiesen wird.

Beispiel:

```
FROM info@opg.de
ATTACHSIZE=999999
ATTACHNAME=MAIL.*
LOCK FROM, DOMAIN, ATTACHSIZE, ATTACHNAME
```

### Optionen für PDF-Generierung

**PDF-LAYOUT={ layout | DEFAULT } , font , fontsize , page , maxlines , mode , code [ ,margin-left] [ ,margin-top ] [ ,background ] [ ,AUTOSIZE= V | H | YES | NO ]**

*layout*

Name des Layouts. Dieser Name kann in der ATTACH-Anweisung im Parameter MODE=PDF-*layout* angegeben werden. Der Name DEFAULT bezeichnet das Standard-Layout, das zum Tragen kommt, wenn in der ATTACH-Anweisung (S. 21) MODE=PDF angegeben wird.

*font*

Zeichensatz: Zulässig sind COURIER (feste Schrift), HELVETICA (Proportionalsschrift) oder TIMES (Proportionalsschrift). Die Namen der Zeichensätze können bis auf 1 Byte abgekürzt werden.

*fontsize*

Schriftgröße: 1-72

*fontwidth/fontheight*

verschiedene Schriftgrößen für die Zeichenbreite und Zeilenhöhe.

*page*

A4 | A4Q | A5 | A5Q | *höhe* x *breite* in mm

Seitengröße:

A4 = DINA4, A4Q = DINA4 quer, A5 = DINA5, A5Q = DINA5 quer oder *höhe* x *breite* in mm, jeweils dreistellig, z.B. 310x210.

Ist eine Zeile länger als die definierte Breite der Seite, wird die Seitenbreite automatisch angepaßt. Beim Ausdruck kann im Acrobat Reader angegeben werden, daß solche Seiten auf DINA4 bzw. DINA5 verkleinert werden sollen (mit der entsprechenden Anpassung der Schriftgröße).

*maxlines*

Max. Anzahl von Zeilen pro Seite. Bei Dateien mit Drucksteuerzeichen kommt dieser Parameter nur zum Tragen, falls wegen der Schriftgröße die Zeilen nicht auf eine Seite passen.

<i>mode</i>	ASA   SIE   NO [ : <i>col1-col2</i> : ] [ ;CHchannel= <i>line</i> [ ;CHchannel= <i>line</i> ..... ] ] [ ;NCTRL ]
SIE	druckaufbereitete Datei mit Siemens-Steuerzeichen. Das Steuerzeichen wird aus der ersten Spalte der <i>col</i> -Angabe entnommen.
ASA	druckaufbereitete Datei mit ASA-Steuerzeichen. Das Steuerzeichen wird aus der ersten Spalte der <i>col</i> -Angabe entnommen.
NO	Datei ohne Druck-Steuerzeichen. Jeweils nach dem Erreichen der max. Zeilenanzahl wird eine neue Seite erzeugt.
<i>:col1-col2:</i>	Spaltenangabe. Für die PDF-Erzeugung werden nur Daten von Spalte <i>col1</i> bis <i>col2</i> benutzt. Bei <i>mode</i> SIE oder ASA muß das Druck-Steuerzeichen in <i>col1</i> stehen. Bei fehlender Spaltenangabe wird der ganze Datensatz benutzt.
<i>channel</i>	Kanal 1 bis 11. Für die Vorschübe auf Kanal 1 - 11 kann für jeden Kanal eine Zeile definiert werden, auf die bei dem entsprechenden Steuerzeichen positioniert werden soll (SIE: vor dem Drucken = X'C1' bis X'CB', nach dem Drucken = X'81' bis X'8B', ASA: C'1' bis C'9', C'A' und C'B').
<i>line</i>	Zeile 1- 999, auf die bei Kanalvorschub positioniert werden soll. Beispiele: <i>sie</i> ;ch1=3;ch2=10;ch3=30 <i>sie</i> :1-50:;ch1=3;ch5=15 <i>no</i> :10-120: <i>asa</i> ;ch1=3;ch2=10;ch3=30
NCTRL	No Control format character: Die Steuerzeichen X'11' bis X'1F' werden nicht ausgewertet.

Außer den Vorschubsteuerzeichen in Spalte 1 werden in den Daten folgende Format-Steuerzeichen berücksichtigt, falls nicht NCTRL angegeben wurde:

X'11' = Schrift höher stellen  
X'12' = Schrift tiefer stellen  
X'13' = hochgestellt bzw. tiefgestellt ausschalten  
X'14' = Sperrschrift ein  
X'15' = Sperrschrift aus  
X'16'### = Beginn Oktalzeichen, danach abdruckbar 3 Bytes Oktalzeichen  
z.B.  
200 = €-Zeichen  
274 = ¼-Zeichen  
275 = ½-Zeichen  
251 Copyright ©  
256 Registered ®  
231 Trademark ™  
X'17'+C'R' = Schrift Rot  
X'17'+C'B' = Schrift Blau  
X'17'+C'G' = Schrift Grün  
X'17'+C'N' = Normalschrift (Schwarz)  
X'1A###' Neue Fontgröße ##, z.B. X'1A08' = Fontgröße 8  
X'1A00' Originalfontgröße wiederherstellen  
X'1B' = Proportionalschrift Helvetica ein  
X'1C' = Fettschrift ein  
X'1D' = Kursivschrift ein  
X'1E' = Fett-kursiv ein  
X'1F' = Reset (alles auf Ausgangswert, auch implizit 13, 15, 17 und 1a00)

*code*

DE | DK | INT | *usercode*

Angaben für die Code-Umwandlung von 7-Bit-EBCDIC nach ASCII bezüglich der Umlaute und Sonderzeichen. Das Problem besteht darin, daß bestimmte internationale Sonderzeichen und deutsche Umlaute nicht bzw. dänische Sonderzeichen gleichzeitig dargestellt werden können. Dabei geht es um folgende Zeichen:

Dänische Sonderzeichen:	æ	ø	å	Æ	Ø	Å	ü	Ü
Deutsche Umlaute:	ä	ö	ü	Ä	Ö	Ü	ß	
Hexa-Wert	FB	4F	FD	BB	BC	BD	FF	6A
Internationale Zeichen:	{		}	[	\	]	~	^

DE Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN = EDF03IRV, deutsche Variante), aus den entsprechenden EBCDIC-Zeichen sollen Umlaute (ÄÖÜäöüß) erzeugt werden.

DK Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN = EDF03IRV, dänische Variante), aus den entsprechenden EBCDIC-Zeichen sollen dänische Sonderzeichen (ÆØÅæøå) erzeugt werden.

INT Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN = EDF03IRV internationale Variante) oder 8-Bit-EBCDIC (CCSN = EDF041), aus den entsprechenden EBCDIC-Zeichen sollen die Zeichen "{ | } [ \ ] ~" erzeugt werden. Beim 8-Bit-Code EDF041 können die Daten sowohl die Sonderzeichen als auch die Umlaute enthalten.



*usercode* Name eines User-Codes. Die von der Standard-Codierung abweichenden Translate-Anweisungen sind in einer Datei definiert, die mit der Anweisung `CODEFILE` (S. 51) zugewiesen wird.

*margin-top* Seitenrand vom oberen Rand der Seite in mm, Standard = 4.

*margin-left* Seitenrand vom linken Rand der Seite in mm, Standard = 4.

### Hintergrundbilder und Wasserzeichen

Hier können Hintergrundbilder und ein Wasserzeichen angegeben werden. Das Wasserzeichen wird über ev. angegebenen Hintergrundbilder gedruckt.

*background* *jpg-file*[*x,y,w,h*] [ ; *jpg-file*[*x,y,w,h,p*] ... [ ; `WATERMARK:watermark`] ... ]

*jpg-file* Name einer JPG-Datei mit dem Hintergrundbild. Folgende Formate sind zulässig:

<i>bs2-file</i>	Name einer BS2000-Datei.
<i>lib</i> ( <i>typ,elem</i> [ , <i>version</i> ])	Bibliothek, Elementtyp und Elementname.
<i>/posix-file</i>	Name einer POSIX-Datei.

### Position des Bildes:

*x* Entfernung des linken oberen Ecks der Hintergrundbildes vom linken Seitenrand in mm. Bei fehlender Angabe gilt der Wert 0.

*y* Entfernung des linken oberen Ecks der Hintergrundbildes vom oberen Seitenrand in mm. Bei fehlender Angabe gilt der Wert 0.

### Breite und Höhe des Bildes:

Bei fehlender Angabe von Höhe und Breite werden die in den Headerangaben der JPG-Datei gespeicherte Werte benutzt. Das Bild wird dann in der Originalgröße eingefügt. Ist das Bild breiter oder höher wie die ganze Seite, wird es abgeschnitten.

Soll die Breite oder Höhe angepaßt werden, können entweder die beiden Werte oder nur einer der Werte angegeben werden. Wird nur ein Wert angegeben, so wird automatisch der andere Werte im gleichen Verhältnis angepaßt, damit das Bild nicht verzerrt wird. Soll das Bild verzerrt werden, müssen immer die Breite und die Höhe angegeben werden.

Ist zum Beispiel in der JPG-Datei ein Wert von 200 mm Breite und 100 mm Höhe gespeichert und als Breite wird 100 mm angegeben, so wird die Höhe automatisch auf 50 mm geändert. Das Gleiche gilt, wenn nur die Höhe angegeben wird.

*w* Width. Breite des Hintergrundbildes in mm.

*h* Height. Höhe des Hintergrundbildes in mm.

### Seiten, auf denen das Hintergrundbild plziert werden soll.

*p* *page-from* [ - *page-to* ] [ E[VEN] | O[DD] ]

*page-from* Nr. der Seite bzw. erste Seite, auf der das Bild plziert werden soll. \$ bedeutet alle Seiten.

<i>page-to</i>	Letzte Seite, auf der das Bild plaziert werden soll. Die Angabe \$ bedeutet, bis zum Ende der Datei.
EVEN	Das Bild soll nur auf den geraden Seiten plaziert werden.
ODD	Das Bild soll nur auf den ungeraden Seiten plaziert werden.

### Wasserzeichen

<i>watermark</i>	<FS= <i>fontsize</i> > <i>watermark-text</i>
<i>fontsize</i>	Schriftgröße für das Wasserzeichen: 1-256
<i>watermark-text</i>	Text für das Wasserzeichen, max. 32 Bytes

### Automatische Größenanpassung

AUTOSIZE= V | H | YES | NO

H	Horizontal. Falls eine Zeile länger ist als die Seitenbreite, wird die Seitenbreite für die aktuelle Seite entsprechen vergrößert. Die Anpassung gilt nur für die aktuelle Seite.
V	Vertikal. Falls noch Zeilen auf der aktuellen Seite gedruckt werden sollen, die über den Seitenrand hinausgehen, wird die Seitenlänge für die aktuelle Seite entsprechen vergrößert. Die Anpassung gilt nur für die aktuelle Seite.
YES	Falls notwendig, wird die Seitenbreite und die Seitenlänge angepaßt.
NO	Eine automatische Anpassung wird nicht durchgeführt. Falls aufgrund der Anzahl der Zeichen bzw. Zeilen und der eingestellten Font-Größe die Seitenränder überschritten werden, gehen die Zeichen verloren (die Daten sind zwar im Dokument enthalten, werden aber vom Acrobat Reader nicht angezeigt). Diese Option gilt als Standard, falls dieser Parameter nicht angegeben wird.

### Beispiele:

```
pdf-layout=default,courier,8,a4q,60,sie,de,autosize=yes
pdf-layout=default,courier,8,320x170,60,sie,de
pdf-layout=a4pde,courier,10,a4,50,sie,de,25,15
pdf-layout=a4de,courier,10,a4,50,no,de,25,15,t1.jpg,10,10,,,1;t2.jpg,10,10,,,
  2-$EVEN;t3.jpg,10,10,,,2-$ODD
pdf-layout=a4helv,helvetica,8,a4,50,sie,de
pdf-layout=a4helv,helvetica,8,a4,50,sie,de,,,t1.jpg;watermark:<fs=50>Test
pdf-layout=a4qp,courier,8,a4q,60,sie,de,25,15,t1.jpg,10,10;t2.jpg,200,10
```

### Hinweis:

Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### Priorität festlegen

PRIORITY=HIGH | NORM | LOW

Über Priority HIGH/LOW kann erreicht werden, dass die Mail beim Empfänger entsprechend markiert angezeigt wird.

Dieses Kommando kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### Antwort-Adresse

**REPLY**=*recipient*

E-Mail-Adresse, die vom Mail-Client des Empfängers automatisch als Adresse für die Beantwortung des E-Mails vorgeschlagen wird.

Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### Quittung des Empfängers anfordern

**RR**=*recipient*

(Return to Recipient) E-Mail-Adresse für die Quittung des Empfängers, daß das E-Mail empfangen wurde. Die Behandlung dieser Anforderung kann je nach den Einstellungen des Mail-Client unterschiedlich behandelt werden:

- a) Die Quittungsanforderung wird von der Software unterdrückt, d.h. es kommt keine Quittung zurück.
- c) Die Quittung wird automatisch zurückgeschickt.
- c) Das Versenden der Quittung wird vom Empfänger angefordert, wenn die Nachricht geöffnet wird.

Es ist also nicht sichergestellt, daß die Quittung wirklich erstellt wird.

Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### Servername

**SERVER**=*servername* [,AUTH=LOGIN, *username*, *password*]

BCAM-Name des Mailservers bzw. des Relays.

Falls zum Empfangen und Senden verschiedene Server verwendet werden sollen, können im Abschnitt <RECEIVE> ein oder mehrere Server eingetragen werden, die nur für das Empfangen von Mails benutzt werden sollen.

Falls mehrere Server eingetragen sind, gilt folgendes:

Die Server werden in der angegebenen Reihenfolge geprüft, wobei der erste gültige Server benutzt wird. Dadurch ist es möglich, alternative Mailserver anzugeben, die nur benutzt werden, wenn die anderen vorher definierten Server nicht ansprechbar sind. Ist in der privaten Parameterdatei oder in den SYSDTA-Datei ein gültiger Server angegeben, wird dieser benutzt, auch wenn vorher schon ein gültiger Server erkannt wurde.

### Authentifizierung beim Senden von Mails

LOGIN

Username und Password werden mit "basx64" verschlüsselt. Andere Verschlüsselungsmethoden sind im Moment nicht vorgesehen.

*username*

Benutzername, falls vom Server eine Authentifizierung verlangt wird.

*password*

Passwort, falls vom Server eine Authentifizierung verlangt wird.

### Port zum Senden

**SMTPPORT**=*port* Wahlweise kann ein vom Standardport 25 abweichender Port angegeben werden.

*port* Port-Nummer.

### SYSDTA-Anweisungen

Folgende SYSDTA-Anweisungen des Programms MAIL (Senden) können auch als Standardvorgabe in der Parameterdatei MAIL.PAR, Abschnitt <SEND> angegeben werden:

SUBJECT Betreff-Angabe (S. 38)  
FROM Absender-Adresse (S. 28)  
REPLY Antwort-Adresse (S. 37)  
RR Adresse für die Quittung des Empfängers (S. 37)  
CRYPTKEY Standard-Passwort für die Verschlüsselung (S. 27)  
TRTAB Datei mit Translate-Tabelle (nur bis MAIL Version 1.2 siehe S. 38)

### Betreffzeile

**SUBJECT**=*string* Beliebiger String, der als Standard beim Empfänger unter "Subject" zur Identifizierung bzw. als Kurzbeschreibung angezeigt wird, falls beim Versenden das Kommando SUBJECT über SYSDTA fehlt.

*string* Beliebiger Text

Diese Anweisung kann auch als SYSDTA-Anweisung für das Programm MAIL angegeben werden.

### Dateiname im Zielsystem

**TARGETNAME**=*string*|\*SAME [ , OPGC=Y|N]

Standardeinstellung für den Dateinamen auf dem Zielsystem, falls in der Anweisung ATTACH (S. 21) kein Name vergeben wird. Diese Einstellung gilt nicht, soweit die Dateien im BS2000 vom Programm MAILR empfangen werden. Der Dateiname wird in diesem Fall aus Sicherheitsgründen immer vom Programm MAILR erzeugt.

*string* Standard-Dateinamen für die Dateinamen auf dem Zielsystem. Enthält die Zeichenfolge an beliebiger Stelle die Zeichen "###", so werden diese Zeichen durch eine laufende Nummer ersetzt.

