Deutsch

Fujitsu Software BS2000

SPOOL V4.9

Spool & Print Makros (Ergänzung)
Makros zur Konvertierung in PDF

Benutzerhandbuch

Stand der Beschreibung:
CONF2PDF V1.0B
SPOOL V4.9A

Ausgabe Juni 2017

Kritik… Anregungen… Korrekturen…

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

Sie können uns Ihre Kommentare per E-Mail an bs2000.info@fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2015

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001: 2015 erfüllt.

Copyright und Handelsmarken

Inhalt

Ergänzung zu „Spool & Print Makros und Exits“

Die Makros PDFCVT, PDFSPO und PDFDIR bieten die Funktionalität aus Textdateien PDF-Dateien zu erstellen. Mit dem Makro PDFTMP können PDF-Vorgabeseiten mit Hintergrundbildern für die Ausgabe mit dem Makro PDFCVT gestaltet werden.

Die Makros, die ursprünglich der Komponente SPOOLSYS zugeordnet waren, gehören jetzt zu der neuen Komponente CONV2PDF. Damit ändert sich der Name der Parameterdatei entsprechend.

Mit CONV2PDF erweitert sich der Makroumfang von BS2000/OSD ≥ V6.0.

Nachfolgend ist die Funktionalität dieser Makros als Erweiterung zu dem Handbuch „Spool & Print Makros und Exits“ beschrieben.

1 Makros zur Konvertierung in PDF

Dieses Dokument beschreibt die Programmschnittelle für die Konvertierung von Textdateien in PDF-Format. Die Schnittstelle ist nur für P1-Anwendungen verfügbar.

Die folgende Tabelle listet die Makros für die verschiedenen Programmiersprachen auf (SPL ist nur für interne Zwecke verfügbar):

Name des Makros	Funktion	Seite
PDFCVT (Assembler-Schnittstelle) PDFCVTC (CPP-Schnittstelle) PDFCVT.H (C-Include) PDFCVTY (COBOL-Schnittstelle) PDFCVTI (SPL-Schnittstelle intern)	Erzeugt eine PDF-Datei durch zeilenweise Konvertierung von Textdaten. Das Layout der PDF-Datei wird entweder durch SPOOL-Parameter (wie z.B. FORM und LOOP) oder durch direkte Angabe der Parameter festgelegt.	Seite 8
PDFDIR (Assembler-Schnittstelle) PDFDIRC (CPP-Schnittstelle) PDFDIR.H (C-Include) PDFDIRY (COBOL-Schnittstelle) PDFDIRI (SPL-Schnittstelle intern)	Legt die Layout-Parameter für das Makro PDFCVT direkt fest.	Seite 23
PDFSPO (Assembler-Schnittstelle) PDFSPOC (CPP-Schnittstelle) PDFSPO.H (C-Include) PDFSPOY (COBOL-Schnittstelle) PDFSPOI (SPL-Schnittstelle intern)	Definiert die Layout-Parameter für das Makro PDFCVT in Form von SPOOL-Parametern (analog zum Kommando PRINT-DOCUMENT).	Seite 35
PDFTMP (Assembler-Schnittstelle) PDFTMPC (CPP-Schnittstelle) PDFTMP.H (C-Include) PDFTMPY (COBOL-Schnittstelle) PDFTMPI (SPL-Schnittstelle intern)	Definiert ein Template zur Gestaltung von PDF-Seiten. Im Makro PDFCVT können vordefinierte Vorgabeseiten des Templates einzelnen PDF-Seiten zugewiesen werden.	Seite 46

Diese Makros sind in der Laufzeitbibliothek SYSPRG.CONV2PDF.010.RTE enthalten. Diese Bibliothek enthält auch die das LLM (Binde-Lade-Modul) PDFCVRT, um mit der aufrufenden Anwendung zu binden. PDFCVRT muss beim Binden explizit angegeben werden.

Hinweis

Für die COBOL-Umgebung müssen dieselben Regeln angewendet werden wie für die Spool&Print Makros, siehe Handbuch **Spool & Print - Makros und Exits (BS2000)**.

PDFCVT - Textdaten in PDF-Format konvertieren

Benutzergruppe: Nichtprivilegierter Benutzer
Programmiersprachen: Assembler, CPP, C, Cobol
Makrotyp: S

Das Makro PDFCVT bietet die Möglichkeit, Daten im Textformat zeilenweise in PDF-Format zu konvertieren und damit eine PDF-Datei zu erzeugen. Das Layout der PDF-Datei kann wahlweise über die direkte Angabe der Parameter (siehe Makro PDFDIR) oder über SPOOL-Parameter (siehe Makro PDFSPO) festgelegt werden. Die Seiten der PDF-Datei können wahlweise im Layout von Vorgabeseiten eines PDF-Templates (Definition siehe Makro PDFTMP) erstellt werden.

Zusätzlich können Lesezeichen für die PDF-Datei erzeugt werden.

Format (Assembler)

Operation	Operanden
PDFCVT	MF=C/D/E/L/M/I ,PREFIX=S / <name 1..1> ,MACID=PDF / <name 1..3> ,PARAM=<name 1..27> ,VARIANT=001 / <c-string 3..3> ,CALLER=*USER ,ACTION=*OPEN / *ADDTEXT / *ADDBOOKMARK / *ENDSECTION / *CLOSE / <var: enum-of action_set> ,OUTNAME=(*pointer*,*length*) *pointer*: *NONE / <var:pointer> / (reg:pointer) *length*: *STD / <integer:1..54> / <var: int:2> / (reg: int:2) ,WRMODE=*CREATE / *REPLACEONLY / *ANY / <var: enum-of write_mode_set>

Operation	Operanden
PDFCVT	CCS=*NONE / <var:pointer> / (reg:pointer) ,TRUNCAT = *YES / *NO / <var: enum-of truncation_set> ,TEMPLAT = *NONE / <var:pointer> / (reg:pointer) ,FFORMAT = *STD / *SAM / *PAM / <var:pointer> / (reg:pointer) ,TEXT=(*pointer*,*length*) *pointer*: *NONE / <var:pointer> / (reg:pointer) *length*: *STD / <integer:1..32767> / <var: int:2> / (reg: int:2) ,BOOKMARK=(*pointer*,*length*) *pointer*: *NONE / <var:pointer> / (reg:pointer) *length*: *STD / <integer:1..54> / <var: int:2> / (reg: int:2) ,NXTSECT = (*name*,*namelength*) *name*: *NONE / <var:pointer> / (reg:pointer) *namelength*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2) ,HANDLE=<var:pointer> / (reg:pointer) ,SPOPAR=*NONE / <var:pointer> / (reg:pointer) ,DIRPAR=*NONE / <var:pointer> / (reg:pointer)

Operandenbeschreibung

ACTION=
gibt an, welche Aktion durchgeführt werden soll. Es bestehen Abhängigkeiten zwischen der Aktion und den Angaben bei anderen Operanden, siehe Abschnitt „Abhängigkeiten der Operanden“ auf Seite 20.

ACTION=*OPEN
Öffnet eine PDF-Datei, Standardwert. Der Schreibmodus (neue Datei erzeugen oder Datei überschreiben) hängt davon ab, welcher Wert beim Operanden WRMODE angegeben wird.

ACTION=*ADDTEXT
Fügt eine Zeile zu der PDF-Datei hinzu. Die Zeile wird zunächst gesichert. Die PDF-Datei wird erst beim Schließen erzeugt (ACTION=*CLOSE).

ACTION=*ADDBOOKMARK
Fügt ein Lesezeichen zu der PDF-Datei hinzu. Das Lesezeichen wird zunächst gesichert. Die PDF-Datei wird erst beim Schließen erzeugt (ACTION=*CLOSE).

ACTION=*ENDSECTION
Beendet den aktuellen Seitenabschnitt der PDF-Datei. Die aktuelle Seite wird noch in den bisherigen Template-Abschnitt geschrieben. Für den nächsten Text gilt:
– Ohne Angabe eines Template-Abschnitts im Operanden NXTSECT wird der nächste Text in den nächsten Template-Abschnitt geschrieben. Falls kein weiterer Template-Abschnitt existiert, geht der Text verloren.
– Bei Angabe eines Template-Abschnitts im Operanden NXTSECT wird der nächste Text in den angegebenen Template-Abschnitt geschrieben. Wenn der angegebene Template-Abschnitt nicht existiert, wird die PDF-Erstellung beendet.

ACTION=*CLOSE
Alle gesicherten Texte und Lesezeichen werden in die PDF-Datei geschrieben und die Datei wird dann geschlossen.

ACTION=<var: enum-of action_set>
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*OPEN
2	*ADDTEXT
3	*ADDBOOKMARK
4	*CLOSE
5	*ENDSECTION

BOOKMARK=(*pointer*,*length*)
Gibt den Namen des Lesezeichens und die Länge des Lesezeichens an.

***pointer*: *NONE / <var: pointer> / (<reg: pointer>)**
Gibt das Lesezeichen an.

***pointer*: *NONE**
Kein Lesezeichen angegeben, Standardwert. In diesem Fall darf beim Operanden ACTION der Wert *ADDBOOKMARK nicht angegeben werden.

***pointer*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***length*: *STD / <integer 1..32767> / <var: int: 2> / (<reg: int:2>)**
Gibt die Länge des Lesezeichens an.

***length*: *STD**
Die Länge des Lesezeichens beträgt 80 Zeichen, Standardwert.

***length*: <integer 1..32767>**
Angabe eines ganzzahligen Wertes für die Länge des Lesezeichens.

***length*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Lesezeichens interpretiert wird.

CALLER=
Aufrufer des Makros.

CALLER=*USER
Aufrufer ist Anwender (TU).

CCS=
Name des Zeichensatzes, der für die PDF-Datei verwendet werden soll.

CCS=*NONE
Kein Zeichensatz angegeben. Standardmäßig wird der Zeichensatz EDF03IRV verwendet.

CCS = <var:pointer> / (reg:pointer)
Adresse eines Speicherplatzes, an dem der Name des Zeichensatzes hinterlegt ist (A(feld) oder Angabe eines Registers).

Hinweis

In der PDF-Datei werden nur die Zeichen verwendet, die der in der Code-Tabelle Windows CP1252 Partial (CCSN=WCP1252P) enthalten sind. Diese Zeichen entsprechen dem Zeichensatz CP1252 ohne Zeichen wie z.B. ¼, ½, ¾, ... Bitte beachten Sie, dass Zeichen aus EBCDF041 wie ¼, ½,... nicht korrekt dargestellt werden können.

Die Beschreibung von Windows CP1252 Partial finden Sie im Handbuch „**XHCS (BS2000)**“.

DIRPAR=
Gibt an, ob die Parameter mit den Layout-Informationen direkt bereitgestellt werden.

DIRPAR=*NONE
Die Parameter nicht direkt angegeben, Standardwert. In diesem Fall muss der Operand SPOPAR mit einem Wert ungleich *NONE versorgt werden.

DIRPAR=<var:pointer> / (reg:pointer)
Adresse einer Parameterliste, welche die Layout-Informationen der PDF-Datei enthält (A(feld) oder Angabe eines Registers). Diese Parameterliste wird mit dem Makro PDFDIR erzeugt, siehe Seite 23.

FFORMAT=
Bestimmt das Dateiformat der PDF-Datei.

Eine bereits bestehende PDF-Datei kann nachträglich mit dem SAM/PAM-Konverter in das andere Dateiformat konvertiert werden (Aufruf mit dem Kommando START-SAM-PAM-CONVERTER, siehe Handbuch „BS2ZIP“).

FFORMAT=*STD
Verwendet das Dateiformat, das in der Parameterdatei SYSPAR.CONV2PDF festgelegt ist, Standardwert. Die Parameterdatei wird an den folgenden Ablageorten gesucht (Suche in der angegebenen Reihenfolge):

1. Benutzerkennung des Aufrufers
2. Benutzerkennung TSOS

Falls keine Parameterdatei gefunden wird, gilt FFORMAT=*SAM.

Unter der Installationskennung von CONV2PDF finden Sie die Vorlage einer Parameterdatei mit dem Dateinamen SYSPAR.CONV2PDF.<version>.

FFORMAT=*SAM
Die PDF-Datei wird im Dateiformat SAM mit REC-FORM=U erstellt.

FFORMAT=*PAM
Die PDF-Datei wird im Dateiformat PAM mit BLOCK-CONTROL=NO erstellt.

