



OPGCom

Benutzerhandbuch

Ausgabe Februar 2016 (OPGCom V2.44)

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, 2001 - 2016
Sendlinger Str. 28, 80331 München,
Tel. 089/26 78 31, Fax 089/260 9929, E-Mail info@opg.de, <http://www.opg.de>
Alle Rechte vorbehalten.

Änderungsprotokoll

Im folgenden sind die Erweiterungen des Programms OPGCom zusammengestellt:

Februar 2016:

- V2.44 EventScan: Neue Ersetzungstring-Varianten zur Parametrisierung der Verarbeitungsaktionen !MSG-STR, !MSG-INS1, !MSG-INS2, ..., !MSG-INS9 (S. 21). Damit wird es einfacher, bestimmte Teile aus der Konsolmeldung an die Verarbeitung weiterzugeben.

Oktober 2010:

- V2.35 EventScan: Neuer Parameter *host* zum Abschnitt <SCAN [*host*]> (S. 22): Die Angabe des Hostnamens bewirkt, daß dieser Abschnitt nur für die Verarbeitung auf dem angegebenen Host gültig ist.

Februar 2010:

- V2.27 FSend: Mit der Prozedur OPGCOM.LIB(J/REP) kann die Standard-CCSN eingestellt werden, die verwendet werden soll, wenn der Datei kein CCSN zugewiesen ist und CODE=*DEFAULT angegeben wird.

Februar 2009:

- V2.19 URLServer/FSend/PDFGEN/OPGCRYPT: POSIX-Dateinamen sind bis zu einer Länge von 256 Bytes zulässig
Damit SDF-P auch in Subtask-Verarbeitungen (CGI) verfügbar ist, werden die Subtasks jetzt als S-Prozedur erzeugt und mit /ENTER-PROC gestartet.

Januar 2009

- V2.17 Ab Version 2.17 wird die Socket-Schnittstelle Version 2.0 (Subsystem SOC6) verwendet.

November 2008

- V2.15 Die Funktionen FSend, Tsend und PDFGEN werden auch über einen ASTI-Service (S. 17) zur Verfügung gestellt. Die Kommunikation mit dem Service erfolgt über die folgenden neuen Kommandos:

TSend: Aufruf TSend mit SDF-Kommando SEND-TRAP (S. 30).

FSend: Aufruf FSend mit SDF-Kommando SEND-FILE (S. 35) und SEND-EDTWFIL (S. 36).

PDFGEN: Aufruf PDFGEN mit SDF-Kommando CREATE-PDF-FILE (S. 101).

Bei der Ausführung dieser Kommandos wird kein Programm aufgerufen, so daß diese Kommandos auch in Programmen benutzt werden können, ohne daß sie entladen werden.

April 2008

- V2.10 URLServer: URL-Adressen in Links können auch die Link-Variablen (S. 73) !CGILIB enthalten.

- V2.04 **September 2007:**

URLServer/PDFGEN: Die Auswertung der Steuerzeichen X'11' bis X'1F' im Text kann mit der Option NCTRL ausgeschaltet werden. Die Anweisungen PDF-MODE (S. 62) und PDF-LAYOUT (S. 63) sowie die Funktion PDF (S. 84) wurden entsprechend erweitert.

V2.02

März 2007:

URLServer/PDFGEN: Bei der PDF-Generierung kann neben den Hintergrundbildern auch ein Wasserzeichen angegeben und die automatische Seitenanpassung eingestellt werden. Die Anweisungen `PDF-BACKGROUND` (S. 60), `PDF-AUTOSIZE` (S. 61) und `PDF-LAYOUT` (S. 63) sowie die Funktion `PDF` (S. 84) wurden entsprechend erweitert.

PDFGEN/PDFGENUP (S. 101): Neue Unterprogramm-Schnittstelle für die PDF-Generierung.

FSend: Bei dem Parameter `EXT=cmdpar` (S. 40) können nun auch Zeichenfolgen angegeben werden, die Anführungszeichen enthalten.

FSendSRV: Dateien, die über die Funktion `EDTWUPD` des URLServers oder über `FSend` zu einem PC gesendet und dann in den `EDTW` eingelesen werden, können nun auch über das neue Programm `FSendSRV` (S. 112) empfangen und/oder zurückgesandt werden. Dadurch ist es möglich, Dateien zu einem PC zu senden, der im BS2000 nicht bekannt ist. Ebenfalls ist es möglich zum Empfangen im BS2000 statt dynamischer Ports nur einen zentralen Port zu verwenden.

V1.96

Dezember 2006:

URLServer: Bei der PDF-Generierung werden auch Vorschübe auf Kanal 2-11 berücksichtigt. Die Anweisungen `PDF-MODE` (S. 62) und `PDF-LAYOUT` (S. 63) sowie die Funktion `PDF` (S. 84) wurden um die Angaben zum Kanalvorschub erweitert.

V1.95

November 2006:

URLServer: Beim Aufruf eines Benutzermoduls (S. 96) mit der Funktion `CGI-BIN` wird der neue Bereich `CGIINFO` als 6. Parameter übergeben, damit wird der externe Bereich `HTMGLOB` nicht mehr benötigt, ist aber aus Kompatibilitätsgründen noch verfügbar. Die Bibliothek `OPGCOM.LIB` wurde um zwei Beispiel-Module `S/HALLOCOB.COB` und `S/ZEILENCO.COB` ergänzt.

V1.87

August 2006:

URLServer: Die Übersetzung der Daten von EBCDIC nach IS8859 wurde um benutzerdefinierte Codes erweitert. Mit der Anweisung `CODEFILE` (S. 50) kann eine Datei angegeben werden, in der eigene Translate-Tabellen definiert sind. Bei folgenden Code-Angaben können außer den bisherigen Varianten auch User-Codes angegeben werden:

Anweisungen `CODE` (S. 49) und `PDF-CODE` (S. 61)

Anweisung `PDF-LAYOUT` (S. 63) Parameter `CODE`

Funktionen `VIEW / XVIEW` (S. 89) Parameter `TRANS`

Funktion `PDF` (S. 84) Parameter `CODE`

URLServer: Bei der Funktion `FSTAT` (S. 81) kann auch eine Datei angegeben werden, in der die Dateinamen für die `FSTAT`-Liste enthalten sind.

URLServer: Neue Optionen `GETGROUP` und `GETUSER` zur CFS-Funktion `*RUN(URLOUT)` (S. 74), um die aktuelle Gruppe und den aktuellen User zu ermitteln.

V1.83

Januar 2006:

URLServer: Die Option `CODE` zur Anweisung `PDF-LAYOUT` (S. 63) und die Anweisung `PDF-CODE` (S. 61) wurde um die Variante `DK` (Dänische Sonderzeichen) erweitert.

URLServer: Die Funktion `FSTAT` und die Anzeige-Funktionen berücksichtigen FGGs. Über den Parameter `FGG=YES/NO` (S. 52) kann gesteuert werden, ob FGGs mit ausgewählt und angezeigt werden sollen (`STD:YES`).

EventScan: Mit der Anweisung ADD-MSG-ID (S. 20) können bis zu 32 Meldungsschlüssel angegeben werden, für die ein MODIFY-MSG-SUBSCRIPTION ADD-MSG-ID=.... abgesetzt wird.

V1.81 **November 2005:**

Falls beim Senden von Daten Fehler auftreten (Port not ready oder WOULDLOCK) wird nach einer Wartezeit von 100 Millisekunden die SEND-Aktion wiederholt (bisher Wartezeit von einer Sekunde).

V1.80 **Oktober 2005:**

URLServer: Bisher wurden vom Main-Task nur so viel Aufträge angenommen, wie Sub-Tasks gestartet wurden. Darüber hinaus gehende Anfragen wurden nach einer Timeout-Zeit abgewiesen. Nun werden bis zu 24 Aufträge angenommen und gespeichert und dann erst an die Sub-Tasks verteilt.

V1.76 **Juni 2005:**

FSend: Ändern von Host-Dateien mit EDT für Windows:

Die Daten werden Nach dem Übertragen automatisch in den Arbeitsbereich des EDT für Windows eingelesen und können von dort mit dem EDT-Kommando WRITE wieder zurück übertragen werden. Es wurden folgende neue Anweisungen und Optionen vorgesehen:

- a) Neue Option "EDTW" zum Parameter *mode* (S. 40).
- b) Mit der neuen Option "ENTER" (S. 41) kann die Verarbeitung (Senden und ggf. Empfangen von Daten) in einem eigenen Enter-Job durchgeführt werden.
- c) Mit dem neuen LLM FILESEND kann die Funktion FSEND direkt von der Kommandozeile aus geladen werden (bisher gab es nur eine UP-Schnittstelle). Zum Laden des LLM werden zwei SDF-Kommandos zur Verfügung gestellt: FILESEND (S. 35) und EDTW (S. 36). Im Programm CFS steht zum Aufruf von EDTW auch der neue Actioncode EDTW zur Verfügung.

TSend: Mit dem neuen LLM TRAPSEND kann der Modul TSENDUP direkt von der Kommandozeile aus geladen werden (bisher gab es nur eine UP-Schnittstelle). Zum Laden des LLM wird das SDF-Kommando TRAPSEND (S. 30) zur Verfügung gestellt.

URLServer: Neue Funktion EDTWUPD (S. 77) zum Editieren von Dateien mit dem Programm EDT für Windows. Dateien, die editiert werden dürfen, müssen mit der Anweisung FILEACCESS freigegeben werden (mit Option UPDATE=YES).

FSend und Tsend: Wird keine Empfangsadresse (Prozessor) angegeben, wird der Prozessorname aus einer Jobvariablen oder vom Terminal benutzt, siehe Parameter Prozessor der Funktion TRAPSEND (S. 32) und FILESEND (S. 39).

V1.74 **Januar 2005:**

URLServer: Bei der Anweisung PDF-LAYOUT (S. 63) und der Funktion PDF (S. 84) wurde der Parameter MODE erweitert: Es können auch druckaufbereitete Dateien mit ASA-Steuerzeichen verarbeitet werden und es ist die Auswahl eines Spaltenbereichs, in dem die Druckdaten enthalten sind, möglich.

EventScan: In der Anweisung SEND (S. 22) wurde ein erweiterter Suchbegriff eingeführt. Der Suchbegriff kann beliebig viele Suchargumente enthalten, die mit die mit und (+), oder (,) bzw. Wildcard (*) miteinander verknüpft werden können.

V1.73 **Dezember 2004:**

URLServer: In der AUTHFILE (S. 45) kann auch die Anweisung `CODE=DE|INT` (S. 49) angegeben werden.

Juli 2004

V1.69 URLServer: Nach der Verarbeitung einer Funktion können über den Benutzer-Modul URLEXIT (S. 99) zusätzliche Aktionen ausgeführt werden, wie z.B. Protokollieren von speziellen Aktionen. Der Modul wird nach Abschluß einer Funktion aufgerufen.

V1.68 URLServer: Bei der Anweisung `PDF-LAYOUT` (S. 63) und der Funktion PDF (S. 84) kann beim Parameter PAGE neben den vordefinierten Seitengrößen A4, A4Q, A5 und A5Q auch die Größe in Millimeter angegeben werden.

April 2004

V1.65 URLServer: Das Inhaltsverzeichnis und die Elemente von BS2000-Zip-Archive können angezeigt werden. Das Inhaltsverzeichnis kann mit der Funktion `LIBTOC/*ZIP:zip-archiv` (S. 81) und die Elemente mit der Funktion `VIEW/*ZIP:zip-archiv(element)` (S. 89) angezeigt werden.

URLServer: Die Funktion `VIEW` überprüft die anzuzeigende Datei. Falls die Datei ein ZIP-Archiv oder eine PLAM-Bibliothek enthält, wird das Inhaltsverzeichnis mit der Funktion `LIBTOC` angezeigt.

Dezember 2003

V1.59 EventScan: Mit dem Parameter `ENTERPAR` (S. 26) im Abschnitt `<SNMP>` kann definiert werden, welche Parameter beim Starten von Enter-Prozeduren für das Kommando `ENTER-JOB` verwendet werden sollen.

September 2003

V1.57 URLServer: Bei der Erzeugung von PDF-Daten (S. 84) kann auch ein Hintergrundbild angegeben werden zum Einfügen von Wasserzeichen, Formularen oder Logos.

URLServer: Mit der Anweisung `PDF-LAYOUT` (S. 63) können verschiedene Parametersätze für die Funktion PDF definiert werden.

EventScan: Mit der neuen Anweisung `FILE=file` (S. 22) im Abschnitt `<ACTION>` kann eine Datei angegeben werden, in der die `SEND`-Anweisungen stehen.

Juli 2003:

V1.55 URLServer: Mit der neuen Funktion `LS` (S. 81) kann eine Liste von Posix-Dateien erzeugt werden. Wie bei der `FSTAT`-Liste kann festgelegt werden, welche Spalten (S. 55) ausgegeben und wie die Liste sortiert werden soll. Die Anweisungen `USERLINK` (S. 68) gelten auch für die `LS`-Liste. Das Layout kann in der Datei `TEMPLATE.LS` (S. 92) festgelegt werden.

Juni 2003:

V1.52 URLServer: Mit der neuen Anweisung `ENTER_TIMEOUT` (S. 52) kann festgelegt werden, nach wieviel Sekunden der `ENTER`-Job gestartet sein muß.

EventScan: In der Anweisung `ACTION` (S. 21) kann der Suchbegriff als Variable angegeben werden.

URLServer: Bei der Anweisung `FILEACCESS` (S. 53) können mehrere Auswahlstrings miteinander verknüpft werden

URLServer: Mit den neuen Anweisungen `CGIACCESS` (S. 47), `CFSACCESS` (S. 48) und `ENTERACCESS` (S. 51) kann der Zugriff auf Benutzermodule, CFS-Prozeduren und Enter-Prozeduren global und pro Benutzergruppe eingeschränkt werden.

V1.51 FSEND: Der Übertragungsmodus, Parameter *mode* (S. 38) wurde um die Variante `TXTD` für Daten mit deutschen Umlauten erweitert.

Mai 2003:

V1.50 URLServer: Mit der neuen Funktion `CFS` (S. 74) können CFS-Prozeduren ausgeführt werden. Zum Senden von Daten an den Browser wurde ein neuer CFS-Modul erstellt, der mit der CFS-Anweisung `*RUN (URLOUT)` aufgerufen wird. Für die Änderung der `SYSOUT`-Zuweisung wurde ebenfalls ein neuer CFS-Modul erstellt (Aufruf `*RUN (SYSOUT)`). Die `CFSLIB` wird mit der Anweisung `CFSLIB` (S. 49) zugewiesen.

URLServer: Die Funktion `FSTAT` und `LIBTOC` wurden um den Parameter `SORT` (S. 81) erweitert. Damit kann die `FSTAT`- bzw. `LIBTOC`-Liste nach allen Spalten sortiert werden. Die Standard-Sortierung kann mit der Anweisung `FSTAT_SORT` (S. 55) bzw. `LIBTOC_SORT` festgelegt werden. Die Anforderung einer neuen Liste mit geänderter Sortierung kann auch durch einen Mausklick auf die entsprechende Spaltenüberschrift erfolgen.

V1.33 URLServer: Der URLServer-Task (S. 44) kann beliebig oft gestartet werden. Einzige Voraussetzung ist, daß für jeden Server ein eigener Port und eigene LOG-Dateien verwendet werden. Die Portnummer kann in der ausgelieferten Musterprozedur über den Parameter `PORT` geändert werden.

April 2003

V1.30 URLServer: URL-Adressen in Links können auch die Link-Variablen (S. 73) `!AUTHGROUP` und `!AUTHUSER` enthalten.

März 2003

V1.22 URLServer: Die Konvertierung der Umlaute von EBCDIC nach ISO8859-1 kann mit der Anweisung `CODE` (S. 49) und den Optionen `TRANS=D` bzw. `TRANS=I` bei den Funktionen `VIEW` (S. 89) und `XVIEW` gesteuert werden.

V1.21 URLServer: Die Syntax für die Angabe von Bibliothekselementen wurde geändert auf: *lib/typ/element[/vers]* (S. 71). Das alte Format (*lib (typ, element[, vers])*) ist ebenfalls noch erlaubt. Das neue Format bietet den Vorteil, daß ein relativer Verweis auf Elementebene möglich ist und daß vom Browser der Dateityp erkannt wird.

V1.19 URLServer: Kommt in einer Datei, die keine HTML-Datei ist und mit `VIEW` angezeigt wird, das Zeichen "<" oder "&" vor, führt dies zur Interpretation durch den Browser. Diese Zeichen werden nun maskiert, so daß sie unverändert angezeigt werden. Sollen diese Zeichen ausnahmsweise angezeigt werden, so kann diese mit der Anweisung `VIEWTAG` (S. 70) bzw. der Option `VIEWTAG=Y` zur Funktion `VIEW` (S. 89) erreicht werden.

Februar 2003

V1.18 URLServer: Bei der Funktionen `XVIEW` (S. 89) können die Parameter `REC` (Record-Type: DOS, UNIX, BS2 oder Satzlängengebiet) und `EXT` (Extension) angegeben werden.

URLServer: Mit den neuen Funktionen `EDTW` (S. 77) und `EDTWBIN` können Dateien direkt mit dem Programm `EDT` für Windows geöffnet werden. Im Binärmodus werden die Daten in einem speziellen Format mit Satzlängengebiet übertragen, so daß alle Daten einschl. Satzendezeichen, wie `X'0A'`, `X'0D'` usw. übertragen werden können.

URLServer: Mit der neuen Anweisung `FILEACCESS` (S. 53) kann die Zugriffsberechtigung flexibler gesteuert werden. Hier kann, ähnlich wie mit der Anweisung `HTMDIR`, mit erweiterten Suchbegriffen der Zugriff auf Dateien erlaubt werden.

URLServer: Mit der neuen Anweisung `TEMPLATE` (S. 68) kann der Prefix für die Template-Dateien definiert werden.

Januar 2003

V1.16 URLServer: In den Template-Dateien (S. 92) `TEMPLATE.FSTAT` und `TEMPLATE.LIBTOC` bzw. in der Parameterdatei, Anweisung `USERLINK` (S. 68) und `FSTAT_par` (S. 55) können folgende Einstellungen für den Aufbau der Liste definiert werden:

- Die Spalten, die in der `FSTAT`- und `LIBTOC` - Liste ausgegeben werden sollen.
- Zusätzliche Attribute, wie Schriftart und Schriftgröße.
- Zusätzliche variable Spalten, in denen in Abhängigkeit des Dateityps weitere Links erzeugt werden können.

Dezember 2002

V1.14 URLServer: Bei der Funktionen `XVIEW` (S. 89) kann der Parameter `MIME-TYPE` angegeben werden. Dieser Mime-Type wird im HTTP-Header als Content-Type gesendet. Dadurch wird die entsprechende Datei mit dem Programm angezeigt, das mit diesem Mime-Type verknüpft ist. Bestimmte Dateien (z.B. PDF, OPGCRYPT, DOC und XLS) werden automatisch mit dem zutreffenden Content-Type gesendet.

URLServer: Bei der Funktion `XVIEW` (S. 89) kann der Parameter `TRANS=Y` angegeben werden. Damit werden die Daten von EBCDIC nach ISO8859 übersetzt und ohne weitere Änderungen an den Browser gesendet.

Oktober 2002

V1.11 URLServer: Die neue Funktion `PDF` (S. 84) erzeugt aus einer Druckdatei oder aus einer Datei ohne Drucksteuerzeichen eine PDF-Datei und veranlaßt den Browser, die Daten mit Acrobat Reader anzuzeigen. Die Optionen (z.B. Schriftart, Schriftgröße, Seitengröße, Zeilenanzahl) für die Konvertierung können in der Parameterdatei (S. 61) als Standard definiert und zusätzlich beim Aufruf der Funktion angegeben werden.

URLServer: Neue Parameter `ACTION` und `ACTIONPAR` für die Funktion `FSTAT` (S. 81). Damit kann der Aufbau des Links beeinflusst werden, der für jedes Element in der `FSTAT`-Liste erzeugt wird.

September 2002

V1.09 URLServer: Mit der Funktion `ENTER` (S. 80) können aus Sicherheitsgründen nur noch Prozeduren aus der `CGILIB` gestartet werden.

August 2002

V1.06 URLServer: Für die Funktion `VIEW`, `FSTAT` und `LIBTOC` können Template-Dateien (S. 92) zur Anpassung der Darstellungsattribute für Textdateien bereitgestellt werden.

Juli 2002

V1.05 `EVENTSCAN`: Neuer Parameter-Abschnitt `<VALIDATION>` (S. 26), in dem definiert werden kann, für welchen Zeitraum die einzelnen Aktionen gültig sein sollen, z.B. `TRAP` während der Arbeitszeit, `MAILTO` außerhalb der Arbeitszeit.

EVENTSCAN: Die Parameter (S. 19) können auch von einer Datei eingelesen werden (LINK=EVENTPAR). Bisher konnten die Parameter nur von SYSDTA gelesen werden.

EVENTSCAN: Mit INTR *tsn*, STOP (S. 19) kann das Programm beendet werden. Mit INTR *tsn*, RELOAD kann das Programm beendet und neu gestartet werden.

URLServer: Einführung eines Authentifizierungs-Verfahrens (S. 45). Es können pro Benutzergruppe geschützte Bereiche eingerichtet werden. Der Zugriff ist nur nach Eingabe des zutreffenden User-Namens und des dazugehörigen Paßwortes möglich.

URLServer: Bei der Funktionen VIEW (S. 89) und XVIEW und FSTAT (S. 81) kann auch ein PLAM-Element in der Form `lib(typ,elem)` angegeben werden.

Juni 2002

V1.03 URLServer: Mit der neuen Funktion ENTER (S. 80) kann eine Prozedur als Enter gestartet werden und das Ergebnis (SYSOUT, SYSLST oder beliebige Datei) angezeigt werden.

März 2002

V1.02 URLServer: Neue Anweisungen DEFDIRPOS (Standard-User-ID POSIX), DEFDIRBS2 (Standard-Verzeichnis POSIX) und SUBTASKS (Anzahl Sub-Tasks).

Steuerung des URL-Servers über Console-Kommando INTR: ADDTASKS, REMTASKS, RELOAD und STOP.

Erweiterung Parameter für Benutzermodule zum URL-Server: Adresse Ausgabebereich, der wahlweise mit EBCDIC oder ASCII-Daten gefüllt werden kann.

Falls der URL-Server unter TSOS läuft, ist der Zugriff nur auf Dateien mit dem Attribut SHARE erlaubt. Sollen alle Dateien zugreifbar sein, kann dies mit der Anweisung VIEW=ALL (S. 69) erreicht werden.

Inhaltsverzeichnis

1	ALLGEMEINE VEREINBARUNGEN	14
2	KURZBESCHREIBUNG	15
	Komponenten im BS2000	15
	Komponenten im Windows	15
	Systemvoraussetzungen	15
3	INSTALLATION	16
	Windows-Installation EventList	16
	Windows-Installation FServer	16
	BS2000-Installation	16
4	EVENTLIST / EVENTSCAN	19
	Steuerung des Servers über INTR	19
	Abschnitt <CONS>	19
	USERID User-ID für Console-Zugang	19
	PASS Passwort für Console-Zugang	19
	ROLE Operator-Role für Console-Zugang	19
	ADD-MSG-ID Hinzufügen von Meldungsschlüsseln	20
	Abschnitt <GROUP>	20
	GROUP Empfänger-Gruppen	20
	Abschnitt <ACTION>	20
	ACTION Definition von Actions	21
	Abschnitt <SCAN>	22
	FILE Datei mit SEND-Anweisungen	22
	SEND Trap senden	22
	Abschnitt <SNMP>	25
	COMMUNITY Community-String	25
	ENTERPAR Parameter für ENTER-JOB	26
	NAC Net-Administration-Center	26
	LICENSE Lizenz-Nummer	26
	Abschnitt <VALIDATION>	26
	Valid-Profil Definition Gültigkeit	26
5	TSEND - TRAP SENDEN	30
	Aufruf TSend über Kommando SEND-TRAP	30
	Aufruf TSend über CFS	30
	Aufruf TSENDUP von Benutzerprogramm	31
	Parameter für TSENDUP bzw. TRAPSEND	32

6	FSEND - DATEI SENDEN	34
	Aufruf FSend über Kommando SEND-FILE	35
	Aufruf FSend über Kommando SEND_EDTWFIL.....	36
	Aufruf FSend über CFS	37
	Aufruf FSENDUP von Benutzerprogramm	38
	Parameter für FSENDUP bzw. FILESEND	39
7	URL-SERVER	44
	Steuerung des Servers über INTR	44
	SYSDTA-Anweisungen.....	45
	AUTHFILE Daten für Authentifizierungsverfahren	45
	CGIACCESS Zugriff auf Module erlauben	47
	CGILIB Bibliothek für Benutzermodule	48
	CFSACCESS Zugriff auf CFS-Prozeduren erlauben	48
	CFSLIB CFS-Bibliothek	49
	CODE Code-Variante	49
	CODEFILE Datei mit User-Codes	50
	DEFDIRBS2 Standard-Verzeichnis für BS2000	51
	DEFDIRPOS Standard-Verzeichnis für POSIX.....	51
	ENTERACCESS Zugriff Enter-Pozeduren erlauben	51
	ENTER-TIMEOUT Maximale Zeit zum Starten von Enter-Job	52
	FGG File Generation Group.....	52
	FILEACCESS Datenzugriff erlauben	53
	FSTAT_par Layout für FSTAT, LS und LIBTOC.....	55
	HTMDIR Datenzugriff erlauben	57
	HTMINDEX Index-Datei.....	58
	HTMEXT Extension HTML-Dateien	58
	LIBTOC_par Layout für LIBTOC	58
	LS_par Layout für LS	59
	LICENSE Lizenz-Nummer	59
	LOGFILE Logging-Datei.....	59
	ERRLOGFILE Error-Logging-Datei.....	59
	PDF-AUTOSIZE Automatische Größenanpassung für Funktion PDF	59
	PDF-BACKGROUND Standard-Hintergrundbild für Funktion PDF	60
	PDF-CODE Standard-Code-Konvertierung für Funktion PDF	61
	PDF-FONT Standard-Font für Funktion PDF	61
	PDF-FONTSIZE Standard-Fontsize für Funktion PDF	61
	PDF-MARGIN-LEFT Standard-Seitenrand links für Funktion PDF	62
	PDF-MARGIN-TOP Standard-Seitenrand oben für Funktion PDF	62
	PDF-MAXLINES Standard-Zeilenzahl für Funktion PDF	62
	PDF-MODE Standard-Modus für Funktion PDF.....	62
	PDF-PAGE Standard-Seitengröße für Funktion PDF	63
	PDF-LAYOUT Optionen für PDF-Generierung	63
	PORT Port-Nummer.....	67
	POST_EXIT User-Exit-Modul	68
	SUBTASKS Anzahl der Sub-Tasks	68
	TEMPLATE Prefix für Template-Dateien	68
	USERLINK Links in FSTAT/LIBTOC/LS-Liste.....	68
	VIEW Zugriff auf alle Dateien	69
	VIEW-PASSWORD Passwörter in BS2-Kommandos maskieren	70
	VIEWTAG HTML-Sonderzeichen	70
	XVIEWEXT Extension für binäre Übertragung	70
	Format der Browser-Eingabe.....	71
	URL-Adresse in einem Link	73

Variable URL-Adresse	73
Beschreibung der Funktionen	74
CGI-BIN	Benutzer-Modul starten 74
CFS	CFS-Prozedur starten 74
EDTW / EDTWBIN	Datei mit EDTW öffnen..... 77
EDTWUPD	Datei mit EDTW ändern 77
ENTER	Enter starten..... 80
FSTAT	Dateienliste erzeugen..... 81
LIBTOC	Inhaltsverzeichnis Bibliothek erzeugen 81
LS	POSIX-Dateienlist erzeugen 81
PDF	PDF-Datei erzeugen und anzeigen 84
VIEW	Datei anzeigen 89
XVIEW	Binärdatei anzeigen / öffnen..... 89
Template-Datei	92
Schnittstelle User-Modul Funktion CGI-BIN	96
Parameter	96
Returncode	97
Funktionen	97
Beispiele	97
Schnittstelle User-Modul URLEXIT.....	99
Parameter	99
Returncode	100
8 PDFGEN.....	101
Aufruf PDFGEN	101
Aufruf PDFGEN über Kommando CREATE-PDF-FILE.....	101
Parameter PDFGEN	102
AUTOSIZE	Automatische Größenanpassung 102
BACKGROUND	Hintergrundbild und Wasserzeichen..... 103
CODE	Code-Konvertierung 104
CODEFILE	Datei mit User-Codes 105
FONT	Font 106
FONTSIZE	Schriftgröße..... 106
INPUT	Eingabedatei 106
LICENSE	Lizenz-Nummer 107
MARGIN-LEFT	Seitenrand links..... 107
MARGIN-TOP	Seitenrand oben..... 107
MAXLINES	Maximale Zeilenanzahl..... 107
MODE	Drucksteuerzeichen..... 107
PAGE	Seitengröße..... 108
PDF	Ausgabedatei 108
VIEW-PASSWORD	Passwörter in BS2-Kommandos maskieren 108
Schnittstelle Modul PDFGENUP.....	110
Parameter	110
Returncode	110
Makro PDFPAR	110
Headerfile PDFPAR.H	111
9 FSENDsrv	112
BS2000-Programm FSendSRV	112
Parameter	112
PC-Programm FSendSRV (Windows/Linux).....	113

Parameter	113
A ANLAGE1: CODE-UMWANDLUNG ZWISCHEN EBCDIC UND ASCII	114
Code-Umwandlung	114
7-bit-Code	114
Erweiterte Zeichensätze	114
Aktivierung der 8-bit-Umgebung	115
Translate-Tabelle EBCDIC → ASCII	116
Translate-Tabelle ASCII → EBCDIC	116
Code-Tabelle EBCDIC.DF.03 (CCSN: EDF03IRV)	117
Code-Tabelle EBCDIC.DF.04.1 (CCSN: EDF041)	118
Code-Tabelle EBCDIC.DF.04-DRV (CCSN: EDF04DR)	119
Code-Tabelle ASCII ISO-8859-1	120
S STICHWORTVERZEICHNIS	121

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 test@opg.de Einzugeben ist: from test@opg.de
<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

Komponenten von OPGCom im BS2000

EventScan	Das Programm analysiert Console-Meldungen und sendet in Abhängigkeit der definierten Aktionen in der Parameterdatei Trap's an EventList bzw. an ein anderes Net Administration Center oder E-Mail's.
TSend	Über die Unterprogramm-Schnittstelle TSENDUP können Trap's an EventList bzw. an ein anderes Net Administration Center versandt werden. Das Unterprogramm kann auch über CFS (Kommando (TSEND) oder Prozedurkommando *RUN) aufgerufen werden. Der Aufruf ist auch über das Kommando SEND-TRAP möglich.
FSend	Über die Unterprogramm-Schnittstelle FSENDUP können Dateien binär oder im Text-Format an einen PC versandt werden. Das Unterprogramm kann auch über CFS (Kommando (FSEND) oder Prozedurkommando *RUN) aufgerufen werden. Der Aufruf ist auch über das Kommando SEND-FILE und SEND-EDTWFILE möglich.
URL-Server	Web-Server im BS2000 mit der Möglichkeit, Benutzermodule aufzurufen, die beliebige Daten aus dem BS2000 oder POSIX für den Browser bereitstellen.
PDFGEN	Aus einer beliebigen BS2000-Datei / PLAM-Element / ZIP-Element oder POSIX-Datei wird eine PDF-Datei im PAM-Format erzeugt. Falls die Ursprungsdatei Drucksteuerzeichen enthält (Siemens oder ASA), werden diese für den Zeilen- und Seitenvorschub benutzt. Ansonsten wird nach einer festgelegten Anzahl von Zeilen ein Seitenvorschub erzeugt.. Der Aufruf ist auch über das Kommando CREATE-PDF-FILE möglich.
FSendSRV	Dateien, die über die Funktion EDTW des URLServers oder über das BS2000-Programm FILESEND zu einem PC gesendet und dann in den EDTW eingelesen werden, können nun auch über einen Proxy-Server empfangen und/oder zurückgesandt werden. Dadurch ist es möglich, Dateien zu einem PC zu senden, der im BS2000 nicht bekannt ist. Ebenfalls ist es möglich zum Empfangen im BS2000 statt dynamischer Ports nur einen zentralen Port zu verwenden.