\*SAME Die Dateien erhalten auf dem Zielsystem den gleichen Namen wie im BS2000 (ohne Cat-ID und User-ID).

OPGC=Y|N

Y: Soweit ein Attachment komprimiert oder verschlüsselt wird (COMP=Y bzw. KRYPTKEY=... in Anweisung ATTACH (S. 21)), wird an den Dateinamen die Endung ".OPGC" angehängt. Dies hat den Vorteil, daß eine solches Attachment auf einem PC als OGPC-Datei erkannt wird und durch Doppelklick automatisch das Programm OPGCRYPTW geladen wird. Ist in der Anweisung ATTACH explizit ein Dateiname für das Zielsystem angegeben, wird dieser unverändert übernommen. Diese Einstellung gilt als Standard.

N: Die Endung ".OPGC" wird nicht angehängt.

Fehlt die Anweisung, so gilt der Standard "ATT##.TXT".  
Die Dateien erhalten den Namen ATT1.TXT, ATT2.TXT, usw.

### Abschnitt <RECEIVE>

#### Format der Attachment-Datei in Sonderfällen

ATTACH-APPLICATION= I [[F]reclsize] | B

Enthält ein E-Mail ein Attachment, dessen Datenformat nicht erkennbar ist und den Content-Type "application/..." enthält, kann festgelegt werden, wie die Daten zu speichern sind.

Folgende Daten können automatisch identifiziert werden:

- Daten vom Programm MAIL mit MODE=BS2 und COMP=Y
- Daten vom Programm MAIL ohne MODE-Angabe falls das Mail direkt vom BS2000 empfangen wird
- PDF-Dateien (S. 21)
- ZIP-Dateien (S. 31)

Bei diesen Dateien werden die Einstellungen der ATTACH-APPLICATION-Anweisung nicht verwendet.

T

Text. Die Daten werden von ASCII nach EBCDIC übersetzt und in Abhängigkeit der enthaltenen Satzende-Kennzeichen satzweise in einer SAM-Datei (RECFORM=V) gespeichert. Die maximale Satzlänge kann nachfolgend mit dem Parameter *reclsize* angegeben werden. Ist ein Satz länger, als die maximale Satzlänge, werden die Daten in mehrere Sätze aufgeteilt. Fehlt die *reclsize*-Angabe, wird der Wert aus der Anweisung RECORD-LENGTH benutzt.

Falls die Anweisung fehlt, wird "T" (Text) als Standard verwendet, weil manche Mail-Clients Dateien mit einer bestimmten Endung (z.B. TXT) als "application/..." versenden, obwohl es sich um reinen ASCII-Text handelt.

F

Fix. Die Daten werden als ein Datenstrom ohne Strukturierung betrachtet. Ev. vorkommende Satzendezeichen werden nicht ausgewertet. Aus den Daten werden Datensätze in der Länge von *reclsize* erzeugt.

*reclsize*

a) ohne Option "F":

Maximale Satzlänge 1-32.000. Innerhalb dieser Länge wird nach dem Satzende-Kennzeichen in den Daten der Attachments gesucht. Ist ein Satz länger als die maximale Satzlänge, so wird der Satz aufgeteilt.

b) mit Option "F":

Aus dem Datenstrom werden Sätze in der Länge von *resize* erzeugt. Ev. vorkommende Satzendezeichen werden nicht ausgewertet.

Der Blockungsfaktor wird anhand der *resize*-Angabe errechnet. Fehlt die *resize*-Angabe, wird der Wert aus der Anweisung `RECORD-LENGTH` benutzt. Ist auch diese Anweisung nicht vorhanden, gilt der Standardwert von 2028.

B

Binär. Die Daten werden in einer PAM-Datei ohne Satzstruktur gespeichert. Eine Code-Konvertierung wird nicht durchgeführt. Diese Einstellung sollte gewählt werden, wenn Windows-Dateien von Anwendungen, wie WORD oder EXCEL unverändert gespeichert werden sollen, z.B. zur Weiterversendung an einen PC oder zum Anzeigen mit dem URLServer.

Diese Anweisung kann auch im Abschnitt `USER username` für einen bestimmten User definiert werden. Die Angaben im Abschnitt `USER` haben in diesem Fall Vorrang.

### Zyklus zum Prüfen der Mailbox

Dieser Abschnitt enthält allgemeine Anweisungen für das Programm `MAILR`.

`CYCLE=sec` | 0

In Abstand von *sec* Sekunden wird geprüft, ob in der Mailbox neue E-Mails für die in der Parameterdatei enthaltenen User eingegangen sind und ggf. verarbeitet. Fehlt diese Anweisung oder ist der Wert 0 angegeben, so wird die Mailbox nur einmal nach dem Aufruf des Programms geprüft und danach das Programm beendet.

Dieses Kommando kann auch über `SYSDTA` angegeben werden.

### Protokollierung Mailbox-Check

`CYCLE-MSG= Y` | N

In Abhängigkeit der Anweisung `CYCLE` wird in bestimmten Zeitabständen geprüft, ob E-Mails vorhanden sind. Soweit E-Mails vorhanden sind, werden entsprechende Meldung auf `SYSOUT` ausgegeben.

Y

Falls keine E-Mails vorhanden sind, wird die Meldung "Mailbox-Check at *date, time*" auf `SYSOUT` ausgegeben.

N

Falls keine E-Mails vorhanden sind, wird keine Meldung auf `SYSOUT` ausgegeben.

### Dateiname für Index-File

`INDEXFILE=filename`

Sollen die E-Mail's auf dem Server nach dem Abholen nicht gelöscht werden (S. 74), ist es notwendig, daß die internen Mail-ID's in einer Datei gespeichert werden. Der Name dieser Datei kann mit dieser Anweisung bestimmt werden. Fehlt diese Anweisung, wird der Name `MAIL.INDEX` benutzt.

### Port zum Empfangen

**POP3PORT=*port*** Wahlweise kann ein vom Standardport 110 abweichender Port angegeben werden.

*port* Port-Nummer.

### Maximale Satzlänge

**RECORD-LENGTH=[F]*resize***

F Fix. Die Daten werden als ein Datenstrom ohne Strukturierung betrachtet. Ev. vorkommende Satzendezeichen werden nicht ausgewertet. Aus den Daten werden Datensätze in der Länge von *resize* erzeugt.

*resize*

a) ohne Option "F":

Maximale Satzlänge 1-32.000. Innerhalb dieser Länge wird nach dem Satzende-Kennzeichen in den Daten der Attachments gesucht. Ist ein Satz länger als die maximale Satzlänge, so wird der Satz aufgeteilt.

b) mit Option "F":

Aus dem Datenstrom werden Sätze in der Länge von *resize* (1-32.000) erzeugt. Ev. vorkommende Satzendezeichen werden nicht ausgewertet.

Standard = 2028.

Der Blockungsfaktor wird anhand der *resize*-Angabe errechnet.

Enthält ein E-Mail ein Attachment mit dem Content-Type "application/...", so kann mit der Anweisung ATTACH-APPLICATION (S. 61) für diese Gruppe von Attachments ein abweichender Wert definiert werden.

Diese Anweisung kann auch im Abschnitt USER (S. 69) *username* für einen bestimmten User definiert werden. Die Angaben im Abschnitt USER haben in diesem Fall Vorrang.

### Anforderung an MAILR über JV

**REQUEST-JV=*qv-name*** Job-Variablen für Nachrichten an MAILR. Folgende Aktionen sind möglich:

HOLD Die Verarbeitung wird unterbrochen. Ein ev. gerade laufender Verarbeitungszyklus wird erst unterbrochen, wenn die E-Mails aller Pop3-User empfangen wurden. In die Status-JV wird als Bestätigung der Wert HOLD eingetragen.

RESUME Die Verarbeitung wird nach einem HOLD-Request wieder fortgesetzt. In die Status-JV wird als Bestätigung der Wert ACTIVE eingetragen.

TERM Die Verarbeitung wird beendet. Ein ev. gerade laufender Verarbeitungszyklus wird erst unterbrochen, wenn die E-Mails aller Pop3-User empfangen wurden. In die Status-JV wird als Bestätigung der Wert TERM eingetragen.

## Servername zum Empfangen

**SERVER**=*servername*

BCAM-Name des Mailservers bzw. des Relays zum Empfangen.

Falls zum Empfangen und Senden verschiedene Server verwendet werden sollen, können im Abschnitt <RECEIVE> ein oder mehrere Server eingetragen werden, die nur für das Empfangen von Mails benutzt werden sollen. Sind im Abschnitt <RECEIVE> keine Anweisungen SERVER enthalten, gelten zum Empfangen die gleichen Server, wie im Abschnitt <GLOBAL>.

Falls mehrere Server eingetragen sind, werden alle Postfächer auf allen Servern abgefragt.

## Status MAILR über JV

**STATUS-JV**=*jv-name*

Job-Variablen für mit den Status von MAILR. Folgende Statusangaben werden von MAILR gesetzt:

HOLD

Die Verarbeitung wurde nach einer HOLD-Anforderung über die REQUEST-JV unterbrochen.

ACTIVE

Die Verarbeitung der E-Mails ist aktiviert. Dieser Status wird nach dem Start von MAILR bzw. nach einer RESUME-Anforderung über die REQUEST-JV gesetzt.

TERM

Die Verarbeitung wurde beendet.

## Abschnitt <RESTRICT \$*user-id*>

Dieser Abschnitt enthält Einschränkungen pro BS2000-User-ID bzw. TSO-User-ID im z/OS für das Programm MAIL (Senden) bezüglich folgender sicherheitsrelevanten Angaben:

1. zulässige Empfänger-Adressen (TO, CC, BCC)
2. zulässige FROM-Adressen
3. zulässige Dateinamen für Attachments
4. zulässige Dateigröße von Attachments

Alle Anweisungen dieses Abschnitts sind nur in der globalen Parameterdatei erlaubt.

*user-id*

BS2000-User-ID bzw. TSO-User-ID im z/OS, in der das Programm MAIL gestartet wurde: Die durch die nachfolgenden Anweisungen definierten Einschränkungen gelten für die angegebene User-ID.

User-ID:

Die durch die nachfolgenden Anweisungen definierten Einschränkungen gelten für die angegebene User-ID.

OTHERUSER

Die durch die nachfolgenden Anweisungen definierten Einschränkungen gelten für alle User-ID's, für die kein Abschnitt RESTRICT-*user-id* existiert.



Beispiel:

```
<RESTRICT $USER1>          BS2000/TSO-Userid USER1
<RESTRICT $OTHERUSER>     alle anderen User
```

## Zulässige Dateien

**ATTACHNAME=[-]ausw [, [-]ausw.....]**

Mit dieser Anweisung kann festgelegt werden, welche Dateien mit der Anweisung `ATTACH` und `MSGFILE` versandt werden dürfen. Sie kann beliebig oft in diesem Abschnitt vorkommen. Fehlt diese Anweisung, werden alle Dateien zum Versenden zugelassen.

**[-] ausw**

**[ :cat-id: ] [ user-id. | \$\* . ] file | prefix\* | \*suffix | \*string\* | \***

-

Ein Minuszeichen vor dem Auswahlbegriff bedeutet, daß eine zuvor definierte Auswahl von Dateien eingeschränkt werden soll. Alle Dateien der angegebenen User-ID bzw. der RESTRICT-User-ID, die den Auswahlbedingungen entsprechen, sind nicht zulässig. Die negative Auswahl gilt nur für die Dateinamen und nicht für die User-ID.

### Cat-ID und User-ID (gilt nur für BS2000):

**:cat-id:**

Cat-ID für die nachfolgende Auswahlbedingungen. Fehlt die Cat-ID, gelten die Auswahlbedingungen für alle Cat-ID's.

**user-id.**

User-ID für die nachfolgende Auswahlbedingungen. Fehlt die User-ID, gelten die Auswahlbedingungen für die User-ID aus dem Abschnittsnamen RESTRICT.

**\$\* .**

Die Auswahlbedingungen gelten für alle User-ID's.

**file**

*bs2file | lib (typ/elem[/vers]) | zip-archiv (elem) | mvs-file*

### BS2000-Dateien

**bs2file**

Voll- oder teilqualifizierter Dateiname, ggf. mit Cat-ID und User-ID in der Syntax des BS2000-Kommandos `FSTAT`.

**lib (typ/elem[/vers])**

Name einer PLAM-Bibliothek, Elementtyp, Elementnamen und ggf. Version.

**zip-archiv (elem)**

Name eines ZIP-Archivs und der Elementname.

Es ist zu beachten, daß hier der Prefix "`*ZIP:`" nicht anzugeben ist.

### z/OS-Dateien

**mvs-file**

*data-set | pds(member) | 'hlq.data-set' | 'hlq.pds(member)'*

**data-set**

Dateiname auf dem fernen Rechner ohne `HLQ` (high level qualifier). Vom Host-System wird der Dataset-Name automatisch um den `HLQ` ergänzt.

<i>pds(member)</i>	Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended) ohne HLQ. Vom Host-System wird der Member-Name automatisch um den HLQ ergänzt.
' <i>hlq.data-set</i> '	Vollqualifizierter Dateiname auf dem fernen Rechner. Aus diesem String wird für den Filetransfer der Dateiname ' <i>hlq.data-set</i> ' erzeugt.
' <i>hlq.pds(member)</i> '	Vollqualifiziertes Member eines PDS (Partioned Data Set) oder PDSE (Partioned Data Set Extended). Aus diesem String wird für den Filetransfer der Dateiname ' <i>hlq.pds(member)</i> ' erzeugt.

### Auswahl von mehreren Dateien

<i>prefix</i> *	Teilqualifizierter Dateiangabe mit Angabe des Prefix, z.B. <i>DAT</i> * (alle Dateinamen, die mit "DAT" beginnen). Es ist zu beachten, daß bei Bibliothekselementen
<i>*suffix</i>	Teilqualifizierter Dateiname mit Angabe des Suffix, z.B. <i>*SRC</i> (alle Dateinamen, die mit "SRC" enden).
<i>*string</i> *	Teilqualifizierter Dateiangabe mit Angabe einer Zeichenfolge, die in der Mitte des Dateinamens vorkommt, z.B. <i>*DAT</i> * (alle Dateinamen, die an einer beliebigen Stelle "DAT" enthalten).
*	Alle Dateien einer User-ID.

### Hinweise:

Fehlt die Anweisung in einem Abschnitt, so gilt eine ev. vorhandene Anweisung *ATTACHNAME* des Abschnitts *<SEND>*.

Es können beliebig viele Anweisungen *ATTACHNAME* angegeben werden.

Die positiven Suchbegriffe aller *ATTACHNAME*-Anweisungen werden in der definierten Reihenfolge als Oder-Bedingungen verarbeitet. Die negativen Suchbedingungen stellen Einschränkungen einer positiven Auswahl dar. Die negativen Bedingungen müssen daher nach den positiven Bedingungen definiert werden.

### Beispiel:

<i>ATTACHNAME=DATEI1, *.TXT, *.LST</i>	<i>DATEI1</i> und alle Dateien die mit <i>.LST</i> oder mit <i>.TXT</i> enden. Die Dateiauswahl bezieht sich auf die User-ID der <i>RESTRICT</i> -Anweisung
<i>ATTACHNAME=-SRC*, -MAC*</i>	Alle Dateien der User-ID aus der <i>RESTRICT</i> -Anweisung, die durch die vorhergehende Anweisung ausgewählt wurden außer <i>SRC*</i> und <i>MAC*</i>
<i>ATTACHNAME=\$USR1.*, -\$USR1.SEC*</i>	alle Dateien der User-ID <i>\$USR1</i> , außer <i>SEC*</i>

<code>ATTACHNAME=:LST:\$*.*STATISTIK*</code>	Alle Dateien, die den String <code>STATISTIK</code> enthalten und sich in einer beliebigen User-ID mit der
<code>ATTACHNAME=\$*.LST.*</code>	Cat-ID "LST" befinden. Alle Dateien, die mit <code>LST.</code> beginnen in allen User-ID's
<code>ATTACHNAME=-\$TSOS.*</code>	Von der User-ID <code>\$TSOS</code> dürfen keine Dateien versandt werden, auch nicht die Dateien, die mit einer vorherigen Anweisung zugelassen wurden.
<code>ATTACHNAME=LIB1(S/*</code>	Alle S-Elemente der PLAM-Bibliothek LIB1
<code>ATTACHNAME=LIB1(S/SRC.*</code>	Alle S-Elemente der PLAM-Bibliothek LIB1, deren Namen mit <code>SRC.</code> beginnt.
<code>ATTACHNAME=ZIP1(*</code>	Alle Elemente des ZIP-Archivs Bibliothek LIB1
<code>ATTACHNAME='TEST.DAT*'</code>	Alle Dateien, die mit <code>TEST.DAT</code> beginnen
<code>ATTACHNAME=LIB1(TEST*</code>	Alle Elemente der Lib. LIB1, die mit <code>TEST</code> beginnen
<code>ATTACHNAME='TEST.LIB(TEST*</code>	Alle Elemente der Lib.'TEST.LIB', die mit <code>TEST</code> beginnen

### zulässige Dateigröße

**ATTACHSIZE=*nnn* [M|K|H]** Mit dieser Anweisung kann die Größe von Dateien, die als Attachment oder MSG-File (Anweisungen `ATTACH` und `MSGFILE`) versandt werden dürfen, begrenzt werden.