FFORMAT=<var: enum-of wrmode_set>
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*STD
2	*SAM
3	*PAM

HANDLE=<var:pointer> / (reg:pointer)
Gibt beim Aufruf ACTION=*OPEN die Adresse eines 4 Bytes langen, auf Wortgrenze ausgerichteten Bereichs an, in dem der Identifikator des PDF-Konverters gespeichert wird. Dieser Identifikator muss bei jedem nachfolgenden Aufruf (ACTION=*ADDTEXT, *ADDBOOKMARK oder *CLOSE) angegeben werden.

MACID=PDF / '<name 1..3>'
Bestimmt das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates.

MF=C / D / E / L / M / I
Typ des Makroaufrufs. Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

NXTSECT=(*name*,*namelength*,*fname*,*fnamelength*)
Gibt den Namen des nächsten Template-Abschnitts an. Groß-/Kleinschreibung wird dabei nicht unterschieden. Der Abschnitt muss im PDF-Template definiert sein (siehe Makro PDFTMP).

***name*:*NONE / var:pointer> / (reg:pointer)**
Name des nächsten Template-Abschnitts.

***name*: *NONE**
Kein Template-Abschnitt angegeben, Standardwert. In diesem Fall wird der nächste Abschnitt des PDF-Templates angenommen.

***namelength*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2)**
Länge des Abschnittsnamens.

OUTNAME=(*pointer*,*length*)
Gibt den Namen der PDF-Datei und die Länge des Dateinamens an.

***pointer*: *NONE / <var: pointer> / (<reg: pointer>)**
Name der PDF-Datei, die erzeugt werden soll.

***pointer*: *NONE**
*NONE ist Standardwert und muss angegeben werden, wenn beim Operand ACTION ein Wert ungleich *OPEN angegeben wird.

***pointer*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***length*: *STD / <integer 1..54> / <var: int: 2> / (<reg: int:2>)**
Länge des Dateinamens.

***length*: *STD**
Die Länge des Dateinamens beträgt 54 Zeichen, Standardwert.

***length*: <integer 1..54>**
Angabe eines ganzzahligen Wertes für die Länge des Dateinamens.

***length*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Dateinamens interpretiert wird.

Beispiele

– PDFCVT OUTNAME=(A(VAR1),41)

  Der Name der PDF-Datei wird im Feld VAR1 hinterlegt. Von diesem Feld sind 41 Zeichen als Dateinamenslänge auszuwerten.

– PDFCVT OUTNAME=(A(VAR1))

  Der Name der PDF-Datei wird im Feld VAR1 hinterlegt. Von diesem Feld sind 54 Zeichen (Voreinstellung) als Dateinamenslänge auszuwerten.

– PDFCVT OUTNAME=(A(VAR1),VAR2)

  Der Name der PDF-Datei wird im Feld VAR1 hinterlegt. Die Länge des Dateinamens ist im Feld mit dem Namen VAR2 hinterlegt.

– PDFCVT OUTNAME=(*NONE,*STD)

  Voreinstellung: keine PDF-Datei angegeben, die Standard-Längenangabe wird ignoriert. Diese Angabe ist nur relevant, wenn bei ACTION ein Wert ungleich *OPEN angegeben wurde.

PARAM='<name 1..27>'
Bezeichnet die Adresse der Operandenliste (nur erlaubt bei MF-Format 2 und 3). Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

PREFIX=S / '<name 1..1>'
Bestimmt das erste Zeichen der Feldnamen und Equates.

SPOPAR=
Gibt an, ob die Parameter mit den Layout-Informationen der PDF-Datei in Form von SPOOL-Parametern bereitgestellt werden.

SPOPAR=*NONE
Die Parameter werden nicht per SPOOL-Parameter angegeben, Standardwert. In diesem Fall muss der Operand DIRPAR mit einem Wert ungleich *NONE versorgt werden.

SPOPAR=<var:pointer> / (reg:pointer)
Adresse einer Parameterliste, welche die Layout-Informationen der PDF-Datei in Form von SPOOL-Parametern enthält (A(feld) oder Angabe eines Registers). Diese Parameterliste wird mit dem Makro PDFSPO erzeugt, siehe Seite 35.

TEMPLAT=
Gibt an, ob ein PDF-Template zur Gestaltung der PDF-Seiten verwendet werden soll. Ein Template kann nur beim Öffnen der PDF-Datei (ACTION=*OPEN) vereinbart werden.

TEMPLAT=*NONE
Es wird kein PDF-Template verwendet.

TEMPLAT=<var:pointer> / (reg:pointer)
Adresse eines PDF-Templates, in dem die Parameter zur Gestaltung einer PDF-Seite definiert sind (A(feld) oder Angabe eines Registers). Das PDF-Template muss zuvor mit dem Makro PDFTMP erzeugt werden, siehe Seite 46.

TEXT=(*pointer,length*)
Textzeile, die zur PDF-Datei hinzugefügt werden soll, sowie Länge der Zeile.

> ***pointer*: *NONE / <var: pointer> / (<reg: pointer>)**
> Gibt die Textzeile an.
>
> ***pointer*: *NONE**
> keine Text angegeben, Standardwert. *NONE muss angegeben werden, wenn beim Operand ACTION ein Wert ungleich *ADDTEXT angegeben wird.
>
> ***pointer*: <var: pointer> / (<reg: pointer>)**
> Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem die Textzeile hinterlegt ist (A(feld) oder Angabe eines Registers).
>
> ***length*: *STD / <integer 1..32767> / <var: int: 2> / (<reg: int:2>)**
> Länge der Textzeile.

***length*: *STD**
Die Textzeile hat eine Länge von 80 Zeichen, Standardwert.

***length*: <integer 1..32767>**
Angabe eines ganzzahligen Wertes für die Länge der Textzeile.

***length*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge der Textzeile interpretiert wird.

TRUNCAT=
Gibt an, ob Datenzeilen, die über den rechten Seitenrand hinausgehen, abgeschnitten werden (siehe auch Festlegung des rechten Seitenrands im Operanden MARGINS).

TRUNCAT=*YES
Längere Datenzeilen werden abgeschnitten, Standardwert.

TRUNCAT=*NO
Längere Datenzeilen werden umbrochen. Der Zeilenumbruch erfolgt mit dem Wort, das über den Seitenrand hinausgeht. Dabei ist ein Wort eine Zeichenfolge, die von Leer-, Satz- oder Sonderzeichen begrenzt wird.

TRUNCAT=<var: enum-of truncation_set>
Gibt den Operandenwert indirekt über ein Feld mit konstantem Inhalt (EQUATE) an. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*YES
2	*NO

VARIANT=001 / <c-string 3..3>
Bezeichnet die Variante der generierten Parameterliste.

WRMODE=
Bestimmt den Schreibmodus für die zu erstellende PDF-Datei.

WRMODE=*CREATE
Die Datei wird neu erstellt, Standardwert. Wenn die Datei bereits existiert, dann wird die Operation mit einem Fehlercode abgelehnt.

WRMODE=*REPLACEONLY
Die Ausgabedatei muss bereits existieren und wird bei der Konvertierung überschrieben. Andernfalls wird die Operation mit einem Fehlercode abgelehnt.

WRMODE=*ANY
Die Ausgabedatei wird neu erstellt. Eine bereits existierende Datei wird überschrieben.

WRMODE=<var: enum-of wrmode_set>
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*CREATE
2	*REPLACEONLY
3	*ANY

Returncodes

SC1	Bedeutung
0	Kein Fehler
1	Parameterfehler
64	Aufruf korrigieren und erneut versuchen

SC2	Bedeutung
0	Kein Fehler
1	Fehler
2	Warnung

MC	Bedeutung
0	Erfolgreiche Verarbeitung
1	Ungültige Aktion
2	Name der Ausgabedatei nicht angegeben
3	Länge des Dateinamens ungültig
4	Ungültiger Schreibmodus
5	Ungültiger Identifikator (Operand HANDLE)
6	Keine Layoutparameter angegeben (weder in DIRPAR noch in SPOPAR)
7	Layoutparameter doppelt angegeben (sowohl in DIRPAR als auch in SPOPAR)
8	Fehler beim Erzeugen der Umgebung
9	Es existieren mehrere Ausgabedateien
10	Ungültiger Name der Ausgabedatei
11	Ausgabedatei existiert schon
12	Ausgabedatei muss schon existieren

MC	Bedeutung
13	Löschen der Datei fehlgeschlagen
14	Ungültiger Wert bei RECPART FIRST
15	Ungültiger Wert bei RECPART LAST
16	Ungültiger Bereich bei RECPART
17	Ungültiger Wert bei LINEPP
18	Ungültiger Wert bei LINESPACING
19	Ungültiger Wert bei CCPOS
20	Ungültiger Wert bei PRINTER TYPE
21	Ungültiger Wert bei LEFT MARGIN
22	Ungültiger Wert bei ROTATION
23	Ungültiger Wert bei MEDIA
24	Ungültiger Wert bei PAGESIZE HEIGHT
25	Ungültiger Wert bei PAGESIZE WIDTH
26	Ungültige PAGESIZE internal 1
27	Ungültige PAGESIZE internal 2
28	Ungültiger Wert bei LPI
29	Ungültiger Wert bei FONT NAME
30	Ungültiger Wert bei FONT STYLE
31	Ungültiger Wert bei FONT SIZE
32	Ungültiger TEXT Parameter
33	Ungültiger TEXT Länge
34	Ungültiger BOOKMARK Parameter
35	Ungültige BOOKMARK Länge
37	Ungültiger Wert bei RIGHT MARGIN
38	Ungültiger Wert bei TOP MARGIN
39	Ungültiger Wert bei BOTTOM MARGIN
40	Fehler bei Erzeugen der Arbeitsdatei
41	Interner Fehler bei ADDTEXT
42	Interner Fehler bei ADDBOOKMARK
43	Schreibfehler bei TEXT
44	Schreibfehler bei BOOKMARK
45	Fehler bei Löschen der Arbeitsdatei
46	Ungültige Aufruf-Reihenfolge

MC	Bedeutung
47	Version der Parameterliste ungültig
48	Ungültiger Wert im Operanden TRUNCAT
49	Schreibfehler bei ACTION=*ENDSECTION
50	Interner Fehler bei ACTION=*ENDSECTION
51	Ungültiger Wert bei TEMPLAT
52	Ungültige TEMPLAT Länge
99	Interner Fehler
101	PDF DMS Fehler
102	PDF: Fehler beim Öffnen
103	PDF-Datei existiert nicht
104	PDF-Datei existiert schon
105	PDF: interner Fehler
106	Erzeugen der PDF-Datei misslungen
111	Ungültiger Wert bei RECPART
113	PDF: ungültiger FCB-Typ
115	PDF: Dateiname abgeschnitten
117	PDF: Ungültige Seitengröße
118	SPOOL-System nicht geladen
119	Ungültiger CHAR-Parameter
121	FORM- und LOOP-Größe inkonsistent
122	FORM oder LOOP ungültig
123	Lesezeichen fehlt
208	Aktivierung des Trace misslungen
214	PDF: Zeile abgeschnitten
220	PDF: Zeilenüberlappung

Abhängigkeiten der Operanden

Die Angaben in den Operanden des PDFCVT-Makros hängen teilweise von der Aktion ab, die im Operand ACTION angegeben wurde. Die folgende Tabelle zeigt Abhängigkeiten zwischen der Angabe bei ACTION und den restlichen Operanden.