Windows-Komponenten von OPGCom

EventList	Empfangen und Anzeigen der Meldungen von EventScan und Tsend.
FServer	Das Program empfängt die von dem Programm FSend versandten Dateien, erstellt eine temporäre Datei und startet das dazugehörige Anzeigeprogramm.
FSendSRV	Das Server-Programm leitet Daten vom BS2000-Server FSendSRV an das Programm FServer auf dem Benutzer-PC weiter und empfängt die Daten von EDTW auf einem zentralen Port und sendet diese wiederum an den BS2000-Server FSendSRV zum Schreiben der Daten zurück.

Systemvoraussetzungen

- BS2000-Betriebssystem ab Version OSD1 und Subsystem `SOCKETS` ab Version 1.1.
- Das Subsystem `SOCKETS` muß geladen sein.

3 Installation

Installation und Aufruf PC-Programm EventList.EXE

Installation:

Installation mit `SetupEventList.exe`

Programmaufruf:

`c:\programme\opg\eventlist\eventlist.exe`

Installation und Aufruf PC-Programm FServer.EXE

Installation:

b) Installation mit `SetupFServer.exe`

c) Datei `FSERVER.INI` anpassen:

Die Parameter sind in der ausgelieferten Datei `FSERVER.INI` beschrieben. Insbesondere können hier die Port-Nummer, der Dateipräfix für die temporären Dateien, die zulässigen Dateiendungen und die zulässigen Hosts definiert werden.

Programmaufruf:

`c:\programme\opg\fserver\fserver.exe`

Installation und Aufruf BS2000-Programme

Installation:

a) Starten `opgcom.exe`. Die Installationsdateien werden in ein beliebiges Verzeichnis übertragen (Standard = `c:\programme\opg\opgcom`). Danach wird das Programm `SETUP.EXE` gestartet. Damit werden neue Dateitypen für den URLServer registriert.

b) Datei `opgcom.exp` im Text-Modus ins BS2000 übertragen (OPENFT oder FTP)

c) Datei mit dem CFS-Kommando `FILIMP opgcom.exp,opgcom.LIB` in PLAM-Bibliothek umwandeln.

d) Enter-Prozedur `J/EVENTSCAN` anpassen.

e) Enter-Prozedur `J/URLSERVER` anpassen:

Als Wert für den Prozedur-Parameter `DEFAULTDIR` müssen Sie eine Kennung eintragen, auf die der Zugriff erlaubt sein soll. Der Wert dieses Parameters wird für die Anweisung `DEFDIRBS2` (S. 51) und `FILEACCESS` (S. 53) verwendet, d.h. der Zugriff auf diese Kennung ist erlaubt und die Kennung wird als Standard-Kennung für BS2000-Dateien benutzt. Die Bibliothek `OPGCOM.LIB` muß sich entweder in der User-ID `DEFAULTDIR` befinden oder Sie müssen beim Prozedur-Parameter `LIB` die User-ID ergänzen.

f) Die Installationsdateien `S/SYSSII.OPGCOM` und `X/SYSSDF.OPGCOM` aus der Bibliothek `OPGCOM.LIB` selektieren..

g) Installation IMON und Erzeugen der SDF-Kommandos:

Mit der Prozedur `J/INSTALL` wird die IMON-Registrierung durchgeführt und folgende SDF-Kommandos erzeugt:

<code>SEND-FILE</code>	Datei zu einem PC senden und anzeigen
<code>SEND-EDTWFIL</code> ,	Datei zu einem PC senden und mit EDTW bearbeiten
<code>SEND-TRAP</code> ,	Trap senden
<code>OPGCRYPT-FILE</code>	Datei verschlüsseln oder entschlüsseln
<code>CREATE-PDF-FILE</code>	PDF-Datei erzeugen
<code>START-OPG-SERVICE</code>	ASTI-Service starten
<code>STOP-OPG-SERVICE</code>	ASTI-Service beenden

Der Installationspfad kann als Parameter angegeben werden. Standard = `OPGCOM`.

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

i) Mit der Prozeduren `J/REP` Lizenznummer und ev. weitere Optionen für `FSEND` festlegen.

ASTI-Services:

Die Kommandos `SEND-FILE`, `SEND-EDTWFIL`, `SEND-TRAP`, `OPGCRYPT-FILE` und `CREATE-PDF-FILE` wenden 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. Der ASTI-Service muß vorher, in der Regel beim System-Start, gestartet werden. Die Services können mit dem Kommando `START-OPG-SERVICE` gestartet und mit dem Kommando `STOP-OPG-SERVICE` beendet werden. Das Subsystem ASTI ist immer vorhanden, weil es auch vom System benutzt wird.

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`).

Dadurch erübrigt sich in der Regel die Verwendung der Unterprogramm-Schnittstelle.

Programmaufruf:

```
/ENTER OPGCOM.LIB (EVENTSCAN)
```

bzw.

```
/ENTER-PROC OPGCOM.LIB (URLSERVER)
```

Die Prozedur geht von folgenden Voraussetzung aus: Die Bibliothek `OPGCOM.LIB` muß sich in der Kennung befinden, die im Prozedur-Parameter als `DEFAULTDIR` angegeben ist. Für den Test sollten sich in diesem Verzeichnis einige `SAM`-Dateien befinden, die mit dem Browser betrachtet werden können. Durch die Eingabe "`http://host[:port]`" im Browser wird die Demo-Indexseite (`opgcom.lib(t,index.html)`) angezeigt.

Parameter für `J/URLSERVER`:

DEFAULTDIR		Standard-User-ID für Dateien und Bibliotheken
LIB	OPGCOM.LIB	Bibliothek zum Laden von URLSERVER und Benutzermodulen sowie Test-Dateien T/.....
HTMINDEX	OPGCOM.LIB/T/INDEX.HTML	Index-Seite, die benutzt wird, wenn nur der Hostname angegeben wird.
AUTHFILE	*NO	Datei mit Authentifizierungs-Einstellungen (deaktiviert).
PORT	80	Portnummer.
TEMPLATE	DEMO.TEMPLATE.	Prefix für Template-Dateien.
VIEW	ALL	ALL: Zugriffsberechtigung für alle Dateien von DEFAULTDIR. SHARE Zugriffsberechtigung nur auf SHARE-Dateien.
CODE	DE/INT	DE: Daten enthalten Umlauten INT: Daten enthalten Zeichen "{ } [\] ~"
SUBTASKS	0	Anzahl der zusätzlichen Server-Tasks

4 EventList / EventScan

EventScan analysiert alle Console-Meldungen und sendet in Abhängigkeit der Einstellungen in der Parameterdatei bei bestimmten Meldungen TRAP's (Kurznachrichten) oder E-Mails an festgelegte Benutzer. Die Aktionen können auch über die UP-Schnittstelle initiiert werden. Über das BS2000-Kommando /TYPE oder dem CMD-Makro gibt es wiederum eine einfache Schnittstelle zu \$CONSOLE.

Steuerung des Servers über INTR

INTR *tsn*, STOP | RELOAD

Zur Steuerung des Servers können folgende INTR-Kommandos an der Console eingegeben werden.

STOP

Beenden des URL-Servers.

RELOAD

Beenden und neu Starten des URL-Servers. In der ENTER-Prozedur muß nach dem Programmaufruf der Prozeßschalter 1 abgefragt werden. Ist der Prozeßschalter "ON", muß wieder zum Programmaufruf verzweigt werden (siehe Musterprozedur in der Bibliothek OPGCOM.LIB).

Parameter für EVENTSCAN

Die Parameter werden entweder SYSDTA oder von der Datei, die mit dem LINK-Namen EVENTPAR zugewiesen ist, gelesen. Die maximale Satzlänge beträgt 2.048 Bytes. Fortsetzungszeilen sind nicht vorgesehen.

Abschnitt <CONS>

Dieser Abschnitt enthält Angaben für den Console-Zugang für YOPNCON auf \$CONSOLE.

User-ID für Console

USERID=*user-id*

User-ID für den Console-Zugang.

Passwort für Console

PASS=*password*

Passwort für den Console-Zugang. Ist kein Passwort vergeben, muß "PASS=" angegeben werden.

Operator-Role für Console

ROLE=*operator-role*

Name der Operator-Role für den Console-Zugang.

Hinzufügen von Meldungsschlüsseln

ADD-MSG-ID=(*id*[,*id*....]) Es können bis zu 32 Meldungsschlüssel angegeben werden, für die ein MODIFY-MSG-SUBSCRIPTION ADD-MSG-ID=.... abgesetzt wird. Dadurch erhält die Eventscan-Konsole die angegebenen Meldungen.

id Meldungsschlüssel (7-stellig).

Beispiel:

ADD-MSG-ID=(FTR0341,FTR0307,FTR0038,FTR0264)

Siehe dazu auch Dokumentation openFT

Abschnitt <GROUP>

Dieser Abschnitt enthält Angaben zur Definition von Gruppen von Empfängern.

Empfänger-Gruppen

grp-name=partner,partner[,*partner*....]

grp-name Name der Gruppe.

partner Empfänger der Nachricht:

- Prozessorname eines PC's oder
- NAC-Name oder
- *action:param*.

Die Angabe *param* ersetzt bei der Erzeugung der Enter-Prozedur den symbolischen Parameter !TO (siehe Abschnitt <ACTION>). In der Regel wird *param* Empfängerangabe enthalten, z.B. eine E-Mail-Adresse.

Diese Anweisung kann mehrmals vorkommen.

Beispiel:

grp1=pc1,pc2,mailto:info@opg.de

Abschnitt <ACTION>

Dieser Abschnitt enthält ENTER-Prozeduren zum Starten von beliebigen Applikationen, z.B. des Programms MAILS. Falls im Abschnitt <GROUP> oder in der Anweisung SEND als Partner die Aktion angegeben ist, z.B. *MAILTO:mail-adresse*, wird aus den Angaben dieses Abschnitts eine ENTER-Prozedur erzeugt und gestartet, wobei die Platzhalter durch die aktuellen Werte ersetzt werden.

Definition von Aktionen (Versenden von Mails usw.)

ACTION=*action-name*
[VALID=...]

Name der Action, die in der Anweisung `SEND` oder im Abschnitt `<GROUP>` angegeben wird und ggf. ein Validation-Profil.

`/LOGON`
`bs2-cmd-oder-sysdta`
`bs2-cmd-oder-sysdta`

Danach folgen alle Anweisungen für eine Enter-Prozedur (Ende = nächste Anweisung `ACTION` oder nächster Abschnitt. Diese Daten werden in eine temporäre Datei geschrieben und mit dem Makro `ENTER` gestartet. Alle gewünschten Enter-Parameter müssen im `LOGON`-Kommando definiert werden. Die Folge von Kommandos bzw. `SYSDTA`-Anweisungen muß mit `/LOGON` beginnen und mit `/LOGOFF` enden. Sollen die gesamten Anweisungen für das Programm EventScan über `SYSCMD` eingelesen werden, müssen `BS2000`-Kommandos mit Blank + "/" beginnen. Ein Schrägstrich in Spalte 1 würde EOF für `SYSDTA` erzeugen.

.....
`/LOGOFF`

oder

ACTION=*action-name*
[VALID=...]
FILE=*enterfile*

Name der Action
 Name einer Datei mit einer Enter-Prozedur. Die Datei wird als Muster für die Erzeugung der endgültigen Enter-Datei benutzt.

oder

ACTION=*action-name*
TRAP=YES
VALID=...

Bei dieser Action wird ein TRAP gesendet unter Berücksichtigung der `VALID`-Attribute. Die Partner werden in der `SEND`-Anweisung (S. 22) als Parameter nach dem Doppelpunkt angegeben.

Beispiel:

```
<ACTION>
ACTION=TRAP1
TRAP=YES
VALID=TRAPVAL1
<VALIDATION>
TRAPVAL1=0/08:10/14:05,0/14:10/23:00
<SCAN>
SEND=N;XZZZ;!MSG;TRAP1:PC1,PC2,GROUP1
```

In der Enter-Prozedur können folgende Platzhalter enthalten sein, die mit den aktuellen Werten ersetzt werden:

<code>!TO</code>	Parameter aus <code>SEND</code> -Anweisung, z.B. E-Mail-Adresse oder sonstige Empfänger-Angaben, die nach dem Action-Namen durch Doppelpunkt getrennt angegeben werden
<code>!MSG</code>	Vollständige Original-Console-Meldung
<code>!MSG-STR</code>	wie <code>!MSG</code> , jedoch mit verdoppelten Hochkommas
<code>!MSG-INS1</code>	Inhalt des ersten Meldungs-Inserts der Console-Meldung
<code>!MSG-INS2</code>	Inhalt des zweiten Meldungs-Inserts der Console-Meldung
<code>!MSG-INS3 - !MSG-INS9</code>	Inhalt des entsprechenden Meldungs-Inserts
<code>!DATE</code>	aktuelles Datum
<code>!TIME</code>	aktuelle Uhrzeit
<code>!TSN</code>	aktuelle TSN
<code>!MSG7</code>	7-stelliger Meldungsschlüssel
<code>!LEVEL</code>	Text Message-Level
<code>!HOST</code>	Host-Name
<code>!SEARCH</code>	Suchbegriff aus der aktuellen <code>SEND</code> -Anweisung

Beispiel siehe unten und ausgelieferte Muster-Enter-Prozedur (Element `S/EVENTSCAN.PAR` in der PLAM-Bibliothek `OPGCOM.LIB`).

Zuweisung eines Validation-Profils

VALID=*Validation-name*

Name eines Validation-Profils (S. 26), in dem definiert werden kann, in welchem Zeitraum die Aktion gültig sein soll.

Abschnitt <SCAN [*host*] >

In diesem Abschnitt werden die Filter und die dazugehörigen Aktionen definiert.

host

Die wahlweise Angabe des Hostnamens bewirkt, daß dieser Abschnitt nur für die Verarbeitung auf dem angegebenen Host gültig ist. Dadurch ist es möglich eine zentrale Parameterdatei für mehrere Hosts zu erstellen.

Der Abschnitt SCAN kann mehrmals vorkommen.

Datei mit weiteren SEND-Anweisungen

FILE= *send-file*

send-file

Datei, die SEND-Anweisungen enthält. In der Datei dürfen nur SEND-Anweisungen und Kommentarzeilen vorkommen. Die Anweisung FILE kann mehrmals vorkommen. Die Anweisungen SEND können auch teilweise in der SYSOUT-Datei und teilweise in der SEND-FILE enthalten sein.

Trap an EventList oder E-Mail senden

SEND= *level; string | ext-search; message; partner[, partner]*

string

Suchstring, der in jeder Console-Meldung gesucht wird. Die Zeichenfolge wird, wie angegeben in Klein- Großschreibung verglichen. Der String wird nicht in Hochkommas eingeschlossen. Kommt in einem Suchstring das Zeichen ";" vor, muß es verdoppelt werden. Der String darf maximal 256 Bytes lang sein.

ext-search

`SEARCH:such [vk such] | (s-dat)`

Erweiterter Suchstring, der in jeder Console-Meldung gesucht wird.

Der Suchstring kann beliebig viele Suchargumenten enthalten, die mit und (+), oder (,) bzw. Wildcard (*) miteinander verknüpft sind. Für jedes Suchargument kann eine Spaltenbereich und ein Vergleichs-Operator (>|<|-) angegeben werden.

such

`[col] [r] item`

<i>col</i>	<p>Spaltenbereich, in dem die gesuchte Zeichenfolge beginnen muß.</p> <p><i>:col1-col2:</i></p> <p>Das erste Zeichen der gesuchten Zeichenfolge muß im Spaltenbereich zwischen <i>col1</i> und <i>col2</i> beginnen.</p> <p><i>:col1:</i></p> <p>Die Zeichenfolge wird nur an der angegebenen Spalte <i>col1</i> gesucht und muß dort beginnen.</p> <p><i>>:col1: <:col1:</i></p> <p>Die Zeichenfolge wird im Bereich ab Spalte <i>col1</i> bis Satzende (<i>>:col1:</i>) bzw. vom Satzanfang bis Spalte <i>col1</i> gesucht (<i><:col1:</i>).</p> <p>Standard: Suche in gesamten Spaltenbereich (von Spalte 1 bis Zeilenende).</p>
<i>r</i>	<p><i>> < -</i></p> <p><i>></i> Suche nach einer Zeichenfolge <i>> item</i> <i><</i> Suche nach einer Zeichenfolge <i>< item</i> <i>-</i> Suche nach einer Zeichenfolge ungleich <i>item</i></p> <p>Standard: Suche nach einer Zeichenfolge = <i>item</i>.</p>
<i>item</i>	Suchzeichenfolge: <i>C'string' X'string' A'string' L'string'</i>
<i>C'string'</i>	Zeichenfolge in Großbuchstaben. Vor dem Vergleich wird die Zeichenfolge in Großbuchstaben umgewandelt. <i>C'string'</i> kann zu <i>'string'</i> abgekürzt werden.
<i>A'string'</i>	Die Zeichenfolge wird unabhängig von der Klein-/ Großschreibung gesucht.
<i>L'string'</i>	Die Zeichenfolge werden, wie geschrieben mit Klein-/ Großschreibung gesucht.
<i>X'string'</i>	Die Zeichenfolge enthält einen hexadezimalen Zeichenfolge. Enthält <i>string</i> Hochkommas ('), so müssen diese verdoppelt werden (").
<i>vk</i>	<p><i>, + *</i></p> <p>Verknüpfungsoperator mit dem vorausgegangenen Suchargument <i>such</i>.</p> <p><i>,</i> Suche das vorausgegangene oder das nachfolgende Suchargument. Die Suchbedingung gilt als erfüllt, wenn zumindest eines der beiden Suchitems enthalten ist.</p> <p>+ Suche das vorausgegangene und das nachfolgende Suchargument. Die Suchbedingung ist erfüllt, wenn beide Suchargumente enthalten sind. Die Reihenfolge der Suchargumente in der Zeile ist ohne Bedeutung.</p>

- * Suche das vorausgegangene **und** das nachfolgende Suchargument. Die Suchbedingung ist erfüllt, wenn beide Suchargumente enthalten sind. Die Suchargumente müssen in der gleichen Reihenfolge auftreten, wie im Suche-Kommando angegeben.

Es können beliebig viele Konstrukte der Art *vk such* aneinandergereiht werden.

Hinweise:

Die Reihe der angegebenen Suchargumente und Verknüpfungsoperatoren wird linear abgearbeitet. Falls mehrere mit "+" bzw. "*" verknüpfte Suchargumente angegeben wurden und eines von ihnen nicht in der Zeile enthalten ist, so wird der Suchvorgang beendet bzw. beim nächsten, mit oder "," verknüpften Such-Item fortgesetzt.

Beispiele:

`xyz`

Suche im gesamten Arbeitsbereich die Zeichenfolge 'xyz'.

`search:'C' ' ''`

Suche nach der Zeichenfolge 'C_' im gesamten Arbeitsbereich (Hochkommas in dem zu suchenden String müssen verdoppelt werden).

`search::73:-' '`

Suche in Spalte 73 jeder Zeile nach einem Zeichen ungleich Blank. Zeilen mit weniger als 73 Stellen werden hier nicht als Treffer erkannt.

`search::100-200:x'47'`

Suche in den Zeilen 10 bis 20 jeweils im Spaltenbereich 100 - 200 nach 'X47'.

`search:='*' (' , 'DC '*' ('*'))'`

Es werden alle Zeilen gesucht, die eine der beiden Bedingungen erfüllen:

- Zeichen '=' und irgendwann danach Zeichen '(' . z.B. '=A(...)', '=V(...)'
- Zeichenfolge 'DC ' und irgendwo danach die Zeichen '(' und ')'. z.B. 'DC A(...)', 'DC Y(...)'

`search:-L'a'+-L'b'+-L'c'`

Es werden alle Datensätze gesucht, die keinen der Kleinbuchstaben a, b oder c enthalten.

`search:L'a',>L'a'+<L'z',L'z'`

Es werden alle Datensätze gesucht, die mindestens einen Kleinbuchstaben enthalten.

level

Message-Level.

N Note
I Information
W Warning
A ALERT
C Critical
E Emergency
P Panic

message

Meldung, die auf dem PC ausgegeben werden soll. Kommt in dem Message-Text das Zeichen ";" vor, muß es verdoppelt werden. Der String darf maximal 256 Bytes lang sein. Folgende Ersatz-Strings werden durch die entsprechenden Werte ersetzt:

!MSG Vollständige Original-Console-Meldung
!MSG-STR wie !MSG, jedoch mit verdoppelten Hochkommas
!MSG-INS1 Inhalt des ersten Meldungs-Inserts der Console-Meldung
!MSG-INS2 Inhalt des zweiten Meldungs-Inserts der Console-Meldung
!MSG-INS3 - !MSG-INS9 Inhalt des entsprechenden Meldungs-Inserts
!DATE aktuelles Datum
!TIME aktuelle Uhrzeit
!TSN aktuelle TSN
!MSG7 7-stelliger Meldungsschlüssel
!LEVEL Text Message-Level
!HOST Host-Name

partner

Empfänger-Adresse der Nachricht:

- Prozessorname eines PC's oder
- NAC-Name (siehe Abschnitt <SNMP> (S. 25)) oder
- Gruppenname (siehe Abschnitt <GROUP> (S. 20)), oder
- *action:param* (siehe Abschnitt <ACTION> (S. 21))

action:

Name einer im Abschnitt <ACTION> definierten Aktion

param

Empfänger-Angaben, die bei der Definition der Action mit dem Platzhalter !TO verwendet werden können. Bei einer Aktion mit der Eigenschaft "TRAP=YES" sind als *param* ein oder mehrere Prozessornamen oder Gruppennamen anzugeben.

Beispiel:

```
MAILTO:info@opg.de  
TRAP:pc1,pc2,group1
```

Diese Anweisung kann mehrmals vorkommen. Die SEND-Anweisungen können auch in eine eigene Datei geschrieben und mit der Anweisung FILE zugewiesen werden.

Abschnitt <SNMP>

Community-String

COMMUNITY=community-string

community-string

Community-String. Standard = OSDTRAP

Parameter für das Kommando ENTER-JOB**ENTERPAR**=*enterparams**enterparams*

Parameter für das BS2000-Kommando ENTER-JOB zum Starten von Jobs, die im Abschnitt <ACTION> (S. 21) definiert sind. Fehlt der Parameter DELETE=, wird der String um den Parameter DELETE=YES ergänzt. Ist ENTERPAR nicht angegeben, wird der Wert "DELETE=YES" verwendet.

Beispiel: ENTERPAR=JOB-CLASS=JCBATCH2,DELETE=YES

Net-Administration-Center**NAC**=*nac-name*

Name eines Net-Administration-Center, z.B. UC4, TRANSVIEW, usw. Wird als Partner ein NAC-Name angegeben, so wird die Nachricht formatgerecht aufgebaut. Diese Anweisung kann mehrmals vorkommen.

Lizenz-Nummer**LICENSE**=*license-string*

Lizenz-Nummer für die unbeschränkte Nutzung. Ohne gültige Lizenz-Nummer läuft das Programm nur einen Monat im Test-Modus.

Abschnitt <VALIDATION>

In diesem Abschnitt kann festgelegt werden, in welchem Zeitraum eine Aktion gültig ist. Zum Beispiel kann in einer Anweisung SEND (S. 22) als Aktion ein Prozessorname (TRAP senden) und eine Aktion MAILTO angegeben werden. Für die Aktion TRAP wird ein Validation-Profil erstellt, das während der Arbeitszeit gültig ist und für die Aktion MAILTO wird ein Profil erstellt, das außerhalb der Arbeitszeit gültig ist. Zur Ausführung gelangt nur die Aktion, die zum Ereignis-Zeitpunkt gültig ist.

Definition eines Validation-Profiles*Validation-Name*=*time*[, *time*,]*Validation-Name*

Name des Validation-Profiles. Über diesen Name wird ein Validation-Profil im Abschnitt <ACTION> einer Aktion zugeordnet.

time=*day*/*time-from*/*time-to* Mit *time* wird ein Zeitraum definiert, in dem eine Aktion gültig ist, d.h. in dem sie beim Eintreten eines Ereignisses ausgeführt wird.

day

0	jeder Tag
1 - 7	Monat bis Sonntag
tt.	Tag des lfd. Monats
tt.mm.	Tag und Monat des lfd. Jahres
tt.mm.jjjj	Datum

time-from hh:mm Beginn des Zeitraums in Stunden und Minuten.

time-to hh:mm Ende des Zeitraums in Stunden und Minuten.

Beispiele für *time*:

0/07:00/18:00	jeden Tag 7:00 - 18:00
1/07:00/18:00	jeden Montag 7:00 - 18:00
01./07:00/18:00	am 1. jeden Monats 7 - 18:00
01.07./07:00/18:00	am 1.7. eines jeden Jahres 7-18:00
01.07.2002/07:00/18:00	am 1.7.2002 7-18:00

Beispiele für gesamte Anweisung:

```
valtrap=1/07:00/17:59,2/07:00/17:59,3/07:00/17:59,4/07:00/17:59  
,5/07:00/17:59  
valmail=5/18:00/23:59,6/00:00/23:59,7/00:00/23:59,1/00:00/6:59
```

Beispiel Parameterdatei:

```
<CONS>
USERID=TT08
PASS=
ROLE=ALLES
<SNMP>
LICENSE=063DC23E
COMMUNITY=OSDTRAP
NAC=OPG4
<VALIDATION>
TEST=0/09:00/16:00,2/09:00/18:00,21./07/08,22.11./07:30/24:00
TRAPVAL=0/08:10/14:05,0/14:10/23:00
MAILVAL=0/08:00/17:30
<ACTION>
ACTION=TRAP
TRAP=YES
VALID=TRAPVAL
ACTION=MAILTO
VALID=MAILVAL
/.EVNTMAIL LOGON
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/START-PROG *MODULE ($OPGCOM.MAILLIB,MAILUP,PROG-MODE=*ANY,RUN-MODE=*ADV)
SERVER OPGNT2
DOMAIN OPG.DE
FROM EVENTSCAN@OPG.DE
TO !TO
SUBJECT NACHRICHT VON !HOST AM !DATE UM !TIME
MSGBEG TYPE=HTML
<BR>
<BODY BGCOLOR="#D6FFED" BACKGROUND="http://www.opg.de/pics/back_3.gif">
<H2>LEVEL: !LEVEL<BR>
RECHNER: !HOST</H2>
DATUM: !DATE !TIME<BR>
TSN: !TSN<BR>
MELDUNGSSCHLUESSEL: !MSG7<BR>
<H4>MESSAGE:<BR>
<FONT COLOR=RED>
!MSG
</H4>
</FONT>
</BODY>
MSGEND
SEND
/LOGOFF
ACTION=MUSTERJOB
FILE=EVENTSCAN.MUSTERPROC
<GROUP>
GRP1=MANFRED,OPG3
GRP2=OPG3,MANFRED,MAILTO:HARLASS@OPG.DE,MAILTO:PILLER@OPG.DE
<SCAN>
SEND=E;TAS0000;!MSG;MANFRED,MAILTO:HARLASS@OPG.DE
SEND=P;TAS0001;!MSG;GRP1
SEND=C;TAS0002;!MSG;GRP2
SEND=A;TAS0003;!MSG;MANFRED
SEND=W;TAS0004;MESSAGE=!MSG,TSN=!TSN,MSG7=!MSG7;MANFRED
SEND=W;EXC0197;MESSAGE=!MSG,TSN=!TSN,MSG7=!MSG7;MANFRED
```

```
SEND=I;TAS0005;!MSG;GRP1,MAILTO:HARLASS@OPG.DE  
SEND=N;TAS0006;!MSG;GRP1  
SEND=A;OFFSEND;ALERT: !MSG AN GRUPPE1;GRP1  
SEND=I;MUSTER;!MSG;MUSTERJOB:HARLASS  
SEND=N;XZZZ;!MSG;TRAP:MANFRED,OPG3  
<END>
```

5 TSEND (Senden Trap)

Aufruf mit SDF-Kommando SEND-TRAP

SEND-TRAP MSG=*message*, TO= *prozessor*, [, MODE=*TRAP (COMMUNITY=*commstr*) | *EVENTLIST (LEVEL=*level* [, OBJECT=*object*] [, APPL=*appl*])]

Das SDF-Kommando übergibt die Parameter dem ASTI-Service OPGTRAPSEND. Die Standardwerte können in der Prozedur SYSPR.SEND-TRAP angepaßt werden.

Für das Kommando ist auch noch der bisherige Namen TRAPSEND als Alias-Name definiert.

message Meldung, die versandt werden soll.

prozessor Prozessorname PC oder IP-Adresse, Beschreibung siehe S. 32

MODE=*TRAP Meldung für Net-Administration-Center (z.B. UC4, Transview usw.) ausgeben, Parameter *commstr* siehe S. 32

MODE=*EVENTLIST Meldung für EVENTLIST ausgeben, Parameter *level*, *object* und *appl* siehe S. 32.

Als Standard für den Parameter MODE gilt: MODE=*TRAP (COMMUNITY=OSDTRAP)

Aufruf von TSend als CFS-Unterprogramm in der Kommandozeile

(TSEND, [*\$user-id.*]OPGCOM.LIB) *prozessor*, *message* [, *level*, *object*, *appl*]

Die Angabe der Bibliothek *opgcom.lib* ist nur beim ersten Aufruf einer CFS-Sitzung zum Nachladen notwendig. Enthält der Parameter *message* Leerstellen oder Kommas, muß er in Hochkommas eingeschlossen werden.

Aufruf von TSend als CFS-Unterprogramm mit Kommando *RUN

RUN (TSEND, [\$user-id.*]OPGCOM.LIB) &*prozessor*, &*message* [, &*level*, &*object*, &*appl*]

Die Parameter müssen vor Aufruf des RUN-Kommandos in CFS-Variable gespeichert werden.

Aufruf von TSENDUP als Unterprogramm von einem Benutzerprogramm

Der Modul TSENDUP wird entsprechend der Standard-Programmverknüpfungsregeln aufgerufen. Er kann auch von höheren Programmiersprachen aufgerufen werden. Alle Strings müssen mit X'00' abgeschlossen werden. Beim Aufruf des Moduls TSENDUP müssen die Register wie folgt geladen sein:

(R1) A (PARAMETERLISTE)

Die Parameterliste enthält folgende Adressen:

A (COMMSTR) Community-String
A (PROZ) Name des Prozessors
A (MESSAGE) Text der zu sendenden Meldung
die folgenden Parameter dürfen nur angegeben werden, wenn die Nachricht von EVENTLIST empfangen wird.