*nnn* Anzahl der Bytes, Killobytes, Megabytes bzw. Halfpages.

M Die Angabe erfolgt in Megabytes (1.048.576 Bytes).

K Die Angabe erfolgt in Killobytes (1024 Bytes).

H Die Angabe erfolgt in Halfpages (2048 Bytes).

Hinweise:

Fehlt die Anweisung in einem Abschnitt, so gilt eine ev. vorhandene Anweisung `ATTACHSIZE` des Abschnitts `<SEND>`.

### Zulässige eigene Mail-Adressen

**FROM=[-]ausw [, [-]ausw.....]**

Eine oder mehrere Mailadressen, die als FROM-Adresse für diese User-ID zulässig sind. Sie kann beliebig oft in diesem Abschnitt vorkommen. Fehlt diese Anweisung, werden alle FROM-Adressen zugelassen.

**[-]ausw**

*mail-adr | prefix\* | \*suffix | \*string\**

Ein Minuszeichen vor dem Auswahlbegriff bedeutet, daß alle Empfänger-Adressen zulässig sind, die nicht den Auswahlbedingungen entsprechen.

*mail-adr*

Vollständige Adresse einer zulässigen eigenen Mail-Adresse, z.B. info@opg.de.

*prefix\**

Teilqualifizierte Mail-Adresse mit Angabe des Prefix, z.B. info@opg.\* (alle Mail-Adressen, die mit "info@opg." beginnen).

*\*suffix*

Teilqualifizierte Mail-Adresse mit Angabe des Suffix, z.B. \*@opg.de (alle Mail-Adressen, die mit "@opg.de" enden).

*\*string\**

Teilqualifizierte Mail-Adresse mit Angabe einer Zeichenfolge, die in der Mitte der Mail-Adresse vorkommt, z.B. \*@opg.\* (alle Mail-Adressen, die an einer beliebigen Stelle "@opg." enthalten) .

Hinweise:

Es können beliebig viele Anweisungen FROM angegeben werden.

Die Suchbegriffe aller FROM-Anweisungen werden in der definierten Reihenfolge als Oder-Bedingungen verarbeitet. Werden positive und negative Bedingungen für einen Benutzer definiert, ist die Reihenfolge der Suchbegriffe von Bedeutung. Zum Beispiel führt die Auswahlbedingung INFO\*, -\*@OGP.DE zum Ergebnis, daß die Adresse INFO@OPG.DE nicht zulässig ist. Bei der Auswahlbedingung -\*@OGP.DE, INFO\* wäre die Adresse INFO@OPG.DE zulässig.

### Zulässige Mail-Empfänger

**TO=[-]ausw [, [-]ausw.....]**

Eine oder mehrere Mail-Adressen bzw. Suchmuster von Empfängern, die als TO-, BCC- oder CC-Adresse für diese User-ID zulässig sind. Sie kann beliebig oft in diesem Abschnitt vorkommen. Fehlt diese Anweisung, werden alle Empfänger zugelassen.

**[-]ausw**

*mail-adr | prefix\* | \*suffix | \*string\**

Ein Minuszeichen vor dem Auswahlbegriff bedeutet, daß alle Empfänger-Adressen zulässig sind, die nicht den Auswahlbedingungen entsprechen.

*mail-adr*

Vollständige Adresse eines zulässigen Mail-Empfängers, z.B. info@opg.de.

- prefix\** Teilqualifizierte Mail-Adresse mit Angabe des Prefix, z.B. `info@opg.*` (alle Mail-Adressen, die mit "`info@opg.`" beginnen).
- \*suffix* Teilqualifizierte Mail-Adresse mit Angabe des Suffix, z.B. `*@opg.de` (alle Mail-Adressen, die mit "`@opg.de`" enden).
- \*string\** Teilqualifizierte Mail-Adresse mit Angabe einer Zeichenfolge, die in der Mitte der Mail-Adresse vorkommt, z.B. `*@opg.*` (alle Mail-Adressen, die an einer beliebigen Stelle "`@opg.`" enthalten) .

Hinweise:

Es können beliebig viele Anweisungen `TO` angegeben werden.

Die Suchbegriffe aller `TO`-Anweisungen werden in der definierten Reihenfolge als Oder-Bedingungen verarbeitet. Werden positive und negative Bedingungen für einen Benutzer definiert, ist die Reihenfolge der Suchbegriffe von Bedeutung. Zum Beispiel führt die Auswahlbedingung `INFO*, -*@OGP.DE` zum Ergebnis, daß die Adresse `INFO@OPG.DE` nicht zulässig ist. Bei der Auswahlbedingung `-*@OGP.DE, INFO*` wäre die Adresse `INFO@OPG.DE` zulässig.

Beispiel:

`TO=INFO.TEST.DE, *@OPG.*`

### Abschnitt <USER *username*>

Für jeden Benutzer, dessen E-Mails empfangen werden sollen, ist ein eigener Abschnitt <USER *username*> zu erstellen. Dieser Abschnitt enthält Angaben über die Zugangsberechtigung zum MAIL-Server (Anweisung `PASS`), die Folgeverarbeitung (Anweisungen `DEFAULT-PROCESSING` und `SUBJECT-PROCESSING`) sowie den Umfang der Ergebnisdateien (Anweisungen `PREFIX`, `ORIGINAL`, `MAILFILE`, `TEXTFILE` und `ATTACHMENT`).

- username* Benutzername, wie er im MAIL-Server definiert ist. Der Benutzername entspricht dem Teil der E-Mail-Adresse vor dem Zeichen "@"

### Datei erstellen

- ATTACHMENT=Y|N** Behandlung der Daten in der Mail-Nachricht, die als Attachment deklariert sind (Content-Disposition: attachment).

- Y Attachment-Daten als BS2000-Datei speichern. Die Daten sind zusätzlich in der Datei `mail-id.OMSG` enthalten.

Der Namen der Datei(en) mit den Attachments besteht aus:

`mail-id.ATTn` `mail-id = prefix.yyyyymmdd.hhmmss`  
`n = lfd. Nummer`

Beispiel: `MAIL.USER1.19991031.121005.ATT1`

N Attachment-Daten nicht als BS2000-Datei speichern. Die Daten sind nur in der Datei *mail-id.OMSG* enthalten.

### Format der Attachment-Datei in Sonderfällen

**ATTACH-APPLICATION=** I **[*[F]*reclsize]** | B

Enthält ein E-Mail ein Attachment, dessen Datenformat nicht erkennbar ist und den Content-Type "application/...." enthält, kann festgelegt werden, wie die Daten zu speichern sind.

Folgende Daten können automatisch identifiziert werden:

- a) Daten vom Programm MAIL mit MODE=BS2 und COMP=Y
- b) Daten vom Programm MAIL ohne MODE-Angabe falls das Mail direkt vom BS2000 empfangen wird
- c) PDF-Dateien (S. 21)
- d) ZIP-Dateien (S. 30)

Bei diesen Dateien werden die Einstellungen der ATTACH-APPLICATION-Anweisung nicht verwendet.

T Text. Die Daten werden von ASCII nach EBCDIC übersetzt und in Abhängigkeit der enthaltenen Satzende-Kennzeichen satzweise in einer SAM-Datei (RECFORM=V) gespeichert. Die maximale Satzlänge kann nachfolgend mit dem Parameter *reclsize* angegeben werden. Ist ein Satz länger, als die maximale Satzlänge, werden die Daten in mehrere Sätze aufgeteilt. Fehlt die *reclsize*-Angabe, wird der Wert aus der Anweisung RECORD-LENGTH benutzt.

Falls die Anweisung fehlt, wird "T" (Text) als Standard verwendet, weil manche Mail-Clients Dateien mit einer bestimmten Endung (z.B. TXT) als "application/..." versenden, obwohl es sich um reinen ASCII-Text handelt.

F Fix. Die Daten werden als ein Datenstrom ohne Strukturierung betrachtet. Ev. vorkommende Satzendezeichen werden nicht ausgewertet. Aus den Daten werden Datensätze in der Länge von *reclsize* erzeugt.

*reclsize* a) ohne Option "F":

Maximale Satzlänge 1-32.000. Innerhalb dieser Länge wird nach dem Satzende-Kennzeichen in den Daten der Attachments gesucht. Ist ein Satz länger als die maximale Satzlänge, so wird der Satz aufgeteilt.

b) mit Option "F":

Aus dem Datenstrom werden Sätze in der Länge von *reclsize* erzeugt. Ev. vorkommende Satzendezeichen werden nicht ausgewertet.

Der Blockungsfaktor wird anhand der *reclsize*-Angabe errechnet. Fehlt die *reclsize*-Angabe, wird der Wert aus der Anweisung RECORD-LENGTH benutzt. Ist auch diese Anweisung nicht vorhanden, gilt der Standardwert von 2028.

B Binär. Die Daten werden in einer PAM-Datei ohne Satzstruktur gespeichert. Eine Code-Konvertierung wird nicht durchgeführt. Diese Einstellung sollte gewählt werden, wenn Windows-Dateien von Anwendungen, wie WORD oder EXCEL unverändert gespeichert werden sollen, z.B. zur Weiterversendung an einen PC oder zum Anzeigen mit dem URLServer.

Die Anweisung kann auch im Abschnitt `RECEIVE` (S. 61) mit Gültigkeit für alle übrigen User definiert werden.

### Authentifizierung beim Empfangen von Mails

**AUTH=LOGIN**

Username und Password werden bei der Authentifizierung mit "base64" verschlüsselt. Andere Verschlüsselungsmethoden sind im Moment nicht vorgesehen. Fehlt diese Anweisung werden die Daten unverschlüsselt gesendet.

### Passwort für Entschlüsselung

**DECRYPTKEY=*password*** Definition eines Passwortes für die Entschlüsselung von Attachments.

### Passwort für Entschlüsselung in Abhängigkeit des Absenders

**DECRYPTKEY<*from*>=*password***

**DECRYPTKEY<*from-prefix*\*>=*password***

**DECRYPTKEY<\**from-suffix*>=*password***

**DECRYPTKEY<\**from-str*\*>=*password***

Wird ein verschlüsseltes Attachment empfangen, wird geprüft, ob die Datei mit diesem Passwort entschlüsselt werden kann und ggf. entschlüsselt. Ist der Key nicht gültig, wird das Attachment verschlüsselt gespeichert und muß mit dem Programm OPGCRYPT entschlüsselt werden.

**<*from*>**

Das Passwort gilt nur für diesen Absender.

**<*from-prefix*\*>**

Das Passwort gilt nur für die Absender, die mit dem String *from-prefix* beginnen.

**<\**from-suffix*>**

Das Passwort gilt nur für die Absender, die mit dem String *from-suffix* enden.

**<\**from-str*\*>**

Das Passwort gilt nur für die Absender, die den String *\*from-str\** enthalten.

***password***

Passwort für die Entschlüsselung. Das Passwort kann max. 56 Byte lang sein.

Weitere Hinweise zum Verschlüsselungs-Verfahren siehe Anlage 2 (S. 95).

Beispiele:

```
DECRYPTKEY=test1
DECRYPTKEY<test@test.de>=test1
DECRYPTKEY<test*>=test1
DECRYPTKEY<*test.de>=test1
DECRYPTKEY<*test*>=test1
```

### Standard Folgeverarbeitung

**DEFAULT-PROCESSING**=/bs2-cmd | mvs-jcl

<i>bs2-cmd</i>	Das BS2000-Kommando, in der Regel das Kommando /ENTER-PROC, wird nach dem Erstellen aller Dateien ausgeführt. Außer dem DEFAULT-PROCESSING können auch Folgeverarbeitungen in Abhängigkeit vom Inhalt der SUBJECT-Angabe definiert werden.
<i>mvs-jcl</i>	Muster-Batch-Job, der nach dem Erstellen aller Dateien ausgeführt wird.  Falls die Bedingungen des <SUBJECT-PROCESSING> erfüllt sind, wird das DEFAULT-PROCESSING nicht mehr ausgeführt.  Das BS2000-Kommando bzw. der Muster-Batch-Job kann an beliebiger Stelle Platzhalter enthalten, die mit den entsprechenden Inhalten substituiert werden. Die Platzhalter beginnen mit dem Zeichen "!".  Folgende Platzhalter sind vorgesehen:
!MAILID	MAIL-ID (S. 40), bestehend aus dem Prefix (S. 77) sowie Datum und Uhrzeit, z.B. USER1.19991031.120505.
!USER	Name des Benutzers. Der Benutzername entspricht dem Teil der E-Mail-Adresse vor dem Zeichen "@"
!FROM	Der Inhalt der FROM-Adresse des E-Mails.
!REPLY	Der Inhalt der REPLY-Adresse des E-Mails.
!RR	Der Inhalt der RR-Adresse (Return to Recipient) des E-Mails. Dieser Parameter beinhaltet eine E-Mail-Adresse, an die eine Quittung versandt werden soll, wenn die Nachricht vom Empfänger gelesen wurde. Ist keine Quittung angefordert worden, enthält der Parameter den Wert "N".
!SUBJECT	Der Inhalt der SUBJECT-Angabe des E-Mails.
!ERROR	N: kein Fehler Y: Fehler bei der Verarbeitung der Attachments, z.B. Fehler DECRYPT.  Die Folgeverarbeitung wird bei einem fehlerhaften Attachment gestartet, weil der Mail-Text und ev. vorhandene andere Attachments trotzdem richtig verarbeitet wurden.
!#ATT	Die Anzahl der Attachments.
!ATTACHFN1 - !ATTACHFN64	Dateinamen der Attachments ( <i>mail-id.ATTn</i> ) auf dem Host, wie sie von MAILR gespeichert werden.
!ATTACHSFN1 - !ATTACHSFN64	Diese Variablen enthalten die Original-Dateinamen der Attachments. Da die Attachments den Namen <i>mail-id.ATTn</i> erhalten, geht der Originalnamen verloren. Falls dieser noch benötigt wird, kann über diese Variablen der Original-Dateinamen ermittelt werden.  Hinweis:  Diese Informationen sind auch in den Header-Sätzen vor jedem E-Mail enthalten, falls alle E-Mails in eine Datei geschrieben werden, siehe dazu auch die Beschreibung der Anweisung TEXTFILE (S. 80).