ACTION=	Pflichtoperanden	Optionale Operanden	Ignorierte Operanden
*OPEN	OUTNAME HANDLE entweder SPOPAR oder DIRPAR	WRMODE CCS TRUNCAT TEMPLAT	TEXT BOOKMARK
*ADDTEXT	HANDLE TEXT		OUTNAME WRMODE BOOKMARK SPOPAR DIRPAR TRUNCAT NXTSECT TEMPLAT
*ADDBOOKMARK	HANDLE BOOKMARK		OUTNAME WRMODE TEXT SPOPAR DIRPAR TRUNCAT NXTSECT TEMPLAT
*ENDSECTION	HANDLE	NXTSECT	OUTNAME WRMODE TEXT SPOPAR DIRPAR TRUNCAT TEMPLAT
*CLOSE	HANDLE		OUTNAME WRMODE TEXT BOOKMARK SPOPAR DIRPAR TRUNCAT NXTSECT TEMPLAT

Struktur-Layout von PDFCVT (Assembler)

```
.*********************************************************************
.*  BEGIN-INTERFACE   PDFCVT
.*
.*  TITLE             (/ pdfcvt /)
.*  NAME              PDFCVT
.*  DOMAIN            CONV2PDF
.*  LANGUAGE          ASS
.*  COPYRIGHT         (C) Fujitsu Technology Solutions 2016
.*                        ALL RIGHTS RESERVED
.*  COMPILATION-SCOPE USER
.*  INTERFACE-TYPE    CALL
.*  RUN-CONTEXT       TU
.*
.*  PURPOSE           (/ Text to PDF converter /)
.*
.*  SYNTAX            (/ Syntax Variant 1:
.*                           PDFCVT  MF = C|D|E|L|M
.*                         , PREFIX     = [S] | <name>
.*                         , MACID      = [PDF] | <name>
.*                         , PARAM      = <name 1..27>
.*                         , VARIANT    = <c-string_without_quotes 3..3> |
.*                                        default 001
.*                         , CALLER     = *USER |
.*                                        default *USER
.*                         , EQUATES    = [YES] | NO
.*                         , SPOPAR     = <var: pointer> |
.*                                        *NONE |
.*                                        default *NONE
.*                         , DIRPAR     = <var: pointer> |
.*                                        *NONE |
.*                                        default *NONE
.*                         , HANDLE     = <var: pointer>
.*                         , TEMPLAT    = <var: pointer>
.*                         , ACTION     = <var: enum-of:2 _ACTION_SET> |
.*                                        *OPEN |
.*                                        *ADDTEXT |
.*                                        *ADDBOOKMARK |
.*                                        *ENDSECTION |
.*                                        *CLOSE |
.*                                        default *OPEN
.*                         , WRMODE     = <var: enum-of:2 _WRMODE_SET> |
.*                                        *CREATE |
.*                                        *REPLACEONLY |
.*                                        *ANY |
.*                                        default *CREATE
```

```
.*                          , TRUNCAT    = <var: enum-of:1 _TRUNCATION_SET> |
.*                                         *YES |
.*                                         *NO |
.*                                         default *YES
.*                          , CCS        = <var: char 1..8> |
.*                                         *NONE |
.*                                         *USERDEFAULT |
.*                                         default *USERDEFAULT
.*                          , OUTNAME    = list(2):
.*                                         OUTNPTR: <var: pointer> |
.*                                                  *NONE |
.*                                                  default *NONE
.*                                         OUTNLEN: <var: int 0..65535> |
.*                                                  <integer 1..54> |
.*                                                  *STD |
.*                                                  default *STD
.*                          , NXTSECT    = list(2):
.*                                         SECNPTR: <var: pointer> |
.*                                                  *NONE |
.*                                                  default *NONE
.*                                         SECNLEN: <var: int 0..65535> |
.*                                                  <integer 1..8> |
.*                                                  *STD |
.*                                                  default *STD
.*                          , TEXT       = list(2):
.*                                         TXTPTR: <var: pointer> |
.*                                                 *NONE |
.*                                                 default *NONE
.*                                         TXTLEN: <var: int 0..65535> |
.*                                                 <integer 1..4000> |
.*                                                 *STD |
.*                                                 default *STD
.*                          , BOOKMRK    = list(2):
.*                                         BMRKPTR: <var: pointer> |
.*                                                  *NONE |
.*                                                  default *NONE
.*                                         BMRKLEN: <var: int 0..65535> |
.*                                                  <integer 1..54> |
.*                                                  *STD |
.*                                                  default *STD /)
.*
.*  REMARKS           (/ LID definition /)
.*
.*********************************************************************
```

PDFDIR - Layout der PDF-Datei direkt festlegen

Benutzergruppe: Nichtprivilegierter Benutzer
Programmiersprachen: Assembler, C, CPP, COBOL
Makrotyp: S

Mit dem Makro PDFDIR können Sie die Parameter für das Seitenlayout einer PDF-Datei in direkter Form festlegen und abspeichern. Die gespeicherten Einstellungen können beim Makro PDFCVT angegeben werden.

Format (Assembler)

Operation	Operanden
PDFDIR	MF=C/D/E/L/M/I ,PREFIX=S / <name 1..1> ,MACID=PDF / <name 1..3> ,PARAM=<name 1..27> ,VARIANT=001 / <c-string 3..3> ,RECPART=(*first*,*last*) *first*: 1 / <integer 1..32767> / <var: int:2> / (<reg: int:2>) *last*: *STD / <integer 1..32767> / <var: int:2> / (<reg: int:2>) ,LINESP=(*spacing*,*position*) *spacing*: *SPACE_1 / *SPACE_2 / *SPACE_3 / *BY_ASA_CONTROL / *BY_EBCDIC_CONTROL / *BY_IBM_CONTROL / <var: enum-of space_set:1> *position*: 1 / <integer 1..2040> / <var: int: 2> / (<reg: int:2>) ,PAGESZ=(*media*,*height*,*width*) *media*: *NONE / *A4 / *A4_LANDSCAPE / *A3 / *A3_LANDSCAPE / *A5 / *A5_LANDSCAPE / *A6 / *A6_LANDSCAPE / <var: enum-of pagesz_set:1> *height*: *NONE / <integer 2..2040> / <var: int:2> / (<reg: int:2> *width*: *NONE / <integer 2..2040> / <var: int:2> / (<reg: int:2>

Operation	Operanden
PDFDIR	,MARGIN=(*left*,*right*,*top*,*bottom*) *left*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *right*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *top*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *bottom*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) ,LPI=6 / <integer 3..24> / <var: int: 2> / (<reg: int:2>) ,FONT=(*name*,*style*,*size*) *name*: *COURIER / *HELVETICA / *TIMES / <var: enum-of fontname_set:1> *style*: *NORMAL / *BOLD / *ITALIC / *BOLD_ITALIC / <var: enum-of fontstyle_set:1> *size*: 8 / <integer 1..72> / <var: int:2> / (<reg: int:2>)

Operandenbeschreibung

FONT=(*name*,*style*,*size*)
Bestimmt die zu verwendende Schrift.

***name:* *COURIER / *HELVETICA / *TIMES / <var: enum-of fontname_set:1>**
Gibt den Namen der Schrift an.

***name:* *COURIER**
Es wird die Schrift Courier verwendet, Standardwert.

***name:* *HELVETICA**
Es wird die Schrift Helvetica verwendet.

***name:* *TIMES**
Es wird die Schrift Times verwendet.

***name:* <var: enum-of fontname_set:1>**
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

0	*COURIER
1	*HELVETICA
2	*TIMES

style: **<u>*NORMAL</u> / *BOLD / *ITALIC / *BOLD_ITALIC / <var: enum-of fontstyle_-set:1>**
Gibt den Schriftstil an.

style: **<u>*NORMAL</u>**
Der Text wird in normalem Schriftstil ausgegeben, Standardwert.

style: ***BOLD**
Der Text wird halbfett ausgegeben.

style: ***ITALIC**
Der Text wird kursiv ausgegeben.

style: ***BOLD_ITALIC**
Der Text wird halbfett und kursiv ausgegeben.

style: **<var: enum-of fontstyle_set:1>**
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

0	*NORMAL
1	*BOLD
2	*ITALIC
3	*BOLD_ITALIC

size: **<u>8</u> / <integer 1..72> / <var: int:2> / (<reg: int:2>)**
Gibt die Schriftgröße an.

size: **<u>8</u>**
Die Schriftgröße beträgt 8 Punkt (pt), Standardwert.

size: **<integer 1..72>**
Angabe eines ganzzahligen Wertes für die Schriftgröße in pt.

size: **<var: int:2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Schriftgröße interpretiert wird.

MACID=<u>PDF</u> / '<name 1..3>'
Bestimmt das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates.

MF=C / D / E / L / M / I
Typ des Makroaufrufs. Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

LINESP=(*spacing*, *position*)
Bestimmt die Anzahl der Zeilenvorschübe bzw. die Art der Steuerzeichenauswertung und die Position des Steuerzeichens.

spacing: **<u>*SPACE_1</u> / *SPACE_2 / *SPACE_3 / *BY_ASA_CONTROL / *BY_EBCDIC_CONTROL / *BY_IBM_CONTROL / *NO / <var: enum-of space_set:1>**
Gibt die Anzahl der Zeilenvorschübe bzw. die Art der Steuerzeichenauswertung an.

spacing: **<u>*SPACE_1</u>**
Der Text wird mit einem Abstand von einer Zeile ausgegeben, Standardwert.

spacing: ***SPACE_2**
Der Text wird mit einem Abstand von zwei Zeilen ausgegeben, d.h. nach dem Text wird eine Leerzeile eingefügt.

spacing: ***SPACE_3**
Der Text wird mit einem Abstand von drei Zeilen ausgegeben., d.h. nach dem Text werden zwei Leerzeilen eingefügt.

spacing: ***BY_ASA_CONTROL**
Der Inhalt des Vorschubsteuerzeichens (siehe ***position***) wird als ASA-Vorschubsteuerzeichen interpretiert.

spacing: ***BY_EBCDIC_CONTROL**
Der Inhalt des Vorschubsteuerzeichens (siehe ***position***) wird als EBCDIC-Vorschubsteuerzeichen interpretiert.

spacing: ***BY_IBM_CONTROL**
Der Inhalt des Vorschubsteuerzeichens (siehe ***position***) wird als IBM-Vorschubsteuerzeichen interpretiert.

spacing: **<var: enum-of space_set:1>**
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*SPACE_1
2	*SPACE_2
4	*SPACE_3
8	*BY_EBCDIC_CONTROL
16	*BY_ASA_CONTROL
32	*BY_IBM_CONTROL

position: **1 / <integer 1..2040> / <var: int: 2> / (<reg: int:2>)**
Nummer des Datenbytes, das als Vorschubsteuerzeichen ausgewertet wird. Bei Datensätzen variabler Länge werden die Felder, in denen die Länge steht, nicht zu den Daten gerechnet, d.h. nicht mitgezählt.

Dieser Parameter wird ignoriert, wenn für ***spacing*** einer der Werte *SPACE_1, *SPACE_2 oder *SPACE_3 angegeben wird.

position: **1**
Das erste Datenbyte einer Textzeile wird als Vorschubsteuerzeichen interpretiert, Standardwert.

position: **<integer 1..2040>**
Ganzzahliger Wert für die Position des Vorschubsteuerzeichens.

position:**<var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Position des Vorschubsteuerzeichens interpretiert wird.

LPI=
Definiert die Zeilendichte der PDF-Seite in Zeilen pro Zoll (lines per inch).

LPI=6
Die Zeilendichte beträgt 6 Zeilen pro Zoll, Standardwert.

LPI=<integer 3..24>
Ganzzahliger Wert für die Zeilendichte.

LPI=<var: int: 2> / (<reg: int:2>)
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Zeilendichte interpretiert wird.

MARGIN=(*left*,*right*,*top*,*bottom*)
Definiert die Abstände zu den Rändern der PDF-Seite. Die Werte werden in mm angegeben.

left**: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Abstand zum linken Seitenrand in mm.

left**: 20**
Der Abstand beträgt 20 mm, Standardwert.

left**: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***left*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum linken Rand interpretiert wird.

***right*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Abstand zum rechten Seitenrand in mm.

***right*: 20**
Der Abstand beträgt 20 mm, Standardwert.

***right*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***right*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum rechten Rand interpretiert wird.

***top*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Abstand zum oberen Seitenrand in mm.

***top*: 20**
Der Abstand beträgt 20 mm, Standardwert.

***top*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***top*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum oberen Rand interpretiert wird.

***bottom*: 20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Abstand zum unteren Seitenrand in mm.

***bottom*: 20**
Der Abstand beträgt 20 mm, Standardwert.

***bottom*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***bottom*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum unteren Rand interpretiert wird.

PAGESZ=(*media*,*height*,*width*)
Gibt die Größe einer PDF-Seite an.

***media*: *NONE / *A4 / *A4_LANDSCAPE / *A3 / *A3_LANDSCAPE / *A5 / *A5_-LANDSCAPE / *A6 / *A6_LANDSCAPE / <var: enum-of pagesz_set:1>**
Format der PDF-Seite.