A (LEVEL) Message-Level (Spalte Level in EventList)
A (OBJECT) Nachrichtentyp (Spalte Object in EventList)
A (APPL) Name der Application (Spalte Application in EventList)
A (0) Ende Parameter

COMMSTR	DC	C ' <i>commstr</i> '
	DC	X '00'
PROZ	DC	C ' <i>prozessor</i> '
	DC	X '00'
MESSAGE	DC	C ' <i>message</i> '
	DC	X '00'
LEVEL	DC	C ' <i>level</i> '
	DC	X '00'
OBJECT	DC	C ' <i>object</i> '
	DC	X '00'
APPL	DC	C ' <i>appl</i> '
	DC	X '00'

(R13) A (SAVEAREA)

Register-Sicherstellungsbereich (18 Worte, DC 18F'0'), der vom Aufrufer erstellt werden muß. TSENDUP sichert dort die Register.

(R14) A (RETURN)

Rücksprungadresse im rufenden Programm. Nach Ausführung der Funktionen wird das Programm an dieser Adresse fortgesetzt.

(R15) V (TSENDUP)

Beschreibung der Parameter für TRAPSEND

<i>prozessor</i>	<i>proz-name</i> <i>IP-addr</i> *DEFAULT Adresse des PC's als BCAM-Name oder als IP-Adresse, auf dem die Datei angezeigt werden soll.
<i>proz-name</i>	Prozessorname (max. Länge 8), wie in BCAM definiert.
<i>IP-addr</i>	IP-Adresse (max. Länge 15) des PC's in der Form <i>nnn.nnn.nnn.nnn</i> . Führende Nullen können entfallen, z.B. <i>195.21.5.5</i> .
*DEFAULT	Die IP-Adresse wird wie folgt, automatisch ermittelt: 1. Spalten 1-15 der Jobvariablen mit dem Linknamen *OPGCOM. 2. Spalten 1-15 der Jobvariablen OPGCOM. <i>jobname</i> 3. Spalten 1-15 der Jobvariablen OPGCOM 4. Ist keine Jobvariable vorhanden, wird der Prozessorname des Terminals verwendet.
<i>message</i>	Meldung, die versandt werden soll. Soweit die Nachricht nicht für EventList bestimmt ist (kein Parameter Level vorhanden), gilt folgendes: Die Nachricht darf max. 127 Bytes lang sein. Ggf. wird sie automatisch gekürzt.
<i>commstr</i>	Community-String für Net-Administration-Center (z.B. UC4, Transview usw.). Für Nachrichten an EventList wird der Community-String "OSDTRP" verwendet. Für Nachrichten an ein Net-Administration-Center sind die Vorgaben der entsprechenden Anwendung zu beachten. Maximale Länge: 8
<i>level</i>	Message-Level für die Einordnung der Kategorien und die Spalte "Level" des Programms EventList. N Note I Information W Warning A ALERT C Critical E Emergency P Panic Standard = I.
<i>object</i>	Bezeichnung des Types der Meldung für die Spalte "Object" des Programms EventList. Maximale Länge: 16
<i>appl</i>	Name der Applikation für die Spalte "Application" des Programms EventList. Maximale Länge: 8

Beispiel:

a) Aufruf mit SDF-Kommando TRAPSEND:

```
trapsend to=testpc,msg='Test-Message1 an TestPC'  
trapsend to=testpc,msg='Test-Message2 an TestPC' (a,Gruppel,Testappl)
```

b) Aufruf aus CFS-Kommandozeile:

```
(tsend,opgcom.lib)testpc,'Test-Message an TestPC',i,Gruppel,Testappl
```


c) CFS-Kommando *RUN

```
*&prozessor='testpc'  
*&message='Test-Message an TestPC'  
*&level='i'  
*&object='gruppel'  
*&appl='TestAppl'  
*run (tsend, opgcom.lib) &prozessor, &message, &level, &object, &appl
```

d) Aufruf aus Benutzerprogramm:

```
TRAPSAD DS V (TSENDUP)  
SAVAREA DS 18F  
PARAM DC A (COMMSTR)  
DC A (PROZ)  
DC A (MESSAGE)  
DC A (LEVEL)  
DC A (OBJECT)  
DC A (APPL)  
DC A (0) UNUSED  
  
*  
COMMSTR DC C 'EVENTLIST'  
DC X '00 '  
RPOZ DC C 'TESTPC'  
DC X '00 '  
MESSAGE DC C 'Test-Message an TestPC'  
DC X '00 '  
LEVEL DC C 'I'  
DC X '00 '  
OBJECT DC C 'Gruppel'  
DC X '00 '  
APPL DC C 'TestAppl'  
DC X '00 '
```

6 FSEND (Senden Datei und Anzeigen unter Windows)

Der Modul FILESEND bzw. FSENDUP liest eine BS2000- oder POSIX-Datei bzw. ein PLAM-Bibliothekselement und sendet sie über einen speziellen Port zu einem Windows-PC. Wahlweise werden die Daten binär oder im Textformat mit Code-Konvertierung gesandt. Mit dem Programm EDT für Windows können Daten auch wieder zum BS2000 zurück übertragen werden.

Der Modul kann auch vom CFS aus mit dem Kommando (FSEND), mit einem User-Actioncode oder mit dem Prozedur-Kommando *RUN aufgerufen werden. Auf dem PC muß das Programm FSERVER geladen sein, das die Daten empfängt, eine temporäre Datei erstellt und das dazugehörige Windows-Anzeigeprogramm aufruft.

Grundsätzlich sind drei Anwendungsfälle zu unterscheiden:

- a) Windows-Dateien, wie z.B. EXCEL-, WORD- oder PDF-Dateien werden binär zum BS2000 oder ins POSIX-Filesystem übertragen. Von dort werden sie zentral für die Benutzer zur Verfügung gestellt, z.B. in einer UTM-Anwendung. Von FSend werden solche Dateien ebenfalls binär zurückübertragen und können dann von dem Windows-Programm angezeigt werden.
- b) Im BS2000 oder POSIX werden Dateien im EBCDIC-Code erzeugt, z.B. EXCEL-Dateien (Spalten durch Strichpunkt getrennt), Textdateien oder HTML-Dateien. Diese Dateien werden von FSend im Textmodus übertragen, d.h. die Daten werden von EBCDIC nach ISO8859 konvertiert. Nach jedem Satzende wird das Satzendezeichen X'0D0A' erzeugt. Soweit noch eine zusätzliche Formatierung der Daten notwendig ist, kann dies z.B. durch Office-Makros erfolgen, die als Kommando-Parameter beim Aufruf mitgegeben werden können.
- c) Eine BS2000-Datei, eine POSIX-Datei oder ein PLAM-Bibliothekselement wird binär zum PC übertragen und im EDT für Windows editiert. Mit dem Kommando WRITE werden die Daten direkt zum BS2000 zurück übertragen und in der Ursprungsdatei oder wahlweise in einer neuen Datei gespeichert.

Proxy-Server FSendSRV für BS2000 und Windows/Linux

Dateien können auch über einen Proxy-Server empfangen und/oder zurückgesandt werden. Dadurch ist es möglich, Dateien zu einem PC zu senden, der im BS2000 nicht bekannt ist. Ebenfalls ist es möglich zum Empfangen im BS2000 statt dynamischer Ports nur einen zentralen Port zu verwenden. Siehe hierzu den Abschnitt FSendSRV (S. 112).

Anpassungen FSEND (Prozedur REP)

Anpassungen für den Modul FSend können über die DO-Prozedur REP in der PLAM-Bibliothek OPGCOM.LIB vorgenommen werden. In der Prozedur (PROC-Anweisung) sind die Standard-Einstellungen vorgegeben. Folgende Modifikationen sind möglich:

- a) Port-Nummer (&PORT=000024DE):

Das Programm verwendet standardmäßig den Port 9438 (Hexadezimal = 24DE). Soll ausnahmsweise ein anderer Port verwendet werden, ist die Port-Nummer hexadezimal mit führenden Nullen, immer mit 8 Stellen, einzugeben. In der Parameterdatei FServer.ini auf dem PC muß die gleiche Port-Nummer eingestellt werden.

b) Wiederholungsversuche (&RETRY=00000005):

Tritt beim Verbindungsaufbau zum PC ein Fehler auf, wird standardmäßig 5 mal versucht, die Verbindung herzustellen. Der Fall kann vor allem dann auftreten, wenn FSEND in einer Prozedur mehrmals aufgerufen wird und auf dem PC durch das Laden einer Office-Anwendung die CPU zu 100 % ausgelastet ist und deshalb die Anforderung nicht angenommen wird. Die Wiederholungsanzahl ist 8-stellig, hexadezimal mit führenden Nullen einzugeben.

c) Lizenznummer (&LICENSE=11111111) :

Als Standard ist der Wert für eine 4-wöchige Test-Lizenz vorgegeben. Fall Sie eine unbeschränkte Lizenz erwerben, müssen Sie hier die mitgeteilte Lizenznummer eintragen. Die Lizenznummer muß immer in der Länge von 8 Bytes eingetragen werden.

d) Standard-CCSN (&DEFCCSN=EDF03DRV) :

Als Standard ist EDF03DRV (EBCDIC 7-Bit mit deutschen Umlauten) vorgegeben. Falls eine Datei ohne CCSN mit dem Kommando SEND-EDTW-FILE bzw. dem Actioncode EDTW gesendet wird und der Parameter CODE nicht angegeben ist, so wird als CCSN dieser Wert verwendet.

Aufruf mit SDF-Kommando SEND-FILE

SEND-FILE FILENAME=*file*, [PC= *DEFAULT | *prozessor*] [, MODE=*mode*] [, EXT=*cmdpar*]
[, UPDATE=*YES | *NO | *newfile*] [, ENTER=*NO | *YES]

Das SDF-Kommando übergibt die Parameter dem ASTI-Service OPGFILESEND. Die Standardwerte können in der Prozedur SYSPR.SEND-FILE angepaßt werden. Beschreibung der Parameter siehe S. 39.

Für das Kommando ist auch noch der bisherige Namen FILESEND als Alias-Name definiert.

file Eingabedatei, Beschreibung siehe S. 39

prozessor Adresse PC, Beschreibung siehe S. 39

mode Übertragungsmodus TXT | TXTD | BIN | EDTW, Beschreibung siehe S. 40.

cmdpar Angaben zum Aufruf des Anzeige-Programms auf dem PC, Beschreibung siehe S. 40.

Falls der Parameter Anführungszeichen enthält, muß die ganze Zeichenfolge in Hochkommas eingeschlossen werden, z.B:

```
' .pdf /p /h %1 "HP DeskJet 970Cxi" "HP DeskJet 970Cxi" "IP_194.221.128.5" '
```

UPDATE=*YES | *NO | *newfile*

Auf Update warten, Beschreibung siehe S. 41

ENTER=*YES Zum Senden und Empfangen von Daten Enter-Job starten, Beschreibung siehe S. 41

Aufruf mit SDF-Kommando SEND-EDTW-FILE (Aliasname: SENDEDTW)

SENDEDTW FILENAME=*file*, [PC= *DEFAULT | *prozessor*] [, UPDATE=*YES | NO | *newfile*]
 [, CODE=*DEFAULT | DE | INT | *ccsn*] [, ENTER=*NO | *YES]

Das SDF-Kommando übergibt die Parameter dem ASTI-Service OPGFILESEND. Als Parameter für MODE wird immer EDTW(WAIT=..,CODE=*ccsn*) benutzt. Die Standardwerte können in der Prozedur SYSPR.SEND-FILE angepaßt werden.

Datenformat:

Den Daten wird ein Header vorangestellt, in dem alle notwendigen Informationen für die Verwaltung enthalten sind. Diese Informationen sind nicht sichtbar, können jedoch mit dem Kommando STA FSEND angezeigt werden. Die Daten werden ohne Code-Konvertierung in EBCDIC übertragen und auch als EBCDIC-Daten angezeigt. Der Anzeige-Code wird automatisch eingestellt. Damit alle Zeichen dargestellt werden können, z.B. auch X'0D' oder X'0A' werden die Daten in einem besonderen Format übertragen: Sätze variabler Länge mit Satzlängenfeld.

Dateiformat:

Im Gegensatz zum OPEN-FT und FTP können auch **ISAM-Dateien, PAM-Dateien, delta-gespeicherte LMS-Elemente, POSIX-Dateien und BS2000-ZIP-Elemente** editiert werden.

ISAM-Dateien mit den Standard-Isamkey (KEYPOS=5,LEYLENGTH=8) werden im EDTW mit der Option "KEY" eingelesen, so daß die ISAM-Schlüssel als Zeilennummer angezeigt werden. Aber selbst ISAM-Dateien mit Nicht-Standard-Schlüsseln können im EDTW bearbeitet werden. Beim Zurückschreiben wird die Datei im BS2000 auch richtig geschrieben, wenn die ISAM-Schlüssel nicht aufsteigend sind.

Die Daten werden immer mit der Option "OVERWRITE" geschrieben.

Prüfung auf Änderung außerhalb des EDTW:

Wie bei lokalen Dateien wird zyklisch und beim Wechsel in einen anderen Arbeitsbereich bzw. vor dem Schreiben geprüft, ob die BS2000-Datei inzwischen außerhalb des EDTW geändert worden ist und ggf. eine Warnung ausgegeben.

Die Datei wird binär übertragen und an das Windows-Programm EDTW übergeben. Im EDTW wird automatisch der entsprechende CODE eingestellt. Die Daten können im EBCDIC-Code editiert und mit dem Kommando WRITE wieder zurück übertragen werden.

file Eingabedatei (BS2000-Datei, PLAM-Element, POSIX-Datei oder ZIP-Element), Beschreibung siehe S. 39

prozessor Adresse PC, Beschreibung siehe S. 39

UPDATE=*YES | *NO | *newfile*

- *NO Die Eingabedatei kann nicht geändert. Nach dem Senden der Datei wird die Funktion beendet.
- *YES Die Eingabedatei kann mit EDT für Windows geändert werden. Nach dem Senden der Datei wartet das Kommando EDTW auf Nachrichten des EDT für Windows und überschreibt auf Anforderung die Eingabedatei. Das Kommando EDTW wird erst beendet, wenn das Datenfenster im EDT für Windows geschlossen wird.

newfile Die Eingabedatei kann mit EDT für Windows geändert und in eine neue Datei geschrieben werden. Als Datei kann eine BS2-Datei, ein PLAM-Bibliothekselement, eine POSIX-Datei oder ein ZIP-Element angegeben werden (Beschreibung siehe Parameter *file* (S. 39)) Nach dem Senden der Datei wartet das Kommando EDTW auf Nachrichten des EDT für Windows und schreibt auf Anforderung die Daten in die Datei *newfile*. Das Kommando EDTW wird erst beendet, wenn das Datenfenster im EDT für Windows geschlossen wird.

CODE=*DEFAULT | DE | INT | EDF041 | *ccsn*

Angaben für die Darstellung des EBCDIC-Codes im EDT für Windows. Bei dem EBCDIC-Code **EDF03IRV**, der zur Zeit als Standard gilt, gibt es zwei Varianten. Das Problem 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
Hexa-Wert	FB	4F	FD	BB	BC	BD	FF
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. Dieser Zeichensatz wird als Standard verwendet.

INT Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN= EDF03IRV internationale Variante), aus den entsprechenden EBCDIC-Zeichen sollen die Zeichen "{ | } [\] ~" erzeugt werden.

EDF041: 8-Bit-EBCDIC-Code. Die Datei enthält unter anderem als Sonderzeichen sowohl die Zeichen "{ | } [\] ~" als auch die Umlaute und das "scharfe S" (**ÄÖÜäöüß**). Die Umlaute sind allerdings anders codiert als beim Zeichensatz EDF03IRV, deutsche Variante.

ccsn Zeichensatzname (CCSN Coded Character Set Name), z.B. EDF042. In der Datei CODEPAGE.TXT für das Programm EDT für Windows muß in diesem Fall eine Code-Tabelle mit diesem Namen definiert werden. Zur Zeit werden nur die Code-Tabellen für EDF03IRV und EDF041 ausgeliefert.

*DEFAULT Es wird eine ev. vorhandene CCSN-Angabe aus dem Datei-Katalog verwendet. Falls im Datei-Katalog kein CCSN gespeichert ist, wird der Standardwert aus den globalen Einstellungen verwendet, siehe Prozedur REP (S. 34).

ENTER=*YES Zum Senden und Empfangen der Datei wird ein Enter-Job gestartet.

Aufruf von FSend als CFS-Unterprogramm in der Kommandozeile

(FSEND, [*\$user-id.*]OPGCOM.LIB) *prozessor*, *file* [,*mode*, *cmdpar*, *update*, *enter*]

Die Angabe der Bibliothek *opgcom.lib* ist nur beim ersten Aufruf einer CFS-Sitzung zum Nachladen notwendig.

prozessor Adresse PC, Beschreibung siehe S. 39

<i>file</i>	Eingabedatei, Beschreibung siehe S. 39
<i>mode</i>	Übertragungsmodus TXT BIN EDTW, Beschreibung siehe S. 40
<i>cmdpar</i>	Angaben zum Aufruf des Anzeige-Programms auf dem PC, Beschreibung siehe S. 40
<i>update</i>	Auf Update warten, Beschreibung siehe S. 41
<i>enter</i>	Zum Senden und Empfangen von Daten Enter-Job starten, Beschreibung siehe S. 41

Aufruf von FSend als CFS-Unterprogramm mit Kommando *RUN

RUN (FSEND, [\$user-id.*]OPGCOM.LIB) *prozessor*, *file* [, *mode*, *cmdpar*, *update*, *enter*]

Die Parameter müssen vor Aufruf des RUN-Kommandos in CFS-Variable gespeichert werden.

Beschreibung der Parameter siehe S. **39**.

Aufruf von FSENDUP als Unterprogramm von einem Benutzerprogramm

Der Modul FSENDUP wird entsprechend der Standard-Programmverknüpfungsregeln aufgerufen. Er kann auch von höheren Programmiersprachen aufgerufen werden. Alle Strings müssen mit X'00' abgeschlossen werden. Beim Aufruf des Moduls FSENDUP müssen die Register wie folgt geladen sein:

(R1) A (PARAMETERLISTE)

Die Parameterliste enthält folgende Adressen:

A (PROZ)	Name des PC's		
A (DATEI)	Name der zu sendenden Datei		
A (MODE)	wahlweise Übertragungs-Modus oder 0		
A (CMDPAR)	wahlweise Aufruf-Parameter für das Anzeige-Programm oder 0		
A (UPDATE)	wahlweise Option Update oder 0		
A (ENTER)	wahlweise Option Enter oder 0		
A (0)	unbenutzt für Erweiterungen		
PROZ	DC	C ' <i>prozessor</i> '	max. Länge 15
	DC	X'00'	
FILE	DC	C ' <i>file</i> '	max. Länge 128
	DC	X'00'	
MODE	DC	C ' <i>mode</i> '	max. Länge 128
	DC	X'00'	
CMDPAR	DC	C ' <i>cmdpar</i> '	max. Länge 128
	DC	X'00'	
UPDATE	DC	C ' <i>update</i> '	max. Länge 128
	DC	X'00'	
ENTER	DC	C '*YES'	max. Länge 4
	DC	X'00'	

Beschreibung der Parameter siehe S. **39**.

- (R13) A (SAVEAREA)
 Register-Sicherstellungsbereich (18 Worte, DC 18F'0'), der vom Aufrufer erstellt werden muß. FSENDUP sichert dort die Register.
- (R14) A (RETURN)
 Rücksprungadresse im rufenden Programm. Nach Ausführung der Funktionen wird das Programm an dieser Adresse fortgesetzt.
- (R15) V (FSENDUP)
 Beschreibung der Parameter siehe S. 39.

Parameter für FSENDUP bzw. FILESEND

Adresse des PC's

- prozessor* *proz-name* | *IP-addr* | *DEFAULT
 Adresse des PC's als BCAM-Name oder als IP-Adresse, auf dem die Datei angezeigt werden soll.
- proz-name* Prozessorname (max. Länge 8), wie in BCAM definiert.
- IP-addr* IP-Adresse (max. Länge 15) des PC's in der Form *nnn.nnn.nnn.nnn*. Führende Nullen können entfallen, z.B. 195.21.5.5.
- *DEFAULT Die IP-Adresse wird wie folgt, automatisch ermittelt:
 1. Spalten 1-15 der Jobvariablen mit dem Linknamen *OPGCOM.
 2. Spalten 1-15 der Jobvariablen OPGCOM.*jobname*
 3. Spalten 1-15 der Jobvariablen OPGCOM
 4. Ist keine Jobvariable vorhanden, wird der Prozessorname des Terminals verwendet.

Name der Datei bzw. des Bibliotheks-Elements

- file* *bs2-file* | *lib(typ,elem[, vers])* | */posix-file* | *POSIX:*file* | *ZIP:*zipfile(zip-elem)*
 Name der zu sendenden Datei bzw. des Bibliothekselements.
- bs2-file* Name der BS2000-Datei, ggf. mit Cat-ID und User-ID (max. Länge 54 Bytes).
- lib(typ,elem[, vers])* Name eines Bibliothekselements:
 lib Name der Bibliothek
 typ Element-Typ, z.B. S, J oder M
 elem Name des Bibliothekselements
 vers Version des Elements, Std = höchste Version.
- /posix-file* Vollqualifizierter Name einer POSIX-Datei (max. Länge 256).
- *POSIX:*file* Dateiname einer POSIX-Datei im aktuellen Home-Verzeichnis (max. Länge 256).

*ZIP:*zipfile(zip-elem)* Name eines ZIP-Archives und Name eines Elements. Ein ZIP-Element kann im BS2000 zur Zeit nur gelesen werden.

mode

Übertragungsmodus

BIN Die Datei wird ohne Code-Übersetzung und ohne Satzendezeichen in einem Datenstrom übertragen. Das bedeutet, die Datei muß im Original-Format (z.B. PDF, XLS, DOC usw.) vorliegen. Als BS2000-Datei muß die Datei entweder das Format PAM oder SAM, RECFORM=U haben.

Am Einfachsten können solche Dateien bereitgestellt werden, in dem sie mit FTP oder über NFS in das POSIX-Dateisystem übertragen werden.

Mit FTP kann eine PC-Datei direkt in das BS2000 im Binär-Modus (Kommando BIN) übertragen werden, wenn anstelle des Kommandos `PUT` das Kommando `SEND` verwendet wird. Im BS2000 wird dann immer eine PAM-Datei erzeugt. Das Kommando `PUT` erstellt nämlich in Abhängigkeit von dem Inhalt der Datei entweder eine SAM-Datei (Recform=V) oder eine PAM-Datei.

Mit OpenFT kann eine PC-Datei direkt in das BS2000 Im Modus "Binär undefined" übertragen werden. Im BS2000 entsteht dann eine Datei mit dem Format `FCBTYPE=SAM, RECFORM=U`.

TXT Textformat international. Die Daten werden von EBCDIC nach ANSI übersetzt. Die 7-Bit Sonderzeichen werden als `[]{}\|~` übersetzt. An jedem Satzende wird das Satzendezeichen `X'0D0A'` angehängt.

TXTD Textformat Deutsch. Die Daten werden von EBCDIC nach ANSI übersetzt. Die 7-Bit Sonderzeichen werden als `ÄÖÜäöüß` übersetzt. An jedem Satzende wird das Satzendezeichen `X'0D0A'` angehängt.

EDTW(CODE=*ccsn* | *DEFAULT)

Die Datei wird binär übertragen und an das Windows-Programm EDTW übergeben. Im EDTW wird automatisch der entsprechende CODE eingestellt. Die Daten können im EBCDIC-Code editiert und mit dem Kommando `WRITE` wieder zurück übertragen werden. Weitere Hinweise siehe SDF-Kommando EDTW (S. 36)

CODE=*ccsn* Zeichensatzname (CCSN **C**oded **C**haracter **S**et **N**ame).

CODE=*DEFAULT Der Zeichensatzname aus dem Catalog-Eintrag wird verwendet. Ist im Catalog kein Eintrag vorhanden, wird der Standard CCSN für die User-ID verwendet.

Die Angabe dieses Parameters ist optional. Fehlt dieser Parameter, gilt als Standard `TXT`.

Angaben zum Aufruf des Programms auf dem PC

cmdpar

extension [par] | `.EXE application [par]`

Angaben zum Aufruf des Programms auf dem PC (maximale Länge: 128). Soweit aus der Extension des Dateinamens die Applikation ermittelt werden kann und keine weiteren Parameter für den Aufruf notwendig sind, ist dieser Parameter nicht erforderlich.

Fehlt dieser Parameter, gilt als Standard die Dateierweiterung im Dateinamen. Damit wird die Applikation gestartet, die in der Registry dieser Dateierweiterung zugewiesen ist.

- extension* Dateinamen-Erweiterung (z.B. `.pdf`, `.xls` usw.) einschl. des Punktes in Klein- oder Großschreibung. Diese Angabe ist normalerweise ausreichend, um den Namen der Applikation aus der Registry zu ermitteln. Falls ausnahmsweise im Windows eine Verknüpfung mit der Applikation nicht vorhanden ist bzw. eine davon abweichende Applikation geladen werden soll, muß der Parameter in der Form `.EXE application [par]` angegeben werden.
- application* Voller Pfadname des Programms, mit dem die Datei angezeigt werden soll.
- par* Parameter für das Anzeige-Programm. Als Platzhalter für den Dateinamen ist `%1` anzugeben.

Beispiele:

```
.doc
.doc %1 /mMakroname
.pdf /p /h %1 "HP DeskJet 970Cxi" "HP DeskJet 970Cxi" "IP_194.221.128.5"
.exe c:\programme\winword\word.exe %1 /mMakroname
```

Falls der Parameter Anführungszeichen enthält und im Kommando FILESEND verwendet wird, muß die ganze Zeichenfolge in Hochkommas eingeschlossen werden, z.B:

```
ext='.pdf /p /h %1 "HP DeskJet 970Cxi" "HP DeskJet 970Cxi"
"IP_194.221.128.5"'
```

Option UPDATE

- update* `*YES|*NO| newfile`
- *NO** Die Eingabedatei kann nicht geändert werden. Nach dem Sender der Datei wird die Funktion beendet.
- *YES** Die Eingabedatei kann mit EDT für Windows geändert werden. Nach dem Senden der Datei wartet das Kommando EDTW auf Nachrichten des EDT für Windows und überschreibt auf Anforderung die Eingabedatei. Das Kommando EDTW wird erst beendet, wenn das Datenfenster im EDT für Windows geschlossen wird.
- newfile* Die Eingabedatei kann mit EDT für Windows geändert und in eine neue Datei geschrieben werden. Als Datei kann eine BS2-Datei, ein PLAM-Bibliothekselement, eine POSIX-Datei oder ein ZIP-Element angegeben werden (Beschreibung siehe Parameter *file* (S. 39)) Nach dem Senden der Datei wartet das Kommando EDTW auf Nachrichten des EDT für Windows und schreibt auf Anforderung die Daten in die Datei *newfile*. Das Kommando EDTW wird erst beendet, wenn das Datenfenster im EDT für Windows geschlossen wird.

Option ENTER

- enter* `*YES|*NO`
- *NO** Das Senden und ggf. Empfangen der Daten wird im aktuellen Task ausgeführt.
- *YES** Zum Senden und ggf. Empfangen der Daten wird ein Enter-Job gestartet.

Beispiele:**a) Aufruf mit SDF-Kommando FILESEND:**

```
filesend testpc,/test/test.pdf,bin
filesend testpc,/test/manual,bin,.pdf
filesend testpc,manual,bin,.exe c:\acroread\acreread.exe %1
```

b) Aufruf mit SDF-Kommando EDTW:

```
edtw testpc,testprog.src
edtw testlib(s,testprog),code=edf041
edtw testlib(s,testprog2),code=edf041,update=*no
edtw testprog2,code=edf041,update=testprog2.t,enter=*yes
```

c) Aufruf aus CFS-Kommandozeile:

```
(fsend,opgcom.lib)testpc,/test/test.pdf,bin
(fsend,opgcom.lib)testpc,/test/manual,bin,.pdf
(fsend)testpc,manual,bin,.exe c:\acroread\acreread.exe %1
```

d) CFS-Kommando *RUN

```
*&prozessor='testpc'
*&file='/test/test.pdf'
*&mode='bin'
*&cmdpar='.pdf'
*run(fsend,opgcom.lib)&prozessor,&file,&bin,&cmdpar
```

e) Aufruf aus Benutzerprogramm:

```
        LA    R1, PARAM
        L     R14, FILESAD
        BALR  R14, R15

FILESAD DS    V(FSENDUP)
SAVAREA DS    18F
PARAM   DC    A(PROZ)
        DC    A(FILE)
        DC    A(MODE)
        DC    A(CDMPAR)
        DC    A(UPDATE)
        DC    A(ENTER)
        DC    A(0)          UNUSED
*
PROZ    DC    C'TESTPC'
        DC    X'00'
FILE    DC    C'/usr/test/testdatei'
        DC    X'00'
MODE    DC    C'BIN'
        DC    X'00'
CDMPAR  DC    C'.PDF'
        DC    X'00'
UPDATE  DC    C'*NO'
        DC    X'00'
ENTER   DC    C'*NO'
        DC    X'00'
```

Ein vollständiges Beispiel-Programm in Assembler finden Sie in der LMS-Bibliothek OPGCOM.LIB, Element S/DEMOCALL.

7 URL-Server (Web-Server) direkt im BS2000

Mit diesem Web-Server können Sie Informationen direkt vom BS2000/OSD für das Internet bereitstellen. Sie benötigen kein CPU-intensives POSIX. Sie können Originaldaten in EBCDIC bereitstellen (Der POSIX-Web-Server benötigt ASCII-Dateien).

Gegenüber Unix-WEB-Servern ergeben sich folgende Vorteile:

- a) Der URL-Server ist nur für Benutzer erreichbar, für die ein BCIN-Eintrag vorhanden ist. Es können also nur bekannte Benutzer auf die Daten zugreifen.
- b) Sie haben direkten Zugriff auf EBCDIC- und ASCII-Dateien, die sich entweder im BS2000 oder im POSIX-Filesystem befinden können.
- c) Mit Hilfe komfortabler UP-Schnittstellen (Assemb, C oder COBOL) können Sie Daten aus Ihren Anwendungen und Datenbanken bereitstellen.

Mehrere Server-Tasks

Der Server-Task kann mehrmals gestartet werden. Einzige Voraussetzung ist, daß für jeden Server ein eigener Port und eigene LOG-Dateien verwendet werden. Die Portnummer kann in der ausgelieferten Musterprozedur über den Parameter PORT geändert werden.

Mehrere Sub-Tasks

Jeder Server-Task kann wiederum bis zu 15 Sub-Tasks starten, in denen die Bereitstellung der Daten durchgeführt wird. Dadurch wird sichergestellt, daß der Haupt-Task immer bereit ist, Anforderungen entgegen zu nehmen und durch aufwendige Datenbereitstellungen der Betrieb nicht gestört wird. Die Anzahl der Sub-Tasks kann in der Anweisung SUBTASKS (S. 68) angegeben werden.

Demo-Seiten

In der PLAM-Bibliothek OPGCOM.LIB sind Demo-Dateien enthalten, mit denen die wichtigsten Funktionen des URLServers getestet werden können. Nach den entsprechenden Anpassungen und Start des Jobs J/URLSERVER (S. 17) kann die Demo durch Eingabe von `http://host[:port]/` gestartet werden.

Steuerung des Servers über INTR

INTR *tsn*, STOP | ADDTASKS *n* | REMTASKS *n*

Zur Steuerung des Servers können folgende INTR-Kommandos an der Console eingegeben werden.