### Beispiel BS2000:

```
/ENTER-PROC EMAIL.SUCC, (!MAILAD, !USER, '!FROM', 'SUBJECT', !#ATT, !ERROR)
```

### Prozedur EMAIL.SUCC:

Die Prozedur EMAIL.SUCC prüft, welche Dateien erstellt worden sind und erzeugt eine Protokolldatei. Danach wird in Abhängigkeit des Parameters &ANSWER ein Antwort-Mail an die Adresse &ADDR gesandt. Zuletzt werden alle Mail-Dateien in eine PLAM-Bibliothek aufgenommen mit dem Namen EMAIL.&USER.YYYY-MM-DD. Diese Beispiel-Prozedur ist auch in der ausgelieferten Datei MAILLIB enthalten.

```
/PROC A, (&MAILID, -  
/      &USER, -  
/      &FROM, -  
/      &SUB, -  
/      &ATTNR, -  
/      &ERROR, -  
/      &ANSWER=N, -  
/      &ADDR='HARLASS@OPG.DE'-  
/      ), SUBDTA=&  
/SYSFILE SYSOUT=PROT.&MAILID  
/REMARK NUMBER OF ATTACHED FILES: &ATTNR  
/SYSFILE SYSDTA=(SYSCMD)  
/.PROCESS REMARK  
/SETSW OFF=(1,31)  
/REMARK *** CHECK EXITING MAIL FILES *****  
/FS &MAILID..  
/STEP  
/.PROTOCOL REMARK PROCESS PROTOCOL OF PROCESSING  
/SYSFILE SYSOUT=(PRIMARY)  
/REMARK SEND SYSOUT TO ADMINISTRATOR  
/SKIP .ANSWER&ANSWER  
/.ANSWERY REMARK  
/EXEC L=SRCLIB.MAIL,E=MAIL  
TO &ADDR  
FROM OSD.client@C80OPG1  
SUBJECT Mail received: protocol see text below  
MSGBEG  
Subject: &SUB  
From: &FROM  
MSGEND  
MSGFILE PROT.&MAILID  
END  
/.ANSWERN REMARK  
/DCLJV #TEMP.DATE  
/SETJV (#TEMP.DATE,1,10),C'&($SYSJV.DATE4)'  
/START-LMS  
//OPEN-LIBRARY LIBRARY=EMAIL.&USER..&(#TEMP.DATE),-  
//      MODE=*UPDATE (STATE=*ANY)  
//ADD-ELEMENT FROM-FILE=PROT.&MAILID,-  
//TO-ELEMENT=*LIBRARY-ELEMENT (ELEMENT=&MAILID..PROT,-  
//      TYPE=S),DELETE-SOURCE=*YES  
//END  
/ENDP
```

### Beispiel z/OS:

```
MAIL.JOB (SUCCP)
```

Prozedur SUCCP:

Die Prozedur SUCCP sendet eine E-Mail an den Empfänger als Empfangsbestätigung.

```
//MAILSUC JOB (ACCNT),MSGCLASS=H,CLASS=A,COND=(4,LT),
//    REGION=(32M)
//* FOLGEVERARBEITUNG FUER EINGEHENDE MAILS
//SET1 SET MAILID=N,FROM=N
//JOB LIB DD DSN='MAIL.LOAD',DISP=SHR
//MAIL1 EXEC PGM=MAILS
//MAILPAR DD DSN=MAIL.PAR,DISP=SHR
//SYSABEND DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
TO &FROM
SUBJECT Re: &SUBJECT
MSGBEG
Ihre Mail

&MAILID
&FROM
```

wurde erfolgreich verarbeitet.

```
MSGEND
END
//
```

### E-Mail auf Server löschen

**DELETE=Y** | N

E-Mail auf dem Server nach dem Abholen löschen / nicht löschen.

Y

Nach dem Abholen des E-Mail's wird es auf dem Server gelöscht.

N

Das E-Mail bleibt auf dem Server gespeichert. In diesem Fall wird eine Datei benötigt, in der die internen Mail-ID's aller bisher abgeholten Mail's gespeichert werden. Der Name dieser Datei kann mit der Anweisung `INDEXFILE` (S. 62) im Abschnitt `<RECEIVE>` festgelegt werden (Standard = `MAIL.INDEX`).

### Behandlung von Leersätzen

**EMPTY-RECORD=Y** | N | B

Leersätze ausgeben / nicht ausgeben / als Blank-Satz schreiben. Im Betriebssystem Windows und Unix sind Leersätze üblich und werden in den Editoren und View-Programmen als leere Zeile dargestellt. Im BS2000 werden Leersätze im EDT nicht angezeigt. Im CFS-Editor werden Leersätze dargestellt.

Y

Leersätze werden in den Text-Dateien und Attachment-Dateien als variabler Satz mit Satzlänge 4 ohne Daten erzeugt.

N

Leersätze werden in den Text-Dateien und Attachment-Dateien nicht erzeugt.

B Blank: Leersätze werden in den Text-Dateien und Attachment-Dateien als Datensatz mit einem Blank erzeugt.

### Original-Dateinamen der Attachments

**FNAMFILE= Y|N|*file***

Diese Datei enthält die Original-Dateinamen der Attachments. Da die Attachments im BS2000 den Namen *mail-id.ATTn* erhalten, geht der Originalnamen verloren. Falls dieser noch benötigt wird, kann der Original-Dateinamen in dieser Datei gespeichert werden und steht hier für weitere Auswertungen zur Verfügung.

Y Die Original-Dateinamen werden in der Datei mit dem Namen *prefix.yyyyymmdd.hhmmss.FNAM* gespeichert.

Diese Datei enthält für jedes Attachment einen Datensatz, der wie folgt aufgebaut ist:

- |         |   |
|---------|---|
| 1-54    | Dateiname der von MAILR erzeugten Attachment-Datei <i>mail-id.ATTn</i>  |
| 55-108  | Original-Dateinamen, wie vom MAIL-Server mitgeteilt. bzw. bei mit OPGCRYPT verschlüsselten Dateien der Originalnamen aus dem OPGCRYPT Datei-Header (Originalname vor der Verschlüsselung).  |
| 109-113 | Format der Datei beim Empfangen durch MAILR:<br>" " Normale Datei, nicht komprimiert und nicht verschlüsselt.<br>"COMP " Mit OPGCRYPT komprimierte Datei.<br>"CRYPT" Mit OPGCRYPT verschlüsselte Datei.   |
| 114-118 | Format der Datei nach einer durch MAILR ev. durchgeführten Konvertierung (entkomprimieren oder entschlüsseln):<br>" " Normale Datei, nicht komprimiert und nicht verschlüsselt.<br>"COMP " Mit OPGCRYPT komprimierte Datei.<br>"CRYPT" Mit OPGCRYPT verschlüsselte Datei. |

N Die **FNAMFILE** wird nicht erstellt.

Beispiel: MAIL.USER1.19991031.121005.FNAM

*file*

Die MAIL-Nachricht wird in der Datei *file* im OPEN-EXTEND-Modus gespeichert. Bei dieser Variante werden die Daten aller E-Mails in die gleiche Datei geschrieben. Vor dem Beginn jedes E-Mails werden zur Trennung folgende Header-Sätze geschrieben:

```
*/OPG-MAIL//START-HEADER
*/OPG-MAIL//AT=date.time (yyyymmdd.hhmmss)
*/OPG-MAIL//PREFIX=prefix (Anw. PREFIX (S. 77))
*/OPG-MAIL//USER=user (Pop3-Benutzer)
*/OPG-MAIL//END-HEADER
```

### HTMLFILE erstellen

**HTMLFILE=Y|N|file**

HTMLFILE erstellen / nicht erstellen. Diese Datei enthält die MAIL-Nachricht einschl. aller SMTP-Anweisungen ohne Daten für die Attachments. Vor allem die Angaben FROM und SUBJECT sind unter anderem in dieser Datei enthalten. Es ist jedoch zu beachten, daß diese Angaben auch in der Folgeverarbeitung als Parameter zur Verfügung stehen.

N

Die HTMLFILE wird nicht erstellt.

Y

Die MAIL-Nachricht wird in der Datei mit dem Namen *prefix.yyyymmdd.hhmmss.HTML* gespeichert.

Beispiel: MAIL.USER1.19991031.121005.HTML

*file*

Die MAIL-Nachricht wird in der Datei *file* im OPEN-EXTEND-Modus gespeichert. Bei dieser Variante werden die Daten aller E-Mails in die gleiche Datei geschrieben. Vor dem Beginn jedes E-Mails werden zur Trennung folgende Header-Sätze geschrieben:

```
* //OPG-MAIL//START-HEADER
* //OPG-MAIL//AT=date.time (yyymmdd.hhmmss)
* //OPG-MAIL//PREFIX=prefix (Anw. PREFIX (S. 77))
* //OPG-MAIL//USER=user (Pop3-Benutzer)
* //OPG-MAIL//END-HEADER
```

### MAILFILE Datei erstellen

**MAILFILE=Y|N|file**

MAILFILE erstellen / nicht erstellen. Diese Datei enthält die MAIL-Nachricht einschl. aller SMTP-Anweisungen ohne Daten für die Attachments. Vor allem die Angaben FROM und SUBJECT sind unter anderem in dieser Datei enthalten. Es ist jedoch zu beachten, daß diese Angaben auch in der Folgeverarbeitung als Parameter zur Verfügung stehen.

N

Die Datei mit der MAIL-Nachricht wird nicht erstellt.

Y

Die MAIL-Nachricht wird in der Datei mit dem Namen *prefix.yyyymmdd.hhmmss.MAIL* gespeichert.

Beispiel: MAIL.USER1.19991031.121005.MAIL

*file*

Die MAIL-Nachricht wird in der Datei *file* im OPEN-EXTEND-Modus gespeichert. Bei dieser Variante werden die Daten aller E-Mails in die gleiche Datei geschrieben. Vor dem Beginn jedes E-Mails werden zur Trennung folgende Header-Sätze geschrieben:

```
* //OPG-MAIL//START-HEADER
* //OPG-MAIL//AT=date.time (yyymmdd.hhmmss)
* //OPG-MAIL//PREFIX=prefix (Anw. PREFIX (S. 77))
* //OPG-MAIL//USER=user (Pop3-Benutzer)
* //OPG-MAIL//END-HEADER
```

## Datei mit Originalnachricht erstellen

**ORIGINAL=Y|N|*file***

Original-Datei erstellen / nicht erstellen. Diese Datei enthält alle Daten der ursprünglichen MAIL-Nachricht, wie sie auch auf dem MAIL-Server gespeichert ist. Der einzige Unterschied besteht darin, daß die Daten in EBCDIC codiert sind. Sie ist hauptsächlich für Zwecke der Analyse vorgesehen.

N

Die Datei mit der Original-Nachricht wird nicht erstellt.

Y

Die Original-Nachricht wird in der Datei mit dem Namen *prefix.yyyyymmdd.hhmmss.OMSG* gespeichert.

Beispiel: MAIL.USER1.19991031.121005.OMSG

*file*

Die Original-Nachricht wird in der Datei *file* im OPEN-EXTEND-Modus gespeichert. Bei dieser Variante werden die Daten aller E-Mails in die gleiche Datei geschrieben. Vor dem Beginn jedes E-Mails werden zur Trennung folgende Header-Sätze geschrieben:

```
*//OPG-MAIL//START-HEADER
*//OPG-MAIL//AT=date.time (yyyymmdd.hhmmss)
*//OPG-MAIL//PREFIX=prefix (Anw. PREFIX (S. 77))
*//OPG-MAIL//USER=user (Pop3-Benutzer)
*//OPG-MAIL//END-HEADER
```

## Paßwort für Benutzer

**PASS=*password***

Passwort für den im Abschnittsnamen definierten User für den Zugang zum Mail-Server.

## POP3 Benutzer für Anmeldung

**POP3USER=*username***

Falls der User-Name im Abschnitt <USER ...> nicht mit dem Anmeldenamen für das POP3-Postfach übereinstimmt, so kann hier der zu verwendende Anmeldenamen für den Zugang zum Mail-Server angegeben werden.

## PREFIX für Dateinamen

**PREFIX=*prefix* | '*prefix*'**

Prefix für die Dateinamen, die von MAILR erzeugt werden.

*prefix*

Aus dem Prefix, dem Datum und der Uhrzeit wird die MAIL-ID gebildet und zwar in der Form *prefix.yyyyymmdd.hhmmss.suffix*

'*prefix*'

Absoluter Dateiname einschl. HLQ für z/OS. Aus dem Prefix, dem Datum und der Uhrzeit wird die MAIL-ID gebildet und zwar in der Form '*prefix.yyyyymmdd.hhmmss.suffix*'..

Aus der MAIL-ID und einem zusätzlichen Suffix werden die Dateinamen gebildet. Als *suffix* wird die jeweilige Dateiarart wie z.B. "ATT1", "TEXT", "HTML" usw. verwendet.

## Maximale Satzlänge

### RECORD-LENGTH=[F]*resize*

**F** Fix. Die Daten werden als ein Datenstrom ohne Strukturierung betrachtet. Ev. vorkommende Satzendezeichen werden nicht ausgewertet. Aus den Daten werden Datensätze in der Länge von *resize* erzeugt.

*resize*

a) ohne Option "F":

Maximale Satzlänge 1-32.000. Innerhalb dieser Länge wird nach dem Satzende-Kennzeichen in den Daten der Attachments gesucht. Ist ein Satz länger als die maximale Satzlänge, so wird der Satz aufgeteilt.

b) mit Option "F":

Aus dem Datenstrom werden Sätze in der Länge von *resize* (1-32.000) erzeugt. Ev. vorkommende Satzendezeichen werden nicht ausgewertet.

Standard = 2028.

Der Blockungsfaktor wird anhand der *resize*-Angabe errechnet.

Enthält ein E-Mail ein Attachment mit dem Content-Type "application/...", so kann mit der Anweisung ATTACH-APPLICATION (S. 58) für diese Gruppe von Attachments ein abweichender Wert definiert werden.

Die Anweisung kann auch im Abschnitt RECEIVE (S. 61) mit Gültigkeit für alle übrigen User definiert werden.

## Subject-Folgeverarbeitung

### SUBJECT-PROCESSING<*search*>=*bs2-cmd* | *mvs-jcl*

Neben dem DEFAULT-PROCESSING können zusätzlich in Abhängigkeit des Inhalts der SUBJECT-Angabe Folgeverarbeitungen definiert werden. Falls die Bedingung für eine SUBJECT-Folgeverarbeitung erfüllt ist, wird die DEFAULT-Folgeverarbeitung nicht mehr ausgeführt.

*search*

Die Folgeverarbeitung wird nur durchgeführt, falls die Suchbedingung erfüllt ist. Format: [\*]*string*[\*]

*string* Der Suchbegriff muß exakt mit dem Inhalt von SUBJECT übereinstimmen.

*string*\* Der Inhalt von SUBJECT muß mit dem Suchbegriff beginnen.

\**string* Der Inhalt von SUBJECT muß mit dem Suchbegriff enden.

\**string*\* Der Suchbegriff muß an einer beliebigen Stelle im SUBJECT vorkommen.

*bs2-cmd*

Das BS2000-Kommando, in der Regel das Kommando /ENTER-PROC, wird nach dem Erstellen aller Dateien ausgeführt, falls die Suchbedingung erfüllt ist.

*mvs-jcl*

Muster-Batch-Job, der nach dem Erstellen aller Dateien ausgeführt wird.

Das BS2000-Kommando bzw. der Muster-Batch-Job kann an beliebiger Stelle Platzhalter enthalten, die mit den entsprechenden Inhalten substituiert werden. Die Platzhalter beginnen mit dem Zeichen "!".

Folgende Platzhalter sind vorgesehen:

!MAILID	MAIL-ID (S. 38), bestehend aus dem Prefix (S. 74) sowie Datum und Uhrzeit, z.B. USER1.19991031.120505.
!USER	Name des Benutzers. Der Benutzername entspricht dem Teil der E-Mail-Adresse vor dem Zeichen "@"
!FROM	Der Inhalt der FROM-Adresse des E-Mails.
!REPLY	Der Inhalt der REPLY-Adresse des E-Mails.
!RR	Der Inhalt der RR-Adresse (Return to Recipient) des E-Mails. Dieser Parameter beinhaltet eine E-Mail-Adresse, an die eine Quittung versandt werden soll, wenn die Nachricht vom Empfänger gelesen wurde. Ist keine Quittung angefordert worden, enthält der Parameter den Wert "N".
!SUBJECT	Der Inhalt der SUBJECT-Angabe des E-Mails.
!ERROR	N: kein Fehler Y: Fehler bei der Verarbeitung der Attachments, z.B. Fehler DECRYPT.  Die Folgeverarbeitung wird bei einem fehlerhaften Attachment gestartet, weil der Mail-Text und ev. vorhandene andere Attachments trotzdem richtig verarbeitet wurden.
!#ATT	Die Anzahl der Attachments.
!ATTACHFN1 - !ATTACHFN64	Dateinamen der Attachments ( <i>mail-id.ATTn</i> ) auf dem Host, wie sie von MAILR gespeichert werden.
!ATTACHSFN1 - !ATTACHSFN64	

Diese Variablen enthalten die Original-Dateinamen der Attachments. Da die Attachments den Namen *mail-id.ATTn* erhalten, geht der Originalnamen verloren. Falls dieser noch benötigt wird, kann über diese Variablen der Original-Dateinamen ermittelt werden.

Hinweis:

Diese Informationen sind auch in den Header-Sätzen vor jedem E-Mail enthalten, falls alle E-Mails in eine Datei geschrieben werden, siehe dazu auch die Beschreibung der Anweisung TEXTFILE (S. 80).