***media*: *NONE**
Es wird kein Format vorgegeben, Standardwert. In diesem Fall muss die Größe der PDF-Seite mit den Parameter ***hight*** und ***width*** explizit festgelegt werden.

***media*: *A4 / *A4_LANDSCAPE / *A3 / *A3_LANDSCAPE / *A5 / *A5_LANDSCAPE / *A6 / *A6_LANDSCAPE**
Gibt ein vordefiniertes Seitenformat an:

Operandenwert	DIN-Format	Breite x Höhe (mm)
*A4	DIN-A4	210 x 297
*A4_LANDSCAPE	DIN-A4 quer	297 x 210
*A3	DIN-A3	297 x 420
*A3_LANDSCAPE	DIN-A3 quer	420 x 297
*A5	DIN-A5	148 x 210
*A5_LANDSCAPE	DIN-A5 quer	210 x 148
*A6	DIN-A6	105 x 148
*A6_LANDSCAPE	DIN-A6 quer	148 x 105

***media*: <var: enum-of pagesz_set:1>**
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

0	*NONE
1	*A4
2	*A4_LANDSCAPE
3	*A3
4	*A3_LANDSCAPE
5	*A5
6	*A5_LANDSCAPE
7	*A6
8	*A6_LANDSCAPE

***height*: *NONE / <integer 2..2040> / <var: int:2> / (<reg: int:2>**
Gibt die Seitenhöhe an, wobei Folgendes zu beachten ist:
– Wird im Parameter ***media*** ein vordefiniertes Seitenformat angegeben, dann ist hier *NONE Pflicht.
– Wenn im Parameter ***media*** kein vordefiniertes Seitenformat angegeben wird, dann muss hier ein Wert ungleich *NONE angegeben werden.

***height*: *NONE**
Keine Angabe der Seitenhöhe, Standardwert.

***height*: <integer 2..2040>**
Ganzzahliger Wert für die Seitenhöhe in mm.

***height*: <var: int:2> / (<reg: int:2>**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Seitenhöhe (in mm) interpretiert wird.

***width*: *NONE / <integer 2..2040> / <var: int:2> / (<reg: int:2>**
Gibt die Seitenbreite an, wobei Folgendes zu beachten ist:
– Wird im Parameter ***media*** ein vordefiniertes Seitenformat angegeben, dann ist hier *NONE Pflicht.
– Wenn im Parameter ***media*** kein vordefiniertes Seitenformat angegeben wird, dann muss hier ein Wert ungleich *NONE angegeben werden.

***width*: *NONE**
Keine Angabe der Seitenbreite, Standardwert.

***width*: <integer 2..2040>**
Ganzzahliger Wert für die Seitenbreite in mm.

***width*: <var: int:2> / (<reg: int:2>**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Seitenbreite (in mm) interpretiert wird.

PARAM='<name 1..27>'
Bezeichnet die Adresse der Operandenliste (nur erlaubt bei MF-Format 2 und 3). Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

PREFIX=S / '<name 1..1>'
Bestimmt das erste Zeichen der Feldnamen und Equates.

RECPART=(*first*,*last*)
Gibt an, ob eine Textzeile vollständig (Standardwert) oder nur zu einem bestimmten Teil zur PDF-Datei hinzugefügt wird. Mit dieser Angabe können z.B. der ISAM-Schlüssel oder Steuerzeichen in der PDF-Datei weggelassen werden.

Wurde bei LINESP einer der Werte *BY_ASA_CONTROL, *BY_EBCDIC_CONTROL oder *BY_IBM_CONTROL angegeben, dann wird das Vorschubsteuerzeichen gemäß der in LINESP(..., ***position)*** angegebenen Position ausgewertet und aus der Textzeile entfernt. Die Textzeile wird ohne Vorschubsteuerzeichen neu erzeugt. Die Angaben bei ***first*** und ***last*** bezieht sich daher auf die neu erzeugte Textzeile, siehe „Beispiele“ auf Seite 32.

Das Vorschubsteuerzeichen wird unabhängig von der Angabe bei ***first*** ausgewertet, solange der Wert für ***first*** kleiner ist als die Länge der Textzeile. Ist der angegebene Wert größer als die Länge der Textzeile, wird von dieser Zeile nichts in die PDF-Datei ausgegeben.

***first*: 1 / <integer 1..32767> / <var: int:2> / (<reg: int:2>)**
Gibt die Byte-Nummer (Satzspalte) an, ab der die Textzeile in die PDF-Datei ausgegeben wird. Die Bytes einer Textzeile sind von links nach rechts – mit 1 beginnend – durchnummeriert; ISAM-Schlüssel und Steuerzeichen sind Bestandteile einer Textzeile.

***first*: 1**
Die Ausgabe beginnt mit dem ersten Byte einer Textzeile, Standardwert.

***first*: <integer 1..32767>**
Ganzzahliger Wert für die Byte-Nummer (Satzspalte), ab der die Textzeile ausgegeben wird.

***first*: <var: int:2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Satzspalte interpretiert wird.

***last*: *STD / <integer 1..32767> / <var: int:2> / (<reg: int:2>)**
Gibt die Byte-Nummer des letzten Bytes an, das von einer Textzeile noch in die PDF-Datei ausgegeben wird.

Ist eine Textzeile länger als nach der Formulardefinition erlaubt, dann wird sie abgeschnitten.

***last*: *STD**
Gibt die Textzeile bis zum Ende aus, Standardwert.

***last*: <integer 1..32767>**
Ganzzahliger Wert für das letzte Byte, das von einer Textzeile noch ausgegeben wird.

***last*: <var: int:2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als letztes Byte interpretiert wird.

VARIANT=001 / <c-string 3..3>
Bezeichnet die Variante der generierten Parameterliste.

Beispiele

1. Die Textzeile lautet c123YYYYYYYYYYYYYYYYYYYY, wobei c das Vorschubsteuerzeichen an Position 1 darstellt.

Angabe bei PDFDIR	Text in der PDF-Datei
LINESP=(*BY_IBM_CONTROL,1) ,RECPART=(1,*STD)	123YYYYYYYYYYYYYYYYYYYY
LINESP=(*BY_IBM_CONTROL,1) ,RECPART=(4,*STD)	YYYYYYYYYYYYYYYYYYYY

2. Die Textzeile lautet 123cXXXXXXX9876543210, wobei c das Vorschubsteuerzeichen an Position 4 darstellt.

Angabe bei PDFDIR	Text in der PDF-Datei
LINESP=(*BY_IBM_CONTROL,4) ,RECPART=(1,*STD)	123XXXXXXX9876543210
LINESP=(*BY_IBM_CONTROL,4) ,RECPART=(4,*STD)	XXXXXXX9876543210
LINESP=(*BY_IBM_CONTROL,4) ,RECPART=(4,10)	XXXXXXX

Struktur-Layout von PDFDIR (Assembler)

```
.*****************************************************************
.*  BEGIN-INTERFACE   PDFDIR
.*
.*  TITLE             (/ pdfdir /)
.*  NAME              PDFDIR
.*  DOMAIN             CONV2PDF
.*  LANGUAGE           ASS
.*  COPYRIGHT          (C) Fujitsu Technology Solutions 2016
.*                         ALL RIGHTS RESERVED
.*  COMPILATION-SCOPE USER
.*  INTERFACE-TYPE     CALL
.*  RUN-CONTEXT        TU
.*
.*  PURPOSE           (/ PDF conversion based on direct parameters /)
.*
.*  SYNTAX            (/ Syntax Variant 1:
.*                           PDFDIR  MF = C|D|L|M
```

```
.*                          , PREFIX     = [S] | <name>
.*                          , MACID      = [DIF] | <name>
.*                          , VARIANT    = <c-string_without_quotes 3..3> |
.*                                         default 001
.*                          , CALLER     = *USER |
.*                                         default *USER
.*                          , EQUATES    = [YES] | NO
.*                          , RECPART    = list(2):
.*                                         FIRSTCH: <var: int 0..65535> |
.*                                                  <integer 1..32767> |
.*                                                  default: 1
.*                                         LASTCH: <var: int 0..65535> |
.*                                                 <integer 1..32767> |
.*                                                 *STD |
.*                                                 default *STD
.*                          , LINESP     = list(2):
.*                                          SPACING: <var: enum-of:1 _SPACE_SET>
.*                                                   *SPACE_1 |
.*                                                   *SPACE_2 |
.*                                                   *SPACE_3 |
.*                                                   *BY_ASA_CONTROL |
.*                                                   *BY_EBCDIC_CONTROL |
.*                                                   *BY_IBM_CONTROL |
.*                                                   default *SPACE_1
.*                                         CCPOS: <var: int 0..65535> |
.*                                                <integer 1..2040> |
.*                                                *STD |
.*                                                 default *STD
.*                          , PAGESZ     = list(3):
.*                                         MEDIA: <var: enum-of:1 _PAGESZ_SET> |
.*                                                 *A4 |
.*                                                 *A4_LANDSCAPE |
.*                                                 *A3 |
.*                                                 *A3_LANDSCAPE |
.*                                                 *A5 |
.*                                                 *A5_LANDSCAPE |
.*                                                 *A6 |
.*                                                 *A6_LANDSCAPE |
.*                                                 *NONE |
.*                                                 default *A4
.*                                         HEIGHT: <var: int 0..65535> |
.*                                                 <integer 2..2040> |
.*                                                 *NONE |
.*                                                 default *NONE
.*                                         WIDTH: <var: int 0..65535> |
.*                                                <integer 2..2040> |
.*                                                *NONE |
.*                                                default *NONE
```

```
.*                          , FONT       = list(3):
.*                                         NAME: <var: enum-of:1 _FONTNAME_SET>
.*                                                *COURIER |
.*                                               *HELVETICA |
.*                                               *TIMES |
.*                                               default *COURIER
.*                                        STYLE: <var: enum-of:1 _FONTSTYLE_SET
.*                                                 *NORMAL |
.*                                                 *BOLD |
.*                                                 *ITALIC |
.*                                                 *BOLD_ITALIC |
.*                                                 default *NORMAL
.*                                          SIZE: <var: int 0..65535> |
.*                                                <integer 1..72> |
.*                                                *STD |
.*                                                default: 8
.*                          , MARGIN     = list(4):
.*                                          LEFT: <var: int 0..65535> |
.*                                                <integer 0..2040> |
.*                                                default: 20
.*                                          RIGHT: <var: int 0..65535> |
.*                                                 <integer 0..2040> |
.*                                                 default: 20
.*                                          TOP: <var: int 0..65535> |
.*                                               <integer 0..2040> |
.*                                               default: 20
.*                                          BOTTOM: <var: int 0..65535> |
.*                                                  <integer 0..2040> |
.*                                                  default: 20
.*                          , LPI        = <var: int 0..65535> |
.*                                         <integer 3..24> |
.*                                         default: 6 /)
.*
.*  REMARKS           (/ LID definition /)
.*
.*********************************************************************
.*
.*  END-INTERFACE     PDFDIR.
.*********************************************************************
```

PDFSPO - Layout der PDF-Datei über SPOOL-Parameter festlegen

Benutzergruppe: Nichtprivilegierter Benutzer
Programmiersprachen: Assembler, C, CPP, COBOL
Makrotyp: S

Mit dem Makro PDFSPO können Sie die Parameter für das Seitenlayout einer PDF-Datei in Form von SPOOL-Parametern wie z.B. FORM, LOOP etc. festlegen und abspeichern. Die gespeicherten Einstellungen können beim Makro PDFCVT angegeben werden.