STOP Beenden des URL-Servers.

ADDTASKS *n* Add Tasks. *n* Sub-Tasks hinzufügen. Falls die Gesamtzahl der Tasks zusammen mit den bestehenden Tasks 15 überschreitet, wird die Anzahl automatisch beschränkt.

REMTASKS *n* Remove Tasks. *n* Sub-Tasks beenden. Falls die Gesamtanzahl der geladenen Subtasks überschritten wird, werden alle SUB-Tasks beendet und der URL-Server im Single-Mode weiterbetrieben.

Authentifizierungsverfahren

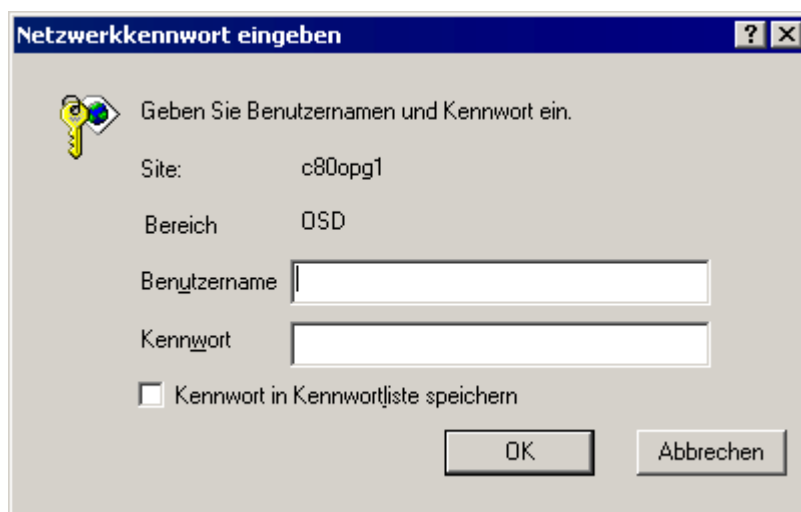
Für die Administration der geschützten Bereiche werden Benutzergruppen eingerichtet. Pro Benutzergruppe können dir Zugriffsrechte (HTMDIR), die Homepage (HTMINDEX), das Default-Directory (DEFDIRBS2 / DEFDIRPOS) und das Attribut SHARE (VIEW) zugewiesen werden. Der Zugriff ist nur nach Eingabe des zutreffenden User-Namens und des dazugehörigen Paßwortes möglich. Bei jedem Zugriff auf eine geschützte Datei wird anhand der festgelegten Rechte geprüft, ob der Zugriff für diesen User zulässig ist.

Soweit Daten durch ein Authentifizierungsverfahren gesichert sind, muß bei einer Datenanforderung durch den Browser gleich nach dem HOST-Namen der Name eine Benutzergruppe (S. 71) angegeben werden.

Die Rechte dieser Gruppe sind in der Datei beschrieben, die mit dem Parameter AUTHFILE (S. 45) zugewiesen ist. Beim ersten Zugriff auf eine geschützte Datei wird der Username und das Paßwort angefordert. Das Paßwort gilt bis zur Beendigung des Browsers und ist für alle Zugriffe dieser Benutzergruppe gültig.

Beispiel: c80opg1/grp1/view/test.html

Der Paßwort-Dialog ist vom verwendeten Browser abhängig und sieht z.B. so aus:



SYSDTA-Anweisungen

AUTHFILE: Rechte für das Authentifizierungsverfahren

AUTHFILE=filename | *NO

filename

Name der Datei, in der die Daten für das Authentifizierungsverfahren und die Rechte der Benutzer enthalten sind. In dieser Datei werden sowohl die Benutzergruppen mit den dazugehörigen Rechten als auch die User-Daten definiert. Bei jedem Zugriff auf eine geschützte Datei wird anhand der festgelegten Rechte geprüft, ob der Zugriff für diesen User zulässig ist.

***NO** Keine AUTHFILE benutzen.

Inhalt AUTHFILE:

<GROUP *group*>

Abschnitt für die Berechtigungen einer Usergruppe. Dieser Gruppe können mehrere User mit dem entsprechenden Passwort zugewiesen werden (siehe Abschnitt **<USER>**).

group Name der Benutzergruppe.

Die Beschreibung der Rechte erfolgt in der gleichen Weise, wie für die allgemeinen Berechtigungen in der Parameterdatei. Folgende Rechte können definiert werden:

DEFDIRBS2= <i>userid</i>	Default-User-ID BS2000 (S. 51).
DEFDIRPOS= <i>directory</i>	Default-User-ID POSIX (S. 51).
CGIACCESS= <i>ausw</i> [+ <i>ausw</i>]	Zugriffsrechte für Benutzermodule (S. 47)
CFSACCESS= <i>ausw</i> [+ <i>ausw</i>]	Zugriffsrechte für CFS-Prozeduren (S. 48).
CODE=DE INT	Sonderzeichen Deutsch oder International (S. 49).
ENTERACCESS= <i>ausw</i> [+ <i>ausw</i>]	Zugriffsrechte für Enter-Prozeduren (S. 51).
HTMDIR= [<i>:cat-id:</i>] <i>\$user-id</i> [<i>.file</i>]	Zugriffsrechte (S. 57).
FILEACCESS= [<i>:cat-id:</i>] [<i>\$user-id.</i>] [<i>ausw</i>] <i>posix-dir</i> /[<i>ausw</i>]	Zugriffsrechte (S. 53).
HTMINDEX= <i>indexfile</i>]	Indexfile - Homepage (S. 58).
TEMPLATE= <i>prefix</i>	Prefix für Template-Dateien (S. 68).
VIEW=ALL SHARE	Zugriff auf NONSHARE-Dateien (S. 69).

AUTH=NO Für diese Gruppe wird keine Authentifizierung durchgeführt, d.h. es wird kein Passwort angefordert. Diese Einstellung ist für Anwendungen gedacht, die zwar keinen Passwortschutz benötigen aber für die verschiedenen Gruppen unterschiedliche Index-Seiten, Template-Dateien, Code-Einstellungen oder Default-User-ID's benötigen.

<USER *user*>

Abschnitt für die Beschreibung der User und die Zuordnung zu einer Benutzergruppe.

user Name des Benutzers.

PASSWORD=*password*

password Paßwort des Benutzers.

GROUP=*group* Name der Benutzergruppe. Diese Anweisung kann beliebig oft angegeben werden. Der Benutzer kann somit mehreren Gruppen angehören.

<END> Ende der Beschreibungen.

Beispiel (siehe auch Datei D/HTACCESS in der OPGCOM.LIB):

```
<GROUP grp1>
VIEW=SHARE
FILEACCESS=$GRP1.*
HTMINDEX=GRP1IND.HTM
DEFDIRBS2=$GRP1
TEMPLATE=TEMPLATE1.
<GROUP grp2>
VIEW=SHARE
FILEACCESS=$GRP2.*
HTMINDEX=GRP2IND.HTM
DEFDIRBS2=$GRP2
TEMPLATE=TEMPLATE2.
<GROUP grp3>
AUTH=NO
DEFDIRBS2=$GRP3
<USER user1>
PASSWORD=pswuser1
GROUP=prg1
<USER user2>
PASSWORD=pswuser2
GROUP=grp2
<USER admin>
PASSWORD=pswadmin
GROUP=grp1
GROUP=grp2
<END>
```

Zugriff auf Module erlauben

CGIACCESS= *ausw* [**+***ausw...*]

Mit dieser Anweisung wird festgelegt, welche Benutzer-Module nachgeladen werden dürfen. Die Anweisung kann mehrmals vorkommen. Die Zugriffsrechte können global in der SYSDTA-Datei und pro Gruppe in der AUTHFILE (S. 45) festgelegt werden.

Der Zugriff auf Dateien, CFS-Prozeduren und BS2000-Prozeduren wird mit den Anweisungen FILEACCESS (S. 53), CFSACCESS (S. 48) und ENTERACCESS (S. 51) gesteuert.

ausw *module* | *prefix** | **suffix* | **string** | *

Ein Auswahlstring kann max. 54 Byte lang sein.

Es können beliebig viele Auswahlstrings miteinander verknüpft werden. Der Zugriff ist nur dann erlaubt, wenn alle mit "+" verknüpften Bedingungen erfüllt sind.

Fehlt die Anweisung, ist aus Kompatibilitätsgründen der Zugriff auf alle Module in der CGILIB erlaubt (wie Auswahl "*").

module Vollständiger Modulnamen.

*prefix** Modulauswahl durch Angabe des Prefix, z.B. a* (alle Module, die mit "a" beginnen).

**suffix* Modulauswahl durch Angabe des Suffix, z.B. *1st (alle Module, die mit "1st" enden).

string Modulauswahl durch Angabe einer Zeichenfolge, die an einer beliebigen Stelle des Modulnamens vorkommt, z.B. **test**. (alle Module, die an einer beliebigen Stelle, auch am Anfang oder am Ende, "test" enthalten).

***** Alle Module der CGILIB.

Beispiele:

```
CGIACCESS MODUL1 Zugriff auf den Modul MODUL1.  
CGIACCESS LST* Zugriff auf alle Module, die mit "LST" beginnen.
```

Bibliothek für Benutzermodule, CFS-Prozeduren und BS2000-Prozeduren

CGILIB=library PLAM-Bibliothek, in der die Benutzermodule und BS2000-Prozeduren (Typ J) für die Funktion `ENTER` gespeichert sind. In dieser Bibliothek müssen ebenfalls die CFS-Prozeduren (Typ S) für die Funktion `CFS` enthalten sein. In der Muster-Prozedur enthält dieser Parameter die Zuweisung `OPGCOM.LIB`, damit die Beispiel-Module nachgeladen werden können. Sobald Sie eigene Module schreiben bzw. Musterprozeduren abändern, sollten Sie eine andere Bibliothek angeben, damit bei einem Update die `OPGCOM.LIB` überschrieben werden kann.

Der Zugriff auf Module, CFS-Prozeduren und BS2000-Prozeduren wird mit den Anweisungen `CGIACCESS` (S. 47), `CFSACCESS` (S. 48) und `ENTERACCESS` (S. 51) gesteuert.

Zugriff auf CFS-Prozeduren erlauben

CFSACCESS= *ausw* [*+ausw...*]

Mit dieser Anweisung wird festgelegt, welche CFS-Prozeduren gestartet werden dürfen. Die Anweisung kann mehrmals vorkommen. Die Zugriffsrechte können global in der `SYSDTA`-Datei und pro Gruppe in der `AUTHFILE` (S. 45) festgelegt werden.

Der Zugriff auf Dateien, Benutzermodule und BS2000-Prozeduren wird mit den Anweisungen `FILEACCESS` (S. 53), `CGIACCESS` (S. 47), und `ENTERACCESS` (S. 51) gesteuert.

ausw *cfsproc* | *prefix** | **suffix* | **string** | ***

Ein Auswahlstring kann max. 54 Byte lang sein.

Es können beliebig viele Auswahlstrings miteinander verknüpft werden. Der Zugriff ist nur dann erlaubt, wenn alle mit "+" verknüpften Bedingungen erfüllt sind.

Fehlt die Anweisung, ist aus Kompatibilitätsgründen der Zugriff auf alle CFS-Prozeduren in der CGILIB erlaubt (wie Auswahl "").

cfsproc Vollständiger Prozedurnamen.

*prefix** Prozedurauswahl durch Angabe des Prefix, z.B. `grp1.*` (alle Prozeduren, die mit "grp1." beginnen).

**suffix* Prozedurauswahl durch Angabe des Suffix, z.B. `*1st` (alle Prozeduren, die mit "1st" enden).

string Prozedurauswahl durch Angabe einer Zeichenfolge, die an einer beliebigen Stelle des Prozedurnamens vorkommt, z.B. **test**. (alle Module, die an einer beliebigen Stelle, auch am Anfang oder am Ende, "test" enthalten).

* Alle CFS-Prozeduren der CGILIB.

Beispiele:

CFSACCESS PROC1	Zugriff auf die Prozedur PROC1.
CFSACCESS SHOW**G1	Zugriff auf alle Prozeduren, die mit "SHOW" beginnen und mit "G1" enden..

Bibliothek zum Nachladen von CFS-Modulen

CFSLIB=library User-ID und Name der CFSLIB mit allen CFS-Modulen. Die Bibliothek wird zum Nachladen der CFS-Module für die Funktion CFS benötigt.

Code-Variante

CODE=DE | INT | DK | *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 gleichzeitig dargestellt werden können. Dabei geht es um folgende Zeichen:

Deutsche Umlaute:	ä ö ü Ä Ö Ü ß
Dänische Umlaute:	æ ø å Æ Ø Å ù Û
Hexa-Wert EBCDIC	F'B 4F F'D BB BC BD F'F 6A
Internationale Zeichen:	{ } [\] ~ ^

Die Einstellung gilt für die Funktionen VIEW und XVIEW. Bei der Funktion PDF (S. 84) gilt die Anweisung PDF-CODE (S. 61), da in diesem Fall die Umlaute bereits bei der Erzeugung der PDF-Datei nach der PDF-Syntax kodiert werden müssen. Bei der Anzeige dieser PDF-Dateien mit der Funktion XVIEW wird automatisch die Einstellung "International" verwendet, da sonst die Zeichen "[", "]" und "\" in den "PDF-Anweisungen" zerstört würden.

Bei den Funktionen VIEW (S. 89) und XVIEW kann mit der Option "TRANS" eine abweichende Code-Variante gewählt werden.

Die Anweisung kann auch in der AUTHFILE (S. 45) pro Gruppe angegeben werden.

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

DK Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN= EDF03IRV dänische Variante) , aus den entsprechenden EBCDIC-Zeichen sollen die Zeichen "æ ø å Æ Ø Å ü Ü" erzeugt werden.

usercode Name eines User-Codes. Die von der Standard-Codierung abweichenden Translate-Anweisungen sind in einer Datei definiert, die mit der Anweisung `CODEFILE` (S. 50) zugewiesen wird.

Datei mit User-Codes

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

Als Standard werden EBCDIC-Zeichen mit der Translate-Tabelle EDF041 to ISO8859-1 (S. 116) in "ASCII-Zeichen" umgewandelt. Intern sind drei verschiedene Varianten vordefiniert:

Codename	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	Ü

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. 116) 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 Translat-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, z.B. `VIEW file?TRANS=codename`, anzugeben.

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

ee EBCDIC-Zeichen hexadezimal

aa ASCII-Zeichen hexadezimal

remark Bemerkung

Die User-Codes können in folgende Anweisungen bzw. Funktionen benutzt werden:

- Anweisungen `CODE` (S. 49) und `PDF-CODE` (S. 61)
- Anweisung `PDF-LAYOUT` (S. 63) Parameter `CODE`
- Funktionen `VIEW` / `XVIEW` (S. 69) Parameter `TRANS`
- Funktion `PDF` (S. 84) Parameter `CODE`

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 = |
```

Default Directory BS2000/OSD
DEFDIRBS2=userid

User-ID, die als Standard benutzt wird, falls bei einem FSTAT, VIEW- oder XVIEW-Aufruf keine User-ID angegeben wird. Ein Dateiname ohne User-ID (*file* oder *:cat-id:file*) wird um die User-ID ergänzt, falls es sich nicht um eine POSIX-Datei handelt.

Default Directory POSIX
DEFDIRPOS=directory

Posix-Verzeichnis, das als Standard benutzt wird, falls bei einem VIEW- oder XVIEW-Aufruf für eine POSIX-Datei ein relativer Dateiname benutzt wird. Ein relativer POSIX-Dateiname beginnt mit dem String `"*posix:"`. Ein absoluter POSIX-Dateiname beginnt mit dem Zeichen `"/`.

Zugriff auf Enter-Prozeduren erlauben
ENTERACCESS= ausw [+ausw...]

Mit dieser Anweisung wird festgelegt, welche BS2000-Prozeduren mit der Funktion `ENTER` gestartet werden dürfen. Die Anweisung kann mehrmals vorkommen. Die Zugriffsrechte können global in der SYSDTA-Datei und pro Gruppe in der AUTHFILE (S. 45) festgelegt werden.

Der Zugriff auf Dateien, Benutzermodule und CFS-Prozeduren wird mit den Anweisungen `FILEACCESS` (S. 53), `CGIACCESS` (S. 47) und `CFSACCESS` (S. 48) gesteuert.

ausw *bs2proc* | *prefix** | **suffix* | **string** | *

Ein Auswahlstring kann max. 54 Byte lang sein.

Es können beliebig viele Auswahlstrings miteinander verknüpft werden. Der Zugriff ist nur dann erlaubt, wenn alle mit "+" verknüpften Bedingungen erfüllt sind.

Fehlt die Anweisung, ist aus Kompatibilitätsgründen der Zugriff auf alle BS2000-Prozeduren in der CGILIB erlaubt (wie Auswahl "").

<i>Bs2proc</i>	Vollständiger Prozedurnamen.
<i>prefix*</i>	Prozedurauswahl durch Angabe des Prefix, z.B. <code>grp1.*</code> (alle Prozeduren, die mit "grp1." beginnen).
<i>*suffix</i>	Prozedurauswahl durch Angabe des Suffix, z.B. <code>*1st</code> (alle Prozeduren, die mit "1st" enden).
<i>*string*</i>	Prozedurauswahl durch Angabe einer Zeichenfolge, die an einer beliebigen Stelle des Prozedurnamens vorkommt, z.B. <code>*test*</code> . (alle Module, die an einer beliebigen Stelle, auch am Anfang oder am Ende, "test" enthalten).
*	Alle BS2000-Prozeduren der CGILIB.

Beispiele:

<code>ENTERACCESS PROC1</code>	Zugriff auf die Prozedur PROC1.
<code>ENTERACCESS SHOW**G1</code>	Zugriff auf alle Prozeduren, die mit "SHOW" beginnen und mit "G1" enden..

Maximale Zeit zum Starten von Enter-Prozeduren

ENTER-TIMEOUT=seconds

seconds Anzahl von Sekunden, die nach dem Starten von Enter-Prozeduren maximal gewartet wird, bis der Job anläuft. Insbesondere, falls der Job wegen Erreichen des Job-Class-Limits nicht gestartet werden kann oder der zuständige Jobstream nicht läuft, kann es sein, daß sich Jobs auf unbestimmte Zeit im Wartezustand befinden. Deshalb wird nach Erreichen der max. Wartezeit der bereits gestartete Enter-Job gecancelt, falls der Job bis dahin noch nicht angelaufen ist, damit der Servertask nicht zu lange belegt ist.

Die Timeout-Angabe bezieht sich nicht auf die Ausführungszeit des Jobs. Die maximale Laufzeit des Jobs kann mit den Optionen für die Jobklasse gesteuert werden.

Fehlt die Anweisung, gilt als Standard 10 Sekunden.

FGG (File Generation group) berücksichtigen

FGG=YES | NO

YES Es werden Informationen zu den einzelnen Dateigenerationen einer Dateigenerationsgruppe ausgegeben (Option GENERATIONS=*YES zum BS2000-Kommando SHOW-FILE-ATTRIBUTES).
Standard = YES.

NO Es werden keine Informationen zu den einzelnen Dateigenerationen einer Dateigenerationsgruppe ausgegeben.

Datenzugriff erlauben

FILEACCESS= [:*cat-id*:] [\$*user-id*.] [*ausw* [+*ausw...*]] | *posix-dir*/[*ausw* [+*ausw...*]] [,UPDATE=YES]

Mit dieser Anweisung wird festgelegt, auf welche Dateien ein Datenzugriff erlaubt ist.

Die Anweisung kann mehrmals vorkommen. Die Zugriffsrechte können global in der SYSDTA-Datei und pro Gruppe in der AUTHFILE (S. 45) festgelegt werden.

Es kann entweder der Zugriff auf eine User-ID, oder ein POSIX-Verzeichnis mit zusätzlichen Auswahlkriterien bezüglich des Dateinamens zugelassen werden.

Der Zugriff auf Module, CFS-Prozeduren und BS2000-Prozeduren wird mit den Anweisungen CGIACCESS (S. 47), CFSACCESS (S. 48) und ENTERACCESS (S. 51) gesteuert.

<i>cat-id</i>	CAT-ID.
<i>user-id</i>	Benutzerkennung der zulässigen Dateinamen. Die User-ID muß immer vollständig angegeben und mit "." abgeschlossen werden. Fehlt die User-ID, gilt die nachfolgende Auswahl für die User-ID der Anweisung DEFDIRBS2 (S. 51).
<i>posix-dir</i>	Beginnt das POSIX-Verzeichnis nicht mit dem Zeichen "/" (absoluter Pfad), handelt es sich um einen relativen Pfad und wird deshalb um das Verzeichnis der Anweisung DEFDIRPOS (S. 51) ergänzt. Das Zeichen "/" am Ende des <i>posix-dir</i> muß angegeben werden. Die Zugriffsberechtigung gilt auch für alle Unterverzeichnisse. Eine ev. nachfolgende Auswahlbedingungen bezieht sich aber immer nur auf den Dateinamen.
<i>ausw</i>	<i>file</i> <i>prefix</i> * <i>suffix</i> <i>string</i> * [*]lib(<i>prefix_elem</i> * [*]lib(<i>typ, elem, vers</i>) *
	Ein Auswahlstring kann max. 256 Byte lang sein. Der Auswahlstring wird nur im Dateinamen gesucht bzw. bei POSIX-Dateien wird ggf. im Unterverzeichnis einschl. Dateinamen gesucht.
	Es können beliebig viele Auswahlstrings miteinander verknüpft werden. Der Zugriff ist nur dann erlaubt, wenn alle mit "+" verknüpften Bedingungen erfüllt sind.
	Der Auswahlstring wird bei POSIX-Dateien nur im Dateinamen und nicht in ev. vorhandenen Unterverzeichnissen des <i>posix-dir</i> gesucht.
	Fehlt der Auswahlstring, ist der Zugriff auf alle Dateien dieser User-ID bzw. dieses POSIX-Verzeichnisses einschl. aller Unterverzeichnisse erlaubt (wie Auswahl "*").
<i>file</i>	Vollständiger Dateiname.
<i>prefix</i> *	Dateiauswahl durch Angabe des Prefix, z.B. <i>src.*</i> (alle Dateien, die mit "src." beginnen).
<i>suffix</i>	Dateiauswahl durch Angabe des Suffix, z.B. <i>*1st</i> (alle Dateien, die mit "1st" enden).
<i>string</i> *	Dateiauswahl durch Angabe einer Zeichenfolge, die an einer beliebigen Stelle des Dateinamens vorkommt, z.B. <i>*test*</i> . (alle Dateien, die an einer beliebigen Stelle, auch am Anfang oder am Ende, "test" enthalten).

Auswahl von Elementen einer LMS-Bibliothek:

*prefix_elem** Soll bei LMS-Bibliotheken der Zugriff nur auf einen Teil der Elemente zulässig sein, kann dies durch Angabe eines Präfixes für die Elementangabe erreicht werden. Eine Element-Angabe kann aus Typ, Elementnamen und Version bestehen. Es ist darauf zu achten, daß Type, Element und Version durch Komma und nicht durch Schrägstrich getrennt werden. Die schließende Klammer darf nicht angegeben werden, z.B.

LIB(S, *	alle Elemente der Bibliothek LIB mit Typ S
LIB(S, A*	alle Elemente "A*" der Bibliothek LIB mit Typ S
LIB(S, TEST, *	nur Element TEST vom Typ S der Bibliothek LIB
LIB(S, TEST, 1*	nur Element TEST vom Typ S und Version "1*" der Bibliothek LIB
*(S, TEST, *	nur Element TEST vom Typ S aus allen Bibliotheken

lib(typ, elem, vers) Der Zugriff ist nur auf das Element *elem* vom Typ *typ* mit der Version *vers* zulässig.

* Alle Dateien und Bibliotheken dieser User-ID bzw. dieses POSIX-Verzeichnisses.

UPDATE=YES Für diese Dateien ist ein UPDATE mit EDTW erlaubt ist (siehe auch Funktion EDTWUPD). Die Update-Berechtigung schließt die Berechtigung zum Lesen mit ein.

Hinweis:

Das POSIX-Verzeichnis, das in der Anweisung `DEFDIRPOS` (S. 51) bzw. die User-ID, die in der Anweisung `DEFDIRBS2` (S. 51) enthalten ist, wird nicht automatisch freigegeben. Auch für dieses Verzeichnis bzw. diese User-ID ist eine FILEACCESS-Anweisung notwendig. So ist es auch möglich, nur einen Teil der Dateien einer User-ID bzw. eines Verzeichnisses freizugeben.

Beispiele:

`FILEACCESS $TEST.SRC**TST`
Alle Dateien der User-ID \$TEST, die mit "SRC" beginnen und mit ,
"TST" enden.

`FILEACCESS $TEST.HTMLLIB`
Die Datei HTMLIB in der User-ID \$TEST. Soweit es sich um eine
Bibliothek handelt, ist der Zugriff auf alle Elemente der Bibliothek
erlaubt.

`FILEACCESS *(X,*`
Elemente vom Typ X in allen Bibliotheken

`FILEACCESS *TESTLIB(S,*`
Elemente vom Typ S in allen Bibliotheken, deren Namen mit
TESTLIB enden.

`FILEACCESS $TEST.HTMLLIB(S,*,UPDATE=YES`
Elemente vom Typ S in der Bibliothek \$TEST.HTMLIB dürfen ge-
lesen und mit der Funktion EDTWUPD geändert werden.

`FILEACCESS $TEST.HTMLLIB(X,TEST*,UPDATE=YES`
Elemente vom Typ X, deren Name mit "TEST" beginnen in der
Bibliothek \$TEST.HTMLIB dürfen gelesen und mit der Funktion
EDTWUPD geändert werden.

`FILEACCESS $TEST.SRC.*`
Alle Dateien der User-ID \$TEST, die mit SRC. beginnen

`FILEACCESS $TEST.*STR1*`
Alle Dateien der User-ID \$TEST, die STR1 im Dateinamen ent-
halten

`FILEACCESS /home/test/*`
Alle Dateien die unter /home/test und allen und Unterverzeich-
nissen gespeichert sind.

`FILEACCESS /home/test/*.c`
Alle Dateien im Verzeichnis /home/test und allen Unterverzeich-
nissen, die mit ".c" enden.

`FILEACCESS /home/test/src*`
Alle Dateien im Verzeichnis /home/test und allen Unterverzeich-
nissen, die mit "src" beginnen.

Layout-Optionen für FSTAT, LS und LIBTOC

FSTAT *_par=opt*
LIBTOC *_par=opt*
LS *_par=opt*

Mit diesen Anweisungen können die Spalten für die FSTAT/LIBTOC-Liste und die Schriftart ausgewählt werden. Diese Angaben gelten global und können durch die Angaben in der PAR-Anweisung der TEMPLATE-Datei (S. 92) überschrieben werden.

Parameter für FSTAT

Parameter für die Spaltenauswahl in der FSTAT-Liste:

FSTAT_BKL=Y N	Backup-Class A, B, C, D oder E
FSTAT_FREE_PAGES=Y N	Freie PAM-Seiten
FSTAT_FCB-TYPE=Y N	FCB-Type SAM PAM ISAM NONE
FSTAT_PAM_PAGES=Y N	Anzahl freie PAM-Seiten
FSTAT_SEC.ALLOC=Y N	Anzahl Second Allocation
FSTAT_SHARE=Y N	Share =YES NO
FSTAT_ACCESS=Y N	Access=WRITE READ
FSTAT_PASS=Y N	Password = RE WR EX
FSTAT_EXT.=Y N	Anzahl Extends
FSTAT_VOLUME=Y N	Name Volume
FSTAT_CHANGE_DATE=Y N	Datum der letzten Änderung
FSTAT_CHANGE_TIME=Y N	Uhrzeit der letzten Änderung
FSTAT_SORT=NAME AGE SIZE	Standard-Sortierreihenfolge
NAME =	Sortierung nach Dateinamen Diese Sortier-Option gilt als Standard
AGE =	Sortierung nach Datum und Uhrzeit absteigend
SIZE =	Sortierung nach Dateigröße absteigend

Die Sortierreihenfolge kann auch direkt bei der Funktion `FSTAT` (S. 81) mit dem Parameter `SORT` angegeben werden.

Parameter für Schriftart und Schriftgröße:

FSTAT_FONT= <i>font</i>	Schriftart
FSTAT_FONTSIZE= <i>size</i>	Schriftgröße

Parameter für LIBTOC

Parameter für die Spaltenauswahl in der LIBTOC-Liste:

LIBTOC_CREATION-DATE=Y N	Erstellungsdatum
LIBTOC_ELEMENTSIZE=Y N	Größe Element
LIBTOC_MODIFICATION-DATE=Y N	Änderungsdatum
LIBTOC_TYPE=Y N	Element-Typ
LIBTOC_USER-DATE=Y N	User-Date
LIBTOC_VERSION=Y N	Versionsnummer
LIBTOC_SORT=TYPE/NAME AGE SIZE	Standard-Sortierreihenfolge
TYPE =	Sortierung nach Typ, Elementname und Version Diese Sortier-Option gilt als Standard
NAME =	Sortierung nach Elementnamen und Version
AGE =	Sortierung nach Datum und Uhrzeit absteigend
SIZE =	Sortierung nach Dateigröße absteigend

Die Sortierreihenfolge kann auch direkt bei der Funktion `LIBTOC` (S. 81) mit dem Parameter `SORT` angegeben werden.

Parameter für Schriftart und Schriftgröße:

LIBTOC_FONT= <i>font</i>	Schriftart
LIBTOC_FONTSIZE= <i>size</i>	Schriftgröße

Parameter für LS (Posix-Dateien)

Parameter für die Spaltenauswahl in der LIBTOC-Liste:

LS_LINK=Y N	Anzahl der Links
LS_ATTR=Y N	Dateiattribute
LS_SIZE=Y N	Dateigröße
LS_DATE=Y N	Änderungsdatum/Uhrzeit
LS_GROUP=Y N	Gruppenname
LS_USER=Y N	Benutzername
LS_SORT=NAME AGE SIZE	Standard-Sortierreihenfolge
	NAME =
	Sortierung nach Typ (Verzeichnis/Datei) und Name
	Diese Sortier-Option gilt als Standard
	AGE =
	Sortierung nach Datum und Uhrzeit absteigend
	SIZE =
	Sortierung nach Dateigröße absteigend

Die Sortierreihenfolge kann auch direkt bei der Funktion `LS` (S. 81) mit dem Parameter `SORT` angegeben werden.

Parameter für Schriftart und Schriftgröße:

LS_FONT= <i>font</i>	Schriftart
LS_FONTSIZE= <i>size</i>	Schriftgröße

Sonstige Parameter:

LS_FILE= <i>file</i>	temporäre Ausgabedatei für Kommando LS
	Standard = <code>/var/tmp/ls_out</code> . An diesen Namen wird die aktuelle TSN angehängt.

Datenzugriff erlauben

HTMDIR=[*:cat-id:*] *\$user-id* [*.file*] | \$HOME\$ | */posix-dir*

Diese Anweisung wird nur noch aus Kompatibilitätsgründen unterstützt. Sie wurde durch die Anweisung `FILEACCESS` (S. 53) ersetzt, mit der zusätzliche Auswahlbedingungen möglich sind.

Mit dieser Anweisung wird festgelegt, auf welche Dateien ein Datenzugriff erlaubt ist. Die Anweisung kann mehrmals vorkommen. Es kann entweder der Zugriff auf eine User-ID, auf Dateien mit einem bestimmten Prefix einer User-ID oder auf eine Datei einer User-ID zugelassen werden.

<i>cat-id</i>	CAT-ID. Fehlt diese Angabe, wird für die Prüfung die User-ID um die Default-Cat-ID ergänzt.
<i>user-id</i>	Benutzerkennung der zulässigen Dateinamen. Die User-ID muß immer vollständig angegeben werden. Falls der gesamte String keinen Punkt enthält, wird intern für die Prüfung ein Punkt als Abschluß der User-ID angehängt.
<i>file</i>	Prefix der Dateinamen oder vollständiger Dateiname.
\$HOME\$	Das Home-Verzeichnis der User-ID, unter der das Programm gestartet wurde.
<i>/posix-dir</i>	POSIX-Verzeichnis. Die Freigabe gilt auch für alle Unterverzeichnisse.

Das POSIX-Verzeichnis, das in der Anweisung `DEFDIRPOS` (S. 51) bzw. die User-ID, die in der Anweisung `DEFDIRBS2` (S. 51) enthalten ist, wird nicht automatisch freigegeben. Auch für dieses Verzeichnis bzw. diese User-ID ist eine `HTMDIR`-Anweisung notwendig. So ist es auch möglich, nur einen Teil der Dateien einer User-ID bzw. eines Verzeichnisses freizugeben.

Beispiele:

<code>\$TEST</code>	Alle Dateien der User-ID <code>\$TEST</code>
<code>\$TEST.SRC.</code>	Alle Dateien der User-ID <code>\$TEST</code> , die mit <code>SRC.</code> beginnen
<code>\$TEST.TEST1</code>	Alle Dateien der User-ID <code>\$TEST</code> , die mit <code>TEST1</code> beginnen oder Datei <code>TEST1</code>
<code>/home/test</code>	Alle Dateien und Unterverzeichnisse die unter <code>/home/test</code> gespeichert sind.
<code>\$HOME\$</code>	Alle Dateien, des aktuellen HOME-Verzeichnisses.

Index-Datei

HTMINDEX=*indexfile*]

Ist als URL-Adressen im Browser nur der Name des Rechners angegeben, z.B. `www.opg.de`, wird die Datei *indexfile* angezeigt. Fehlt diese Anweisung, wird die Datei `INDEX.HTM` von der aktuellen User-ID angezeigt.

indexfile Name der Index-Datei einschl. User-ID und ggf. Cat-ID.

Extension HTML-Dateien

HTMEXT=*ext* [,*ext,ext...*]

Dateien, deren Namen mit dem Suffix `htm` oder `html` enden, werden immer als Dateien im HTML-Format interpretiert, d.h. sie werden ggf. von EBCDIC nach ASCII konvertiert und ohne weitere Änderungen an den Browser weitergegeben. Alle anderen Dateien, die nicht im Binär-Modus verarbeitet werden (siehe Funktion `XVIEW` und Anweisung `XVIEWEXT`) werden vor der Weitergabe an den Browser in das HTML-Format umgewandelt.

Der Parameter darf mehrfach angegeben werden. Derzeit sind maximal 300 Extensions zugelassen. Die Länge der Extension darf (ohne Punkt) maximal 5 Zeichen betragen.

ext Datei-Extension.

Layout-Optionen für LIBTOC

LIBTOC_*par*=*opt*

Mit diesen Anweisungen können die Spalten und die Schriftart für die LIBTOC-Liste ausgewählt werden. Beschreibung siehe S. 55.

Layout-Optionen für LS (POSIX-Dateienliste)

LS_par=opt

Mit diesen Anweisungen können die Spalten und die Schriftart für die LS-Liste ausgewählt werden. Beschreibung siehe S. 55.

Lizenz-Nummer

LICENSE=license-string

Lizenz-Nummer für die unbeschränkte Nutzung. Ohne gültige Lizenz-Nummer läuft das Programm nur einen Monat im Test-Modus.

Logging-File

LOGFILE=file-prefix | *NO

file-prefix In der Logging-File werden alle URL-Pfade mit Datum und Uhrzeit protokolliert. Der Dateiname wird um die Extension `pYYYYddd` ergänzt (p=Portnummer, yyyy=Jahr, ddd=Tag des Jahres). Läuft der URL-Server länger als einen Tag, wird um 0:00 Uhr eine neue Log-Datei erzeugt.

*NO Es wird keine Logging-File angelegt.

Logging-File Fehlermeldung

ERRLOGFILE=file-prefix | *NO

file-prefix In der Error-Logging-File werden Meldungen ausgegeben, falls Beim Zugriff auf Dateien Fehler auftreten Der Dateiname wird um die Extension `pYYYYddd` ergänzt (p=Portnummer, yyyy=Jahr, ddd=Tag des Jahres). Läuft der URL-Server länger als einen Tag, wird um 0:00 Uhr eine neue Log-Datei erzeugt.

*NO Es wird keine Error-Logging-File angelegt.

Standardeinstellung Automatische Größenanpassung für Funktion PDF (S. 84)

PDF-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. Diese Option gilt als Standard, falls dieser Parameter nicht angegeben wird.

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).

Hintergrundbilder und Wasserzeichen für Funktion PDF (S. 84)

PDF-BACKGROUND=*jpg-file*[*x,y,w,h*] [; *jpg-file*[*x,y,w,h,p*].... [; WATERMARK : *watermark*]....]

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

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

bs2-file Name einer BS2000-Datei.

lib(typ,elem[,version]) Bibliothek, Elementtyp und Elementname.

/posix-file 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.

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

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

page-to 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

Das Wasserzeichen wird im Gegensatz zum Hintergrundbild immer auf allen Seiten gedruckt.

watermark <FS=fontsize>*watermark-text*

fontsize Schriftgröße für das Wasserzeichen: 1-256

watermark-text Text für das Wasserzeichen, max. 32 Bytes

Beispiele:

```
pdf-background=t1.jpg,10,10,,280
pdf-background=t1.jpg,10,10;t2.jpg,200,10;t3.jpg,20,250,200,30
pdf-background=t1.jpg;watermark:<fs=50>Test
```

Standard-Code-Konvertierung für Funktion PDF (S. 84)

PDF-CODE=D[E] | I[NT] | DK | *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 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 ù Û
Hexa-Wert EBCDIC	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 (ÆØÅæøåùÛ) 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. 50) zugewiesen wird.

Standard-Font für Funktion PDF (S. 84)

PDF-FONT=C[OURIER] | H[ELVETICA] | T[IMES]

Die Namen der Zeichensätze können bis auf 1 Byte abgekürzt werden.

Standard-Fontsize für Funktion PDF (S. 84)

PDF-FONTSIZE=*width* [/*height*]

width Schriftgröße des Zeichensatzes für die Breite eines Zeichens.

height Schriftgröße für den Zeilenabstand. Damit ist es möglich die Zeilenbreite und Zeilenhöhe aus unterschiedlichen Font-Größen zu benutzen. Bei fehlender Angabe gilt die Fontsize aus *width*.

Standard-Seitenrand für Funktion PDF (S. 84)

PDF-MARGIN-LEFT=*left*

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

PDF-MARGIN-TOP=*top*

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

Standard-Seitengröße für Funktion PDF (S. 84)

PDF-MAXLINES=*lines*

lines 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.

Standard-Aufbereitungsmodus für Funktion PDF (S. 84)

PDF-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 und 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 also auch implizit 13, 15, 17 und 1a00)

Standard-Seitengröße für Funktion PDF (S. 84)

PDF-PAGE=A4 | A4Q | A5 | A5Q | *höhe* × *breite* in mm

Seitengröße A4 = DINA4, A4Q = DINA4 quer, A5 = DINA5, A5Q = DINA5 quer oder *höhe* × *breite* in mm, jeweils dreistellig, z.B. 310×210.

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

Layout für PDF-Generierung

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

Bei der Funktion PDF kann mit dem Parameter LAYOUT=*layout-name* der Name eines Layouts angegeben werden, in dem die Parameter für die PDF-Generierung definiert werden können. Wird bei der Funktion PDF kein Layout angegeben, gelten die Standardwerte aus den Anweisungen PDF-FONT, PDF-FONTSIZE usw.

layout Name des Layouts.

<i>font</i>	Zeichensatz: Zulässig sind COURIER (feste Schrift), HELVETICA (Proportionalschrift) oder TIMES (Proportionalschrift). Die Namen der Zeichensätze können bis auf 1 Byte abgekürzt werden.
<i>fontsize</i>	<i>width</i> [/ <i>height</i>]
<i>width</i>	Schriftgröße des Zeichensatzes für die Breite eines Zeichens.
<i>height</i>	Schriftgröße für den Zeilenabstand. Damit ist es möglich die Zeilenbreite und Zeilenhöhe aus unterschiedlichen Font-Größen zu benutzen. Bei fehlender Angabe gilt die Fontsize aus <i>width</i> .
<i>page</i>	A4 A4Q A5 A5Q <i>höhe</i> × <i>breite</i> in mm Seitengröße: A4 = DINA4, A4Q = DINA4 quer, A5 = DINA5, A5Q = DINA5 quer oder <i>höhe</i> × <i>breite</i> in mm, jeweils dreistellig, z.B. 310×210. Ist eine Zeile länger als die definierte Breite der Seite, kann die Seitenbreite automatisch angepaßt werden (siehe Parameter AUTOSIZE). Beim Ausdruck kann im Acrobat Reader angegeben werden, daß solche Seiten auf DINA4 bzw. DINA5 verkleinert werden sollen (mit der entsprechenden Anpassung der Schriftgröße).
<i>maxlines</i>	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>coll-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>coll-col2</i> : Spaltenangabe. Für die PDF-Erzeugung werden nur Daten von Spalte <i>coll</i> bis <i>col2</i> benutzt. Bei <i>mode</i> SIE oder ASA muß das Druck-Steuerzeichen in <i>coll</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

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
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 (ÆØÅæøåüÜ) 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. 50) zugewiesen wird.

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

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

Hintergrundbilder und Wasserzeichen

Hier können Hintergrundbilder und ein Wasserzeichen angegeben werden. Das Wasserzeichen wird über die ev. vorhandenen 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:

bs2-file Name einer BS2000-Datei.

lib(*typ,elem*[*,**version*]) Bibliothek, Elementtyp und Elementname.

/posix-file 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.

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

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

page-to 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

Das Wasserzeichen wird im Gegensatz zum Hintergrundbild immer auf allen Seiten gedruckt.

watermark <FS=*fontsize*>*watermark-text*

fontsize Schriftgröße für das Wasserzeichen: 1-256

watermark-text 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.

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
```

Port-Nummer

PORT=*portnr* Port-Nummer für die Kommunikation über TCP/IP. Wird eine vom Standard abweichende Portnummer (80) festgelegt, muß beim Verbindungsaufbau die Portnummer, durch ":" getrennt, angegeben werden, z.B. Beispiel: `www.opg.de:8888`.

Der Port 80 ist ein privilegierter Port (Standard = Port 1-2049), der nur unter TSOS oder in Kennungen mit dem Privileg NETADMIN benutzt werden darf. Soll der URL-Server unter einer nicht privilegierten Kennung laufen, muß entweder eine Port-Nummer > 2049 verwendet werden oder der Nummernkreis der privilegierten Ports geändert werden.

Mit dem Parameter `PRIVPORT#` des BS2000-Kommandos `BCMOD` bzw. beim `BCSTART` könnte der Port 80 zum nicht privilegierten Port erklärt werden. Aus Sicherheitsgründen kommt diese Methode in der Regel nicht in Frage.

Soweit SECOS installiert ist, kann mit dem Kommando `SET-PRIV PRIV=*NET-ADMIN, USER-ID=userid` einer beliebigen User-ID das NETADM-Privileg zugeordnet werden. Das Kommando ist nur unter der Kennung `SYS-PRIV` erlaubt.

User-Exit-Modul URLEXIT aktivieren

POST_EXIT=Y, *exit-param*

Nach der Ausführung einer Funktion wird der User-Modul URLEXIT (Funktion USREXPO) aufgerufen, ausführliche Beschreibung siehe S. 99.

exit-param Beliebiger String, der dem Modul übergeben wird.

Anzahl der Sub-Tasks

SUBTASKS=n Der URL-Server startet bis zu 15 Sub-Tasks, in denen die Bereitstellung der Daten durchgeführt wird. Dadurch wird sichergestellt, daß der Haupt-Task immer bereit ist, Anforderungen entgegen zu nehmen und durch aufwendige Datenbereitstellungen der Betrieb nicht gestört wird.

n Anzahl der Sub-Tasks 0-15.

Wird 0 als Anzahl der Subtasks angegeben, startet der Server im Single-Mode-Betrieb, d.h. alle Eingaben werden vom Haupttask selbst bearbeitet. Bei längeren Transaktionen können sich dadurch Wartezeiten für die Benutzer ergeben. Sinnvoll ist der Single-Mode-Betrieb immer dann, wenn Benutzermodule ausgetestet werden sollen, da der Ablauf dediziert im Haupttask erfolgt.

Prefix für Template-Dateien

TEMPLATE=prefix

prefix Prefix für den Namen der Template-Dateien (S. 92). Fehlt die Anweisung, wird der String "TEMPLATE" verwendet. Für die Erzeugung des Dateinamens wird zwischen *prefix* und der Funktion ein "." eingefügt, falls der Prefix nicht mit einem "." endet, z.B. "TEMPLATE.FSTAT". Die Anweisung kann auch in der Datei AUTHFILE (S. 45) für jede Gruppe angegeben werden.

Userlink für FSTAT/LIBTOC-Liste für bestimmte Dateien

USERLINK=ausw | (ausw [,ausw] ...), *link*

Mit dieser Anweisung kann ein Userlink im HTML-Format definiert werden, der in der FSTAT- oder LIBTOC-Liste zusätzlich zum Standard-Link generiert wird. Jede USERLINK-Anweisung besteht aus zwei Parametern:

- a) Dateinamen bzw. Teilstring von Dateinamen als Suchbegriff zur Auswahl der Dateien, für die dieser Userlink generiert werden soll.
- b) HTML-Anweisungen, die bei diesem Link generiert werden sollen.

Insgesamt können bis zu 128 verschiedene Links und 128 verschiedene Auswahlbedingungen definiert werden. Dabei können für die gleichen Datei-Auswahl-Bedingungen auch mehrmals angegeben werden. d.h. für eine Datei werden mehrere Links generiert.

<i>ausw</i>	<i>file</i> <i>prefix*</i> <i>*suffix</i> <i>*string*</i> TYPE: <i>type</i> Ein Suchstring kann max. 54 Byte lang sein. Bei Bibliothekselementen wird der String nur im Elementnamen gesucht. Die Klein- / Großschreibung ist zu beachten, d.h. für BS2000-Dateien und Bibliothekselemente ist der Suchstring immer in Großbuchstaben, für POSIX-Dateien in der Regel in Kleinbuchstaben anzugeben. Der Suchstring kann auch die Variablen (S. 73) !AUTHGROUP und !AUTHUSER enthalten.
<i>file</i>	Vollständiger Dateiname / Elementname.
<i>prefix*</i>	Dateiauswahl durch Angabe des Prefix, z.B. SRC.* (alle Dateien, die mit "src." beginnen).
<i>*suffix</i>	Dateiauswahl durch Angabe des Suffix, z.B. *LST (alle Dateien, die mit "lst" enden).
<i>*string*</i>	Dateiauswahl durch Angabe einer Zeichenfolge, die an einer beliebigen Stelle des Dateinamens vorkommt, z.B. *TEST*. (alle Dateien, die an einer beliebigen Stelle, auch am Anfang oder am Ende, "TEST" enthalten).
<i>type</i>	Element-Typ von Bibliothekselementen.
<i>link</i>	HTML-Anweisungen für den Link, der in der Regel mindestens einen "A-Tag" mit HREF enthält. Der Link kann max. 256 Bytes lang sein. Falls in der BS2000-Prozedur zum Starten des URLServers auch die Parameter enthalten sind und das Zeichen "&" als Escape-Zeichen definiert ist, müssen ev. vorkommende Zeichen "&" verdoppelt werden. Der aktuelle Dateinamen kann in Form des Platzhalters "!NAME" angegeben werden.

Beispiel:

Aus Darstellungsgründen werden die Anweisungen teilweise in mehrere Zeilen aufgeteilt. In der Parameterdatei müssen alle Daten für einen Userlink in einer Zeile enthalten sein.

```
USERLINK=SRC.*, <A HREF=" ../EDTW/!NAME>EDTW</A>
USERLINK=SRC*, <A HREF=" ../EDTW/!NAME>EDTW<IMG SRC=" ../XVIEW/EDTW.GIF"
WIDTH=16 HEIGHT=16 BORDER=0></A>
USERLINK (*LST, *.DAT),
<A HREF=" ../PDF/!NAME?PAGE=A4Q&&MODE=SIE&&MIME-TYPE=APPLICATION/X-
PDFP"><IMG SRC=" ../XVIEW/PRINTER.GIF" WIDTH=16 HEIGHT=16 BORDER=0></A>
USERLINK>(*LST, *LISTE, *ERRFILE*)
<A HREF=" ../PDF/!NAME?PAGE=A4&&MODE=SIE"> <IMG SRC=" ../XVIEW/ACRO.GIF"
WIDTH=16 HEIGHT=16 BORDER=0></A>
USERLINK>(*LST, *LISTE, *ERRFILE*)
<A HREF=" ../PDF/!NAME?PAGE=A4&&MODE=SIE"> <IMG SRC=" ../XVIEW/ACRO.GIF"
WIDTH=16 HEIGHT=16 BORDER=0></A>
```

Zugriff auf alle Dateien**VIEW=ALL | SHARE**

SHARE	Es ist nur der Zugriff auf Dateien mit dem Attribut SHARE erlaubt. Dies ist vor allem dann von Belang, wenn der URL-Server unter TSOS läuft. Diese Einstellung gilt als Standard.
ALL	Soweit das Programm unter TSOS läuft, werden auch Dateien angezeigt, die nicht das Attribut SHARE haben.

Passwörter in BS2-Prozeduren maskieren

VIEW-PASSWORD=SECRET

VIEW-PASSWORD=NO

SECRET oder NO Passwörter, die in BS2000-Prozeduren vorkommen, werden mit dem String "P" überschrieben. Passwörter in SDF-Kommandos werden nur ausgewertet, wenn sie mit dem Schlüsselwort "PASSWORD=" angegeben sind. In folgenden Datensätzen werden Passwörter ausgewertet:

```
/LOGON user,account,password
/PASS[WORD] password
/E[ENTER] file,user,account,password
/sonstigeKommandos .....PASSWORD=password
```

password = 'string' | C'string' | H'x-string' | dezimalzahl

Beispiel:

Datensatz: /LOGON USER1,ACC1,C'GEHEIM'
Anzeige: /LOGON USER1,ACC1,P

Sonderzeichen "<", ">" und "&"

VIEWTAG=Y|N

Empfängt der Browser die Zeichen "<", ">" oder "&", werden diese Zeichen nicht angezeigt sondern als HTML-Anweisungen interpretiert. Kommen in einer Datei, die **keine** HTML-Datei ist und mit VIEW angezeigt wird, diese Zeichen vor, können HTML-Anweisungen wahlweise angezeigt oder interpretiert werden. Bei der Funktion VIEW (S. 89) kann die Einstellung für eine Datei mit der Option VIEWTAG geändert werden.

- Y Die Sonderzeichen werden maskiert, so daß sie unverändert angezeigt werden. Diese Einstellung gilt als Standard. Die gültigen Dateiendungen (in der Regel HTML und HTM) werden mit der Anweisung HTMEXT (S. 58) definiert.
- N Die Sonderzeichen werden unverändert an den Browser weiter gegeben und als HTML-Anweisungen interpretiert.

Extension für binäre Übertragung

XVIEWEXT=ext [,ext,ext....]

Dateien, deren Namen mit dem Suffix gif, jpg, jpeg, class oder png enden, werden immer als ASCII-Dateien im Binärformat interpretiert, d.h. sie werden nicht von EBCDIC nach ASCII konvertiert und ohne weitere Änderungen an den Browser weitergegeben. Der Parameter kann mehrfach angegeben werden. Insgesamt sind derzeit bis zu 300 Extensions zulässig. Die Länge der Extension darf (ohne Punkt) maximal 5 Zeichen betragen.

ext Datei-Extension.

Beispiel: xviewext=xls,pdf,doc

Format der Browser-Eingabe

`http://hostname[:port] [/auth-group] /function /filename [?params[¶m.....]]`

hostname Host-Name oder IP-Adresse.

port Falls in der Parameterdatei die Anweisung `PORT` mit einem anderen Port als 80 enthalten ist, muß hier die entsprechende Portnummer angegeben werden. In diesem Fall ist darauf zu achten, daß der String "`http://`" vor dem Host-Namen angegeben wird, da die sonst übliche Ergänzung durch den Browser unterbleibt.

Wird nur der Hostname und ggf die Port-Nummer eingegeben, wird die in der Anweisung `HTMINDEX` definierte Indexseite angezeigt. Bei Auslieferung ist in der Prozedur `J/URLSERVER` als Indexseite die Datei `OPGCOM.LIB/T/INDEX.HTML` eingetragen.

auth-group Soweit Daten durch ein Authentifizierungsverfahren gesichert werden sollen, muß an dieser Stelle der Name einer Benutzergruppe angegeben werden. Die Rechte dieser Gruppe sind in der Datei beschrieben, die mit dem Parameter `AUTHFILE` (S. 45) zugewiesen ist. Beim ersten Zugriff auf einen geschützten Bereich wird der Username und das Paßwort angefordert. Das Paßwort gilt bis zur Beendigung des Browsers.

function Folgende Funktionen sind zur Zeit eingebaut:

- CGI-BIN** User-Modul starten.
- EDTW** BS2000-Datei, POSIX-Datei oder Bibliothekselement downloaden und mit EDT für Windows öffnen.
- ENTER** BS2000-Prozedur als Enter starten und Ergebnisdatei anzeigen.
- FSTAT** Liste von Dateien mit variablen Links erzeugen.
- PDF** Aus BS2000-Datei, POSIX-Datei oder Bibliothekselement eine PDF-Datei erzeugen und mit Acrobat Reader anzeigen.
- VIEW** Aus BS2000-Datei, POSIX-Datei oder Bibliothekselement eine HTML-Datei erzeugen und anzeigen.
- XVIEW** BS2000-Datei, POSIX-Datei oder Bibliothekselement downloaden und in Abhängigkeit des Dateityps mit der entsprechenden Applikation anzeigen.
- LIBTOC** Liste von Bibliothekselementen mit variablen Links erzeugen..

Damit die relative Adressierung in Links (`<a href=...`) von Bibliothekselementen und POSIX-Dateien möglich ist, kann die Funktion auch als Parameter `F=function` angegeben werden.

Beispiel:

Eingabe in der Kommandozeile des Browsers:

```
host/view/weblib/s/index.html
```

Links in der Datei `index.html`:

```
<a href="test1.lst?f=pdf">
<a href="../p/test2.lst?f=pdf">
```

Vom Browser werden daraus folgende Aufruf erzeugt:

```
http://host/view/weblib/s/test1.lst?f=pdf
http://host/view/weblib/p/test2.lst?f=pdf
```

filename *bs2-file* | *lib/typ/elem[/vers]* | *lib(typ,elem,vers)* | */posix-file* | **posix:posix-file*
Name einer BS2000-Datei, POSIX-Datei oder eines Bibliothekselements.

bs2-file Name einer BS2000-Datei.

lib/typ/elem[/vers]

PLAM-Bibliothek / Element-Typ / Elementname / Version. Dieses Format ist nicht zulässig für die Funktion LIBTOC. Alternativ kann auch das alte Format *lib(typ,elem,vers)* verwendet werden.

lib(typ,elem[,vers])

Dieses Format muß für die Angabe von Suchkriterien für die Funktion LIBTOC verwendet werden. Es kann auch für andere Funktionen benutzt werden. Es ist jedoch folgendes zu beachten: Vom Browser kann in diesem Fall der Datei-Typ wegen der schließenden Klammer nicht erkannt werden. Außerdem ist es nicht möglich, von der aktuellen Position aus in einem Link eine relative URL-Adresse anzugeben ohne den Bibliotheksnamen zu wiederholen.

/posix-file Name einer POSIX-Datei einschl. absoluter Pfadangabe (max. Länge 256).

**posix:posix-file*

Name einer POSIX-Datei mit relativer Pfadangabe bzw. ohne Verzeichnisname. Dem Dateinamen wird der Verzeichnisname aus der Anweisung `DEFDIRPOS` vorangestellt. Bei einer relativen URL-Adresse (`href="../datei"`) darf der String `*posix:` nicht vorangestellt werden, da vom Browser automatisch die URL-Adresse um den Prefix der aktuellen Position ergänzt wird.

Beispiel:

Eingabe in der Kommandozeile des Browsers:

```
host/view//home/test/html/index.html
```

Links in der Datei `index.html`:

```
<a href="test1.lst?f=pdf">  
<a href="../list/test2.lst?f=pdf">
```

Vom Browser werden daraus folgende Aufruf erzeugt:

```
http://host/view//home/test/html/test1.lst?f=pdf  
http://host/view//home/test/list/test2.lst?f=pdf
```

?params[¶m.....]

Nach dem Dateinamen bzw. Elementnamen können wahlweise Parameter angegeben werden. Entsprechend der "Browser-Syntax" wird der erste Parameter mit "?", die weiteren Parameter mit "&" eingeleitet.

URL-Adresse in einem Link

Die Angabe einer URL-Adresse (Verweis auf eine Datei) kann z.B. in den HTML-Tags `<href=...>`, `` oder `<frameset src=...>` vorkommen. Relative Adressen werden vom Browser automatisch zu einer absoluten Adresse vervollständigt.

Absolute Adresse

```
http://hostname[:port] [ /auth-group ] /function /filename [?params[&param.....] ]
ftp://hostname[:port] [ /auth-group ] /function /filename [?params[&param.....] ]
```

Mit dieser Form wird ein Ziel auf einem anderen Rechner angesprochen.

Relative Adresse ab Hostnamen

```
[/auth-group/ ] function /filename [?params[&param.....] ]
```

Mit dieser Form wird ein Ziel auf dem gleichen HOST angesprochen. Die relative Adresse wird vom Browser um den String "`http//:hostname`" ergänzt.

Relative Adresse ab der aktuellen Position

Diese Variante sollte bevorzugt verwendet werden. Die relative Adressierung hat den Vorteil, daß alle Dateien für die WEB-Anwendung leicht in eine andere User-ID bzw. in eine andere Bibliothek oder in ein anderes POSIX-Verzeichnis verschoben werden können.

```
../[.../../]function /filename [?params[&param.....] ]
../[.../../]function /lib/type/element [?params[&param.....] ]
```

Mit dieser Form wird ein relatives Ziel ab der aktuellen Position auf dem gleichen HOST angesprochen. Mit jeweils einem `../` kommt man eine Ebene höher, also z.B. von *filename* zu *function* oder von *element* zur *type*. Mit `../../` würde man von *element* zu *filename* bzw. *lib* gelangen. Mit `../.../../` würde man von *element* zu *function* gelangen. Die relative Adresse wird vom Browser um die aktuelle Position, ohne die Verzeichnisse, die zurück positioniert wurde, ergänzt.

```
filename | element [?params[&param.....] ]
```

Mit dieser Form wird ein Dateiname bzw. ein Element mit dem gleichen Typ in der gleichen Bibliothek mit der gleichen Funktion wie die aktuelle Seite angesprochen. Die relative Adresse wird vom Browser um die aktuelle ergänzt.

Variable URL-Adresse in einem Link

Die URL-Adresse kann Variable enthalten, die vor dem Zugriff auf die Datei / des Elements vom URLServer durch den aktuellen Wert ersetzt werden. Die Variable wird nur im Dateinamen (*file*, *lib/typ/elem/vers* oder *posixfile*) ersetzt. Dadurch ist es möglich, für unterschiedliche Gruppen bzw. Benutzer verschiedene Web-Seiten vorzusehen, ohne daß die Seite, in der die Links vorkommen, mehrfach erzeugt werden muß.

!AUTHGROUP Aktuelle Gruppe (S. 45).

!AUTHUSER Aktueller Benutzer (S. 45).

!CGILIB PLAM-Bibliothek, in der die Benutzermodule und Prozeduren sind (S. 48).

Soweit keine aktuelle Gruppe bzw. kein aktueller User existiert, wird der String !AUTHGROUP bzw. !AUTHUSER als Teile des Dateinamens gelöscht.

Beschreibung der Funktionen

Benutzer-Modul starten

CGI-BIN/*usermod* [*?par1=wert [&par2=wert2.....]*]

Aufruf User-Modul. Der Modul wird aus der Bibliothek geladen, die in der Anweisung CGIBIN definiert ist. Nach dem Modulnamen können wahlweise Parameter angegeben werden. Entsprechend der "Browser-Syntax" wird der erste Parameter mit "?", die weiteren Parameter mit "&" eingeleitet. In der Bibliothek OPGCOM.LIB sind drei Testmodule enthalten: L/HALLOA (Source=S/HALLOA.SRC), L/HALLOC (Source=S/HALLOC.C), L/ZEILENA (Source=S/ZEILENA.SRC), L/ZEILENC (Source=S/ZEILENC.C).

Die Unterprogramm-Schnittstelle wird im Abschnitt "Schnittstelle User-Modul" (S. 96) beschrieben.

Beispiele:

cgi-bin/halloa

cgi-bin/zeilena?ANZ=20

cgi-bin/test?par1=xxx&par2=yyyy

Modul ZEILENA mit Parameter ANZ=20

Modul test mit Parameter par1=xxx und par2=yyyy

CFS-Prozedur starten

CFS/*cfs-proc-elem*[*,proc-par*]

Die Funktion startet eine CFS-Prozedur. Beim ersten Aufruf wird der Modul CFSUP nachgeladen. In den SYSDTA-Anweisungen muß die CFSLIB mit der Anweisung CFSLIB (S. 49) zugewiesen werden. In der CFS-Prozedur können mit der neuen Funktion *RUN (URLOUT) Daten direkt an den Browser gesandt werden.

cfs-proc-elem Elementname einer CFS-Prozedur. Das Element vom Typ S muß sich in der CGILIB (Anweisung CGILIB=*library*) befinden.

proc-par Parameter für die CFS-Prozedur. Mehrere Parameter werden durch Komma getrennt.

CFS-Funktion *RUN (URLOUT)

Diese CFS-Funktion bietet die Möglichkeiten, Daten an den Browser zu senden bzw. Informationen der Browser-Eingabe anzufordern. Die Anweisung kann sich auch über mehrere Sätze erstrecken. Als Fortsetzungszeichen dient das Zeichen "-".

RUN (URLOUT) HTTP, *content-type

An den Browser wird der HTTP-Header mit dem Content-Type *type* gesandt. Fehlt der Parameter *content-type* wird "text/html" verwendet. Der Header beinhaltet folgende Daten:

```
HTTP/1.0 200 OK
Content-Type: type
Leerzeile
```

Content-type Der Content-Type kann entweder als CFS-Variable oder als String angegeben werden. Der String kann in Hochkommas eingeschlossen werden.