Beispiel:

Auszug aus Parameterdatei:

```
DOMAIN=OPG.DE
PREFIX=EMAIL.USER1
SUBJECT-PROCESSING<BESTELLUNG*>=/ENTER-PROC
EMAIL.S1, (ID=!MAILID, FROM=!FROM, SUB='!SUBJECT')
```

Inhalt des E-Mail (auszugsweise) vom 31.10.1999, 12:05:05 Uhr:

Absender: TEST@TEST.DE  
Empfangsadresse: ORDER@OPG.DE  
SUBJECT: BESTELLUNG Software MAIL  
Attachments: 1 Datei mit Bestelldaten

Aufgrund dieser Anweisungen in der Parameterdatei und den E-MAIL-Daten wird folgendes BS2000-Kommando ausgeführt:

```
/ENTER-PROC EMAIL.S1, (ID=EMAIL.USER1.19991031.120505,  
FROM=TEST@TEST.DE, SUB='BESTELLUNG Software MAIL')
```

### Textdatei erstellen

**TEXTFILE=Y|N|*file***

Textdatei erstellen / nicht erstellen. Diese Datei enthält nur den MAIL-Text ohne SMTP-Anweisungen und ohne Daten für die Attachments.

N

Die Textdatei wird nicht erstellt.

Y

Der MAIL-Text wird in der Datei mit dem Namen *prefix.yyyymmdd.hhmmss.TEXT* gespeichert.

Beispiel: MAIL.USER1.19991031.121005.TAEXT

*file*

Der MAIL-Text wird in der Datei *file* im OPEN-EXTEND-Modus gespeichert. Bei dieser Variante werden die Daten aller E-Mails in die gleiche Datei geschrieben. Vor dem Beginn jedes E-Mails werden zur Trennung folgende Header-Sätze geschrieben:

```
* //OPG-MAIL//START-HEADER  
* //OPG-MAIL//AT=date.time (yyyymmdd.hhmmss)  
* //OPG-MAIL//PREFIX=prefix (Anw. PREFIX (S. 77))  
* //OPG-MAIL//USER=user (Pop3-Benutzer)  
* //OPG-MAIL//SUBJECT=subject (Betreff-Text)  
* //OPG-MAIL//FROM=from-adr (Absender-Adresse)  
* //OPG-MAIL//REPLY=reply-adr (nur, falls vorhanden)  
* //OPG-MAIL//RR=rr-adr (nur, falls vorhanden)  
* //OPG-MAIL//END-HEADER
```

Damit stehen für eine ev. automatische Verarbeitung der E-Mails die gleichen Informationen zur Verfügung wie bei der Folgeverarbeitung über DEFAULT-PROCESSING (S. 72) oder SUBJECT-PROCESSING (S. 78) mit Ausnahme des Parameters !#ATT. Der Parameter !MAILID wird getrennt in den Header-Sätzen PREFIX und AT dargestellt. Die Namen von ev. vorhandenen Attachments kann aus den Informationen *prefix.date.time.ATTn* erzeugt werden (*n* = laufende Nummer).

### Automatisch entkomprimieren

**UNCOMPRESS=Y|N**

Y: Wird ein komprimiertes Attachment im OPGC-Format empfangen, wird es automatisch entkomprimiert.

N: Ein komprimiertes Attachment wird unverändert gespeichert und muß ggf. bei Bedarf mit dem Programm OPTCRYPT entkomprimiert werden.

Weitere Hinweise zum Verschlüsselungs-Verfahren siehe Anlage 2 (S. 95).



## Abschnitt <TRTAB-RECEIVE>

Die Code-Umwandlung erfolgt von ASCII ISO 8859 nach BS2000-Code EBCDIC.DF.04-1 (CCSN: EDF041). Code-Tabellen siehe Anlage1 (S. 89).

Soll von dieser Tabelle abgewichen werden, muß für jedes Zeichen, das abweichend übersetzt werden soll, eine Translate-Anweisung angegeben werden. Die Translate-Anweisungen überschreiben die Standardtabelle. Es müssen also nur die Zeichen angegeben werden, die abweichend behandelt werden sollen.

Deutsche Umlaute oder sonstige nationale Sonderzeichen können am Bildschirm nach dieser Tabelle nur dargestellt werden, wenn eine 8-Bit-Terminal eingestellt ist (im BS2000 und in der Terminal-Emulation). In der Regel ist ein nationales 7-Bit-Terminal eingestellt. Für den deutschen 7-Bit-Code sind in der ausgelieferten MAIL.PAR bereits Kommentarzeilen für die Sonderzeichen enthalten. Zum Aktivieren dieser Variante müssen Sie nur die Kommentarzeichen ("\*" in der ersten Spalte) entfernen.

Format der Anweisungen:

*xyyy* [remark]

xx ASCII-Zeichen  
yy = EBCDIC-Zeichen  
remark = wahlweise Bemerkung

Beispiel:

```
C4BB      UMLAUT GROSS A  
D6BC      UMLAUT KLEIN A
```

## Abschnitt <TRTAB-SEND>

Die Code-Umwandlung erfolgt von BS2000-Code EBCDIC.DF.04-1 (CCSN: EDF041) nach ASCII ISO 8859-1 CP1252. Code-Tabellen siehe Anlage1 (S. 89).

Soll von dieser Tabelle abgewichen werden, muß für jedes Zeichen, das abweichend übersetzt werden soll, eine Translate-Anweisung angegeben werden. Die Translate-Anweisungen überschreiben die Standardtabelle. Es müssen also nur die Zeichen angegeben werden, die abweichend behandelt werden sollen.

Deutsche Umlaute oder sonstige nationale Sonderzeichen können am Bildschirm nach dieser Tabelle nur dargestellt werden, wenn ein 8-Bit-Terminal eingestellt ist (im BS2000 und in der Terminal-Emulation). In der Regel ist ein nationales 7-Bit-Terminal eingestellt. Für den deutschen 7-Bit-Code sind in der ausgelieferten MAIL.PAR bereits Kommentarzeilen für die Sonderzeichen enthalten. Zum Aktivieren dieser Variante müssen Sie nur die Kommentarzeichen ("\*" in der ersten Spalte) entfernen.

Hier ist unbedingt darauf zu achten, daß die EBCDIC-Zeichen immer in den Code ISO 8859-1 übersetzt werden, weil diese Code-Variante vom Programm MAIL als Charset im Content-Type benutzt wird.

Hinweis:

Soll der alte Translate-Modus, wie er bis zur Version 1.2 des Programms MAIL verwendet wurde, aktiviert werden, ohne daß eine abweichende TRTAB zugewiesen wird, muß der Abschnittsname <TRTAB\_SEND> in eine Kommentarzeile (\*<TRTAB\_SEND>) geändert oder gelöscht werden. Siehe hierzu auch die Anweisung TRTAB (S. 38).

Format der Anweisungen:

*xxyy* [*remark*]

xx = EBCDIC-Zeichen

yy = ASCII-Zeichen

remark = wahlweise Bemerkung

Beispiel:

```
BBC4    UMLAUT GROSS Ae
BCD6    UMLAUT KLEIN ae
```

### 7 Aufruf von MAIL als Unterprogramm

#### MAILUP oder OPGMAILS (Senden)

**MAILUP** Alle notwendigen Funktionen werden zum aufrufenden Programm dazugebunden. Falls vom aufrufenden Programm andere Socket- und CRTE-Versionen benutzt werden als von MAILUP, können Probleme auftreten. In diesem Fall sollte die Schnittstelle OPGMAILS verwendet werden.

**OPGMAILS** Es wird nur der Verbindungsmodul OPGMAILS zum aufrufenden Programm gebunden. Die Funktionen werden über den ASTI-Service OPGMAILCLNT (S. 17) ausgeführt. Dieser Service muß gestartet sein.

#### Parameter:

Die Module MAILUP und OPGMAILS werden entsprechend der Standard-Programmverknüpfungsregeln aufgerufen. Die Parameter sind für beide Module identisch. Die Module können auch von höheren Programmiersprachen aufgerufen werden. Die Register müssen wie folgt geladen sein:

- (R1) A (PARAMETERLISTE)  
Die Parameterliste enthält folgende Adressen:  
A (RECORD-TABLE) Tabelle aller Anweisungen (siehe unten)  
A (MSGIND) Umfang der Protokollierung  
A (0) oder A (PARAM-TABLE) Tabelle Anweisungen aus Parameterdatei  
A (0) oder A (MSGAREAA) Fehlermeldungen aus MAILUP (nur falls MSGIND = A)  
A (0) oder A (L'MSGAREAA) Länge Bereich Fehlermeldungen (nur falls MSGIND = A)
- (R13) A (SAVEAREA)  
Register-Sicherstellungsbereich (18 Worte, DC 18F'0'), der vom Aufrufer erstellt werden muß. MAILUP sichert dort die Register.
- (R14) A (RETURN)  
Rücksprungadresse im rufenden Programm. Nach Beenden der MAIL-Funktionen wird das Programm an dieser Adresse fortgesetzt.
- (R15) V (MAILUP)  
Nach dem Rücksprung enthält das Register 15 folgende Werte
- |   |  |
|---|--|
| 0 | Verarbeitung ohne Fehler.  |
| 4 | Verarbeitung mit Fehlern, die nicht zum Abbruch führen. Das Mail wurde versandt. |
| 8 | Verarbeitung mit Fehlern. Die Verarbeitung wurde abgebrochen.                    |

#### RECORD-TABLE

Beliebig viele Anweisungen für MAILUP, wie sie im Kapitel 4 beschrieben sind, jeweils mit 4 Byte Satzlängefeld (2 Byte Satzlänge + X'0000'). Am Ende der Tabelle muß X'0000' stehen.

**MSGIND** Umfang der Meldungen, die auf SYSOUT ausgegeben werden.  
N Es werden nur Fehlermeldungen auf SYSOUT ausgegeben.

- Y Außer den Fehlermeldungen werden verschiedene Aktionen protokolliert, wie z.B. die Verarbeitung der Parameterdateien und Attachments.
- L Für Diagnosezwecke werden zusätzlich alle Nachrichten, die mit dem Mailserver ausgetauscht werden, auf SYSOUT ausgegeben.
- R Außer den Standardmeldungen (wie Y) werden alle Anweisungen der RECORD-TABLE protokolliert (wie PROC A in einer DO-Prozedur).
- Axyz Fehlermeldungen in den übergebene Bereich MSGAREA schreiben. Dazu müssen folgende Optionen angegeben werden:
- x Y: Fehlermeldungen nach SYSOUT und in MSGAREA ausgeben.  
N: Fehlermeldungen nur in MSGAREA ausgeben.
  - y L: Fehlermeldungen mit Satzlängenfeld und Control-Character für WROUT ausgeben in der Form X'LLLLL4040CC' (LLLLL=Satzlänge, CC=Control-Character).  
S: Fehlermeldungen als C-String ausgeben, d.h. jede einzelne Fehlermeldung wird mit X'00' abgeschlossen. Die Fehlermeldung enthält am Schluß die Steuerzeichen für einen Zeilenvorschub und kann mit PRINTF ausgegeben werden.  
N: Fehlermeldungen mit "New Line" ausgeben, d.h. jede einzelne Fehlermeldung wird mit den Steuerzeichen für einen Zeilenvorschub abgeschlossen. Nach der letzten Fehlermeldung wird ein X'00' angehängt. Alle Fehlermeldungen können mit einem einzigen PRINTF ausgegeben werden. Nach jeder einzelnen Fehlermeldung wird ein Zeilenvorschub ausgegeben.

### PARAM-TABLE

Beliebig viele Anweisungen aus der Parameterdatei, wie sie im Kapitel 6 beschrieben sind, jeweils mit 4 Byte Satzlängenfeld (2 Byte Satzlänge + X'0000'). Die globale und private Parameterdatei wird in diesem Fall ignoriert. Am Ende der Tabelle muß X'0000' stehen. Dieser Parameter wurde für Anwendungen eingeführt, in denen die Parameter für MAILUP vom aufrufenden Programm erzeugt werden sollen ohne daß eine eigene Parameterdatei für MAILUP notwendig ist.

### MSGAREA

Bereich, in den von UPMail Fehlermeldungen gespeichert werden. Die Länge des Bereichs ist im nachfolgenden Parameter zu definieren. Die einzelnen Fehlermeldungen werden je nach den Angaben im Parameter MSGIND im WROUT-Format (Beginn mit 4 Byte Satzlänge und 1 Byte Control-Character) oder im PRINTF-Format (Abschluß mit den Steuerzeichen für neue Zeile) erstellt und werden mit 4X'00' abgeschlossen.

### L'MSGAREA

Länge von MSGAREA. Fehlermeldungen, die nicht mehr in den Bereich MSGAREA passen, werden abgeschnitten.

Ein vollständiges Beispiel-Programm in Assembler finden Sie in der LMS-Bibliothek MAILIB, Element S/CALLMAIL.SRC.

### Beispiel:

```
MAILAD DS V (MAILUP)
SAVAREA DS 18F
PARAM DC A (RECTAB)
      DC A (MSGIND)
      DC A (0) UNSUSED
      DC A (0) UNSUSED
      DC A (0) UNSUSED
*
MSGIND DC CL1 'Y'
*
RECTAB DS 0A
REC1 DC AL2 (LEN1, 0)
     DC C 'TO TEST.DE '
LEN1 EQU *-REC1
*
REC2 DC AL2 (LEN2, 0)
     DC C 'SUBJECT MESSAGE FROM UP-INTERFACE ,
LEN2 EQU *-REC2
*
REC6 DC AL2 (LEN6, 0)
     DC C 'END '
LEN6 EQU *-REC6
*
      DC AL2 (0, 0) INDICATES END OF RECORD TABLE
```

### Hinweis:

Falls die LMS-Bibliothek MAILLIB nicht in der gleichen User-ID steht wie das Hauptprogramm, muß die globale Parameterdatei MAIL.PAR per File-Anweisung mit dem Linknamen MAILPARG zugewiesen oder der Name der MAILLIB mit IMON registriert werden (S. 15). Der Name der MAILLIB kann mit Hilfe des logischen Namens MAILLIB mit den entsprechenden IMON-Makros ermittelt werden (siehe Musterprogramm S/CALLMAIL.SRC in der MAILLIB).

Beispiel: /FILE \$MAIL.MAI.PAR, LINK=MAILPARG

Falls die Funktion OPEN-ZIP verwendet wird, muß zum Nachladen des Moduls OPGZIP die LMS-Bibliothek MAILLIB per File-Anweisung mit dem Linknamen OPGZIPL zugewiesen werden.

Beispiel: /FILE \$MAIL.MAILIB, LINK=OPGZIPL

### MAILUPR (Empfangen)

Der Modul MAILUPR wird entsprechend der Standard-Programmverknüpfungsregeln aufgerufen. Er kann auch von höheren Programmiersprachen aufgerufen werden. Beim Sprung in den Modul MAILUP müssen die Register wie folgt geladen sein:

- (R13)           A (SAVEAREA)  
Register-Sicherstellungsbereich (18 Worte, DC 18F'0'), der vom Aufrufer erstellt werden muß. MAILUP sichert dort die Register.
  
- (R14)           A (RETURN)  
Rücksprungadresse im rufenden Programm. Nach Beenden der MAIL-Funktionen wird das Programm an dieser Adresse fortgesetzt.
  
- (R15)           V (MAILUPR)

An das Programm MAILUPR werden keine Parameter übergeben. Wird das Programm innerhalb eines Batch-Prozesses aufgerufen, sollte in der Parameterdatei MAIL.PAR die Anweisung "CYCLE=0" enthalten. Damit werden alle zum Zeitpunkt des Programmaufrufs vorhandenen E-Mails abgeholt und das Programm kehrt in das aufrufende Programm zurück.

## Anlage 1: Codeumwandlung zwischen EBCDIC und ASCII

### Code-Umwandlung

Da BS2000-Rechner mit einem EBCDIC-Code arbeiten, die Programme auf PC und Unix jedoch grundsätzlich nur ISO-Codes verarbeiten, findet bei der Übertragung zum Mail-Server eine Umcodierung von ISO-Code in EBCDI-Code bzw. beim Empfangen vom MAIL-Server von EBCDI-Code in ISO-Code statt.

Zu sendende Nachrichten werden, soweit sie nicht binär übertragen werden, immer von **EBCDIC.DF.04-1 (CCSN: EDF041)** nach **ISO8859-1** übersetzt.