Format (Assembler)

Operation	Operanden
PDFSPO	MF=C/D/E/L/M/I ,PREFIX=S / <name 1..1> ,MACID=PDF / <name 1..3> ,PARAM=<name 1..27> ,VARIANT=001 / <c-string 3..3> ,RECPART=(*first*,*last*) *first*: 1 / <integer 1..32767> / <var: int:2> / (<reg: int:2>) *last*: *STD / <integer 1..32767> / <var: int:2> / (<reg: int:2>) ,LINEPP=*STD / <integer 1..32767> / <var: int: 2>/ (<reg: int:2>) ,LINESP=(*spacing*,*position*) *spacing*: *SPACE_1 / *SPACE_2 / *SPACE_3 / *BY_ASA_CONTROL / *BY_EBCDIC_CONTROL / *BY_IBM_CONTROL / <var: enum-of space_set:1> *position*: 1 / <integer 1..2040> / <var: int: 2> / (<reg: int:2>) ,FORM=*STD / <var: char: 6> / (<reg: char:6>) / <c-string: alphanum-name 1..6> ,LOOP=*STD / <var: char: 3> / (<reg: char:3>) / <c-string: alphanum-name 1..3>

Operation	Operanden
PDFSPO	,CHARSET=*STD / <var: char: 3> / (<reg: char:3>) / <c-string: alphanum-name 1..3> ,PRTYPE=*HP90 / *HP / *LP / <var: enum-of prtype_set:1> ,LEFTMAR=20 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) ,ROT=*NO / *YES / <var: enum-of rotation_set:1>

Operandenbeschreibung

CHARSET=
Gibt die Schriftart (Font) an, die für die PDF-Datei verwendet wird.
Unabhängig von der Angabe bei CHARSET wird in der PDF-Datei immer die Schriftart Courier verwendet. Es werden nur die Parameter WEIGHT, CHARACTER-STYLE und LINE-PER-INCH interpretiert.
Die Kombination von WEIGHT und CHARACTER-STYLE bestimmt den Schriftstil, der in der PDF-Datei verwendet wird:

WEIGHT	Schriftstil bei CHARACTER-STYLE =*ITALICS	Schriftstil bei CHARACTER-STYLE =andere Werte
*BOLD	*BOLD-ITALICS	*BOLD
andere Werte	*ITALICS	*NORMAL

CHARSET=*STD
Es wird der Standard-Font verwendet, der sich aus dem angegebenen Druckertyp (PRTYPE) und ggf. dem gewünschten Format (FORM) ergibt. Dieser Font kann mit SHOW-SPOOL-FORMS abgefragt werden.

CHARSET=<c-string: alphanum-name 1..3>
Name des Fonts, der für die PDF-Datei verwendet wird. Die Zeichenfolge muss in Hochkommas eingeschlossen werden.

CHARSET=<var: char: 3> / (<reg: char:3>)
Name eines Feldes, das mit CL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Zeichenfolge (Länge 3 Byte) hinterlegt, die als Name des Fonts interpretiert wird.

FORM=
Bezeichnet das Papierformat, das für die Ausgabe verwendet werden soll (z.B. STD, STDSF1, STDWA4). Für alle Druckertypen müssen Standardformate in der SPOOL-Parameterdatei definiert sein.
Mit SHOW-SPOOL-FORMS können Sie sich die Einträge auf SYSOUT ausgeben lassen. Er werden nur die Formate für die Druckertypen HP, HP90 and LP unterstützt. Der Druckertyp muss im Operanden PRTYPE angegeben werden, damit das korrekte Format verwendet wird.

FORM=*STD
Standardformat, Voreinstellung.

FORM='<c-string: alphanum-name 1..6>'
Name des Formats, mit dem der SPOOLOUT-Auftrag verarbeitet werden soll. Implizit wird mit der Formatangabe ein Loop benannt. Wenn der Operand ROT=*YES angegeben ist, wird implizit auch ein Rotations-Loop für die Seitendrehung benannt.
Der zugeordnete Loop (bzw. die PAGEDEF und FORMDEF) muss in der Druckersteuerdatei $SYSSPOOL.PRFILE enthalten sein.
Der über den FORM-Operanden implizit benannte Loop bzw. Rotations-Loop wird ignoriert, wenn zugleich der Operand LOOP angegeben wird.
Ohne die Operanden FORM und LOOP wird die PDF-Datei mit dem für den jeweiligen Druckertyp eingetragenen Standardformular erstellt.
Ein bei dem Operanden LOOP explizit angegebener Loop muss die gleiche Länge haben wie der dem verwendeten Formular zugeordnete Loop.

FORM=<var: char: 6> / (<reg: char:6>)
Name eines Feldes, das mit CL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Zeichenfolge (Länge 6 Byte) hinterlegt, die als Name des Formulars interpretiert wird.

MACID=PDF / '<name 1..3>'
Bestimmt das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates.

MF=C / D / E / L / M / I
Typ des Makroaufrufs. Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

LEFTMAR=
Gibt an, ob der Ausgabetext eingerückt wird. Der Wert wird in mm angegeben.

LEFTMAR=20
Die Einrückung beträgt 20mm, Standardwert.

LEFTMAR=<integer 0..2040>
Ganzahliger Wert für die Einrückung in mm.

LEFTMAR=<var: int: 2> / (<reg: int:2>)
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Einrückung interpretiert wird.

LINEPP=
Legt die Anzahl der Zeilen fest, die eine Seite der PDF-Datei enthalten darf.

LINEPP=*STD
Wird kein Angabe gemacht (Standardwert), dann errechnet sich die Zahl der Zeilen pro PDF-Seite nach folgender Formel:

Zeilenanzahl = P * Z - A - 6

Dabei bedeuten:
P = Papiergröße in Zoll
Z = Zeilendichte
A = Anzahl der Zeilen vor erstem Vertikaltabulator "Kanal 1"

Hinweis

- Der Vertikaltabulator "Kanal 1" legt die Zeile für den Beginn der PDF-Seite fest. Standardmäßig werden 2 Leerzeilen vor Druckbeginn eingestellt, d.h. im Loop steht "Kanal 1" (CHANNEL 01) in der dritten Zeile.
- Ist der beim Operanden LINEPP angegebene Wert größer als die angegebene Zeilenanzahl im Loop, wird der im Loop vorgegebene Wert genommen.
- Wenn LINEPP zusammen mit dem Operanden LINESP=*SPACE_1, *SPACE_2 oder *SPACE_3 angegeben wird, dann muss der ei LINEPP angegebene Wert mindestens dreimal so groß sein wie der bei LINESP angegebene Zeilenvorschub, .
- Der Parameter LINEPP wird ignoriert, falls bei LINESP einer der Werte *BY-EBCDIC-CONTROL, *BY-ASA-CONTROL oder *BY-IBM-CONTROL angegeben wird.

LINEPP=<integer 1..32767>
Anzahl der Zeilen auf einer Seite.

LINEPP=<var: int: 2>/ (<reg: int:2>)
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Anzahl der Zeilen einer Seite interpretiert wird.

LINESP=(*spacing*, *position*)
Bestimmt die Anzahl der Zeilenvorschübe bzw. die Art der Steuerzeichenauswertung und die Position des Steuerzeichens.

***spacing:* *SPACE_1 / *SPACE_2 / *SPACE_3 / *BY_ASA_CONTROL / *BY_EBCDIC_CONTROL / *BY_IBM_CONTROL / *NO / <var: enum-of space_set:1>**
Gibt die Anzahl der Zeilenvorschübe bzw. die Art der Steuerzeichenauswertung an.

***spacing:* *SPACE_1**
Der Text wird mit einem Abstand von einer Zeile ausgegeben, Standardwert.

***spacing:* *SPACE_2**
Der Text wird mit einem Abstand von zwei Zeilen ausgegeben.

***spacing:* *SPACE_3**
Der Text wird mit einem Abstand von drei Zeilen ausgegeben.

***spacing:* *BY_ASA_CONTROL**
Der Inhalt des Vorschubsteuerzeichens (siehe ***position***) wird als ASA-Vorschubsteuerzeichen interpretiert.

***spacing:* *BY_EBCDIC_CONTROL**
Der Inhalt des Vorschubsteuerzeichens (siehe ***position***) wird als EBCDIC-Vorschubsteuerzeichen interpretiert.

***spacing:* *BY_IBM_CONTROL**
Der Inhalt des Vorschubsteuerzeichens (siehe ***position***) wird als IBM-Vorschubsteuerzeichen interpretiert.

***spacing:* <var: enum-of space_set:1>**
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*SPACE_1
2	*SPACE_2
4	*SPACE_3
8	*BY_EBCDIC_CONTROL
16	*BY_ASA_CONTROL
32	*BY_IBM_CONTROL

***position:* 1 / <integer 1..2040> / <var: int: 2> / (<reg: int:2>)**
Nummer des Datenbytes, das als Vorschubsteuerzeichen ausgewertet wird. Bei Datensätzen variabler Länge werden die Felder, in denen die Länge steht, nicht zu den Daten gerechnet, d.h. nicht mitgezählt.

Dieser Parameter wird ignoriert, wenn für ***spacing*** *SPACE_1, *SPACE_2, *SPACE_3 oder *NO angegeben wird.

position: **1**
Das erste Datenbyte einer Textzeile wird als Vorschubsteuerzeichen interpretiert, Standardwert.

position: **<integer 1..2040>**
Ganzzahliger Wert für die Position des Vorschubsteuerzeichens.

position: **<var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Position des Vorschubsteuerzeichens interpretiert wird.

LOOP=
Name des Loops, der verwendet wird. Der Loop-Name darf die Zeichen '$', '&' und '@' nicht enthalten.

LOOP=*STD
Die Vorschubsteuerung für die PDF-Datei wird mit dem Standard-Loop des verwendeten Formulars realisiert, Standardwert.

LOOP=<c-string: alphanum-name 1..3>
Name des Loops, der den Vorschub steuern soll. Die Länge des angegebenen Loops muss mit der Länge des Standard-Loops des verwendeten Formats übereinstimmen.
Loops sind in der Druckersteuerdatei PRFILE gespeichert. Wird kein Loop angegeben, dann werden die impliziten Angaben beim Operanden FORM benutzt.

Ohne Angabe der Operanden FORM oder LOOP werden Standardwerte eingesetzt.

LOOP=<var: char: 3> / (<reg: char:3>)
Name eines Feldes, das mit CL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Zeichenfolge (Länge 3 Byte) hinterlegt, die als Name des Loops interpretiert wird.

PARAM='<name 1..27>'
Bezeichnet die Adresse der Operandenliste (nur erlaubt bei MF-Format 2 und 3). Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

PREFIX=S / '<name 1..1>'
Bestimmt das erste Zeichen der Feldnamen und Equates.

PRTYPE=
Bestimmt den Druckertyp, dessen Format für das Layout der PDF-Datei verwendet wird.

PRTYPE=*HP90
Es wird das Format für den Druckertyp HP90 verwendet, Standardwert.

PRTYPE=*HP
Es wird das Format für den Druckertyp HP verwendet.

PRTYPE=*LP
Es wird das Format für den Druckertyp LP verwendet.

PRTYPE=<var: enum-of prtype_set:1>
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*HP90
2	*HP
3	*LP

RECPART=(*first*,*last*)
Gibt an, ob eine Textzeile vollständig (Voreinstellung) oder nur zu einem bestimmten Teil zur PDF-Datei hinzugefügt wird. Mit dieser Angabe können z.B. der ISAM-Schlüssel oder Steuerzeichen in der PDF-Datei weggelassen werden.

Wurde bei LINESP einer der Werte *BY_ASA_CONTROL, *BY_EBCDIC_CONTROL oder *BY_IBM_CONTROL angegeben, dann wird das Vorschubsteuerzeichen gemäß der in LINESP(..., ***position)*** angegebenen Position ausgewertet und aus der Textzeile entfernt. Die Textzeile wird ohne Vorschubsteuerzeichen neu erzeugt. Die Angaben bei ***first*** und ***last*** beziehen sich daher auf die neu erzeugte Textzeile, siehe „Beispiele“ auf Seite 43.

Das Vorschubsteuerzeichen wird unabhängig von der Angabe bei ***first*** ausgewertet, solange der Wert für ***first*** kleiner ist als die Länge der Textzeile. Ist der angegebene Wert größer als die Länge der Textzeile, wird von dieser Zeile nichts in die PDF-Datei ausgegeben.

***first*: 1 / <integer 1..32767> / <var: int:2> / (<reg: int:2>)**
Gibt die Byte-Nummer (Satzspalte) an, ab der die Textzeile in die PDF-Datei ausgegeben wird. Die Bytes einer Textzeile sind von links nach rechts – mit 1 beginnend – durchnummeriert; ISAM-Schlüssel und Steuerzeichen sind Bestandteile einer Textzeile.

***first*: 1**
Die Ausgabe beginnt mit dem ersten Byte einer Textzeile, Standardwert.

***first*: <integer 1..32767>**
Ganzzahliger Wert für die Byte-Nummer (Satzspalte), ab der die Textzeile ausgegeben wird.

***first*: <var: int:2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Satzspalte interpretiert wird.

***last*: *STD / <integer 1..32767> / <var: int:2> / (<reg: int:2>)**
Gibt die Byte-Nummer des letzten Bytes an, das von einer Textzeile noch in die PDF-Datei ausgegeben wird.