Beispiele:

```
*&TYPE='text/html'
*RUN (URLOUT) HTTP, &TYPE
*RUN (URLOUT) HTTP, text/plain
*RUN (URLOUT) HTTP, 'text/plain'
```

RUN (URLOUT) HREF, *url

An den Browser wird ein HTTP-Header mit dem Status Code 302 (Andere URL-Adresse anfordern) gesandt. Diese Anweisung ist nur sinnvoll, wenn sie als einzige URLOUT-Anweisung am Ende einer Prozedur verwendet wird, d.h. es wird zuerst eine Datei erzeugt und dann die Anzeige dieser Datei mit ***RUN (URLOUT) HREF** gestartet. Dabei können alle Funktionen des URLServer benutzt werden, wie z.B. VIEW, XVIEW, EDTW, PDF usw. Von der CFS-Prozedur muß also nicht unbedingt eine HTML-Datei aufgebaut werden. Es kann auch eine Text-Datei erzeugt werden, die dann mit den Funktionen des URLServer aufbereitet wird.

```
HTTP/1.0 302 GO TO PAGE
Location: url
Leerzeile
```

url Die URL-Adresse kann entweder als CFS-Variable oder als String angegeben werden. Der String kann in Hochkommas eingeschlossen werden.

Beispiele:

```
Datei TEST.LST erzeugen.
*&GOTO='../PDF/TEST.LST'
*RUN (URLOUT) HREF, &GOTO
*RUN (URLOUT) HREF, ../PDF/TEST.LST
*RUN (URLOUT) HREF, '../PDF/TEST.LST'
```

***RUN (URLOUT) DATA[, *htmltext*] [, &*var1*, &*var2*.....]**

An den Browser wird der Inhalt des Parameters *htmltext* und der Variablen gesandt. Die Daten werden von EBCDIC nach ISO8859 konvertiert.

htmltext String mit Daten und HTML-Tags, die an den Browser gesandt werden sollen. Der String kann in Hochkommas eingeschlossen werden.

&var1,&var2.. Variablen mit den Daten und HTML-Tags, die an den Browser gesandt werden sollen. Eine Variable kann max. 80 Bytes enthalten. Insgesamt darf der Inhalt aller Variablen die max. Länge von 4096 Bytes nicht überschreiten.

Beispiele:

```
*&VAR1='Daten Zeile1<BR>'
*&VAR2='Daten Zeile2<BR>'
*RUN (URLOUT) DATA, &VAR1, &VAR2
*RUN (URLOUT) DATA, 'FSTAT für Datei '
*RUN (URLOUT) DATA, &FILE
*RUN (URLOUT) DATA, <BR>
```

*RUN (URLOUT) DATAFILE, *file*

Der Inhalt der Datei *file* wird an den Browser gesandt. Die Daten werden von EBCDIC nach ISO8859 konvertiert. Die Datei muß entweder Daten im HTML-Format enthalten oder vor der Anweisung *RUN (URLOUT) DATAFILE muß der HTML-Tag <PRE> erzeugt werden oder es muß als Content-Type "text/plain" verwendet werden.

Beispiel:

```
*RUN (URLOUT) DATAFILE, &TESTDATEI
*RUN (URLOUT) DATAFILE, #OUT
```

*RUN (URLOUT) GETGROUP, *cfsvvar*

Die zur Zeit gültige Gruppe wird in die Variable *cfsvvar* eingelesen. Ist beim Aufruf der CFS-Prozedur keine Gruppe angegeben worden, ist die Variable leer.

Beispiel:

```
*RUN (URLOUT) GETGROUP, &GROUP
```

*RUN (URLOUT) GETUSER, *cfsvvar*

Die zur Zeit gültige Benutzer wird in die Variable *cfsvvar* eingelesen. Ist beim Aufruf der CFS-Prozedur keine Gruppe angegeben worden, ist die Variable leer.

Beispiel:

```
*RUN (URLOUT) GETUSER, &USER
```

CFS-Funktion *RUN (SYSOUT)

*RUN (SYSOUT) [**EXTEND**] *&sysout, &oldsysout*

Diese CFS-Funktion bietet die Möglichkeiten, SYSOUT auf eine neue Datei zuzuweisen und den Namen der bisherigen SYSOUT-Datei in einer CFS-Variablen zu speichern. Damit kann nach der Erstellung von SYSOUT-Daten wieder die bisherige SYSOUT-Datei zugewiesen werden.

EXTEND	Die SYSOUT-Datei wird mit OPEN-MODE=*EXTEND geöffnet.
<i>&sysout</i>	Name einer CFS-Variablen, die den Dateiname der neuen SYSOUT-Datei enthält.
<i>&oldsysout</i>	Name einer CFS-Variablen, in die der Dateinamen der alten SYSOUT-Datei übertragen werden soll.

Beispiel:

```
* &SYSOUT=' #OUT '  
* RUN (SYSOUT) , &SYSOUT, &OLDSYSOUT  
* /FSTAT */SHOW-FILE-ATT &FILE, INF=*A  
* RUN (SYSOUT) EXTEND, &OLDSYSOUT, &SYSOUT  
* RUN (URLOUT) DATAFILE, &SYSOUT
```

Datei mit EDT für Windows öffnen

Die Datei/das Bibliothekselement wird vom Browser in einer temporären Datei gespeichert und mit dem Programm EDT für Windows über die DDE-Schnittstelle geöffnet.

Für diese Funktion muß der Mime-Type application/x-edtw und der Dateitype ".EDTW" registriert sein. Diese Registrierung erfolgt bei der Installation automatisch nach dem Entpacken der Installationsdateien mit dem mitgelieferten SETUP.EXE

EDTW*/filename* | *lib/typ/elem*[*/vers*] | *ZIP:*zip-archiv (elem)*

Die Datei/das Bibliothekselement wird im Textmodus in den Arbeitsbereich des EDT für Windows eingelesen und im Standard-Code des EDTW angezeigt. Intern werden folgende Parameter verwendet: ?trans=y&mime-type=application/x-edtw.

EDTWBIN*/filename* | *lib/typ/elem*[*/vers*] | *ZIP:*zip-archiv (elem)*

Die Datei/das Bibliothekselement wird im Binärmodus in den Arbeitsbereich des EDT für Windows eingelesen und im EBCDIC-Code angezeigt. Intern werden folgende Parameter verwendet: ?rec=lf&mime-type=application/x-edtw.

Datei mit EDT für Windows editieren

EDTWUPD*/filename* | *lib/typ/elem*[*/vers*] [?CODE=DE | INT | *ccsn*] [&UPDATE=*YES | *NO | *newfile*]

Die Datei/das Bibliothekselement wird wie mit der Funktion EDTWBIN im Binärmodus in den Arbeitsbereich des EDT für Windows eingelesen und im EBCDIC-Code angezeigt. Zusätzlich wird eine ENTER-Prozedur gestartet, die auf den Update der Datei wartet. Die Enter-Prozedur wird erst beendet, wenn das Arbeitsfenster des EDTW geschlossen wird.

Datenformat:

Den Daten wird ein Header vorangestellt, in dem alle notwendigen Informationen für die Verwaltung enthalten sind. Diese Informationen sind nicht sichtbar, können jedoch mit dem Kommando `STA FSEND` angezeigt werden. Die Daten werden ohne Code-Konvertierung in EBCDIC übertragen und auch als EBCDIC-Daten angezeigt. Der Anzeige-Code wird automatisch eingestellt. Damit alle Zeichen dargestellt werden können, z.B. auch `X'0D'` oder `X'0A'` werden die Daten in einem besonderen Format übertragen: Sätze variabler Länge mit Satzlängenfeld.

Dateiformat:

Im Gegensatz zum OPEN-FT und FTP können auch **ISAM-Dateien, PAM-Dateien, delta-gespeicherte LMS-Elemente, POSIX-Dateien und BS2000-ZIP-Elemente** editiert werden.

ISAM-Dateien mit den Standard-Isamkey (KEYPOS=5,LEYLENGTH=8) werden im EDTW mit der Option "KEY" eingelesen, so daß die ISAM-Schlüssel als Zeilennummer angezeigt werden. Aber selbst ISAM-Dateien mit Nicht-Standard-Schlüsseln können im EDT bearbeitet werden. Beim Zurückschreiben wird die Datei im BS2000 auch richtig geschrieben, wenn die ISAM-Schlüssel nicht aufsteigend sind.

Die Daten werden immer mit der Option "OVERWRITE" geschrieben.

Prüfung auf Änderung außerhalb des EDTW:

Wie bei lokalen Dateien wird zyklisch und beim Wechsel in einen anderen Arbeitsbereich bzw. vor dem Schreiben geprüft, ob die BS2000-Datei inzwischen außerhalb des EDTW geändert worden ist und ggf. eine Warnung ausgegeben.

Proxy-Server FSendSRV für BS2000 und Windows/Linux

Dateien können auch über einen Proxy-Server empfangen und/oder zurückgesandt werden. Dadurch ist es möglich, Dateien zu einem PC zu senden, der im BS2000 nicht bekannt ist. Ebenfalls ist es möglich zum Empfangen im BS2000 statt dynamischer Ports nur einen zentralen Port zu verwenden. Siehe hierzu den Abschnitt FSendSRV (S. 112).

filename

Name einer BS2000- oder POSIX-Datei. Beginnt der Dateiname mit dem Zeichen "/" (absoluter Dateiname) oder dem String "`*posix:`", erfolgt ein Zugriff auf das POSIX-Filesystem. Es ist in diesem Fall darauf zu achten, daß der POSIX-Pfad mit der Anweisung `HTMDIR` (S. 57) freigegeben wird. Ein relativer Pfad wird um das Verzeichnis ergänzt, daß mit der Anweisung `DEFDIRPOS` (S. 51) definiert ist.

Soweit ein BS2000-Dateiname keine User-ID enthält, wird er um die User-ID ergänzt, die in der Anweisung `DEFDIRBS2` (S. 51) definiert ist.

lib/typ/elem[/vers]

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

**ZIP:zip-archiv(elem)*

Name eines ZIP-Archivs und der Elementname. Ein ZIP-Element kann nicht geändert werden.

CODE=DE | INT | EDF041 | *ccsn*

Angaben für die Darstellung des EBCDIC-Codes im EDT für Windows. Bei dem EBCDIC-Code **EDF03IRV**, der zur Zeit als Standard gilt, gibt es zwei Varianten. Das Problem 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
Hexa-Wert	FB	4F	FD	BB	BC	BD	FF
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. Dieser Zeichensatz wird als Standard verwendet.

INT Die Datei enthält Daten in 7-Bit-EBCDIC (CCSN= EDF03IRV internationale Variante), aus den entsprechenden EBCDIC-Zeichen sollen die Zeichen "{ | } [\] ~" erzeugt werden.

EDF041: 8-Bit-EBCDIC-Code. Die Datei enthält unter anderem als Sonderzeichen sowohl die Zeichen "{ | } [\] ~" als auch die Umlaute und das "scharfe S" (**ÄÖÜäöüß**). Die Umlaute sind allerdings anders codiert als beim Zeichensatz EDF03IRV, deutsche Variante.

ccsn Zeichensatzname (CCSN **C**oded **C**haracter **S**et **N**ame), z.B. EDF042. In der Datei CODEPAGE.TXT für das Programm EDT für Windows muß in diesem Fall eine Code-Tabelle mit diesem Namen definiert werden. Zur Zeit werden nur die Code-Tabellen für EDF03IRV und EDF041 ausgeliefert.

Fehlt der Parameter, wird eine ev. vorhandene CCSN-Angabe aus dem Datei-Katalog verwendet. Falls im Datei-Katalog kein CCSN gespeichert ist, wird der Standardwert "DE" verwendet.

UPDATE=*YES | *NO | *newfile*

***NO** Die Eingabedatei kann nicht geändert. Nach dem Sender der Datei wird die Funktion beendet.

***YES** Die Eingabedatei kann mit EDT für Windows geändert werden.

newfile Die Eingabedatei kann mit EDT für Windows geändert und in eine neue Datei geschrieben werden. Als Datei kann eine BS2-Datei, ein PLAM-Bibliothekselement, eine POSIX-Datei oder ein ZIP-Element angegeben werden.

Beispiele:

```
edtw/testdatei           Datei testdatei
edtw/lib/s/elem1        Element elem1 aus Bibliothek lib, Type s
edtw//home/cfstest/test POSIX-Datei
edtwupd/testdate?code=de&update=testdatei.neu
```

ENTER starten

ENTER/*call-proc-elem*[,*proc-par*]?ACTION=*function/params* [&ERASE=NO] [&ENTERPAR=*enterpar*]

Die Funktion erzeugt und startet eine Enter-Prozedur, wartet auf die Beendigung der Prozedur und führt danach die gewünschte Aktion aus.

call-proc-elem Elementname einer Prozedur, die mit dem BS2000-Kommando /CALL-PROC aufgerufen werden kann. Das Element vom Typ J muß sich in der CGILIB (Anweisung CGILIB=*library*) befinden.

proc-par Parameter für die BS2000-Prozedur.

ACTION=*function/params*

Als Aktion, die nach der Beendigung der Enter-Prozedur ausgeführt werden soll, kann eine beliebige Funktion mit den entsprechenden Parametern angegeben werden. Falls ein Benutzermodul angegeben wird, ist darauf zu achten, daß die Parameter zu dem Benutzermodul mit dem Zeichen "&" eingeleitet werden, da als erster Parameter bereits "?action" angegeben wurde.

Als Datei für die Funktion VIEW kann auch SYSOUT oder SYSLST angegeben werden.

TSN in Dateinamen und Parametern:

Damit eindeutige Dateinamen für temporäre Dateien erzeugt werden können, können die Prozedur-Parameter und der Wert des Parameters ACTION den String !TSN enthalten. Dieser String wird beim Erzeugen der Enterdatei durch die aktuelle TSN ersetzt. Dabei handelt es sich um die TSN des Server-Tasks und nicht des ENTER-Tasks.

ERASE=NO Nach dem Anzeigen einer Datei mit der Action VIEW bzw. XVIEW wird die Datei nicht gelöscht. Standardmäßig wird die Datei nach der Anzeige gelöscht.

enterpar Hier können die Enter-Parameter angegeben werden, die beim Starten der Enter-Prozedur gelten sollen. Es sind alle Parameter zulässig, die nach dem Dateinamen der Enter-Prozedur angegeben werden können. Wird als Parameter eine andere User-ID angegeben, um den Job unter einer anderen User-ID laufen zu lassen, wird die SYSOUT- und SYSLST-Datei mit SHARE=YES erstellt.

Beispiele:

```
enter/do.test, (par1=xxx,par2=file1.!tsn)?action=view/sysout
enter/do.test, (file1.!tsn)?action=view/file1.!tsn&erase=no
enter/do.test, (pre=file1.!tsn.)?action=fstat/file1.!tsn.&enterpar=user1,acct,'pass'
```


FSTAT - Liste, Inhaltsverzeichnis PLAM-Bibliothek/ZIP-Archiv oder Liste Posix-Dateien erzeugen

FSTAT/*filename* [, *show-file-par*] | FILELIST : *name* [?ACTION=*function*&ACTIONPAR=*funcnpar*] [&SORT=*sort-fstat*]

LIBTOC/*lib* [(*typ,elem* [, *vers*])] [?ACTION=*function*&ACTIONPAR=*funcnpar*] [&SORT=*sort-libtoc*]

LIBTOC/*ZIP : *zip-archiv* [(*elem*)] [?ACTION=*function*&ACTIONPAR=*funcnpar*] [&SORT=*sort-libtoc*]

LS/*posix-path* [?ACTION=*function*&ACTIONPAR=*funcnpar*] [&SORT=*sort-posix*]

filename Suchmuster für Dateien. Teil eines Dateinamens einschl. Wildcard-Zeichen. Mit den angegebenen Suchmustern zu Dateien und den weiteren Optionen *show-file-par* wird das BS2000-Kommando `SHOW-FILE-ATTR` aufgerufen. Die Parameter sind grundsätzlich in SDF-Syntax anzugeben; der Parameter `output` ist verboten, da er bereits vom FSTAT-Kommando verwendet wird und zu Fehler führen würde.

FILELIST:*name* Name einer Datei, in der pro Satz ein Dateiname steht. Es kann auch eine Datei angegeben werden, die mit dem Kommando "SHOW-FILE-ATTR ... OUTPUT=*file*(*FILE-NAME)" erzeugt wurde. In diesem Fall steht auf Spalte 1 eine Leerstelle und der Dateinamen beginnt ab Spalte 2. Die Datei kann auch mittels einer CFS-Prozedur erzeugt werden. Siehe dazu auch Musterprozedur `S/CFS.FSTAT4` in der Bibliothek `OPGCLOM.LIB`. Gegenüber den Auswahlmöglichkeiten des Kommandos `SHOW-FILE-ATTR` bietet die CFS-Prozedur die Möglichkeit, mehrere Suchbegriffe anzugeben und die Dateinamen aus mehreren Dateienlisten in einer Datei zu speichern.

Aus dem Ergebnis des BS2000-Kommandos `SHOW-FILE-ATTR` bzw. den in der Datei FILELIST:*name* enthaltenen Dateinamen wird eine Tabelle aller Dateien mit Links auf die Funktion `VIEW` (Datei anzeigen) bzw. `LIBTOC` (PLAM-Bibliotheken) erzeugt.

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

Suchmuster für Bibliothekselemente. Vollständiger Bibliotheksname und Suchmuster für Elementtyp, Elementname und Elementversion. Die Suchmuster für *typ*, *elem* und *vers* können auch Wildcard-Zeichen enthalten. Wird für *vers* "*" angegeben, so bedeutet dies, daß z.B. bei delta-gespeicherten Elementen alle Versionen angezeigt werden. Wenn *vers* weggelassen wird, so wird nur die höchste Version ausgewählt. Fehlen die Angaben (*typ,elem* [, *vers*]), wird " (*, *) " als Standard benutzt.

Hier darf nicht das Format *lib/typ/elem* verwendet werden, weil bei der Erzeugung der Links in der LIBTOC-Liste davon ausgegangen wird, daß die aktuelle Position die Ebene nach *filename* ist.

Aus dem Ergebnis wird eine Tabelle aller Elemente mit Links auf die Funktion `VIEW` (Datei anzeigen) erzeugt.

*ZIP : *zip-archiv* [(*elem*)]

Name eines ZIP-Archivs und ggf. ein Suchmuster für die gesuchten Elemente. Das Archiv kann im BS2000 oder mit WINZIP erstellt worden sein.

Aus dem Ergebnis wird eine Tabelle aller Elemente mit Links auf die Funktion `VIEW` (Datei anzeigen) erzeugt.

<i>Posix-path</i>	Suchmuster für Posix-Dateien einschl. Verzeichnis. Teil eines Dateinamens einschl. Wildcard-Zeichen. Mit den angegebenen Suchmustern zu Dateien wird das POSIX-Kommando <code>ls -lad</code> aufgerufen. Die Ausgabedatei für das ls-Kommando kann in der Anweisung <code>LS_FILE</code> definiert werden. Als Standard wird <code>/var/tmp/ls_out</code> benutzt. Aus dem Ergebnis wird eine Tabelle aller Dateien mit Links auf die Funktion <code>VIEW</code> (Datei anzeigen) bzw. <code>LS</code> (Verzeichnisse) erzeugt.																								
<i>function</i>	Hier kann als Aktion der Name einer Funktion angegeben werden, die beim Anklicken eines Elements in der Liste ausgeführt werden soll. Die Funktionen <code>ENTER</code> und <code>CGI-BIN</code> sind nicht zulässig. Standard = <code>VIEW</code> .																								
<i>functpar</i>	Parameter für die Funktion, z.B. Schriftart und Schriftgröße für PDF. Mehrere Parameter sind durch Komma zu trennen.																								
<i>sort-fstat</i>	<table> <tr><td>NAME</td><td>aufsteigend nach Dateiname.</td></tr> <tr><td>AGE</td><td>absteigend nach Datum und Uhrzeit der letzten Änderung</td></tr> <tr><td>SIZE</td><td>absteigend nach der Dateigröße</td></tr> <tr><td>FREE</td><td>absteigend nach der Anzahl der freien PAM-Seiten</td></tr> <tr><td>SEC</td><td>absteigend nach der Anzahl der Anzahl Second Allocation</td></tr> <tr><td>FCB</td><td>aufsteigend nach dem FCB-Type SAM PAM ISAM NONE</td></tr> <tr><td>SHARE</td><td>aufsteigend nach dem Merkmal SHARE=YES NO</td></tr> <tr><td>ACCESS</td><td>aufsteigend nach dem Merkmal READ WRITE</td></tr> <tr><td>PASS</td><td>aufsteigend nach Password = R W E</td></tr> <tr><td>BKL</td><td>aufsteigend nach Backup-Class A, B, C, D oder E</td></tr> <tr><td>EXT</td><td>absteigend nach Anzahl der Extensions</td></tr> <tr><td>VOL</td><td>aufsteigend nach dem Volume-Namen</td></tr> </table>	NAME	aufsteigend nach Dateiname.	AGE	absteigend nach Datum und Uhrzeit der letzten Änderung	SIZE	absteigend nach der Dateigröße	FREE	absteigend nach der Anzahl der freien PAM-Seiten	SEC	absteigend nach der Anzahl der Anzahl Second Allocation	FCB	aufsteigend nach dem FCB-Type SAM PAM ISAM NONE	SHARE	aufsteigend nach dem Merkmal SHARE=YES NO	ACCESS	aufsteigend nach dem Merkmal READ WRITE	PASS	aufsteigend nach Password = R W E	BKL	aufsteigend nach Backup-Class A, B, C, D oder E	EXT	absteigend nach Anzahl der Extensions	VOL	aufsteigend nach dem Volume-Namen
NAME	aufsteigend nach Dateiname.																								
AGE	absteigend nach Datum und Uhrzeit der letzten Änderung																								
SIZE	absteigend nach der Dateigröße																								
FREE	absteigend nach der Anzahl der freien PAM-Seiten																								
SEC	absteigend nach der Anzahl der Anzahl Second Allocation																								
FCB	aufsteigend nach dem FCB-Type SAM PAM ISAM NONE																								
SHARE	aufsteigend nach dem Merkmal SHARE=YES NO																								
ACCESS	aufsteigend nach dem Merkmal READ WRITE																								
PASS	aufsteigend nach Password = R W E																								
BKL	aufsteigend nach Backup-Class A, B, C, D oder E																								
EXT	absteigend nach Anzahl der Extensions																								
VOL	aufsteigend nach dem Volume-Namen																								
<i>sort-libtoc</i>	<table> <tr><td>TYPE</td><td>aufsteigend nach Elementtyp, Elementnamen und Version</td></tr> <tr><td>NAME</td><td>aufsteigend nach Elementnamen und Version</td></tr> <tr><td>SIZE</td><td>absteigend nach der Elementgröße</td></tr> <tr><td>AGE</td><td>absteigend nach User-Datum und User-Uhrzeit</td></tr> <tr><td>VERSION</td><td>aufsteigend nach der Version</td></tr> <tr><td>CREA</td><td>absteigend nach Creation-Date und Creation-Time</td></tr> <tr><td>MOD</td><td>absteigend nach Modification-Date und Modification-Time</td></tr> </table>	TYPE	aufsteigend nach Elementtyp, Elementnamen und Version	NAME	aufsteigend nach Elementnamen und Version	SIZE	absteigend nach der Elementgröße	AGE	absteigend nach User-Datum und User-Uhrzeit	VERSION	aufsteigend nach der Version	CREA	absteigend nach Creation-Date und Creation-Time	MOD	absteigend nach Modification-Date und Modification-Time										
TYPE	aufsteigend nach Elementtyp, Elementnamen und Version																								
NAME	aufsteigend nach Elementnamen und Version																								
SIZE	absteigend nach der Elementgröße																								
AGE	absteigend nach User-Datum und User-Uhrzeit																								
VERSION	aufsteigend nach der Version																								
CREA	absteigend nach Creation-Date und Creation-Time																								
MOD	absteigend nach Modification-Date und Modification-Time																								
<i>sort-posix</i>	<table> <tr><td>NAME</td><td>aufsteigend nach Dateityp (Verzeichnis / sonstige Datei) und Name</td></tr> <tr><td>AGE</td><td>absteigend nach Datum und Uhrzeit der letzten Änderung</td></tr> <tr><td>ATTR</td><td>aufsteigend nach Dateiattributen</td></tr> <tr><td>LINK</td><td>absteigend nach den Anzahl der Links</td></tr> <tr><td>SLZE</td><td>absteigend nach der Dateigröße</td></tr> <tr><td>GROUP</td><td>aufsteigend nach Gruppenname</td></tr> <tr><td>USER</td><td>aufsteigend nach Benutzername</td></tr> </table>	NAME	aufsteigend nach Dateityp (Verzeichnis / sonstige Datei) und Name	AGE	absteigend nach Datum und Uhrzeit der letzten Änderung	ATTR	aufsteigend nach Dateiattributen	LINK	absteigend nach den Anzahl der Links	SLZE	absteigend nach der Dateigröße	GROUP	aufsteigend nach Gruppenname	USER	aufsteigend nach Benutzername										
NAME	aufsteigend nach Dateityp (Verzeichnis / sonstige Datei) und Name																								
AGE	absteigend nach Datum und Uhrzeit der letzten Änderung																								
ATTR	aufsteigend nach Dateiattributen																								
LINK	absteigend nach den Anzahl der Links																								
SLZE	absteigend nach der Dateigröße																								
GROUP	aufsteigend nach Gruppenname																								
USER	aufsteigend nach Benutzername																								

Template-Datei:

Die HTML-Anweisungen für die Darstellung der z.B. Schriftart, Schriftgröße, Farbe, Kopf und Fuß der Seite, können in einer Template-Datei vorgegeben werden. Die Datei muß sich in der Default-User-ID bzw. in der Default-User-ID der Benutzergruppe befinden und muß `template.FSTAT`, `template.LIBTOC` bzw. `template.LS` heißen. Der Prefix `template` wird mit der Anweisung `TEMPLATE` (S. 68) definiert. Ein Muster der Datei befindet sich in der Bibliothek `OPGCOM.LIB`, Element `D/DEMO.TEMPLATE.FSTAT`, `D/DEMO.TEMPLATE.LIBTOC` bzw. `D/DEMO.TEMPLATE.FS`. Weitere Hinweise zur Template-Datei siehe S. 92.

Userlink:

Mit der Anweisung `USERLINK` (S. 68) können Links im HTML-Format definiert werden, die in der `FSTAT`-Liste zusätzlich zum Standard-Link generiert werden, z.B. Generierung PDF-Datei, Anzeigen im EDTW usw.

Auszugebende Spalten und Standard-Sortierung:

Die auszugebenden Spalten und die Standard-Sortierung können Sie mit den Anweisungen `FSTAT_par`, `LIBTOC_par` und `LS_par` (S. 55) oder in der entsprechenden Template-Datei (S. 68) bestimmen.

Beispiele:

```
fstat/$test.
fstat/$test.*,fil-stru=sam,last-ch-d=*today
fstat/*test*
fstat/*.lst?action=pdf&actionpar=page=a4,fontsize=8
fstat/*.lst?sort=age
fstat/filelist:list2
libtoc/lib(*,*)
libtoc/lib(*,*,*)
libtoc/lib(s,src*)
libtoc/lib(s/*.*.h)
libtoc/lib(s/*.*.h)?sort=name
libtoc/lib(p/*)?action=pdf&actionpar=page=a4,fontsize=8
ls//home/test/src/*.c?sort=age
ls//home/test/lst/*?action=pdf&actionpar=page=a4,fontsize=8
cfs/fstat4
```

Prozedur `s/fstat4` in der Bibliothek, die mit der Anweisung `CGILIB` zugewiesen ist: Es wird die Datei `OPGCOM.LIB` ausgewählt, danach wird die Dateienliste um alle Dateien ergänzt, die mit "DEMO." beginnen. Anschließend werden die Dateinamen der Dateienliste mit dem Kommando `ON&DPF` in die Datei `TEMP.FILELIST` geschrieben und der Link für die Funktion `FSTAT` an den Browser übergeben.

```
*PROC N
*&FSTATFILE='TEMP.FILELIST'
NPOPGCOM.LIB,Q
ALDEMO.,Q
ON&DPF &FSTATFILE
AU
*RUN (URLOUT)HREF,.. /FSTAT/FILELIST:&FSTATFILE
END
```

LIBTOC - Inhaltsverzeichnis Bibliothek oder ZIP-Archiv erzeugen

LIBTOC*/lib* (*typ,elem*[, *vers*]) [*?ACTION=function&ACTIONPAR=functpar*] [*&SORT=sort-libtoc*]

LIBTOC**ZIP:zip-archiv* [*(elem)*] [*?ACTION=function&ACTIONPAR=functpar*] [*&SORT=sort-libtoc*]

Beschreibung siehe Funktion `FSTAT` (S. 81)

LS - POSIX-Dateienliste erzeugen

LS*/posix-path* [*?ACTION=function&ACTIONPAR=functpar*] [*&SORT=sort-posix*]

Beschreibung siehe Funktion `FSTAT` (S. 81)

PDF - Datei erzeugen und anzeigen

PDF*filename* | *lib/typ/elem*[*/vers*] [*?LAYOUT=layout-name*] [*&FONT=font*] [*&FONTSIZE=fontsize*] [*&PAGE=pagesize*] [*&MAXLINES=lines*] [*&MODE=mode*] [*&CODE=code*] [*&MARGIN-LEFT=left*] [*&MARGIN-TOP=top*] [*&BACKGROUND=background*] [*&AUTOSIZE=V | H | YES | NO*]

Entsprechend der "Browser-Syntax" wird der erste Parameter mit "?", die weiteren Parameter mit "&" eingeleitet. Die Parameter können in beliebiger Reihenfolge eingegeben werden. Falls Sie z.B. nur den Parameter PAGE und MODE benötigen, können Sie "pdf/file?page=a4&mode=sie" oder "pdf/file?mode=sie&page=a4" angeben.

Die PDF-Datei kann aus einer Datei mit und ohne Drucksteuerzeichen erzeugt werden. Die Standardwerte für die Aufbereitungs-Optionen können in der Parameterdatei mit den Parametern PDF-..... (S. 61) festgelegt werden. Die Optionen können auch bei der Funktion FSTAT (S. 81) angegeben werden. In diesem Fall werden die erzeugten URL-Adressen für die FSTAT-Liste um diese Werte als Parameter ergänzt.

Passwörter können mit der Anweisung VIEW-PASSWORD=SECRET (S. 70) maskiert werden.

filename Name einer BS2000-Datei. Beginnt der Dateinamen mit dem Zeichen "/" (absoluter Dateiname) oder dem String "**posix*:", erfolgt ein Zugriff auf das POSIX-Filesystem. Es ist in diesem Fall darauf zu achten, daß der POSIX-Pfad mit der Anweisung HTMDIR (S. 57) freigegeben wird. Ein relativer Pfad wird um das Verzeichnis ergänzt, daß mit der Anweisung DEFDIRPOS (S. 51) definiert ist.

Soweit ein BS2000-Dateiname keine User-ID enthält, wird er um die User-ID ergänzt, die in der Anweisung DEFDIRBS2 (S. 51) definiert ist.

lib/typ/elem[*/vers*]

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

Optionen für die Erzeugung der PDF-Datei:

Entsprechend der "Browser-Syntax" wird der erste Parameter mit "?", die weiteren Parameter mit "&" eingeleitet, z.B. *host/pdf/datei&font=courier&fontsize=8*. Die Reihenfolge der Parameter ist beliebig. Für alle Optionen können Standardwerte in der Parameterdatei in den Parametern PDF-.... (S. 61) definiert werden. Zusätzlich besteht die Möglichkeit, mit der Anweisung PDF-LAYOUT (S. 63) Standardwerte für verschiedene Anwendungsfälle zu definieren. Diese Werte können dann mit dem Parameter LAYOUT zugewiesen werden.

layout-name Standardwerte des PDF-Layouts *layout-name* verwenden (Parameter PDF-LAYOUT (S. 63)). Fehlt der Parameter LAYOUT, so gelten die Standardwerte der Parameter PDF-.... (S. 61). Diese Standardwerte können mit den folgenden Angaben der einzelnen Optionen überschrieben werden

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

Standard: Wert aus dem Parameter PDF-FONT bzw. PDF-LAYOUT in der Parameterdatei.

fontsize *width* [*/height*]

width Schriftgröße des Zeichensatzes für die Breite eines Zeichens.