Der **ISO 8859-1** ist eine Erweiterung des ASCII-Codes. Als ASCII-Code wird die US-Variante des 7-bit-Codes gemäß ISO646 bezeichnet. Neben dem internationalen ASCII-Code gibt es noch weitere nationale Varianten des 7-bit-Codes gemäß ISO646.

### 7-bit Code

Als Standard ist im BS2000 der 7-bit-Leitungscode ISO646-IRV und der dazugehörige EBCDIC-Code EBCDIC.DF.03-IRV (CCSN: **EDF03IRV**) eingestellt. Wenn Sie im 7-bit-Mode arbeiten, verwenden Sie interne Tabellen. Das hat die Vorteile, daß die Arbeit unabhängig von XHCS (Verfügbarkeit, Änderung von Tabellen) ist und volle Kompatibilität mit älteren Versionen der Anwendung besteht.

Das Problem bei dieser Code-Variante besteht darin, daß bestimmte internationale Sonderzeichen und deutsche Umlaute nicht gleichzeitig dargestellt werden können. Dabei geht es um folgende Zeichen:

Deutsche Umlaute:	ä	ö	ü	Ä	Ö	Ü	ß	
Dänische Sonderzeichen:	æ	ø	å	Æ	Ø	Å	ü	
Hexa-Wert	FB	4F	FD	BB	BC	BD	FF	6A
Internationale Zeichen:	{		}	[	\	]	~	^

Durch Einstellungen des Terminals bzw. der Terminalemulation kann gewählt werden, ob internationalen Zeichen oder die deutschen Zeichen dargestellt werden sollen. Die interne Code-Umsetzung der Programme `MAIL` und `MAILR` benutzt die internationale Variante. Sollen die Umlaute verwendet werden, sind in der Parameterdatei in den Abschnitten `<TRTAB-SEND>` (S. 81) und `<TRTAB-RECEIVE>` (S. 81) Translate-Anweisungen anzugeben. In der ausgelieferten Parameterdatei sind diese Anweisungen bereits als Kommentarzeilen enthalten und können durch Entfernen des Kommentarzeichens ("\*" in der ersten Spalte) aktiviert werden.

### 8-bit-Codes oder Erweiterte Zeichensätze

Erweiterte Zeichensätze werden auch "8-bit-Codes" oder erweiterte Codes genannt, weil die Anzahl der zur Verfügung stehenden Codeplätze verdoppelt wurde. Diese Anzahl der Codeplätze erlaubt es, Zeichen für verschiedene Sprachen zu definieren, die aber trotzdem zu dem ursprünglich verwendeten Code, dem ASCII-Code (**American Standard Code for Information Interchange**), kompatibel sind. Die verschiedenen 8-bit-Codes sind in der internationalen Norm ISO 8859 definiert. Sie haben alle in der "linken" (niederwertigen) Hälfte der Codetabelle einen gemeinsamen Teil, analog zu ASCII, in der "rechten" (höherwertigen) Hälfte unterscheiden sie sich. Einzelne Codes werden zu Gruppen kompatibler Codes zusammengefaßt, die über ihre ISO-Code-Variantennummer identifiziert werden. Folgende Codes sind zur Zeit in ISO 8859 als Standard definiert:

- 8859-1 Latin-1 (West- und Nord-Europa)
- 8859-2 Latin-2 (Ost-Europa, ausgenommen Türkei und die Baltischen Staaten)
- 8859-3 Latin-3 (Mittelmeerraum und Süd-Afrika)
- 8859-4 Latin-4 (Skandinavien und die Baltischen Staaten)
- 8859-5 Kyrillisch
- 8859-6 Arabisch
- 8859-7 Griechisch
- 8859-8 Hebräisch
- 8859-9 Latin-5 (Türkei, West-Europa inklusive Skandinavien)
- 8859-10 Latin-6 (Nord-Europa und die Baltischen Staaten)

Auf der BS2000-Seite wird der Zeichensatz ISO 8859-1 durch den EBCDIC.DF.04-1 dargestellt. Der EBCDIC (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchange **C**ode), den BS2000 verwendet, muß so erweitert werden, daß jedes Zeichen ein Gegenstück im entsprechenden ISO 8859-x hat. Da EBCDIC-Codes nicht standardisiert sind, existieren unterschiedliche Zuordnungen zwischen EBCDI- und ISO-Codes.

Der **EBCDIC.DF.04-n (CCSN: EDF041)** ist eine Erweiterung des EBCDIC.DF.03-IRV (CCSN: EDF03IRV =Internationale **R**eferenz **V**ersion). Eine Variante des EBCDIC.DF.03-IRV ist der EBCDIC.DF.03-DRV (=Deutsche **R**eferenz **V**ersion). Beide Codes unterstützen Zeichensätze in der gleichen Größe. Sie haben einen gemeinsamen Zeichensatz, den sogenannten EBCDIC-Kern und unterscheiden sich nur in einigen Symbolen (siehe Tabelle EBCDIC.DF.03). System- und Anwendungsprogramme, die nicht durch XHCS unterstützt werden, benutzen diese EBCDIC.DF.03-Codes.

Die definierte Zuordnung zwischen ISO 8859-1 und EBCDIC.DF.04-1 spielt eine besondere Rolle. Für jede ISO-Tabelle wird eine EBCDIC-Tabelle definiert, die jeweils den gleichen Zeichenvorrat enthält, aber mit einer anderen Codebelegung. Diese EBCDIC-Tabellen haben die Bezeichnung EDF041 bis EDF049. Sie wurden so gewählt, daß mit ein und derselben Umsetztabelle die Konvertierung von ISO 8859-n nach EDF04n (n=1..9) und umgekehrt möglich ist. Diese Standardumsetzung wird von Netzkomponenten und File-Transfer-Programmen verwendet.

Bei Verwendung der Code-Tabellen EDF041 ist bezüglich der Umlaute und Sonderzeichen die richtige Übersetzung in den ASCII-Code ISO8859-1 gewährleistet, d.h. in der Parameterdatei in den Abschnitten <TRTAB-SEND> (S. 81) und <TRTAB-RECEIVE> (S. 81) sind keine Angaben notwendig.

### Aktivierung der 8-bit-Umgebung für 8-bit-Datensichtstationen

Der 8-bit-Modus kann entweder fest eingestellt (permanenter 8-Bit-Modus) oder nur für ein bestimmtes Programm aktiviert werden.

#### Permanenter 8-bit-Modus

Der permanente 8-bit-Modus kann für TIAM-, DCAM- und UTM-Anwendungen durch Verwendung der VTSU-Betriebsparameterdatei **systemweit** eingestellt werden, indem der Name des Standard-Anwenderzeichensatzes (CCSN) automatisch für die 8-bit-DSS aktiviert wird. Zusätzlich kann für TIAM-Anwendungen dieser CCSN mit Hilfe des TIAM-Kommandos /MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=8-BIT-DEFAULT vom Anwender selbst aktiviert werden.

Eine Code-Gruppe kann Codes enthalten, die zu einem ISO 8859-Code voll kompatibel (z.B. EBCDIC.DF.04-n) oder begrenzt kompatibel sind (z.B. EBCDIC.DF.03-DRV oder ISO646-IRV). Begrenzt kompatible Codes bestehen aus einer Teilmenge der Zeichen des entsprechenden ISO 8859-Codes. Da eine Umwandlung von begrenzt kompatiblen Codes in den Referenzcode immer möglich ist, während eine Umwandlung vom Referenzcode in die begrenzt kompatiblen Codes nicht für alle Zeichen des Gruppenreferenzcodes möglich ist, werden begrenzt kompatible Codes gekennzeichnet.

#### *Beispiel*

Jedes unter EBCDIC.DF.03 definierte Zeichen hat ein Äquivalent im ISO 8859. Die mit Akzent versehenen Buchstaben im ISO 8859-1 haben dagegen kein Gegenstück im EBCDIC.DF.03-IRV sondern in dessen Code-Erweiterung EBCDIC.DF.04-1.



### *Hinweis*

Einige Produkte benutzen die Kennzeichnung von begrenzt kompatiblen Codes zur Feststellung, ob die Daten an einer 7-bit-Datensichtstation dargestellt werden können. Es wird deshalb geraten, 8-bit-Codes nicht als begrenzt kompatible Codes zu definieren.

### **Translate-Tabelle EBCDIC → ASCII**

Im Programm MAIL (Senden) erfolgt die Code-Umwandlung von EBCDIC.DF.04-1 (CCSN: EDF041) nach ASCII ISO 8859-1 nach folgender Tabelle:

#### **EBCDIC → ASCII**

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	00	01	02	03	85	09	86	7F	87	8D	8E	0B	0C	0D	0E	0F
10	10	11	12	13	8F	0A	08	97	18	19	9C	9D	1C	1D	1E	1F
20	80	81	82	83	84	92	17	1B	88	89	8A	8B	8C	05	06	07
30	90	91	16	93	94	95	96	04	98	99	9A	9B	14	15	9E	1A
40	20	A0	E2	E4	E0	E1	E3	E5	E7	F1	60	2E	3C	28	2B	7C
50	26	E9	EA	EB	E8	ED	EE	EF	EC	DF	21	24	2A	29	3B	9F
60	2D	2F	C2	C4	C0	C1	C3	C5	C7	D1	5E	2C	25	5F	3E	3F
70	F8	C9	CA	CB	C8	CD	CE	CF	CC	A8	3A	23	40	27	3D	22
80	D8	61	62	63	64	65	66	67	68	69	AB	BB	F0	FD	FE	B1
90	B0	6A	6B	6C	6D	6E	6F	70	71	72	AA	BA	E6	B8	C6	A4
A0	B5	AF	73	74	75	76	77	78	79	7A	A1	BF	D0	DD	DE	AE
B0	A2	A3	A5	B7	A9	A7	B6	BC	BD	BE	AC	5B	5C	5D	B4	D7
C0	F9	41	42	43	44	45	46	47	48	49	AD	F4	F6	F2	F3	F5
D0	A6	4A	4B	4C	4D	4E	4F	50	51	52	B9	FB	FC	DB	FA	FF
E0	D9	F7	53	54	55	56	57	58	59	5A	B2	D4	D6	D2	D3	D5
F0	30	31	32	33	34	35	36	37	38	39	B3	7B	DC	7D	DA	7E

### Translate-Tabelle ASCII → EBCDIC

Im Programm MAILR (Empfangen) erfolgt die Code-Umwandlung von ASCII ISO 8859-1 nach EBCDIC.DF.04-1 (CCSN: EDF041) nach folgender Tabelle:

#### ASCII → EBCDIC

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	00	01	02	03	37	2D	2E	2F	16	05	15	0B	0C	0D	0E	0F
10	10	11	12	13	3C	3D	32	26	18	19	3F	27	1C	1D	1E	1F
20	40	5A	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
30	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
40	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
50	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	BB	BC	BD	6A	6D
60	4A	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
70	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	FB	4F	FD	FF	07
80	20	21	22	23	24	04	06	08	28	29	2A	2B	2C	09	0A	14
90	30	31	25	33	34	35	36	17	38	39	3A	3B	1A	1B	3E	5F
A0	41	AA	B0	B1	9F	B2	D0	B5	79	B4	9A	8A	BA	CA	AF	A1
B0	90	8F	EA	FA	BE	A0	B6	B3	9D	DA	9B	8B	B7	B8	B9	AB
C0	64	65	62	66	63	67	9E	68	74	71	72	73	78	75	76	77
D0	AC	69	ED	EE	EB	EF	EC	BF	80	E0	FE	DD	FC	AD	AE	59
E0	44	45	42	46	43	47	9C	48	54	51	52	53	58	55	56	57
F0	8C	49	CD	CE	CB	CF	CC	E1	70	C0	DE	DB	DC	8D	8E	DF

EBCDIC Code-Tabelle EBCDIC.DF.03 (CCSN:EDF03IRV)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00-0F																
10-1F																
20-2F																
30-3F																
40-4F	SP										`	.	<	(	+	ö
50-5F	&										!	\$	*	)	;	
60-6F	-	/									^	,	%	_	>	?
70-7F											:	#	@	'	=	"
80-8F		a	b	c	d	e	f	g	h	i						
90-9F		j	k	l	m	n	o	p	q	r						
A0-AF			s	t	u	v	w	x	y	z						
B0-BF												[	Ä	\	Ö	ü
C0-CF		A	B	C	D	E	F	G	H	I						
D0-DF		J	K	L	M	N	O	P	Q	R						
E0-EF			S	T	U	V	W	X	Y	Z						
F0-FF	0	1	2	3	4	5	6	7	8	9		{	ä	}	ü	~ ß
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

## EBCDIC Code-Tabelle EBCDIC.DF.04.1 (CCSN:EDF041)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00-0F																
10-1F																
20-2F																
30-3F																
40-4F	SP	NbS	â	ä	à	á	ã	å	ç	ñ	`	.	<	(	+	
50-5F	&	é	ê	ë	è	í	î	ï	ì	ß	!	\$	*	)	;	
60-6F	-	/	Â	Ä	À	Á	Ã	Å	Ç	Ñ	^	,	%	_	>	?
70-7F	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	¨	:	#	@	'	=	"
80-8F	∅	a	b	c	d	e	f	g	h	i	«	»	ð	ý	þ	±
90-9F	°	j	k	l	m	n	o	p	q	r	<sup>a</sup>	°	æ	.	Æ	¤
A0-AF	µ	—	s	t	u	v	w	x	y	z	i	¿	Ð	Ý	Þ	®
B0-BF	¢	£	¥	·	©	§	¶	¼	½	¾	¬	[	\	]	'	x
C0-CF	ù	A	B	C	D	E	F	G	H	I	-	ô	ö	ò	ó	õ
D0-DF	ı	J	K	L	M	N	O	P	Q	R	<sup>1</sup>	û	ü	Û	ú	ÿ
E0-EF	Ù	÷	S	T	U	V	W	X	Y	Z	<sup>2</sup>	Ô	Ö	Ò	Ó	Õ
F0-FF	0	1	2	3	4	5	6	7	8	9	<sup>3</sup>	{	Ü	}	Ú	~
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

## EBCDIC Code-Tabelle EBCDIC.DF.04-DRV (CCSN:EDF04DR)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00-0F																
10-1F																
20-2F																
30-3F																
40-4F	SP	NbS	â		à	á	ã	å	ç	ñ	`	.	<	(	+	ö
50-5F	&	é	ê	ë	è	í	î	ï	ì		!	\$	*	)	;	—
60-6F	-	/	Â		À	Á	Ã	Å	Ç	Ñ	^	,	%	_	>	?
70-7F	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	¨	:	#	§	'	=	"
80-8F	Ø	a	b	c	d	e	f	g	h	i	«	»	ð	ý	þ	±
90-9F	°	j	k	l	m	n	o	p	q	r	<sup>a</sup>	°	æ	.	Æ	¤
A0-AF	µ	~	s	t	u	v	w	x	y	z	i	¿	Ð	Ý	Þ	®
B0-BF	¢	£	¥	·	©	@	¶	¼	½	¾	¬	Ä	Ö	Ü	'	x
C0-CF	{	A	B	C	D	E	F	G	H	I	-	ô	[	ò	ó	õ
D0-DF	}	J	K	L	M	N	O	P	Q	R	<sup>1</sup>	û	]	ù	ú	ÿ
E0-EF	\	÷	S	T	U	V	W	X	Y	Z	<sup>2</sup>	Ô	Û	Ò	Ó	Õ
F0-FF	0	1	2	3	4	5	6	7	8	9	<sup>3</sup>	ä	ú	ü	ù	ß
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

## ASCII Code-Tabelle ISO 8859-1 Latin Alphabet No. 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00-0F																
10-1F																
20-2F		!	"	¶	§	%	v	'	(	)	*	+	,	-	.	/
30-3F	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40-4F	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50-5F	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60-6F	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70-7F	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80-8F	€															
90-9F																
A0-AF		ı	ç	£	¤	¥		§	¨	©	ª	«	¬	-	®	¯
B0-BF	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C0-CF	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0-DF	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0-EF	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0-FF	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Der **ISO 8859-1** ist eine Erweiterung des ASCII-Codes. Als ASCII-Code wird die US-Variante des 7-bit-Codes gemäß ISO646 bezeichnet. Neben dem internationalen ASCII-Code gibt es noch weitere nationale Varianten des 7-bit-Codes gemäß ISO646.