Ist eine Textzeile länger als nach der Formulardefinition erlaubt, dann wird sie abgeschnitten.

***last*: *STD**
Gibt die Textzeile bis zum Ende aus, Standardwert.

***last*: <integer 1..32767>**
Ganzzahliger Wert für das letzte Byte, das von einer Textzeile noch ausgegeben wird.

***last*: <var: int:2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als letztes Byte interpretiert wird.

ROT=*NO / *YES / <var: enum-of rotation_set:1>
Gibt an, ob die Seiten gedreht ausgegeben werden oder nicht.

ROT=*NO
Keine Seitendrehung, Standardwert. Als Loop wird der im Format definierte Loop verwendet.

ROT=*YES
Die Seite wird um 90 Grad im Uhrzeigersinn gedreht. Als Loop wird der im Format definierte Rotations-Loop verwendet.

ROT=<var: enum-of rotation_set:1>
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

0	*NO
1	*YES

VARIANT=001 / <c-string 3..3>
Bezeichnet die Variante der generierten Parameterliste.

Beispiele

1. Die Textzeile lautet c123YYYYYYYYYYYYYYYYYYYY, wobei c das Vorschubsteuerzeichen an Position 1 darstellt.

Angabe bei PDFSPO	**Text in der PDF-Datei**
LINESP=(*BY_IBM_CONTROL,1) ,RECPART=(1,*STD)	123YYYYYYYYYYYYYYYYYYYY
LINESP=(*BY_IBM_CONTROL,1) ,RECPART=(4,*STD)	YYYYYYYYYYYYYYYYYYYY

2. Die Textzeile lautet 123cXXXXXXX9876543210, wobei c das Vorschubsteuerzeichen an Position 4 darstellt.

Angabe bei PDFSPO	**Text in der PDF-Datei**
LINESP=(*BY_IBM_CONTROL,4) ,RECPART=(1,*STD)	123XXXXXXX9876543210
LINESP=(*BY_IBM_CONTROL,4) ,RECPART=(4,*STD)	XXXXXXX9876543210
LINESP=(*BY_IBM_CONTROL,4) ,RECPART=(4,10)	XXXXXXX

Struktur-Layout von PDFSPO (Assembler)

```
.*****************************************************************
.*  BEGIN-INTERFACE   PDFSPO
.*
.*  TITLE             (/ pdfspo /)
.*  NAME              PDFSPO
.*  DOMAIN             CONV2PDF
.*  LANGUAGE           ASS
.*  COPYRIGHT          (C) Fujitsu Technology Solutions 2016
.*                         ALL RIGHTS RESERVED
.*  COMPILATION-SCOPE USER
.*  INTERFACE-TYPE    CALL
.*  RUN-CONTEXT       TU
.*
.*  PURPOSE           (/ PDF conversion based on SPOOL parameters /)
.*
.*  SYNTAX            (/ Syntax Variant 1:
.*                           PDFSPO  MF = C|D|L|M
.*                         , PREFIX     = [S] | <name>
.*                         , MACID      = [POF] | <name>
.*                        , VARIANT    = <c-string_without_quotes 3..3> |
.*                                       default 001
.*                         , CALLER     = *USER |
.*                                        default *USER
.*                         , EQUATES    = [YES] | NO
.*                         , ROT        = <var: enum-of:1 _ROTATION_SET> |
.*                                       *NO |
.*                                       *YES |
.*                                       default *NO
.*                        , PRTYPE     = <var: enum-of:1 _PRTYPE_SET> |
.*                                       *HP90 |
.*                                       *HP |
.*                                       *LP |
.*                                       default *HP90
.*                         , RECPART    = list(2):
.*                                       FIRSTCH: <var: int 0..65535> |
.*                                                <integer 1..32767> |
.*                                                default: 1
.*                                       LASTCH: <var: int 0..65535> |
.*                                               <integer 1..32767> |
.*                                               *STD |
.*                                               default *STD
.*                         , LINESP     = list(2):
.*                                        SPACING: <var: enum-of:1 _SPACE_SET>
.*                                                 *SPACE_1 |
.*                                                 *SPACE_2 |
.*                                                  *SPACE_3 |
```

```
.*                                                *BY_ASA_CONTROL |
.*                                                *BY_EBCDIC_CONTROL |
.*                                                *BY_IBM_CONTROL |
.*                                                default *SPACE_1
.*                                          CCPOS: <var: int 0..65535> |
.*                                                 <integer 1..2040> |
.*                                                 *STD |
.*                                                 default *STD
.*                           , LEFTMAR    = <var: int 0..65535> |
.*                                          <integer 0..2040> |
.*                                          default: 20
.*                           , LINEPP     = <var: int 0..65535> |
.*                                          <integer 1..32767> |
.*                                          *STD |
.*                                          default *STD
.*                           , FORM       = <var: char 1..6> |
.*                                          <c-string 1..6> |
.*                                          *STD |
.*                                          default *STD
.*                           , LOOP       = <var: char 1..3> |
.*                                          <c-string 1..3> |
.*                                          *STD |
.*                                           default *STD
.*                           , CHARSET    = <var: char 1..3> |
.*                                          <c-string 1..3> |
.*                                          *STD |
.*                                          default *STD /)
.*
.*  REMARKS           (/ LID definition /)
.*
.*  END-INTERFACE     PDFSPO.
.*****************************************************************
```

PDFTMP - Template für PDF-Seiten definieren

Benutzergruppe: Nichtprivilegierter Benutzer
Programmiersprachen: Assembler, C, CPP, COBOL
Makrotyp: S

Mit dem Makro PDFTMP können Sie die Gestaltung von PDF-Seiten in Form eines Templates festlegen und abspeichern. Dabei entspricht eine definierte PDF-Vorgabeseite einem Template-Abschnitt. Mit dem Makro PDFCVT können Sie für die Erstellung der PDF-Seiten innerhalb eines Seitenabschnitts die Vorgabeseite eines Template-Abschnitts verwenden.

Format (Assembler)

Operation	Operanden
PDFTMP	MF=C/D/E/L/M/I ,PREFIX=S / <name 1..1> ,MACID=PDF / <name 1..3> ,PARAM=<name 1..27> ,VARIANT=001 / <c-string 3..3> ,ACTION=*OPEN / *ADDOVERLAY / *ADDSECTION / *ATTOVERLAY / <var: enum-of action_set> ,TMPNAME=(*tmpnptr*,*tmpnlen*) *tmpnptr*: *NONE / <var:pointer> / (reg:pointer) *tmpnlen*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2) ,BOOKMRK=*NO / *YES / <var: enum-of bookmarkuse_set:1> ,OVERLAY = (*name*,*namelength*,*fname*,*fnamelength*) *name:* *NONE / <var:pointer> / (reg:pointer) *namelength*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2) *fname:* *NONE / <var:pointer> / (reg:pointer) *fnamelength*: *STD / <integer:1..54> / <var: int:2> / (reg: int:2)

Operation	Operanden
PDFTMP	,SECTION = (*secnptr*,*secnlen*) *secnptr:* *NONE / <var:pointer> / (reg:pointer) *secnlen*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2) ,OVLLOC=(*name*,*namelength*,*mediabox*,*left*,*right*,*top*,*bottom*,*halign*,*valign*) *name:* *NONE / <var:pointer> / (reg:pointer) *namelength*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2) *mediabox*: *PAGE / *TEXT / *CUSTOM / <var: enum-of ovl_mediabox_set:1> *left*: 0 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *right*: 0 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *top*: 0 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *bottom*: 0 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>) *halign*: *LEFT / *RIGHT / *CENTER / *FIT / <var: enum-of ovl-alignement-set:1> *valign*: *TOP / *BOTTOM / *CENTER / *FIT / <var: enum-of ovl-alignement-set:1> ,TOITEM = (*itemtyp*,*itemptr*,*itemlen*) *itemtyp*: *SECTION / <var: enum-of tmpitem_set:1> *itemptr:* *NONE / <var:pointer> / (reg:pointer) *itemlen*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2) ,HANDLE=<var:pointer> / (reg:pointer)

Operandenbeschreibung

ACTION=
Gibt an, welche Aktion durchgeführt werden soll. Es bestehen Abhängigkeiten zwischen der Aktion und den Angaben bei anderen Operanden, siehe Abschnitt „Abhängigkeiten der Operanden“ auf Seite 57.

ACTION=*OPEN
Erstellt ein PDF-Template, Standardwert. Im Operanden TMPNAME muss der Name angegeben werden.

ACTION=*ADDOVERLAY
Definiert ein Hintergrundbild. Damit die Binärdaten der Bilddatei in die PDF-Datei übernommen werden können, muss das Hintergrundbild mit der Aktion *ATTOVERLAY einem Template-Abschnitt zugewiesen werden.

ACTION=*ADDSECTION
Fügt dem PDF-Template einen neuen Abschnitt hinzu.

ACTION=*ATTOVERLAY
Positioniert ein Hintergrundbild auf der PDF-Seite und weist die Seite einem Template-Abschnitt zu.

ACTION=<var: enum-of action_set>
Die Aktion wird nicht direkt über den entsprechenden Operandenwert, sondern indirekt über ein Feld mit konstantem Inhalt (EQUATE) angegeben. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*OPEN
2	*ADDOVERLAY
3	*ADDSECTION
4	*ATTOVERLAY

BOOKMRK=
Bestimmt die Verwendung von Lesezeichen:
– Bei ACTION=*OPEN bestimmt die Angabe, ob die PDF-Datei mit oder ohne Lesezeichen erstellt wird.
– Bei ACTION=*ADDSECTION bestimmt die Angabe, ob in dem angebenen Template-Abschnitt Lesezeichen aktzeptiert werden.

BOOKMRK=*NO
Keine Verwendung von Lesezeichen, Standardwert.

BOOKMRK=*YES
Verwendung von Lesezeichen. Wenn ein Template-Abschnitt Lesezeichen akzeptiert, wird bei Beginn dieses Seitenabschnitts automatisch ein Lesezeichen gesetzt.

BOOKMRK=<var: enum-of bookmarkuse_set:1>
Gibt den entsprechenden Operandenwert indirekt über ein Feld mit konstantem Inhalt (EQUATE) an. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*NO
2	*YES

HANDLE=<var:pointer> / (reg:pointer)
Gibt beim Aufruf ACTION=*OPEN die Adresse eines 4 Bytes langen, auf Wortgrenze ausgerichteten Bereichs an, in dem der Identifikator des PDF-Konverters gespeichert wird. Dieser Identifikator muss bei jedem nachfolgenden Aufruf (ACTION=*ADDOVERLAY, *ADDSECTION oder *ATTOVERLAY) angegeben werden.

MACID=PDF / '<name 1..3>'
Bestimmt das zweite bis einschließlich vierte Zeichen der Feldnamen und Equates.

MF=C / D / E / L / M / I
Typ des Makroaufrufs. Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil" zu finden.

OVERLAY = (*name*,*namelength*,*fname*,*fnamelength*)
Definiert ein Hintergrundbild mit Overlay-Name und Namenslänge, sowie dem Dateinamen der Bilddatei und der Länge des Dateinamens.

Es werden nur Bilder im JPG-Format unterstützt. Die Übertragung der Bilder vom PC ins BS2000 muss binär erfolgen.

***name*: *NONE / <var:pointer> / (reg:pointer)**
Gibt den Overlay-Namen an.

***name*: *NONE**
Kein Overlay-Name angegeben, Standardwert.

***name*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***namelength*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2)**
Gibt die Länge des Overlay-Namens an.

***namelength*: *STD**
Es wird eine Länge von 8 Zeichen angenommen, Standardwert.

***namelength*: <integer 1..8>**
Angabe eines ganzzahligen Wertes für die Länge des Overlay-Namens.

***namelength*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Overlay-Namens interpretiert wird.

***fname*: *NONE / <var:pointer> / (reg:pointer)**
Gibt den Dateinamen an.

***fname*: *NONE**
Kein Dateinamen angegeben, Standardwert.

***fname*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***fnamelength*: *STD / <integer:1..54> / <var: int:2> / (reg: int:2)**
Gibt die Länge des Dateinamens an.