- height* Schriftgröße für den Zeilenabstand. Damit ist es möglich die Zeilenbreite und Zeilenhöhe aus unterschiedlichen Font-Größen zu benutzen. Bei fehlender Angabe gilt die Fontsize aus *width*.
- Standard: Wert aus dem Parameter `PDF-FONTSIZE` bzw. `PDF-LAYOUT` in der Parameterdatei.
- pagesize* A4 | A4Q | A5 | A5Q | *breite x höhe*
- Seitengröße: A4 = DINA4, A4Q = DINA4 quer, A5 = DINA5, A5Q = DINA5 quer.
breite x höhe = Größe 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).
- Standard: Wert aus dem Parameter `PDF-PAGE` bzw. `PDF-LAYOUT` in der Parameterdatei.
- lines* 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.
- Standard: Wert aus dem Parameter `PDF-MAXLINE` bzw. `PDF-LAYOUTS` in der Parameterdatei.
- mode* ASA | SIE | NO [*:coll-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.
- :coll-col2:* Spaltenangabe. Für die PDF-Erzeugung werden nur Daten von Spalte *coll* bis *col2* benutzt. Bei *mode* SIE oder ASA muß das Druck-Steuerzeichen in *coll* 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 und 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' Orginalfontgröße 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 also auch implizit 13, 15, 17 und 1a00)

Standard: Wert aus dem Parameter PDF-MODE bzw. PDF-LAYOUT in der Parameterdatei

code

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	F B 4 F F D B B B C B D F F
Internationale Zeichen:	{ } [\] ~ ^

- DE Die Datei enthält Daten in 7-Bit-EBCIDC (CCSN = EDF03IRV, deutsche Variante), aus den entsprechenden EBCDIC-Zeichen sollen Umlaute (ÄÖÜäöüß) erzeugt werden.
- DK Die Datei enthält Daten in 7-Bit-EBCIDC (CCSN = EDF03IRV, dänische Variante), aus den entsprechenden EBCDIC-Zeichen sollen dänische Sonderzeichen (ÆØÅæøåüÜ) erzeugt werden.
- INT Die Datei enthält Daten in 7-Bit-EBCIDC (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. 50) zugewiesen wird.

Standard: Wert aus dem Parameter `PDF-CODE` bzw. `PDF-LAYOUT` in der Parameterdatei

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

Standard: Wert aus dem Parameter `PDF-MARGIN-LEFT` bzw. `PDF-LAYOUT` in der Parameterdatei

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

Standard: Wert aus dem Parameter `PDF-MARGIN-TOP` bzw. `PDF-LAYOUT` in der Parameterdatei

Beispiele:

```
pdf/liste1.lst?page=a4&fontsize=10&maxlines=50
pdf/liste1.lst?page=297x210&fontsize=10&maxlines=50
pdf/lib/p/liste2?font=h&page=a4&fontsize=9&mode=sie;nctrl
pdf/prog.src?mode=no&margin-left=25&margin-top=15
```

Hintergrundbilder und Wasserzeichen

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

Hier können Hintergrundbilder und ein Wasserzeichen angegeben werden.

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

bs2-file Name einer BS2000-Datei.

lib(typ,elem[,version]) Bibliothek, Elementtyp und Elementname.

/posix-file 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.

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

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

page-to 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

Das Wasserzeichen wird im Gegensatz zum Hintergrundbild immer auf allen Seiten gedruckt.

watermark <FS=*fontsize*>*watermark-text*

fontsize Schriftgröße für das Wasserzeichen: 1-256

watermark-text 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. Diese Option gilt als Standard, falls dieser Parameter nicht angegeben wird.

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).

Beispiele:

```
background=t1.jpg,10,10,,,1; t2.jpg,10,10,,,2-$EVEN
background=t1.jpg,10,10;t2.jpg,200,10;t3.jpg,20,250,200,30
background=t1.jpg;watermark:<fs=50>Test
```


Datei anzeigen

VIEW*filename* | *lib/typ/elem*[/*vers*] | *ZIP:*zip-archiv(elem)* [?TRANS=Y | N | *usercode*] [&VIEWTAG=Y|N]

Binärdatei anzeigen

XVIEW*filename* | *lib/typ/elem*[/*vers*] | *ZIP:*zip-archiv(elem)* [?MIME-TYPE=*mime-type*] [&EXT=*ext*] [&TRANS=Y | N | *usercode*] [&REC=*rectype*]

Entsprechend der "Browser-Syntax" wird der erste Parameter mit "?", die weiteren Parameter mit "&" eingeleitet. Die Parameter können in beliebiger Reihenfolge eingegeben werden. Falls Sie z.B. nur den Parameter TRANS und REC benötigen, können Sie "xview/file?trans=y&rec=dos" oder "xview/file?rec=dos&trans=y" angeben.

VIEW Die Datei wird im Text-Modus angezeigt.

Dateien mit der Endung HTM oder HTML bzw. mit den Endungen aus der Anweisungen HTMEXT (S. 58) werden ohne Veränderung angezeigt. Andere Daten werden vor der Anzeige um notwendige HTML-Anweisungen ergänzt.

Die HTML-Anweisungen für die Darstellung der z.B. Schriftart, Schriftgröße, Farbe, Kopf und Fuß der Seite, können in einer Template-Datei vorgegeben werden. Die Datei muß sich in der Default-User-ID bzw. in der Default-User-ID der Benutzergruppe befinden und muß *template.VIEW* heißen. Der Prefix *template* wird mit der Anweisung TEMPLATE (S. 68) definiert. Ein Muster der Datei befindet sich in der Bibliothek OPGCOM.LIB, Element D/DEMO.TEMPLATE.VIEW. Weitere Hinweise zur Template-Datei siehe S. 92.

Die Daten werden von EBCDIC nach ISO8859 konvertiert, soweit es sich nicht bereits um ASCII-Dateien (nur bei POSIX-Dateien möglich) handelt.

Sonderzeichen "<", ">" und "&"

VIEWTAG Empfängt der Browser die Zeichen "<", ">" oder "&", werden diese Zeichen nicht angezeigt sondern als HTML-Anweisungen interpretiert. Kommen in einer Datei, die mit VIEW angezeigt wird, diese Zeichen vor, können HTML-Anweisungen wahlweise angezeigt oder interpretiert werden.

Y Die Sonderzeichen werden maskiert, so daß sie unverändert angezeigt werden. Diese Einstellung gilt als Standard für Dateien, die **keine** HTML-Dateien sind. Die Standardeinstellung kann mit der Anweisung VIEWTAG (S. 70) geändert werden. HTML-Dateien werden anhand der Dateiendungen identifiziert. Die gültigen Dateiendungen (in der Regel HTML und HTM) werden mit der Anweisung HTMEXT (S. 58) definiert.

N Die Sonderzeichen werden unverändert an den Browser weiter gegeben und als HTML-Anweisungen interpretiert. Diese Einstellung gilt als Standard für Dateien, die HTML-Dateien sind.

XVIEW Die Datei wird binär angezeigt, d.h. sie wird mit Ausnahme der Änderungen aufgrund der Optionen TRANS und REC unverändert übertragen. In der Regel erfolgt die Anzeige der Daten mit einer Applikation, die aufgrund des Parameters MIME-TYPE oder EXT aktiviert wird, z.B. Acrobat Reader, Word usw. Die Daten werden vom Browser in eine temporäre Datei gespeichert und danach wird die entsprechende Anwendung aufgerufen. Falls für die Anwendung ein Plugin existiert, wird sie im Browser-Fenster aufgerufen. Ansonsten wird ein eigener Task erzeugt.

filename Name einer BS2000-Datei. Beginnt der Dateiname mit dem Zeichen "/" (absoluter Dateiname) oder dem String "*`posix:`", erfolgt ein Zugriff auf das POSIX-Filesystem. Es ist in diesem Fall darauf zu achten, daß der POSIX-Pfad mit der Anweisung `HTMDIR` (S. 57) freigegeben wird. Ein relativer Pfad wird um das Verzeichnis ergänzt, daß mit der Anweisung `DEFDIRPOS` (S. 51) definiert ist.

Soweit ein BS2000-Dateiname keine User-ID enthält, wird er um die User-ID ergänzt, die in der Anweisung `DEFDIRBS2` (S. 51) definiert ist.

lib/typ/elem[/vers]

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

*ZIP:*zip-archiv(elem)*

Name eines ZIP-Archivs und der Elementname.

Hinweis:

Dateien bzw. Bibliothekselemente mit der Endung `.bmp`, `.gif`, `jpg`, `jpeg`, `class` und `.png` werden automatisch binär mit der Funktion `XVIEW` angezeigt. Das Gleiche gilt für Endungen, die mit der Anweisung `XVIEWEXT` (S. 70) als Binärdateien definiert werden.

Mime-Type und Extension:

Im Windows-Explorer können Dateitypen definiert werden, die anhand der Datei-Extension oder anhand des Mime-Typen identifiziert werden können und mit einer Applikation verknüpft sind. Diese Dateitypen können auch von anderen Programmen, z.B. von Internet-Browsern, ausgewertet werden. Die Dateitypen werden in der Regel bei der Installation der entsprechenden Applikation erzeugt. Falls der URLServer den Dateityp nicht automatisch erkennen kann, muß er über die Parameter *mime-type* oder *ext* bestimmt werden.

Folgende Dateien werden automatisch mit dem zutreffenden Mime-Type gesendet:

- a) PDF-Dateien in EBCDIC- oder ASCII-Codierung werden anhand des Datei-Headers erkannt und mit dem Content-Type `application/pdf` gesendet.
- b) Mit dem Programm `OPGCRYPT` verschlüsselte Dateien werden ebenfalls automatisch erkannt und mit dem Content-Type `application/x-opgc` gesendet.
- c) Dateien mit der Endung `.DOC` werden mit dem Content-Type `application/msword` gesendet.
- d) Dateien mit der Endung `.XLS` werden mit dem Content-Type `application/x-msexcel` gesendet.
- e) Dateien mit der Endung `.PPT` werden mit dem Content-Type `application/x-mspowerpoint` gesendet.

Bei der Installation für URLServer werden durch das automatische Starten des Programms `SETUP.EXE` folgende zusätzliche Dateitypen registriert:

- `.EDTW` Datei für EDTW (Mime-Type `application/x-edtw`). Dieser Dateityp wird für die Funktion EDTW und EDTWBIN verwendet.
- `.PDFP` Datei mit Acrobat Reader auf Standarddrucker ausdrucken ohne Dialog (Mime-Type = `application/x-pdfp`).
- `.PDFD` Drucker-Dialog von Acrobat Reader anzeigen (Mime-Type = `application/x-pdfd`)

mime-type Mime-Type für die anzuzeigende Datei. Der Mime-Type wird vom URLServer als Content-Type im HTTP-Header eingetragen. Die entsprechende Applikation wird, falls ein Plugin vorhanden ist, im Browser gestartet, andernfalls als eigener Task.

ext Extension: Dateiendung des Dateinamens. Die Extension wird vom URLServer als Content-Disposition (`Content-Disposition: attachment; filename=file`) im HTTP-Header eingetragen. Die Angabe bewirkt, daß die Applikation immer außerhalb des Browsers gestartet wird, auch wenn ein Plugin vorhanden ist. Falls beide Angaben vorhanden sind, reagieren die verschiedenen Browser unterschiedlich und bevorzugen einen der beiden Parameter.

TRANS=Y | N | *usercode*

Damit kann die Übersetzung von EBCDIC nach ISO8859 gesteuert werden. Das Problem besteht darin, daß bestimmte internationale Sonderzeichen und deutsche Umlaute nicht 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
Internationale Zeichen:	{ } [\] ~ ^

Y Yes. Die Daten werden von EBCDIC nach ISO8859 übersetzt. Bezüglich der Umlaute gilt die Einstellung der Anweisung `CODE` (S. 49). Dies ist die Standard-Einstellung für die Funktion `VIEW`.

D[E] Deutsch. Die Daten werden von EBCDIC nach ISO8859 übersetzt. Die "7-Bit-Umlaute" werden in die ISO8859-Umlaute übersetzt.

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

I[NT] International. Die Daten werden von EBCDIC nach ISO8859 übersetzt. Die "7-Bit-Umlaute" werden in die Sonderzeichen "{ | } [\] ~" des Codes ISO8859 übersetzt.

N No. Die Daten werden nicht übersetzt. Dies ist die Standard-Einstellung für die Funktion `XVIEW`

usercode Name eines User-Codes. Die von der Standard-Codierung abweichenden Translate-Anweisungen sind in einer Datei definiert, die mit der Anweisung `CODEFILE` (S. 50) zugewiesen wird.

Bei der Funktion `VIEW` gilt als Standard `TRANS=Y`, bei der Funktion `XVIEW` gilt als Standard `TRANS=N`.

Bei der Funktion `PDF` (S. 84) gilt die Anweisung `PDF-CODE` (S. 61), da in diesem Fall die Umlaute bereits bei der Erzeugung der PDF-Datei nach der PDF-Syntax kodiert werden müssen. Bei der Anzeige dieser PDF-Dateien mit der Funktion `XVIEW` wird automatisch die Einstellung "International" verwendet, da sonst die Zeichen "[", "]" und "\" in den "PDF-Anweisungen" zerstört würden.

Ist der Parameter `REC` nicht angegeben wird, so wird `REC=DOS` als Standard benutzt. PDF-Dateien im EBCDIC-Format werden automatisch mit "TRANS=I" verarbeitet.

rectype

Soll für die Applikation, mit der die Datei angezeigt werden soll, die Satzstruktur erhalten bleiben, kann mit diesem Parameter die Struktur der Datei gesteuert werden. In der Regel ist dieser Parameter nur mit der Option TRANS sinnvoll, da bei Dateien, die im ISO8859-Format als PAM-Datei vorliegen, bereits die DOS-Satzendekennzeichen enthalten sind.

- NO Die Daten werden ohne zusätzliche Satzendekennzeichen gesendet. Dieser Wert wird bei "TRANS=N" als Standard benutzt.
- DOS Jeder Satz wird mit dem Satzende-Kennzeichen X'0D0A' abgeschlossen. Dieser Wert wird bei "TRANS=Y | D | I" als Standard benutzt.
- UNIX Jeder Satz wird mit dem Satzende-Kennzeichen X'0A' abgeschlossen.
- BS2 Jeder Satz wird mit dem Satzende-Kennzeichen X'15' abgeschlossen.
- LF Für jeden Datensatz wird ein Satzlängenfeld vorangestellt, das die Satzlänge einschließlich des Satzlängenfeldes enthält. Dieses Format kann vom Programm EDT für Windows verarbeitet werden. Dadurch ist es möglich Daten im EBCDIC-Code satzstrukturiert anzuzeigen. In den Daten können auch die Satzendezeichen X'0D' und X'0A' enthalten sein. Zur Kennzeichnung des Formats enthält die Datei einen Datei-Header mit dem String "EDTWFILELENGTHFIELDE". Der Header-String kann im Code EBCDIC oder ASCII codiert sein.

In folgenden Fällen werden die Daten bei fehlenden REC-Parameter automatisch mit folgenden Werten verarbeitet:

- DOS bei TRANS=Y.
- UNIX bei PDF-Dateien im EBCDIC-Format.
- LF bei Funktion EDTWBIN.

Beispiele:

```
view/$test.testdatei
view/:ttt:$test.testdatei
view//home/test/testdatei
view/*posix:testdatei
view/lib/s/testelem
view/lib/s/testelem/0001)
xview/lib/x/logo.gif
xview/testdatei?trans=y&mime-type=application/msword
xview/testdatei?trans=y&ext=xls
xview/testdatei?trans=y&ext=csv
xview/testdatei?trans=y&mime-type=application/x-msexcel
```

Template-Datei

In den Template-Dateien sind Format-Beschreibungen enthalten die dazu dienen, normale Text-Dateien in HTML-Dateien umzuwandeln.

Aufbau der Template-Datei

Eine Template-Datei besteht aus 4 Elementen:

- a) Parameter für das Layout der Dateienliste und die Auswahl der Spalten für die Template-Dateien `TEMPLATE.FSTAT`, `TEMPLATE.LIBTOC` und `TEMPLATE.LS`. Diese Parameter können auch in der Parameterdatei mit der Anweisung `FSTAT_par`, `LIBTOC_par` bzw. `LS_par` definiert werden. Soweit in der `TEMPLATE`-Datei Parameter fehlen bzw. keine `TEMPLATE`-Datei existiert, gelten die Angaben der Anweisungen `FSTAT_par`, `LIBTOC_par` bzw. `LS_par`.
- b) Beginn-Informationen, die vor den eigentlichen Daten ausgegeben werden sollen.
- c) Platzhalter "`<!--BEGIN>`" für die Daten der auszugebenden Datei.
- d) Ende-Informationen die nach den eigentlichen Daten ausgegeben werden sollen.

Parameter für `TEMPLATE.FSTAT`

Parameter für die Spaltenauswahl in der `FSTAT`-Liste:

<code>PAR BKL=Y N</code>	Backup-Class
<code>PAR FREE_PAGES=Y N</code>	Freie PAM-Seiten
<code>PAR FCB-TYPE=Y N</code>	Backup-Class A, B, C, D oder E
<code>PAR PAM_PAGES=Y N</code>	Anzahl freie PAM-Seiten
<code>PAR SEC.ALLOC=Y N</code>	Anzahl Second Allocation
<code>PAR SHARE=Y N</code>	Share =YES NO
<code>PAR ACCESS=Y N</code>	Access=WRITE READ
<code>PAR PASS=Y N</code>	Password = RE WR EX
<code>PAR EXT.=Y N</code>	Anzahl Extends
<code>PAR VOLUME=Y N</code>	Name Volume
<code>PAR SORT=NAME AGE SIZE</code>	Standard-Sortierreihenfolge
NAME =	Sortierung nach Dateinamen Diese Sortier-Option gilt als Standard
AGE =	Sortierung nach Datum und Uhrzeit absteigend
SIZE =	Sortierung nach Dateigröße absteigend

Die Sortierreihenfolge kann auch direkt bei der Funktion `FSTAT` (S. 81) mit dem Parameter `SORT` angegeben werden.

Parameter für Schriftart und Schriftgröße:

<code>PAR FONT=font</code>	Schriftart
<code>PAR FONTSIZE=size</code>	Schriftgröße

Parameter für `TEMPLATE.LIBTOC`

Parameter für die Spaltenauswahl in der `LIBTOC`-Liste:

<code>PAR CREATION-DATE=Y N</code>	Erstellungsdatum
<code>PAR ELEMENTSIZE=Y N</code>	Größe Element
<code>PAR MODIFICATION-DATE=Y N</code>	Änderungsdatum
<code>PAR TYPE=Y N</code>	Element-Typ
<code>PAR USER-DATE=Y N</code>	User-Date
<code>PAR VERSION=Y N</code>	Versionsnummer
<code>PAR SORT=TYPE/NAME AGE SIZE</code>	Standard-Sortierreihenfolge

TYPE =	Sortierung nach Typ, Elementname und Version Diese Sortier-Option gilt als Standard
NAME =	Sortierung nach Elementnamen und Version
AGE =	Sortierung nach Datum und Uhrzeit absteigend
SIZE =	Sortierung nach Dateigröße absteigend

Die Sortierreihenfolge kann auch direkt bei der Funktion `LIBTOC` (S. 81) mit dem Parameter `SORT` angegeben werden.

Parameter für Schriftart und Schriftgröße:

PAR FONT= <i>font</i>	Schriftart
PAR FONTSIZE= <i>size</i>	Schriftgröße

Parameter für `TEMPLATE.LS` (Posix-Dateien)

Parameter für die Spaltenauswahl in der `LIBTOC`-Liste:

PAR LINK=Y N	Anzahl der Links
PAR ATTR=Y N	Dateiattribute
PAR SIZE=Y N	Dateigröße
PAR DATE=Y N	Änderungsdatum/Uhrzeit
PAR GROUP=Y N	Gruppenname
PAR USER=Y N	Benutzername
PAR SORT=NAME AGE SIZE	Standard-Sortierreihenfolge
NAME =	Sortierung nach Typ (Verzeichnis/Datei) und Name Diese Sortier-Option gilt als Standard
AGE =	Sortierung nach Datum und Uhrzeit absteigend
SIZE =	Sortierung nach Dateigröße absteigend

Die Sortierreihenfolge kann auch direkt bei der Funktion `LS` (S. 81) mit dem Parameter `SORT` angegeben werden.

Parameter für Schriftart und Schriftgröße:

PAR FONT= <i>font</i>	Schriftart
PAR FONTSIZE= <i>size</i>	Schriftgröße

Variablen:

In den Daten können folgende Variablen enthalten sein, die während der Verarbeitung ersetzt werden:

&AUTHGROUP	Aktueller Gruppenname (S. 45)
&AUTHUSER	Aktueller Username (S. 45)
&DATE	Aktuelles Datum in der Form <code>dd/mm/yy</code>
&FILENAME	Name der aktuell zu verarbeitenden Datei
&LOGON	User-ID des URL-Servers
&TIME	Aktuelle Uhrzeit in der Form <code>hh:mm:ss</code>
&TSN	Aktuelle TSN des URL-Servers

Verarbeitung der Template-Datei

Die Verarbeitung der Template-Datei läuft wie folgt ab:

- a) Die Beginn-Informationen der Template-Datei werden ausgegeben. Template-Variablen werden substituiert. Die Parameter für die FSTAT- und LIBTOC-Liste werden gespeichert.
- b) Die Daten der anzuzeigenden Datei bzw. die Daten der FSTAT-Liste oder LIBTOC-Liste werden ausgegeben, nachdem Sie ggf. vorher von EBCDIC nach ISO8859 übersetzt worden sind. Bei der Funktion VIEW wird nach jedem Datensatz ein Zeilenumbruch mittels des Strings "
" eingefügt.
- c) Die Ende-Information der Template-Datei werden ausgegeben. Template-Variablen werden substituiert.

Suchreihenfolge und Name der Template-Datei

Die Datei wird, soweit eine Benutzergruppe angegeben ist, zuerst im Default-Directory (S. 51) gesucht und dann im Default-Directory für den nicht geschützten Bereich.

Der Name der Template-Datei ist wie folgt aufgebaut: *prefix.function*

prefix Prefix aus der Anweisung TEMPLATE, falls nicht vorhanden wird der String "TEMPLATE." benutzt.

function Funktionsname, z.B. VIEW.

Beispiel:

```
<HTML>
<TITLE>&FILENAME</TITLE>
<BODY>
<FONT color=blue><B>&FILENAME</B></FONT>
<HR>
<PRE>
<!--BEGIN>
</PRE>
<HR>
<small>Die Ausgabe erfolgte am &DATE um &TIME durch TSN &TSN im
Logon &LOGON</small>
</BODY>
</HTML>
```

Muster der Template-Dateien befinden sich in der Bibliothek OPGCOM.LIB, Elemente
D/DEMO.TEMPLATE.VIEW,
D/DEMO.TEMPLATE.FSTAT,
D/DEMO.TEMPLATE.LIBTOC.
D/DEMO.TEMPLATE.LS.

Schnittstelle User-Modul Funktion CGI-BIN

Der Modul wird vom URLServer nachgeladen und aufgerufen, wenn die Funktion CGI-BIN (S. 74) angegeben ist. Der Modul muß sich in der LMS-Bibliothek befinden, die mit der Anweisung CGILIB (S. 48) zugewiesen ist.

Dem Modul wird der String, der im Browser eingegeben wird, zur Verfügung gestellt. Über den Bereich CGIINFO sind weitere Informationen verfügbar. Die Daten, die vom Benutzermodul bereitgestellt werden sollen, können entweder in den Ausgabebereich übertragen werden (RC = 8 oder 16) und werden dann vom URL-Server an den Browser gesandt oder die Daten werden mit der Funktion SENDTCP direkt ausgegeben (RC = 0).

Der Modul wird entsprechend der Standard-Programmverknüpfungsregeln aufgerufen. Beim Aufruf des Moduls sind die Register wie folgt geladen:

(R1) A (PARAMETERLISTE)

Die Parameterliste enthält folgende Adressen:

A (INPUT)	Adr. Bereich mit Browser-Anforderung
A (LINPUT)	Adr. Länge gesamter Eingabebereich
A (LSTRING)	Adr. Länge Eingabe-String
A (OUTPUT)	Adr. Ausgabebereich (max. 64.000)
A (LOUTPUT)	Adr. Vollwort Länge Ausgabebereich
A (CGIINFO)	Adr. Bereich CGIINFO (s. Makro M/CGIINFO bzw. Header datei S/HTMSERV.H)

INPUT Dieser Bereich enthält den Teil der URL-Adresse, der nach dem HOST-Namen kommt. Falls neben dem Modulnamen noch Parameter übergeben werden, können diese Parameter mit der Funktion GETPARAM aus dem Eingabe-String gelesen werden. Die Benutzung dieser Funktion wird empfohlen, weil der Eingabestring kodiert sein kann, z.B. wenn Umlaute im Parameternamen oder im Wert vorkommen.

Beispiel:

Eingabe im Browser: `get cgi-bin/modul1?par1=wert1&par2=wert2`

Inhalt Eingabebereich: `cgi-bin/modul1?par1=wert1&par2=wert2`

LINPUT Vollwort mit der Länge des Eingabebereichs. Die Länge wird benötigt, falls im Eingabebereich mit RC 4 Daten zurückgegeben werden. Diese Option wird nur noch aus Kompatibilitätsgründen unterstützt und wurde durch die Option 8 und 16 ersetzt.

LSTRING Länge des Eingabestrings.

OUTPUT Ausgabebereich, in dem bei RC 8 und 16 die Daten für den Browser (max. Länge = 64.000) bereitgestellt werden.

LOUTPUT Länge der Daten, die in OUTPUT zurückgegeben werden (notwendig bei RC 8 und 16)

CGIINFO Dieser Bereich enthält weitere Informationen, wie z.B. den Hostnamen, Angaben zur Autorisierung, den Namen der CFSLIB und CGILIB sowie den Namen des Browsers (siehe Element S/HTMSERV.H, M/CGIINFO und Cobol-Musterprogramm S/HALLOCOB.COB).

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

Rückkehrcode(in Assembler = R15, in C = R1):

- | | |
|----|--|
| 0 | Daten wurden bereits mit SENDTCP ausgegeben. |
| 4 | Daten im Bereich INPUT ausgeben und nach ISO8859-1 konvertieren.
Diese Option wird nur noch aus Kompatibilitätsgründen unterstützt und wurde durch die Option 8 und 16 ersetzt. Im Ausgabebereich OUTPUT können mehr Daten zur Verfügung gestellt werden. Außerdem besteht die Möglichkeit, ASCII-Daten auszugeben. |
| 8 | Daten im Ausgabebereich OUTPUT nach ISO8859-1 konvertieren und ausgeben. |
| 16 | Daten im Ausgabebereich OUTPUT nicht konvertieren und ausgeben. |

Funktionen:

Das Format des Funktionsaufrufs ist in der Header-Datei S/HTMSERV.H bzw. M/HTMSERV für Assembler und C beschrieben.

- checkacc** Die Funktion prüft, ob ein Zugriff auf eine Datei in Abhängigkeit der Anweisungen FILEACCESS zulässig ist.
- writelog** Ausgabe eines Satzes in die Log-Datei.
- writeerr** Ausgabe eines Satzes in die Error-Datei.
- sendtcp** Senden eines Strings an den Browser. Wahlweise werden die Daten von EBCDIC nach ASCII übersetzt.
- getparam** Suchen eines Parameters im Eingabebereich und Übertragen in ein Benutzerfeld. Falls der Parameter kodiert ist, z.B. Umlaute, werden die Daten dekodiert zur Verfügung gestellt.

Beispiele

Folgende Beispiel-Module sind jeweils in 3 Varianten (Assemb, COBOL und C) in der Bibliothek OPGCOM.LIB enthalten:

S/HALLOA.SRC (Assemb), S/HALLOCOB.COB (COBOL) und S/HALLOCC.C (C):

In diesem Beispiel wird eine Nachricht im Ausgabebereich übergeben. Der Name des Rechners wird aus dem Bereich CGIINFO ermittelt.

S/ZEILENA.SRC (Assemb), S/ZEILENCO.COB (COBOL) und S/ZEILENC.C (C)

In diesem Beispiel werden variabel viele Zeilen mit SENDTCP ausgegeben. Der Parameter ANZ wird mit getparam ermittelt.

Schnittstelle User-Modul URLEXIT (Entry USREXPO)

Der Modul wird vom URLServer nach Abschluß einer Funktion aufgerufen, falls SYSDTA die Anweisung `POST_EXIT=Y` (S. 68) vorhanden ist. Der Modul muß sich in der LMS-Bibliothek befinden, die mit der Anweisung `CGILIB` (S. 48) zugewiesen ist. Ein Muster des Moduls befindet sich in der Plam-Bibliothek `OPGCOM.LIB`, Element `S/URLEXIT`. Der Modul kann dazu verwendet werden, um nach Abschluß einer Funktion zusätzliche Aktionen auszuführen, die im Funktionsumfang nicht enthalten sind oder aus Sicherheitsgründen von der Browser-Schnittstelle nicht zur Verfügung stehen sollen.

Dem Modul wird u.a. der String, der im Browser eingegeben wird, zur Verfügung gestellt. Über den Bereich `HTMGLOB` sind weitere Informationen verfügbar. Alle Funktionen, die im User-Modul für die Funktion `CGI-BIN` zur Verfügung stehen, wie z.B. **sendtcp**, **writelog**, **writeerr** usw. können auch hier aufgerufen werden, Beschreibung siehe S. 97.

Der Modul wird entsprechend der Standard-Programmverknüpfungsregeln aufgerufen. Beim Aufruf des Moduls sind die Register wie folgt geladen:

- (R1) `A (PARAMETERLISTE)`
 Die Parameterliste enthält folgende Adressen:
- | | |
|-------------------------------|---|
| <code>A (ERRMSG)</code> | Adr. Bereich String aus <code>LOGFILE</code> oder <code>ERRLOGFILE</code> |
| <code>A (INPUT)</code> | Adr. Bereich mit Browser-Anforderung |
| <code>A (EXIT-PAR)</code> | Adr. Bereich mit Exit-Parameter aus Anweisung <code>POST_EXIT</code> |
| <code>A (IP-Adr)</code> | Adr. Bereich mit IP-Adresse |
| <code>A (0) oder A (1)</code> | 0 = Verarbeitung OK, 1 = Fehler aufgetreten |
- `ERRMSG` Vollwort mit der Adresse eines Bereiches, der die Daten enthält, die vom URLServer in die Datei `LOGFILE` oder `ERRLOGFILE` geschrieben wurden.
- `INPUT` Dieser Bereich enthält den Teil der URL-Adresse, der nach dem `HOST`-Namen kommt. Falls neben dem Modulnamen noch Parameter übergeben werden, können diese Parameter mit der Funktion `GETPARAM` aus dem Eingabe-String gelesen werden. Die Benutzung dieser Funktion wird empfohlen, weil der Eingabestring kodiert sein kann, z.B. wenn Umlaute im Parameternamen oder im Wert vorkommen.
- Beispiel:
 Eingabe im Browser: `HOST1/cgi-bin/modul1?par1=wert1&par2=wert2`
 Inhalt Eingabebereich: `cgi-bin/modul1?par1=wert1&par2=wert2`
- `EXIT-PAR` Vollwort mit der Adresse eines Bereiches mit dem String *exit-param* aus der Anweisung `POST_EXIT` (S. 68).
- `IP-ADR` Vollwort mit der Adresse eines Bereiches, in dem die IP-Adresse steht.
- (R13) `A (SAVEAREA)`
 Register-Sicherstellungsbereich (18 Worte, `DC 18F'0'`), der vom Aufrufer erstellt werden muß. Der URL-Server sichert dort die Register.
- (R14) `A (RETURN)`
 Rücksprungadresse im rufenden Programm. Nach Ausführung der Funktionen wird das Programm an dieser Adresse fortgesetzt.
- (R15) `V (USREXPO)`

Rückkehrcode(in Assembler = R15, in C = R1):

- 0 Der Rückkehrcode wird nicht ausgewertet.
Ev. auftretende Fehler können mit der Funktion **writeerr** in die Error-Logdatei geschrieben werden.