### Anlage 2: Verschlüsselung der Attachments

#### Verschlüsseln im BS2000

##### 1. Programm MAILS

Die Verschlüsselung wird mit der Option `CRYPTKEY` der Anweisung `ATTACH` (S. 21) initiiert. Mit der Anweisung `TARGETNAME` kann bestimmt werden, ob der Name der Zielfeile die Endung ".OPGC" erhalten soll. Für die Eingabe des Schlüssels gibt es zwei Möglichkeiten:

- a) Der Schlüssel wird direkt in der Anweisung angegeben (Option `CRYPTKEY=key`).
- b) Der Schlüssel wird in einer vorangehenden Anweisung bzw. in der Parameterdatei definiert (Anweisung `CRYPTKEY=key` (S. 27)) und Option `CRYPTKEY=Y`.

##### 2. Programm OPGCRYPT

Das Programm wird zusammen mit dem Programm `MAILS` bzw. `MAILR` ausgeliefert. Die Optionen für die Verschlüsselung werden über `SYSDTA` eingegeben (siehe Musterprozedur `J/CRYPT` in der Bibliothek `MAILLIB`).

Falls der Dateiname für das Zielsystem nicht festgelegt wurde, erhält die Datei beim Empfänger immer den Suffix ".opgc". Dadurch ist es möglich, daß über eine Verknüpfung in der Registry beim Doppelklick auf eine solche Datei automatisch das PC-Programm `OPGCRYPT` zum Entschlüsseln aufgerufen wird.

##### 3. SDF-Kommando OPGCRYPT-FILE

Beschreibung siehe S. 96

#### Verschlüsseln beim Versenden vom PC

Die Verschlüsselung wird stehen zwei Windows-Programme zur Verfügung:

- a) EDT für Windows:

Mit dem Kommando `WRITE` (Parameter `CRYPTKEY`) können Daten verschlüsselt abgespeichert werden.

- b) `OPGCRYPT` oder `OPGCRYPTW`:

Die Optionen für die Verschlüsselung (S. 98) können direkt beim Aufruf über Schalter mitgegeben werden. Falls nicht explizit der Name der Ausgabedatei festgelegt wird, erhält die Datei den Suffix ".opgc". Dadurch ist es möglich, daß über eine Verknüpfung in der Registry beim Doppelklick auf eine solche Datei automatisch das Windows-Programm `OPGCRYPTW` zum Entschlüsseln aufgerufen wird.

---

## Anlage 2: Verschlüsselung von Attachments

### Entschlüsseln im BS2000

#### 1. Programm MAILR

Für jeden User kann im Abschnitt `<USER-username>` in der Anweisung `DECRYPTKEY` (S. 71) ein Standard-Cryptkey zum Entschlüsseln gespeichert werden. Wird ein verschlüsseltes Attachment empfangen, wird geprüft, ob die Datei mit dem Standard-Cryptkey entschlüsselt werden kann und ggf. entschlüsselt. Ist der Key nicht gültig bzw. ist kein Standard-Cryptkey gespeichert, wird das Attachment verschlüsselt gespeichert und muß mit dem Programm `OPGCRYPT` entschlüsselt werden. Mit der Anweisung `UNCOMPRESS=Y|N` (S. 80) kann bestimmt werden, ob ein Attachment, daß mit `OPGCRYPT` komprimiert wurde, automatisch entkomprimiert werden soll.

#### 2. Programm OPGCRYPT

Das Programm wird zusammen mit dem Program `MAILS` bzw. `MAILR` ausgeliefert. Die Optionen für die Entschlüsselung werden über `SYSDTA` eingegeben (siehe Musterprozedur `J/DECRYPT` in der Bibliothek `MAILLIB`).

#### 3. SDF-Kommando OPGCRYPT-FILE

Für die Installation dieses SDF-Kommandos wird die Bibliothek `OPGCOM.LIB` benötigt. Beschreibung siehe S. 96 und Manual `OPGCOM`.

### Entschlüsseln beim Empfangen im Windows

Die Entschlüsselung wird stehen zwei Windows-Programme zur Verfügung::

a) EDT für Windows:

Mit dem Kommando `READ` (Parameter `CRYPTKEY`) können verschlüsselte Daten in den Arbeitsbereich eingelesen werden. Fehlt der Parameter `CRYPTKEY`, wird der Schlüssel angefordert.

b) `OPGCRYPT` oder `OPGCRYPTW`:

Die Optionen für die Entschlüsselung (S. 98) können direkt beim Aufruf über Schalter mitgegeben werden. Das Programm ist kostenlos erhältlich und kann von unserem Web-Server [www.opg.de](http://www.opg.de) heruntergeladen werden.

### SDF-Kommando OPGCRYPT-FILE

Das SDF-Kommando übergibt die Parameter dem ASTI-Service `OPGCRYPT` (S. 17). Die Standardwerte können ggf. in der Prozedur `SYSPR.CRYPT-FILE` angepaßt werden.

```
OPGCRYPT-FILE      FILENAME= bs2file | lib(typ,elem[,vers]) | *ZIP:zipfile(elem) | /posix-file,
                   OUTPUT= *STD | bs2file | lib(typ,elem) | /posix-file,
                   DATA= *TEXT | *TEXTDE | *TEXTDOS | *TEXTDOSE | *TEXTBIN | *BIN,
                   MODE= *AUTO | *CRYPT | *DECRYPT
                   KEY= *NO | key,
                   OVERWRITE= NO | YES
```

*bs2file*

Dateiname, ggf. mit Cat-ID und User-ID.



---

## Anlage 2: Verschlüsselung von Attachments

*lib (typ, elem[, vers])* Name einer PLAM-Bibliothek, Elementtyp, Elementnamen und ggf. Version.

\*ZIP : *zip-archiv (elem)* Name eines ZIP-Archivs und der Elementname.

*lposix-file* oder \*POSIX:file POSIX-file

**OUTPUT** = \*STD bzw. fehlt der Name der Ausgabedatei, wird der Dateiname wie folgt gebildet:

Verschlüsseln: An den Namen der Eingabedatei wird die Endung `.opgc` angehängt.

Entschlüsseln: Als Dateiname wird der Name der Originaldatei, der im Datei-Header gespeichert ist, benutzt.

**DATA** = \*TEXT | \*TEXTDE | \*TEXTBIN | \*TEXTDOS | \*TEXTDOSE | \*BIN

Beim Verschlüsseln kann die Struktur der Datei bestimmt werden. Diese Angabe wird in den Header-Informationen der verschlüsselten Datei gespeichert und beim Entschlüsseln automatisch berücksichtigt.

-bin

### **Binärformat**

Die Daten werden binär interpretiert, d.h. sie werden als ein Datenstrom ohne Satzstruktur behandelt. Es wird keine Code-Konvertierung und keine Satzbehandlung durchgeführt.

-text

### **Textformat international**

Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit internationale Variante) nach ISO8859-1 und umgekehrt durchgeführt.

-textde

### **Textformat Deutsch**

Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit deutsche Variante) nach ISO8859-1 und umgekehrt durchgeführt.

-textdos

### **Textformat international CP850**

Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit internationale Variante) nach CP850 (MS-DOS) durchgeführt. Diese Option kann nur im BS2000-Programm angegeben werden.

-textdedos

### **Textformat Deutsch CP850**

Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit deutsche Variante) nach CP850 (MS-DOS) und umgekehrt durchgeführt. Diese Option kann nur im BS2000-Programm angegeben werden.

-textbin

### **Textformat mit Sendecode**

Die Daten werden als Textdatei mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Eine Code-Konvertierung wird nicht durchgeführt.

---

## Anlage 2: Verschlüsselung von Attachments

**MODE** = \*AUTO | \*CRYPT | \*DECRYPT

\*AUTO: Anhand des Datei-Inhalts wird festgestellt, ob die Datei bereits verschlüsselt ist und in Abhängigkeit davon, die Funktion \*CRYPT oder \*DECRYPT ausgeführt.

\*CRYPT: Verschlüsseln

\*DECRYPT: Entschlüsseln

**KEY**=\*NO | *key*

\*NO: Die Datei wird nur komprimiert, ohne Verschlüsselung.

*key*: Schlüssel, der für die Verschlüsselung verwendet werden soll.

**OVERWRITE** = \*NO | \*YES

\*NO: Falls die Ausgabedatei bereits existiert, wird das Kommando mit Fehlermeldung beendet.

\*YES: Falls die Ausgabedatei bereits existiert, wird sie ohne Rückfrage überschrieben.

### Optionen für das Programm OPGCRYPT bzw. OPGCRYPTW

**OPGCRYPT** *input* | (STDIN) [*output* | (STDOUT)] [-C | -D] [-O] [-K *key*] [-BIN | -TEXT | -TEXTDE | -TEXTDEDOS | -TEXTDOS | -TEXTBIN] [-U *format*]

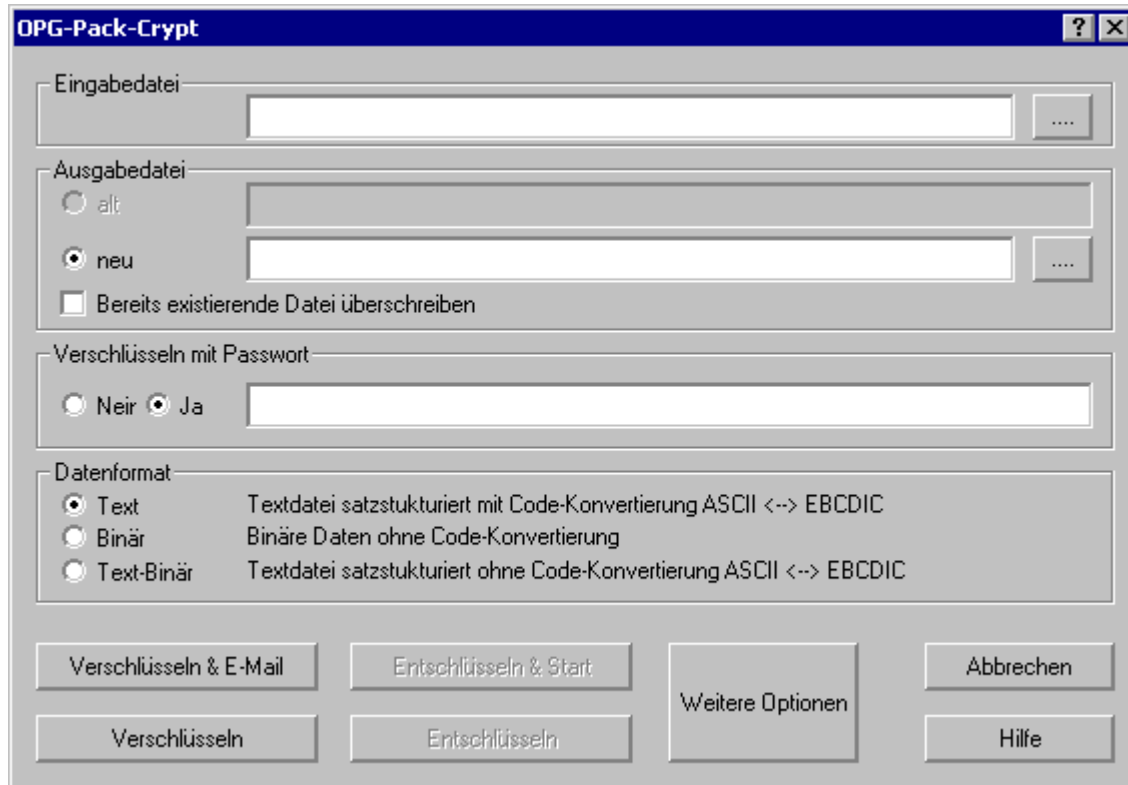
**OPGCRYPTW** *input* | (STDIN) [-O] [-K *key*] [-BIN | -TEXT | -TEXTDE | -TEXTDEDOS | -TEXTDOS | -TEXTBIN]



Unter Windows können die Parameter direkt beim Laden angegeben werden. Im BS2000 werden die Parameter über SYSDTA angefordert. Sie sind wie unter Windows mit Blank getrennt anzugeben.

Das Programm OPGCRYPT kann auch durch Doppelklick auf eine Datei mit der Endung ".opgc" oder durch Drag & Drop auf das OPGCRYPT-Icon auf dem Desktop bzw. über das Menü der rechten Maustaste, Menüpunkt "Senden an" bzw. "Send to" aufgerufen werden.

## Anlage 2: Verschlüsselung von Attachments

Das Programm OPGCRYPTW gibt folgende Dialogbox aus, in der die Parameter ergänzt werden können:



- input*            Eingabedatei.  
Hier ist der Dateiname der Datei anzugeben, die verarbeitet werden soll. Mit der Schaltfläche  kann die Datei ausgewählt werden.
- (STDIN)           Die Eingabedaten werden von der Standard-Eingabe gelesen. Dies ist nur zulässig, wenn die Daten von einer Pipe gelesen werden. Eine Eingabe von Tastatur ist nicht zulässig. Diese Angabe ist im BS2000 nicht zulässig.
- output*            Ausgabedatei.  
Hier ist der Dateiname der Ausgabedatei anzugeben. Mit der Schaltfläche  kann die Datei ausgewählt werden.  
Fehlt der Name der Ausgabedatei, wird der Dateiname wie folgt gebildet:  
Verschlüsseln: An den Namen der Eingabedatei wird die Endung `.opgc` angehängt.  
Entschlüsseln: Als Dateiname wird der Name der Originaldatei, der im Datei-Header gespeichert ist, benutzt.
- (STDOUT)           Die Daten werden in die Systemdatei STDOUT ausgegeben und können über eine Pipe weiterverarbeitet werden. Dies ist nur zulässig, wenn die Daten auf eine Pipe umgelenkt werden. Diese Option ist im BS2000 nicht zulässig.
- Beispiel: `opgencrypt file1.opgc (stdout) | edtw.exe -stdin`

---

## Anlage 2: Verschlüsselung von Attachments

- `-c / -d`      Komprimieren bzw. Dekomprimieren.
- Diese Angabe ist nur nötig bei (STDIN) oder falls eine bereits komprimierte Datei nochmals komprimiert werden soll. Ansonsten erkennt das Programm anhand der Header-Informationen, ob es sich um eine komprimierte Datei handelt und führt automatisch die entsprechende Funktion aus.
- `-o`            Overwrite.
- Falls die Ausgabedatei bereits existiert, wird sie ohne Rückfrage überschrieben.
- `-k key`      Paßwort.
- Schlüssel, der für die Verschlüsselung verwendet werden soll.
- Beim Verschlüsseln kann die Struktur der Datei bestimmt werden. Diese Angabe wird in den Header-Informationen der verschlüsselten Datei gespeichert und beim Entschlüsseln automatisch berücksichtigt.
- `-bin`            **Binärformat**
- Die Daten werden binär interpretiert, d.h. sie werden als ein Datenstrom ohne Satzstruktur behandelt. Es wird keine Code-Konvertierung und keine Satzbehandlung durchgeführt.
- `-text`            **Textformat international**
- Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit internationale Variante) nach ISO8859-1 und umgekehrt durchgeführt.
- `-textde`        **Textformat Deutsch**
- Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit deutsche Variante) nach ISO8859-1 und umgekehrt durchgeführt.
- `-textdos`       **Textformat international CP850**
- Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit internationale Variante) nach CP850 (MS-DOS) durchgeführt. Diese Option kann nur im BS2000-Programm angegeben werden.
- `-textdedos`    **Textformat Deutsch CP850**
- Die Daten werden als Textdatei im lokale Code mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Soweit notwendig, wird eine Konvertierung von EBCDIC (7-Bit deutsche Variante) nach CP850 (MS-DOS) und umgekehrt durchgeführt. Diese Option kann nur im BS2000-Programm angegeben werden.
- `-textbin`       **Textformat mit Sendecode**
- Die Daten werden als Textdatei mit lokaler Satzstruktur interpretiert bzw. wieder hergestellt. Eine Code-Konvertierung wird nicht durchgeführt.