***fnamelength*: *STD**
Die Länge des Dateinamens beträgt 54 Zeichen, Standardwert.

***fnamelength*: <integer 1..54>**
Angabe eines ganzzahligen Wertes für die Länge des Dateinamens.

***fnamelength*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Dateinamens interpretiert wird.

OVLLOC=(*name*,*namelength*,*mediabox*,*left*,*right*,*top*,*bottom*,*halign*,*valign*)
Bestimmt die Position des Hintergrundbildes innerhalb der PDF-Seite. Anzugeben sind Overlay-Name und Namenslänge, der Bildrahmen, die Seitenabstände zum Rahmen (links, rechts, oben, unten), sowie die horizontale und vertikale Ausrichtung im Rahmen.

***name*: *NONE / <var:pointer> / (reg:pointer)**

***namelength*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2)**

***mediabox*: *PAGE / *TEXT / *CUSTOM**
Bestimmt den Rahmen, in dem das Hintergrundbild positioniert wird.

***mediabox*: *PAGE**
Das Hintergrundbild wird innerhalb der physikalischen PDF-Seite positioniert (bestimmt durch den Operanden PAGEZ im Makro PDFDIR).

***mediabox*: *TEXT**
Das Hintergrundbild wird innerhalb des Textrahmens positioniert (bestimmt durch den Operanden MARGIN im Makro PDFDIR).

***mediabox*: *CUSTOM**
Definiert einen Rahmen über die Abstände zu den Seitenrändern. Dieser Rahmen ist unabhängig von dem definierten Textrahmen.

mediabox: **<var: enum-of ovl_mediabox_set:1>**
Gibt den entsprechenden Operandenwert indirekt über ein Feld mit konstantem Inhalt (EQUATE) an. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*PAGE
2	*TEXT
3	*CUSTOM

***left*: <u>0</u> / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Nur wenn mediabox mit *CUSTOM angegeben ist:
Abstand zum linken Seitenrand in mm.

***left*: <u>0</u>**
Der Abstand beträgt 0 mm, Standardwert.

***left*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***left*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum linken Rand interpretiert wird.

***right*: <u>0</u> / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Nur wenn mediabox mit *CUSTOM angegeben ist:
Abstand zum rechten Seitenrand in mm.

***right*: <u>0</u>**
Der Abstand beträgt 0 mm, Standardwert.

***right*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***right*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum rechten Rand interpretiert wird.

***top*: 0 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Nur wenn mediabox mit *CUSTOM angegeben ist:
Abstand zum oberen Seitenrand in mm.

***top*: 0**
Der Abstand beträgt 0 mm, Standardwert.

***top*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***top*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum oberen Rand interpretiert wird.

***bottom*: 0 / <integer 0..2040> / <var: int: 2> / (<reg: int:2>)**
Nur wenn mediabox mit *CUSTOM angegeben ist:
Abstand zum unteren Seitenrand in mm.

***bottom*: 0**
Der Abstand beträgt 0 mm, Standardwert.

***bottom*: <integer 0..2040>**
Ganzzahliger Wert für den Abstand.

***bottom*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Abstand zum unteren Rand interpretiert wird.

***halign*: *LEFT / *RIGHT / *CENTER / *FIT / <var: enum-of ovl-alignement-set:1>**
Bestimmt die horizontale Ausrichtung des Bildes innerhalb des Rahmens.

***halign*: *LEFT**
Das Bild wird ausgerichtet zur linken Seite des Rahmens, Standardwert.

***halign*: *RIGHT**
Das Bild wird ausgerichtet zur rechten Seite des Rahmens.

***halign*: *CENTER**
Das Bild wird horizontal zentriert im Rahmen.

***halign*: *FIT**
Das Bild wird auf die Breite des Rahmens skaliert. Wenn die Größenverhältnisse von Rahmen und Bild voneinander abweichen, kann das Bild verzerrt dargestellt werden.

***halign*: <var: enum-of ovl-alignement-set:1>**
Gibt den entsprechenden Operandenwert indirekt über ein Feld mit konstantem Inhalt (EQUATE) an. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*LEFT
2	*RIGHT
3	*CENTER
4	*FIT

***valign*: *TOP/ *BOTTOM / *CENTER / *FIT / <var: enum-of ovl-alignement-set:1>**
Bestimmt die vertikale Ausrichtung des Bildes innerhalb des Rahmens.

***valign*: *TOP**
Das Bild wird nach oben ausgerichtet, Standardwert.

***valign*: *BOTTOM**
Das Bild wird nach unten ausgerichtet.

***valign*: *CENTER**
Das Bild wird vertikal zentriert im Rahmen.

***valign*: *FIT**
Das Bild wird auf die Höhe des Rahmens skaliert. Wenn die Größenverhältnisse von Rahmen und Bild voneinander abweichen, kann das Bild verzerrt dargestellt werden.

***valign*: <var: enum-of ovl-alignement-set:1>**
Gibt den entsprechenden Operandenwert indirekt über ein Feld mit konstantem Inhalt (EQUATE) an. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*TOP
2	*BOTTOM
3	*CENTER
4	*FIT

PARAM='<name 1..27>'
Bezeichnet die Adresse der Operandenliste (nur erlaubt bei MF-Format 2 und 3). Nähere Erläuterungen sind im Handbuch „Makroaufrufe an den Ablaufteil“ zu finden.

PREFIX=S / '<name 1..1>'
Bestimmt das erste Zeichen der Feldnamen und Equates.

SECTION = (*secnptr*,*secnlen*)
Gibt den Template-Abschnitt mit seinem Namen und der Namenslänge an.

***secnptr*: *NONE / <var:pointer> / (reg:pointer)**
Gibt den Namen des Template-Abschnitts an.

***secnptr*: *NONE**
Kein Abschnittsname angegeben, Standardwert.

***secnptr*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***secnlen*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2)**
Gibt die Länge des Abschnittsnamens an.

***secnlen*: *STD**
Die Länge des Abschnittsnamens beträgt 8 Zeichen, Standardwert.

***secnlen*: <integer 1..8>**
Angabe eines ganzzahligen Wertes für die Länge des Abschnittsnamens.

***secnlen*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Abschnittsnamens interpretiert wird.

TMPNAME=(*tmpnptr*,*tmpnlen*)
Gibt den Template-Namen und die Länge des Template-Namens an.

***tmpnptr*: *NONE / <var:pointer> / (reg:pointer)**
Gibt den Template-Namen an.

***tmpnptr*: *NONE**
Kein Template-Name angegeben, Standardwert. In diesem Fall darf beim Operanden ACTION der Wert *OPEN nicht angegeben werden.

***tmpnptr*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***tmpnlen*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2)**
Gibt die Länge des Template-Namens an.

***tmpnlen*: *STD**
Die Länge des Template-Namens beträgt 8 Zeichen, Standardwert.

***tmpnlen*: <integer 1..8>**
Angabe eines ganzzahligen Wertes für die Länge des Template-Namens.

***tmpnlen*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Template-Namens interpretiert wird.

TOITEM = (*itemtyp*,*itemptr*,*itemlen*)
Item, dem das Hintergrundbild zugewiesen wird.

***itemtyp*: *SECTION / <var: enum-of tmpitem-set:1>**
Gibt den Item-Typ an.

Derzeit ist nur der Item-Typ *SECTION, d.h. die Zuweisung zu einem Template-Abschnitt möglich.

***itemtyp*: <var: enum-of tmpitem-set:1>**
Gibt den entsprechenden Operandenwert indirekt über ein Feld mit konstantem Inhalt (EQUATE) an. In der Konstanten bzw. dem entsprechenden Feld kann ein ganzzahliger Wert hinterlegt werden. Hier besteht folgende Beziehung zwischen Wert und der gewünschten Funktion:

1	*SECTION

***itemptr*: *NONE / <var:pointer> / (reg:pointer)**
Gibt den Namen des Items an.

***secnptr*: *NONE**
Kein Item-Name angegeben, Standardwert.

***secnptr*: <var: pointer> / (<reg: pointer>)**
Es wird ein Zeiger vereinbart, d.h. der Inhalt der Variablen oder des Feldes ist nicht der benötigte Wert selbst, sondern die Adresse eines Speicherplatzes, an dem der Wert hinterlegt ist (A(feld) oder Angabe eines Registers).

***secnlen*: *STD / <integer:1..8> / <var: int:2> / (reg: int:2)**
Gibt die Länge des Item-Names an.

***secnlen*: *STD**
Die Länge des Item-Names beträgt 8 Zeichen, Standardwert.

***secnlen*: <integer 1..8>**
Angabe eines ganzzahligen Wertes für die Länge des Item-Names.

***secnlen*: <var: int: 2> / (<reg: int:2>)**
Name eines Feldes, das mit FL definiert ist oder ein Register, in dem der Wert steht. In diesem Feld bzw. dem Register wird eine Ganzzahl (Länge 2 Byte) hinterlegt, die als Länge des Item-Names interpretiert wird.

VARIANT=001 / <c-string 3..3>
Bezeichnet die Variante der generierten Parameterliste.

Returncodes

SC1	Bedeutung
0	Kein Fehler
1	Parameterfehler
64	Aufruf korrigieren und erneut versuchen

SC2	Bedeutung
0	Kein Fehler
1	Fehler
2	Warnung

MC	Bedeutung
0	Erfolgreiche Verarbeitung
1	Ungültige Aktion
2	Ungültiger Identifikator (Operand HANDLE)
3	Ungültiger Operand OVERLAY
4	Ungültige Adresse des Overlay-Namens
5	Ungültige Länge des Overlay-Namens
6	Ungültige Adresse für Bilddateiname
7	Ungültige Länge des Bilddateinamens
8	Schreibfehler bei ACTION=*ADDOVERLAY
9	Interner Fehler bei ACTION=*ADDOVERLAY
10	Ungültiger Bildrahmen (mediabox)
11	Ungültiger Wert für horizontale Ausrichtung
12	Ungültiger Wert für vertikale Ausrichtung
13	Ungültige Parameter für benutzerspezifischen Bildrahmen (mediabox *CUSTOM)
14	Schreibfehler bei ACTION=*ATTOVERLAY
15	Interner Fehler bei ACTION=*ATTOVERLAY
16	Version der Parameterliste ungültig
17	Nicht genutzt
18	Bilddatei nicht gefunden
19	Fehler beim Lesen der Bilddatei
20	Bild ist nicht im JPG-Format
21	Ungültiger Funktions-Header

MC	Bedeutung
22	Ungültige Sequenz
23	Ungültige Adresse für Template-Abschnittsname
24	Ungültige Länge für Template-Abschnittsname
25	Kein PDF-Template angegeben
26	Ungültiger Lesezeichen-Indikator
27	Name des Template-Abschnitts ungültig
28	Template-Abschnitt existiert bereits
99	Interner Fehler

Abhängigkeiten der Operanden

Die Angaben in den Operanden des PDFTMP-Makros hängen teilweise von der Aktion ab, die im Operand ACTION angegeben wurde. Die folgende Tabelle zeigt Abhängigkeiten zwischen der Angabe bei ACTION und den restlichen Operanden.

ACTION=	Pflichtoperanden	Optionale Operanden	Ignorierte Operanden
*OPEN	TMPNAME	BOOKMRK	OVERLAY OVLLOC SECTION TOITEM
*ADDOVERLAY	HANDLE OVERLAY		OVLLOC SECTION TOITEM BOOKMRK
*ADDSECTION	HANDLE SECTION	BOOKMRK	OVERLAY OVLLOC TOITEM
*ATTOVERLAY	HANDLE OVLLOC TOITEM		OVERLAY BOOKMRK SECTION

2 Beispiele

Die folgenden Abschnitte zeigen den schematischen Aufbau eines Programms sowie Beispiele für Assembler, C und Cobol. Für eine produktive Anwendung müssen die Beispiele noch erweitert werden.

2.1 Aufbau eines Programms

Die Programmschnittstelle ermöglicht es dem Entwickler von P1-Programmen, aus seiner Anwendung heraus - z.B. aus Logging- oder Trace-Dateien - auf direktem Wege PDF-Dateien zu erzeugen.

Die folgende Liste zeigt die für den Programmaufbau relevanten Schritte mit den zugehörigen Parametern, wobei die Assembler-Notation verwendet wird.