8 PDFGEN - Generierung PDF

Aus einer beliebigen BS2000-Datei / PLAM-Element / ZIP-Element oder POSIX-Datei wird eine PDF-Datei im PAM-Format erzeugt. Falls die Ursprungsdatei Drucksteuerzeichen enthält (Siemens oder ASA), werden diese für den Zeilen- und Seitenvorschub benutzt. Ansonsten wird nach einer festgelegten Anzahl von Zeilen ein Seitenvorschub erzeugt.

Über Parameter können im Wesentlichen folgende Anpassungen vorgenommen werden:

- Seitengröße
- Seitenränder links und oben
- Schriftart und Schriftgröße, Zeilenabstand
- Angaben zur Code-Übersetzung von EBCDIC nach ASCII
- Hintergrundbilder und Wasserzeichen (Seite von/bis, gerade oder ungerade Seiten)

In den Daten werden zusätzliche Steuerzeichen für die Umschaltung der Schrift, Schriftgröße, Schriftattribute (Farbe, kursiv, fett, tiefgestellt, hochgestellt, Sperrschrift usw.) ausgewertet.

Aufruf der Funktion PDFGEN

Die PDF-Generierung kann über folgende Schnittstellen aufgerufen werden:

- a) Eigenständiges Programm PDFGEN: Die Parameter werden über SYSDTA eingelesen. Beispiel-Job siehe OPGCOM.LIB(J/PDFGEN).
- b) Unterprogramm PDFGENUP: Die Parameter werden über die Struktur PDFPAR.H bzw. den Makro PDFPAR übergeben.
- c) Programm URLServer, Funktion PDF (S. 84): Die Parameter werden über die Parameterdatei (Anweisung PDF-*param*) und/oder beim Aufruf über den Browser übergeben.
- d) Programm MAILS / MAILUP, Anweisung ATTACH mit MODE=PDF (s. Manual MAIL): In der Parameterdatei MAIL.PAR können Layouts definiert werden. Der Name des gewünschten Layouts wird dann bei der ATTACH-Anweisung angegeben.
- e) Aufruf als CFS-Unterprogramm in der Kommandozeile oder in einer CFS-Prozedur. Die Parameter werden von SYSDTA eingelesen.
Beispiel: (PDFGENUP, OPGCOM.LIB)
- f) Kommando CREATE-PDF-FILE: Die Funktion wird vom ASTI-Service (S. 17) OPGPDFGEN ausgeführt (siehe Prozedur J/SYSSPR.PDFGEN).

Aufruf mit SDF-Kommando CREATE-PDF-FILE

Das SDF-Kommando übergibt die Parameter dem ASTI-Service OPGPDFGEN. Die Standardwerte können in der Prozedur SYSPR.PDFGEN angepaßt werden.

CREATE-PDF-FILE	FILENAME= <i>file</i> ,	(S. 106)
	PDFFILE= <i>pdffile</i>	(S. 108)
	[, PAGE=A4 A4Q A5 A5Q <i>höhe</i> x <i>breite</i> in mm]	(S. 108)
	[, FONT=C[OURIER] H[ELVETICA] T[IMES]]	(S. 106)
	[, FONTSIZE= <i>width</i> [/ <i>height</i>]	(S. 106)
	[, MARGINLEFT= <i>left</i>]	(S. 107)
	[, MARGINTOP= <i>top</i>]	(S. 107)
	[, MAXLINES= <i>lines</i>]	(S. 107)
	[, MODE=ASA SIE <u>NO</u> [: <i>col1-col2</i> :] [;CH <i>channel</i>]]	(S. 107)
	[, CODE=D[E] I[NT] DK <i>usercode</i>]	(S. 104)
	[, AUTOISIZE=V H YES <u>NO</u>]	(S. 102)
	[, LICENSE= <i>license-string</i>]	(S. 107)
	[, VIEWPASSW= <u>YES</u> NO]	(S. 108)
	[, CODEFILE= <i>file</i> <i>lib(typ,elem)</i>]	(S. 105)
	[, BACKGROUND= <i>background</i>]	(S. 103)

Parameter PDFGEN / PDFGENUP

Die Parameter werden entweder über SYSDTA eingelesen bzw. über SDF-Parameter bereitgestellt oder müssen in den Parameterbereich PDFPAR übertragen werden. Im Parameterbereich enthalten alle Felder Zeichenfolgen mit Ausnahme des Feldes BACKGROUND bzw. BACKG. Die Anweisungen sind wie folgt aufgebaut:

- a) SYSDTA oder Parameterbereich: *name params* *oder*
 b) SDF-Parameter: *name=params*

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.

Hintergrundbilder und Wasserzeichen

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

Hier können Hintergrundbilder und ein Wasserzeichen angegeben werden.

jpg-file Name einer JPG-Datei mit dem Hintergrundbild. Folgende Formate sind zulässig:
bs2-file Name einer BS2000-Datei.
lib(typ,elem[, version]) Bibliothek, Elementtyp und Elementname.
/posix-file 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.

page-to Letzte Seite, auf der das Bild plziert werden soll. Die Angabe \$ bedeutet, bis zum Ende der Datei.

EVEN Das Bild soll nur auf den geraden Seiten plziert werden.

ODD Das Bild soll nur auf den ungeraden Seiten plziert werden.

Wasserzeichen

Das Wasserzeichen wird im Gegensatz zum Hintergrundbild immer auf allen Seiten gedruckt.

watermark <FS=*fontsize*>*watermark-text*

fontsize Schriftgröße für das Wasserzeichen: 1-256.
Standard: 60

watermark-text Text für das Wasserzeichen, max. 128 Bytes

Beispiele:

```
background t1.jpg,10,10,,280
background t1.jpg,10,10;t2.jpg,200,10;t3.jpg,20,250,200,30
background t1.jpg;watermark:<fs=50>Test
```

Code-Konvertierung

CODE D[E] | I[NT] | DK | *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 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	F B 4 F F D B B B C B D F F
Internationale Zeichen:	{ } [\] ~ ^

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

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

INT Die Datei enthält Daten in 7-Bit-EBCIDC (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. 105) zugewiesen wird.

Datei mit User-Codes

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

Als Standard werden EBCDIC-Zeichen mit der Translate-Tabelle EDF041 to ISO8859-1 (S. 116) in "ASCII-Zeichen" umgewandelt. Intern sind drei verschiedene Varianten vordefiniert:

Codename	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	ù	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. 116) 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 Translat-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, z.B. VIEW *file*?TRANS=*codename*, anzugeben.

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

ee EBCDIC-Zeichen hexadezimal

aa ASCII-Zeichen hexadezimal

remark Bemerkung

Die User-Codes können in der Anweisungen CODE (S. 104) angegeben werden.

Beispiel Codefile:

```

* Tranlate-Tabelle fuer deutsche Umlate 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 = |

```

Zeichensatz

FONT C[OURIER] | H[ELVETICA] | T[IMES]

Die Namen der Zeichensätze können bis auf 1 Byte abgekürzt werden.

Schriftgröße

FONTSIZE *width* [/height]

width Schriftgröße des Zeichensatzes für die Breite eines Zeichens.

height Schriftgröße für den Zeilenabstand. Damit ist es möglich die Zeilenbreite und Zeilenhöhe aus unterschiedlichen Font-Größen zu benutzen. Bei fehlender Angabe gilt die Fontsize aus *width*.

Eingabedatei

INPUT *filename* | *lib/typ/elem*[/vers] | *ZIP:*zip-archiv(elem)*

filename Name einer BS2000-Datei. Beginnt der Dateinamen mit dem Zeichen "/" (absoluter Dateiname) oder dem String "*posix:", erfolgt ein Zugriff auf das POSIX-Filesystem.

lib/typ/elem[/vers]

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

*ZIP:*zip-archiv(elem)*

Name eines ZIP-Archivs und des Elementnamens.

Lizenz-Nummer**LICENSE** *license-string*

Lizenz-Nummer für die unbeschränkte Nutzung. Ohne gültige Lizenz-Nummer läuft das Programm nur einen Monat im Test-Modus.

Seitenränder links und oben**MARGIN-LEFT** *left*

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

MARGIN-TOP *top*

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

Maximale Zeilenanzahl**MAXLINES** *lines*

lines 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.

Aufbereitungsmodus**MODE** ASA | SIE | NO[:*col1-col2*:] [;CH*channel=*line [;CH*channel=*line.....]]

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

Außer den Vorschubsteuerzeichen in Spalte 1 werden in den Daten folgende Format-Steuerzeichen berücksichtigt:

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 und danach abdruckbar 3 Bytes Oktalzeichen
 z.B.
 200 = EURO-Zeichen
 274 = 1/4-Zeichen
 275 = 1/2-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 also auch implizit 13, 15, 17 und 1a00)

Seitengröße

PAGE A4 | A4Q | A5 | A5Q | *höhe* × *breite* in mm

Seitengröße A4 = DINA4, A4Q = DINA4 quer, A5 = DINA5, A5Q = DINA5 quer oder *höhe* × *breite* in mm, jeweils dreistellig, z.B. 310×210.

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

Ausgabedatei

PDF *filename* Name einer BS2000-Datei für die Ausgabe der PDF-Daten.

Passwörter in BS2000-Prozeduren maskieren

VIEW-PASSWORD YES | NO

NO Passwörter, die in BS2000-Prozeduren vorkommen, werden mit dem String "P" überschrieben. Passwörter in SDF-Kommandos werden nur ausgewertet, wenn sie mit dem Schlüsselwort "PASSWORD=" angegeben sind. In folgenden Datensätzen werden Passwörter ausgewertet:

```
/LOGON user,account,password  
/PASS[WORD] password  
/E[ENTER] file,user,account,password  
/sonstigeKommandos .....PASS[WORD]=password  
password = 'string' | C'string' | H'x-string' | dezimalzahl
```

YES

Daten werden nicht auf vorkommende Passwörter überprüft.

Beispiel:

Datensatz: /LOGON USER1,ACC1,C'GEHEIM'

Anzeige: /LOGON USER1,ACC1,P

Schnittstelle Modul PDFGENUP

Die Funktion kann alternativ auch über das Kommando `CREATE-PDF-FILE` aufgerufen werden.

Beim Aufruf des Moduls sind die Register wie folgt geladen:

- (R1) A (PARAMETERLISTE)
Die Parameterliste enthält folgende Adressen:
A (PDFPAR) Adr. Bereich PDFPAR (pdfpar.h bzw. Makro PDFPAR)
- (R13) A (SAVEAREA)
Register-Sicherstellungsbereich (18 Worte, `DC 18F'0'`), der vom Aufrufer erstellt werden muß. Der URL-Server sichert dort die Register.
- (R14) A (RETURN)
Rücksprungadresse im rufenden Programm. Nach Ausführung der Funktionen wird das Programm an dieser Adresse fortgesetzt.
- (R15) V (PDFGENUP)

Rückkehrcode (Assembler = R15, C = R1):

0 OK, ohne Fehler.
>0 Adresse einer Fehlermeldung, die wie folgt aufgebaut ist:
 2 Byte Länge
 2 Byte X'4040'
 1 Byte Vorschubsteuerzeichen X'40'
 max. 256 Bytes Fehlermeldung, abgeschlossen mit X'00' für Ausgabe in C-Programm

Makro PDFPAR für die Parameterübergabe

name **PDFPAR** MF=L | D [, P=*prefix*]

Der Makro generiert einen Datenbereich oder eine DSECT mit den Parametern für den Modul PDFGENUP. Der Makro wird jeweils mit dem Längenequate PDFPARL beendet. Bis auf das Adressfeld PDFBACKG enthalten alle Felder Zeichenfolgen. Falls eine Zeichenfolge kürzer ist als die Maximallänge, kann sie wahlweise mit Blank oder X'00' beendet werden.

- name* Name der DSECT oder des Parameterbereichs
- MF = L Es wird nur der Datenbereich mit Feldnamen und Standardwerten generiert.
- MF = D Es wird eine DSECT mit Feldnamen generiert.
- prefix* Prefix in der Länge 1 bis 3 für die Feldnamen, Standard = PDF.

Folgende Felder und Standardwerte sind in dem Makro enthalten:

&P.HDR	DC	CL6'PDFPAR'	Header
&P.INPUT	DC	CL155' '	Anweisung INPUT (S. 106)
&P.PDF	DC	CL155' '	Anweisung PDF (S. 108)
&P.PAGE	DC	CL8'A4'	Anweisung PAGE (S. 108)
&P.FONT	DC	CL32'COURIER'	Anweisung FONT (S. 106)
&P.FONTS	DC	CL8'8'	Anweisung FONTSIZE (S. 106)
&P.TOP	DC	CL8'4'	Anweisung MARGIN-TOP (S. 107)
&P.LEFT	DC	CL8'4'	Anweisung MARGIN-TOP (S. 107)
&P.LINES	DC	CL8'64'	Anweisung MAXLINES (S. 107)
&P.MODE	DC	CL128'NO '	Anweisung MODE (S. 107)
&P.CODE	DC	CL32'INT'	Anweisung CODE (S. 104)
&P.AUTO	DC	CL4'YES'	Anweisung AUTOSIZE (S. 102)
&P.BACKG	DC	A(0)	Adr. von Anweisung BACKGROUND (S. 103)
&P.LICS	DC	CL8' '	Anweisung LICENSE (S. 107)
&P.VPSW	DC	CL4'YES'	Anweisung VIEW-PASSWORD (S. 108)
&P.CODF	DC	CL155' '	Anweisung CODEFILE (S. 105)
&P.HDRE	DC	CL6'PDFEND'	Header end
&P.PARL	EQU	*-&P.HDR	

Headerfile PDFPAR.H für die Parameterübergabe

Die Headerfile PDFPAR.H enthält die Struktur `pdfpar` und `#define`-Anweisungen zum Löschen der Struktur. Bis auf das Adressfeld `background` enthalten alle Felder Zeichenfolgen. Falls eine Zeichenfolge kürzer ist als die Maximallänge, kann sie wahlweise mit Blank oder `X'00'` beendet werden. Die Struktur kann wie folgt gelöscht werden:

```
struct pdfpar pdfpar1 = {CPDFPAR1,CPDFPAR2,CPDFPAR3,CPDFPAR4};
```

Folgende Felder und Standardwerte sind in `pdfpar.h` enthalten:

```
struct pdfpar
{
char hdrstr[6];           /* Header*/
char input[155];         /* Anweisung INPUT (S. 106)
char pdf[155];           /* Anweisung PDF (S. 108)
char page[8];            /* Anweisung PAGE (S. 108)
char font[32];           /* Anweisung FONT (S. 106)
char fontsize[8];       /* Anweisung FONTSIZE (S. 106)
char top[8];             /* Anweisung MARGIN-TOP (S. 107)
char left[8];            /* Anweisung MARGIN-TOP (S. 107)
char maxlines[8];       /* Anweisung MAXLINES (S. 107)
char mode[128];          /* Anweisung MODE (S. 107)
char code[32];           /* Anweisung CODE (S. 104)
char autosize[4];        /* Anweisung AUTOSIZE (S. 102)
char *background;       /* Adr. von Anweisung BACKGROUND (S. 103)
char license[8];         /* Anweisung LICENSE (S. 107)
char viewpassword[4];    /* Anweisung VIEW-PASSWORD (S. 108)
char codefile[155];     /* Anweisung CODEFILE (S. 105)
char hdrstre[6];        /* Header end*/
```

9 FSendSRV - Proxy-Server für FileSend und URLServer

Dateien, die über die Funktion EDTW des URLServers oder über das BS2000-Programm FILESEND zu einem PC gesendet und dann in den EDTW eingelesen werden, können nun auch über einen Proxy-Server empfangen und/oder zurückgesandt werden. Dadurch ist es möglich, Dateien zu einem PC zu senden, der im BS2000 nicht bekannt ist. Ebenfalls ist es möglich zum Empfangen im BS2000 statt dynamischer Ports nur einen zentralen Port zu verwenden.

BS2000-Programm FSendSRV

Das Server-Programm leitet Daten von FSEND an den externen Server (PC-FSendSRV oder FServer) weiter und empfängt die Daten von EDTW bzw. dem externen Server auf einem zentralen Port und sendet diese wiederum an FSEND zum Schreiben der Daten zurück.

Das Programm erhält über SYSDTA jeweils durch Blank getrennt folgende Parameter:

BS2PORT	Zentraler Port zum Empfangen der Daten vom PC
PCPORT	Port für das Programm FServer, das auf dem jeweiligen Benutzer-PC läuft und die Daten empfängt.
EXTSRV	Wahlweise Name oder IP-Adresse eines externen Servers, der zentral die Daten vom BS2000 entgegennimmt und an die Benutzer-PC's weiterleitet. Diese Programm kann entweder auf einem Windows- oder auf einem LINUX-PC laufen.
EXTPORT	Wahlweise PORT des externen Servers.

Beispiel (Musterprozedur J/FSENDSRV in der Plam-Bibliothek OPGCOM.LIB):

```
/PROC N, (&LIB=$OPGCOM.OPGCOM.LIB, -  
/      &BS2PORT='9438', -  
/      &PCPORT='9438', -  
/      &EXTSRV=' ', -  
/      &EXTPORT=' '-  
/      ), SUBDTA=&  
/SYSFILE SYSDTA=(SYSCMD)  
/START-PROG FROM-FILE=*MODULE (LIBRARY=&LIB, -  
/                                     ELEMENT-OR-SYMBOL=FSENDSRV, -  
/                                     PROGRAM-MODE=*ANY, RUN-MODE=*ADVANCED)  
&BS2PORT &PCPORT &EXTSRV &EXTPORT  
/ENDP
```


Windows/Linux-Programm FSendSRV

Das Server-Programm leitet Daten vom BS2000-Server FSendSRV an das Programm FServer auf dem Benutzer-PC weiter und empfängt die Daten von EDTW auf einem zentralen Port und sendet diese wiederum an den BS2000-Server FSendSRV zum Schreiben der Daten zurück.

Das Windows- oder Linux-Programm kann von unserem Web-Server <http://www.opg.de/download/opgcom.html> heruntergeladen werden. Der Download für Linux besteht nur aus einer ausführbaren Datei `fendsrv`.

Das Programm erhält über die Kommandozeile jeweils durch Blank getrennt folgende Parameter:

PORT Port zum Empfangen der Daten vom BS2000. Hier muß der gleiche Port angegeben werden, der beim Start des BS2000-Server im Parameter EXTPORT angegeben wurde.

Beispiel Aufruf FSendSRV auf Windows oder Linux:

```
FSENDSRV 9438
```

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 Browser eine Umcodierung 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:

Dänische Sonderzeichen	æ ø å Æ Ø Å ü Ü
Deutsche Umlaute:	ä ö ü Ä Ö Ü ß
Hexa-Wert	F8 4F FD BB BC BD FF
Internationale Zeichen:	{ } [\] ~ ^

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.

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

Die Code-Umwandlung von EBCDIC.DF.04-1 (CCSN: EDF041) nach ASCII ISO 8859-1 erfolgt 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

Die Code-Umwandlung von ASCII ISO 8859-1 nach EBCDIC.DF.04-1 (CCSN: EDF041) erfolgt 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	Ø	a	b	c	d	e	f	g	h	i	<<	>>	ð	ý	þ	±
90-9F	°	j	k	l	m	n	o	p	q	r	^a	°	æ	.	Æ	¤
A0-AF	µ	—	s	t	u	v	w	x	y	z	ı	¿	Đ	Ý	Ð	®
B0-BF	¢	£	¥	·	©	§	¶	¼	½	¾	¬	[\]	´	×
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	¹	û	ü	Û	ú	ÿ
E0-EF	Ù	÷	S	T	U	V	W	X	Y	Z	²	ô	ö	ò	ó	õ
F0-FF	0	1	2	3	4	5	6	7	8	9	³	{	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	^a	°	æ	.	Æ	¤
A0-AF	µ	~	s	t	u	v	w	x	y	z	j	¿	Đ	Ý	Ð	®
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	¹	û]	ù	ú	ÿ
E0-EF	\	÷	S	T	U	V	W	X	Y	Z	²	Ô	Û	Ò	Ó	Õ
F0-FF	0	1	2	3	4	5	6	7	8	9	³	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.

Stichwortverzeichnis

7

7-bit-Code [114](#)
 7-bit-Leitungscodes ISO646-IRV [49](#), [114](#)

8

8-bit-Codes [114](#)
 8-bit-Umgebung aktivieren [115](#)

A

Abschnitt ACTION [20](#)
 Abschnitt CONS [19](#)
 Abschnitt GROUP [20](#)
 Abschnitt SCAN [22](#)
 Abschnitt SNMP [25](#)
 Abschnitt VALIDATION [26](#)
 Aktivierung 8-bit-Umgebung [115](#)
 ASA [107](#)
 ASCII [114](#), [116](#), [120](#)
 ASTI-Service [17](#)
 ASTI-Service OPGFILESEND [30](#), [35](#), [36](#)
 ASTI-Service OPGPDFGEN [101](#)
 Aufruf PDFGEN [101](#)
 Ausgabedatei [108](#)
 AUTHFILE [45](#)

B

Beispiel EventScan [28](#)
 Benutzer-Codes [50](#), [105](#)
 Benutzermodul [96](#)
 Benutzermodul URLEXIT [99](#)
 Berechtigung zum EDTW-Update [54](#)
 Bibliothekselemente [81](#)
 Binäre Übertragung [70](#)
 Browser [71](#)
 BS2000-ZIP-Dateien - EDTWUPD - [78](#)
 BS2000-ZIP-Dateien - FILESEND - [36](#)

C

CCSN EDF03IRV [49](#), [114](#), [117](#)
 CCSN EDF041 [114](#), [118](#)
 CCSN EDF04DR [119](#)
 CCSN ISO8859 [114](#)
 CFS [74](#)
 CFSACCESS [48](#)
 CFS-Kommando *RUN(FSend) [38](#)
 CFS-Kommando *RUN(SYSOUT) [76](#)
 CFS-Kommando *RUN(TSend) [30](#)
 CFS-Kommando *RUN(URLOUT) [75](#)
 CFSLIB [49](#)
 CFS-Unterprogramm FSend [37](#)
 CFS-Unterprogramm TSend [30](#)
 CGIACCESS [47](#)
 CGI-BIN [74](#)
 CGILIB [48](#)
 checkacc [97](#)
 CODE [49](#)
 CODEFILE [50](#), [105](#)
 Code-Tabelle EBCDIC.DF.03 (CCSN EDF03IRV) [117](#)

Code-Tabelle EBCDIC.DF.04.1 (CCSN EDF041) [118](#)
 Code-Tabelle EBCDIC.DF.04-DRV (CCSN EDF04DR) [119](#)
 Code-Tabelle ISO 8859-1 [120](#)
 Code-Umwandlung [114](#)
 COMMUNITY [25](#)
 CREATE-PDF-FILE [17](#)
 CREATE-PDF-FILE - SDF-Kommando [101](#)
 CRYPT-FILE [17](#)

D

Dänische Sonderzeichen [49](#)
 Dateiformat EDTWUPD - [78](#)
 Dateiformat FILESEND - [36](#)
 Datenformat EDTWUPD - [78](#)
 Datenformat FILESEND - [36](#)
 DEFDIRBS2 [51](#)
 DEFDIRPOS [51](#)
 delta-gespeicherte LMS-Elemente - EDTWUPD - [78](#)
 delta-gespeicherte LMS-Elemente - FILESEND - [36](#)
 Deutsche Umlaute [49](#), [61](#), [91](#), [114](#)
 Drucksteuerzeichen [107](#)

E

EBCDIC [114](#), [116](#), [117](#), [118](#), [119](#)
 EBCDIC.DF.03 (CCSN EDF03IRV) [117](#)
 EBCDIC.DF.04.1 (CCSN EDF041) [118](#)
 EBCDIC.DF.04-DRV (CCSN EDF04DR) [119](#)
 EDF03IRV [114](#), [117](#)
 EDF041 [114](#), [118](#)
 EDF04DR [119](#)
 EDTW [77](#)
 EDTW - SDF-Kommando [36](#)
 EDTWUPD - Dateiformat [78](#)
 EDTWUPD - Datenformat [78](#)
 EDTWUPD - delta-gespeicherte LMS-Elemente [78](#)
 EDTWUPD - FILEACCESS [54](#)
 EDTWUPD - ISAM-Dateien [78](#)
 Eingabedatei [108](#)
 ENTER [80](#)
 ENTER_TIMEOUT [52](#)
 ENTERACCESS [51](#)
 ENTERPAR [26](#)
 ERRLOGFILE [59](#)
 Erweiterte Zeichensätze [114](#)
 Exit User-Modul URLEXIT [99](#)

F

FGG [52](#)
 FILEACCESS [53](#)
 FILESEND - Dateiformat [36](#)
 FILESEND - Datenformat [36](#)
 FILESEND - delta-gespeicherte LMS-Elemente [36](#)
 FILESEND - ISAM-Dateien [36](#)
 FILESEND - SDF-Kommando [35](#)
 FSend - CFS-Unterprogramm [37](#)
 FSend - Jobvariable OPGCOM.jobname [39](#)
 FSendSRV [112](#)
 FSTAT [81](#)
 FSTAT_par [55](#)
 Funktionen Benutzermodul [97](#)

G

getparam [97](#)
GROUP [20](#)

H

Headerfile PDFPAR.H [111](#)
Hintergrundbilder [66](#)
HTMDIR [57](#)
HTMEXT [58](#)
HTMINDEX [58](#)

I

IMON [17](#)
Input [108](#)
INSTALL [17](#)
Installation BS2000 [16](#)
Installation EventList [16](#)
Installation FSendSRV [113](#)
Installation FServer [16](#)
INTR-Kommando [19, 44](#)
ISAM-Dateien - EDTWUPD - [78](#)
ISAM-Dateien - FILESEND - [36](#)
ISO 8859-1 [120](#)
ISO646 [114](#)
ISO8859 [114, 120](#)
ISO-Code [114](#)

J

Jobvariable OPGCOM.jobname [32, 39](#)

L

LIBTOC [81](#)
LIBTOC_par [58](#)
License [26, 59, 107](#)
Linux - Programm FSendSRV [113](#)
Lizenz [26, 59, 107](#)
LMS-Elemente - EDTWUPD - [78](#)
LMS-Elemente - FILESEND - [36](#)
LOGFILE [59](#)
LS [81](#)
LS_par [59](#)

M

MAIL [21](#)
Makro PDFPAR [110](#)
Margin-Left [107](#)
Margin-Top [107](#)
maxlines [107](#)
Modul FSendSRV BS2000 [112](#)
Modul PDFGENUP [110](#)

N

NAC [26](#)
Net Administration Center [26](#)

P

Page [63, 108](#)
PAM-Dateien - EDTWUPD - [78](#)
PAM-Dateien - FILESEND - [36](#)

Parameter PDFGEN [102](#)
PASS [19, 20](#)
Password [108](#)
PDF [84, 108](#)
PDF-AUTOSIZE [59](#)
PDF-BACKGROUND [60](#)
PDF-CODE [61](#)
PDF-FONT [61](#)
PDF-FONTSIZE [61](#)
PDFGEN Aufruf [101](#)
PDFGEN Parameter [102](#)
PDFGENUP - Headerfile PDFPAR.H [111](#)
PDFGENUP - Makro PDFPAR [110](#)
PDFGENUP - Return-Code [110](#)
PDF-LAYOUT [63](#)
PDF-MARGIN-LEFT [62](#)
PDF-MARGIN-TOP [62](#)
PDF-MAXLINES [62](#)
PDF-MODE [62](#)
PDF-PAGE [63](#)
PORT [67](#)
POSIX-Dateien - EDTWUPD - [78](#)
POSIX-Dateien - FILESEND - [36](#)
POSIX-Dateienliste [81](#)
POST_EXIT [68](#)
Programm FSendSRV Windows/Linux [113](#)
Prosysserver [112](#)

R

Return-Code [97, 100](#)
Return-Code PDFGENUP [110](#)

S

Schnittstelle Benutzermodul [96](#)
Schnittstelle Benutzermodul URLEXIT [99](#)
Schnittstelle Modul PDFGENUP [110](#)
SDF-Kommando CREATE-PDF-FILE [101](#)
SDF-Kommando FILESEND [35](#)
SDF-Kommando SEND-EDTWFIL [36](#)
SDF-Kommando SEND-FILE [35](#)
SDF-Kommando SEND-TRAP [30](#)
SDF-Kommando TRAPSEND [30](#)
Seitengröße [63, 108](#)
Seitenrand links [107](#)
Seitenrand open [107](#)
SEND [22](#)
SEND-EDTWFIL [17](#)
SEND-EDTWFIL - SDF-Kommando [36](#)
SEND-FILE [17](#)
SEND-FILE - SDF-Kommando [35](#)
sendtcp [97](#)
SEND-TRAP [17](#)
SEND-TRAP - SDF-Kommando [30](#)
Server [112](#)
Service [17](#)
SIE [107](#)
SUBTASKS [68](#)
SYSDTA-Anweisungen [45](#)
SYSSDF.OPGCOM [16](#)
SYSSII.OPGCOM [16](#)
SYSSPR.CRYPT-FILE [17](#)
SYSSPR.PDFGEN [17](#)
SYSSPR.SEND-EDTWFIL [17](#)
SYSSPR.SEND-FILE [17](#)
SYSSPR.SEND-TRAP [17](#)

T

TEMPLATE [68](#)
Template-Datei [92](#)
Translate-Tabelle ASCII to EBCDIC [116](#)
Translate-Tabelle EBCDIC to ASCII [116](#)
Translate-Tabellen [50](#), [105](#)
TRAPSEND - SDF-Kommando [30](#)
TSend - CFS-Unterprogramm [30](#)
TSend - Jobvariable OPGCOM.jobname [32](#)

U

Umlaute [49](#), [61](#), [91](#), [114](#)
UPDATE mit EDTW [54](#)
URL-Adresse [73](#)
URL-Adresse variabel [73](#)
USERID [19](#)
USERLINK [68](#)
User-Modul [96](#)

V

Validation [26](#)
Variable URL-Adresse [73](#)

VIEW [69](#), [89](#)
VIEW-PASSWORD [70](#), [108](#)
VIEWTAG [70](#)

W

Wasserzeichen [60](#), [66](#)
Watermark [60](#), [66](#)
writeerr [97](#)
writelog [97](#)

X

XVIEW [89](#)
XVIEWEXT [70](#)

Z

Zeilenanzahl [107](#)
ZIP-Dateien BS2000 - EDTWUPD - [78](#)
ZIP-Dateien BS2000 - FILESEND - [36](#)