## Anlage 2: Verschlüsselung von Attachments

-u *format*

Beim Verschlüsseln von Daten werden automatisch die Dateiattribute in den Header-Angaben gespeichert. Falls die Daten von einem anderen System kommen und auf dem aktuellen System erst verschlüsselt werden, kann hier das Format der Ursprungsdaten eingetragen werden, damit sie auf dem Empfangssystem zutreffend erstellt werden können. Zur Zeit kann nur das Format für Dateien, die im BS2000 entschlüsselt werden, eingetragen werden. Für *format* ist folgender String anzugeben:

OSDFILE/F/R/bb/B/11111/iiii/kkk

F	FCB-Type	S=SAM, I=Isam, P=PAM
R	RECFORM:	V= variabel, F=Fix
11111	RECSIZE :	Satzlänge oder 00000
bb	BLKSIZE	Blockgröße als Anzahl der Standardblöcke 1-16
B	BLKCNTL	Block-Control = DATA/PAMKEY/NONE (D/P/N)
iiii	KeyPos	Position Isam-Key oder 00000
kkk	KeyLength	Länge Isam-Key oder 000

Beispiel:

Die Ursprungsdaten wurden auf einer IBM-Anlage erstellt mit dem Format Fix, Satzlänge 80 und mit FTP auf einen PC binär übertragen. Die Daten liegen in der Form eines Strings vor ohne Satzende-Kennzeichen. Diese Datei soll zum BS2000 gesandt werden und mit folgenden Attributen erstellt werden:

FCBTYPE	SAM
RECFORM	F
RECSIZE	80
BLKSIZE	(STD,1)
BLKCNTL	DATA

Als Format-Angabe wäre einzutragen:

OSDFILE/S/F/00080/01/D/00000/000

Verschlüsseln & E-Mail

Die Datei wird verschlüsselt. In Abhängigkeit von den eingestellten Optionen wird die Verschlüsselung mit oder ohne Paßwort durchgeführt. Anschließend wird das E-Mail-Programm aufgerufen und die verschlüsselte Datei wird als Anlage für ein neues Mail eingetragen.

Verschlüsseln

Die Datei wird verschlüsselt. In Abhängigkeit von den eingestellten Optionen wird die Verschlüsselung mit oder ohne Paßwort durchgeführt.

Entschlüsseln & Start

Die Datei wird entschlüsselt. Soweit die Ursprungsdatei mit Paßwort verschlüsselt wurde, muß das Paßwort angegeben werden. Anschließend wird das Programm, das mit der Dateiendung verknüpft ist, aufgerufen. Heißt z.B. die entschlüsselte Datei TEST.DOC, wird das Programm WORD aufgerufen und die Datei TEST.DOC eingelesen.

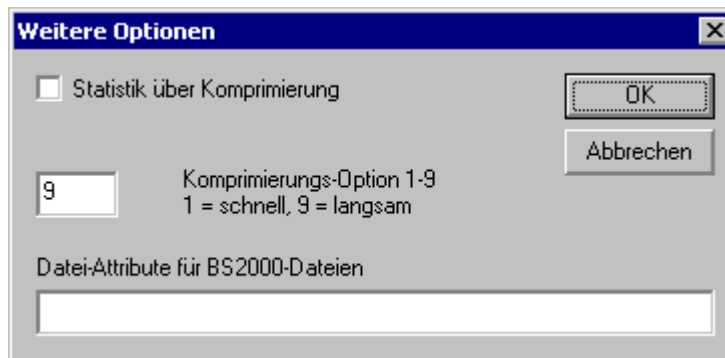
## Anlage 2: Verschlüsselung von Attachments

Entschlüsseln

Die Datei wird entschlüsselt. Soweit die Ursprungsdatei mit Paßwort verschlüsselt wurde, muß das Paßwort angegeben werden.

Weitere Optionen

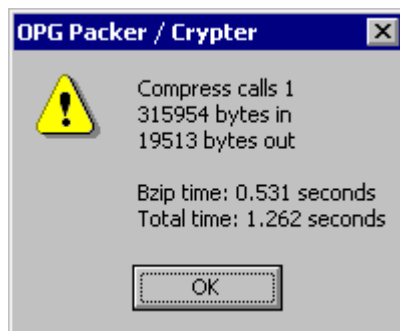
Es wird folgende Dialogbox mit weiteren Optionen angezeigt:



### Statistik über Komprimierung

Nach der Verarbeitung wird eine Messagebox mit Einzelheiten zur Komprimierung angezeigt.

Beispiel:



### Komprimierungs-Option

Der Wert "1" bedeutet, daß die Komprimierung so schnell wie möglich erfolgt und die Komprimierungsrate geringer ist. Der Wert "9" bedeutet, daß die höchste Komprimierungsrate erreicht wird, dafür eine höhere Laufzeit notwendig ist. Die Werte "2" bis "8" stellen entsprechende Abstufungen dar. Normalerweise ist die Einstellung "9" zu bevorzugen, da der CPU-Verbrauch in der Regel keine Rolle spielt. Nur bei sehr großen Dateien ist es sinnvoll eine Einstellung von "1" bis "8" zu wählen.

Der Unterschied des CPU-Verbrauchs und der Dateigröße hängt vom Dateiinhalt ab.

## Anlage 2: Verschlüsselung von Attachments

---

### Dateiattribute für BS2000-Dateien

Beim Verschlüsseln von Daten werden automatisch die Dateiattribute in den Header-Angaben gespeichert. Falls die Daten von einem anderen System kommen und auf dem aktuellen System erst verschlüsselt werden, kann hier das Format der Ursprungsdaten eingetragen werden, damit sie auf dem Empfangssystem zutreffend erstellt werden können. Zur Zeit kann nur das Format für Dateien, die im BS2000 entschlüsselt werden, eingetragen werden.

OSDFILE/F/R/bb/B/11111/iiii/kkk

F	FCB-Type	S=SAM, I=Isam, P=PAM
R	RECFORM:	V= variabel, F=Fix
11111	RECSIZE :	Satzlänge oder 00000
bb	BLKSIZE	Blockgröße als Anzahl der Standardblöcke 1-16
B	BLKCNTL	Bloc-Contrl = DATA/PAMKEY/NONE (D/P/N)
iiii	KeyPos	Position Isam-Key oder 00000
kkk	KeyLength	Länge Isam-Key oder 000

Beispiel:

Die Ursprungsdaten wurden auf einer IBM-Anlage erstellt mit dem Format Fix, Satzlänge 80 und mit FTP auf einen PC binär übertragen. Die Daten liegen in der Form eines Strings vor ohne Satzende-Kennzeichen. Diese Datei soll zum BS2000 gesandt werden und mit folgenden Attributen erstellt werden:

```
FCBTYPE    SAM
RECFORM    F
RECSIZE    80
BLKSIZE    (STD,1)
BLKCNTL    DATA
```


Als Format-Angabe wäre einzutragen:

OSDFILE/S/F/00080/01/D/00000/000

Abbrechen

Die Dialogbox wird geschlossen. Es wird keine Verarbeitung durchgeführt.

Hilfe

Sie erhalten Hilfe zu jedem Eingabefeld bzw. zu jeder Option, indem Sie auf das Symbol  im oberen rechten Ecke der Dialogbox klicken und dann im Help-Modus (Cursor erhält die Form eines Fragezeichens) auf die entsprechende Stelle in der Dialogbox klicken.

Bei Eingabefeldern können Sie auch den normalen Cursor in ein Feld positionieren und die Taste F1 drücken.

### **Algorithmus für die Verschlüsselung**

#### **Methode OPGCRYPT**

Als Basis wird der bekannte und als äußerst sicher geltende Blowfish-Algorithmus verwendet. Diese Methode wurde wie folgt ergänzt:

- a) Das Passwort wird auf die maximale Länge von 56 Bytes (448 Bits) ergänzt. Dabei wird das Passwort zusätzlich mit zeitabhängige Elementen modifiziert, so daß die gleiche Datei zu unterschiedlichen Zeitpunkten und mit dem gleichen Passwort immer anders verschlüsselt wird.
- b) Vor dem Verschlüsseln werden die Daten nach einem ZIP-ähnlichen Algorithmus komprimiert.
- c) Nach dem Verschlüsseln werden in die Daten Zufallswerte mit Zufallslänge und Zufallsrythmus eingestreut.

Die Sicherheit der Methode wird durch diese zusätzlichen Elemente noch erhöht. Die Datenmenge wird durch die Komprimierung auf ein Minimum beschränkt.

#### **Blowfish-Algorithmus**

Blowfish ist ein von Bruce Schneier 1994 vorgestellter Algorithmus, der als Alternative zu IDEA, Triple-DES oder DES zur symmetrischen Verschlüsselung gedacht ist. Blowfish ist wie IDEA ein Blockverschlüsselungsalgorithmus, der mit 64- Bit-Blöcken arbeitet. Dabei erlaubt er variable Schlüssellängen bis zu 448 Bit. Bisher konnten trotz intensiver Analysen keine Schwächen des Algorithmus gefunden werden.

Die Implementierung entspricht den Vorgaben und Definitionen von Bruce Schneier im Artikel "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", veröffentlicht in "Fast Software Encryption - Proceedings of the Cambridge Security Workshop, Cambridge, United Kingdom, Lectures Notes in Computer Science 809, pp. 191-204, Springer Verlag, 1994".

Eine Schlüsselexpansion wandelt den bis zu 448 Bit großen Schlüssel in verschiedene Teilschlüssel  $P_i$  und S-Boxen  $S\text{-Box}[i,j]$  um, die zusammen 4168 Bit ergeben. Die Verschlüsselung besteht aus einer Funktion, die in sechzehn Runden durchlaufen wird. Jede Runde besteht aus einer schlüssel- und datenabhängigen Substitution und aus einer schlüssel-abhängigen Permutation.



**Stichwortverzeichnis**

\*

\* [45](#)**7**7-bit-Code [87](#)7-bit-Leitungscode ISO646-IRV [87](#)**8**8-bit-Codes [87](#)8-bit-Umgebung aktivieren [88](#)**A**Abschnitt <GLOBAL> [45](#)Abschnitt <RECEIVE> [60](#)Abschnitt <RESTRICT \$user-id> [64](#)Abschnitt <SEND> [47](#)Abschnitt <TRTAB-RECEIVE> [80](#)Abschnitt <TRTAB-SEND> [81](#)Abschnitt <USER> [69](#)Aktivierung 8-bit-Umgebung [88](#)Algorithmus Blowfish [104](#)ASCII [38](#), [87](#), [89](#), [94](#)ASTI-Service [17](#)ATTACH [21](#)ATTACH-APPLICATION [60](#), [69](#)ATTACHMENT [69](#)ATTACHNAME [48](#), [64](#)ATTACHSIZE [47](#), [48](#), [67](#)AUTH [70](#)**B**BCC [25](#)Benutzer-Codes [26](#), [51](#)bin [100](#)Blowfish Algorithmus [104](#)Button Abbrechen [103](#)Button Entschlüsseln [101](#)Button Entschlüsseln/Starten [101](#)Button Verschlüsseln [101](#)Button Verschlüsseln/E-Mail [101](#)Button Weitere Optionen [102](#)**C**CC [25](#)CCSN EDF03IRV [87](#), [91](#)CCSN EDF041 [87](#), [92](#)CCSN EDF04DR [93](#)CCSN ISO8859 [87](#)CFS-Prozedur EMAIL [19](#)CLOSE-ZIP [25](#)CODEFILE [26](#), [51](#)Code-Tabelle EBCDIC.DF.03 (CCSN EDF03IRV) [91](#)Code-Tabelle EBCDIC.DF.04.1 (CCSN EDF041) [92](#)Code-Tabelle EBCDIC.DF.04-DRV (CCSN EDF04DR) [93](#)Code-Tabelle ISO 8859-1 [94](#)Code-Umwandlung [87](#)CONTENT-DESCRIPTION [52](#)CRYPTKEY [27](#), [52](#)CYCLE [61](#)CYCLE-MSG [62](#)**D**Dänische Sonderzeichen [87](#)Dateien – von MAILR erzeugt [40](#), [41](#)DECRYPTKEY [70](#)Default-Processing [41](#)DEFAULT-PROCESSING [71](#)DELETE [74](#)Deutsche Umlaute [87](#)DO.MAIL [18](#)DOMAIN [27](#), [45](#)DOMAINADD [27](#), [53](#)**E**EBCDIC [38](#), [87](#), [89](#), [91](#), [92](#), [93](#)EBCDIC.DF.03 (CCSN EDF03IRV) [91](#)EBCDIC.DF.04.1 (CCSN EDF041) [92](#)EBCDIC.DF.04-DRV (CCSN EDF04DR) [93](#)EDF03IRV [87](#), [91](#)EDF041 [87](#), [92](#)EDF04DR [93](#)EMAIL [19](#)EMPTY-RECORD [74](#)EMPTY-SUBJECT [53](#)END [28](#)Entschlüsselung im BS2000 [96](#)Entschlüsselung PC [96](#)ERRORON [28](#), [45](#)Erweiterte Zeichensätze [87](#)erzeugte Dateien [40](#), [41](#)**F**FNAMFILE [75](#)Folgeverarbeitung [41](#)FROM [28](#), [53](#), [67](#)**G**GLOBAL - Abschnitt Parameterdatei [45](#)**H**Hilfe [103](#)Hintergrundbilder [35](#), [56](#)HTMLFILE [75](#)

## I

ID für MAIL-Nachrichten [40](#)  
IMON [15, 44](#)  
INDEXFILE [62](#)  
Input [99](#)  
Installation [14](#)  
Installationsmonitor [15, 44](#)  
Installation-Unit (IMON) [15](#)  
ISAMMODE [53](#)  
ISO 8859-1 [94](#)  
ISO646 [87](#)  
ISO8859 [87, 94](#)  
ISO-Code [87](#)

## K

Kommandos Programm MAIL [21](#)

## L

LICENSE [46](#)  
LOCK [54](#)  
LOGFILE [20](#)  
Logical-ID (IMON) [15](#)

## M

MAIL (Programm) [18](#)  
MAILFILE [76](#)  
Mail-ID [77](#)  
MAIL-ID [40](#)  
MAILUP - Unterprogramm [83](#)  
MAILUPR - Unterprogramm [86](#)  
MSGBEG [28](#)  
MSGEND [29](#)  
MSGFILE [29](#)  
MSGLOG [31, 46](#)

## O

OPEN-ZIP [31](#)  
OPGCRYPT-FILE [17](#)  
Optionen [98](#)  
ORIGINAL [76](#)  
Output [99](#)

## P

Parameterdatei MAIL.PAR [44](#)  
PASS [77](#)  
PDF-LAYOUT [32, 54](#)  
POP3PORT [62](#)  
POP3USER [77](#)  
PREFIX [77](#)  
PRIORITY [36, 58](#)  
Programm MAIL [18](#)  
Programmaufruf MAIL [18](#)  
Programmaufruf MAILR [42](#)  
Programmbeendigung MAILR [43](#)  
Protokollierung der versandten Mails [20](#)

Prozedur DO.MAIL [18](#)

## R

RECEIVE - Abschnitt Parameterdatei [60](#)  
RECORD-LENGTH [62, 77](#)  
REPLY [37, 58](#)  
REQUEST-JV [63](#)  
RESTRICT \$user-id – Einschränkungen pro BS2000-  
User-ID [64](#)  
RR [37, 58](#)

## S

SEND [37](#)  
SEND - Abschnitt Parameterdatei [47](#)  
SEND-EDTWFIL [17](#)  
SEND-FILE [17](#)  
SEND-TRAP [17](#)  
SERVER [37, 46, 59, 63](#)  
Service [17](#)  
SMRPPORT [59](#)  
STATUS-JV [63](#)  
STDIN [99](#)  
STDOUT [99](#)  
SUBJECT [38, 60](#)  
Subject-Processing [41](#)  
SUBJECT-PROCESSING [78](#)  
SYSDTA [21](#)  
SYSDTA-Anweisungen [59](#)

## T

TARGETNAME [60](#)  
text [97, 100](#)  
textbin [97, 100](#)  
TEXTFILE [79](#)  
TO [38, 68](#)  
Translate-Tabelle ASCII to EBCDIC [89](#)  
Translate-Tabelle EBCDIC to ASCII [89](#)  
Translate-Tabellen [26, 51](#)  
TRTAB [38](#)  
TRTAB-ADDR [47](#)  
TRTAB-RECEIVE - Abschnitt Parameterdatei [80](#)  
TRTAB-SEND [80, 81](#)  
TRTAB-SEND - Abschnitt Parameterdatei [81](#)

## U

Umlaute [87](#)  
UNCOMPRESS [80](#)  
Unterprogramm MAILUP [83](#)  
Unterprogramm MAILUPR [86](#)  
USER - Abschnitt Parameterdatei [69](#)

## V

Verschlüsseln im BS2000 [95](#)  
Verschlüsselung PC [95](#)

### W

WARNINGON [39](#), [47](#)

Wasserzeichen [35](#), [56](#)

Watermark [35](#), [56](#)