1. Optional: PDF-Template definieren

   Wenn Sie PDF-Seiten mit Hintergrundbildern gestalten wollen, können Sie über das Makro PDFTMP in einem PDF-Template die Parameter zur Gestaltung von PDF-Vorgabeseiten festlegen. Dabei definiert ein Template-Abschnitt die Gestaltung einer PDF-Vorgabeseite, die für alle Seiten eines Abschnitts der zu erstellenden PDF-Datei verwendet wird. Mit mehreren Template-Abschnitten können Sie verschieden gestaltete PDF-Seiten vorgeben und einzelnen Seitenabschnitten der zu erstellenden PDF-Datei zuweisen.

   Um ein Template zu erstellen, gehen Sie wie folgt vor:

   1. Template erstellen

      - Geben Sie bei ACTION die Option *OPEN an.
      - Vereinbaren Sie bei TMPNAME den Template-Namen.
      - Optional können Sie bei BOOKMRK angeben, ob die PDF-Datei, die mit dem Template erstellt wird, Lesezeichen enthält.
      - Rufen Sie das Makro PDFTMP auf.

   2. Hintergrundbild in das Template aufnehmen

      - Geben Sie bei ACTION die Option ADDOVERLAY an.

- ► Vereinbaren Sie im Operanden OVERLAY einen Overlay-Namen für das Hintergrundbild und geben Sie den Pfadnamen der Bilddatei an.
- ► Geben Sie in HANDLE die Adresse des Identifikators an, der beim Erstellen des Templates (OPEN-Aufruf) geliefert wird.
- ► Rufen Sie das Makro PDFTMP auf.

Wiederholen Sie diesen Schritt für jedes weitere Hintergrundbild, das Sie im Template verwenden wollen.

3. Template-Abschnitt definieren
   - ► Geben Sie bei ACTION die Option *ADDSECTION an.
   - ► Vereinbaren Sie im Operanden SECTION Sie den Abschnittsnamen.
   - ► Optional geben Sie im Operanden BOOKMRK an, ob Sie bei der Ausgabe in diesem Template-Abschnitt Lesezeichen zulassen.
   - ► Geben Sie in HANDLE die Adresse des Identifikators an, der beim Erstellen des Templates (OPEN-Aufruf) geliefert wird.
   - ► Rufen Sie das Makro PDFTMP auf.

   Wiederholen Sie diesen Schritt für jeden weiteren Template-Abschitt.
4. Hintergrundbild auf PDF-Seite positionieren und einem Template-Abschnitt zuweisen
   - ► Geben Sie bei ACTION die Option *ATTOVERLAY an.
   - ► Geben Sie im Operanden OVLLOC die Parameter zur Positionierung des Hintergrundbildes in der PDF-Seite an. Notwendige Parameter sind:
     - – Overlay-Name, mit dem das Bild in das Template aufgenommen wurde
     - – Rahmen, in das Bild positioniert wird
     - – Abstände zum Bildrahmen
     - – vertikale und horizontale Ausrichtung des Bildes
   - ► Geben Sie im Operanden TOITEM den Item-Typ *SECTION und die Adresse des Template-Abschnitts an, dem die PDF-Seite zugeordnet werden soll.
   - ► Geben Sie in HANDLE die Adresse des Identifikators an, der beim Erstellen des Templates (OPEN-Aufruf) geliefert wird.
   - ► Rufen Sie das Makro PDFTMP auf.

   Wiederholen Sie diesen Schritt für jede weitere PDF-Seite. Sie können auf der PDF-Seite eines Template-Abschnitts auch weitere Hintergrundbilder positionieren.

2. Layout-Parameter der PDF-Datei definieren.

   Die Layout-Parameter können Sie über das Makro PDFDIR oder das Makro PDFSPO festlegen:

   - Für die direkte Parameterangabe verwenden Sie das Makro PDFDIR mit den folgenden Optionen:
     - Seitengröße, Seitenränder, Zeilendichte, Zeilenvorschub
     - Angabe ob Textzeilen vollständig oder teilweise ausgegeben werden
     - Schrift (Art, Stil, Größe)
   - Für die Spool-Parameterangabe verwenden Sie das Makro PDFSPO mit den folgenden Optionen:
     - FORM, LOOP, Druckertyp, ROT(ATION)
     - Einrückung, Zeilen pro Seite, Zeilenvorschub
     - Angabe ob Textzeilen vollständig oder teilweise ausgegeben werden
     - CHARSET (Schrift)

3. Eigenschaften der PDF-Datei festlegen:

   Die Eigenschaften der PDF-Datei definieren Sie im ersten Aufruf des Makros PDFCVT:

   - Geben Sie bei ACTION die Option *OPEN an.
   - Legen Sie über DIRPAR oder SPOPAR fest, welche Layout-Parameter der PDF-Datei verwendet werden, siehe Schritt2. DIRPAR referenziert die Einstellungen im Makro PDFDIR, SPOPAR die Einstellungen in PDFSPO. Sie dürfen immer nur einen der beiden Parameter angeben.
   - Definieren Sie mit OUTNAME den Namen der PDF-Datei und die Namenslänge.
   - Geben Sie in HANDLE die Adresse eines 4 Bytes langen, auf Wortgrenze ausgerichteten Bereichs an, in dem der Identifikator des PDF-Konverters gespeichert wird.
   - Optional können Sie in WRMODE und CCS einen vom Standard abweichenden Schreibmodus und Zeichensatz einstellen.
   - Optional können Sie in TEMPLAT die PDF-Vorgabeseiten eines zuvor definierten PDF-Templates für die Ausgabe verwenden, siehe Schritt 1.
   - Rufen Sie das Makro PDFCVT auf.

4. Lesezeichen angeben (optional)

   Wenn Sie Lesezeichen in die PDF-Datei einfügen möchten, dann müssen Sie das erste Lesezeichen direkt nach den OPEN-Aufruf einfügen, d.h. vor der ersten Textzeile. Anderfalls führt das zu einem Fehler.

   Ein Lesezeichen fügen Sie wie folgt mit dem Makro PDFCVT ein:

   - Geben Sie bei ACTION die Option *BOOKMARK an.
   - Definieren Sie mit BOOKMARK den Namen des Lesezeichens und die Namenslänge.
   - Geben Sie in HANDLE die Adresse des Identifikators an, der beim OPEN-Aufruf geliefert wird, siehe Schritt 3.
   - Rufen Sie das Makro PDFCVT auf.

5. Textzeile in die PDF-Datei einfügen

   Eine Textzeile fügen Sie wie folgt mit dem Makro PDFCVT ein:

   - Geben Sie bei ACTION die Option *ADDTEXT an.
   - Geben Sie bei TEXT die Adresse und die Länge der Textzeile an.
   - Geben Sie in HANDLE die Adresse des Identifikators an, der beim OPEN-Aufruf geliefert wird, siehe Schritt 3.
   - Rufen Sie das Makro PDFCVT auf.

6. Wiederholen Sie Schritt 5 für alle Textzeilen, die in die PDF-Datei eingefügt werden sollen.

   Wiederholen Sie Schritt 4, wenn Sie weitere Lesezeichen einfügen möchten.

7. Aktuellen Seitenabschnitt in der PDF-Datei beenden (optional):

   Wenn die PDF-Datei mit einem PDF-Template definiert wurde, wird die PDF-Vorgabeseite des ersten Template-Abschnitts zur Erstellung der aktuellen PDF-Seiten verwendet. Soll die Vorgabeseiten eines anderen Template-Abschnitts verwendet werden, müssen Sie den aktuellen Seitenabschnitt beenden und den neuen Template-Abschnitt zuweisen.

   Den Abschnitt einer PDF-Datei beenden Sie wie folgt mit dem Makro PDFCVT:

   - Geben Sie bei ACTION die Option *ENDSECTION an.
   - Geben Sie bei NXTSECT einen Template-Abschnitt an, der für die Ausgabe der nächsten PDF-Seiten (des nächsten Abschnitts) verwendet werden soll.
     Ohne diese Angabe wird die Vorgabeseite des nächsten Template-Abschnitt verwendet.

- ► Geben Sie in HANDLE die Adresse des Identifikators an, der beim OPEN-Aufruf geliefert wird, siehe Schritt 3.
- ► Rufen Sie das Makro PDFCVT auf.
- ► Wiederholen Sie die Schritte 4 bis 5 für den neuen Seitenabschnitt der PDF-Datei.

8. PDF-Datei erzeugen und schließen:

   Die PDF-Datei wird erst beim Schließen erzeugt, d.h. die Textzeilen und Lesezeichen werden erst dann in die PDF-Datei geschrieben.

   Die PDF-Datei schließen Sie wie folgt mit dem Makro PDFCVT:

   - ► Geben Sie bei ACTION die Option *CLOSE an.
   - ► Geben Sie in HANDLE die Adresse des Identifikators an, der beim OPEN-Aufruf geliefert wird, siehe Schritt 3.
   - ► Rufen Sie das Makro PDFCVT auf.

2.2 Assembler-Beispiel

Die Layout-Parameter werden in diesem Beispiel direkt angegeben (Makro PDFDIR).

```
##BAL    OPSYN ##BAS
##BALR   OPSYN ##BASR
TESTASC  AMODE ANY
TESTASC  RMODE ANY
TESTASC  @ENTR TYP=M,LOCAL=DMGWA,ENV=SPLSPEC,TITLE=NO
*
         USING STAT,11
         L     11,=A(STAT)
         MVC   0(4,4),=X'00000000'
*
* DIRPL
*
         LA    1,DIRPL
         USING DIRPL,1
         LR    3,1
         MVC   DIRPL(DIRPLL),DIRPLC
         PDFDIR MF=M,                                                  C
               RECPART=(1,50),                                         C
               LINESP=(*SPACE_3,3),                                    C
               PAGESZ=(*A4_LANDSCAPE,*NONE,*NONE),                     C
               FONT=(*HELVETICA,*BOLD,8),                              C
               MARGIN=(20,40,10,30),                                   C
               LPI=8


*
         LA    2,PDFPL
         ST    2,PDFPLA
         MVC   PDFPL(PDFPLL),PDFPLC
         PDFCVT MF=M,                                                  C
               ACTION=*OPEN,                                           C
               OUTNAME=(A(FN),10),                                     C
               WRMODE=*CREATE,                                         C
               HANDLE=A(HDL),                                          C
               SPOPAR=(4),                                             C
               DIRPAR=(1)
*
         PDFCVT MF=E,PARAM=PDFPLA
*
         PDFCVT MF=M,                                                  C
               ACTION=*ADDTEXT,                                        C
               TEXT=(A(LINE1),9),                                      C
               DIRPAR=(3)
```

```
*
         PDFCVT MF=E,PARAM=PDFPLA
*
         PDFCVT MF=M,                                                  C
               ACTION=*ADDTEXT,                                        C
               TEXT=(A(LINE2),9),                                      C
               DIRPAR=(3)
*
         PDFCVT MF=E,PARAM=PDFPLA
*
         PDFCVT MF=M,                                                  C
               ACTION=*ADDBOOKMARK,                                    C
               BOOKMRK=(A(BKMRK1),13),                                 C
               DIRPAR=(3)
*
         PDFCVT MF=E,PARAM=PDFPLA
*
*
         PDFCVT MF=M,                                                  C
               ACTION=*CLOSE,                                          C
               TEXT=(*NONE,*STD),                                      C
               OUTNAME=(*NONE,*STD),                                   C
               SPOPAR=(4),                                             C
               DIRPAR=(3)
*
         PDFCVT MF=E,PARAM=PDFPLA
         @EXIT
         @END
*

DMGWA    @PAR  D=YES
         PRINT GEN
WA       DS    OF
*
PDFPLA   DS    A
PDFPL    PDFCVT MF=C
DIRPL    PDFDIR MF=C
*
DMGWA    @PAR  LEND=YES
*
STAT     DS    OF
PDFPLC   PDFCVT MF=L
PDFPLL   EQU   *-PDFPLC
*
DIRPLC   PDFDIR MF=L
DIRPLL   EQU   *-DIRPLC
*
LINE1    DC    C'LINE 0001'
```

```
LINE1L   EQU   *-LINE1
LINE2    DC    C'LINE 0002'
LINE2L   EQU   *-LINE2
BKMRK1   DC    C'BOOKMARK 0001'
BKMRK1L  EQU   *-BKMRK1
FN       DC    C'MYFILE.PDF'
FNL      EQU   *-FN
HDL      DS    A
*
         PDFCVT MF=D,PREFIX=X
*
         PDFSPO MF=D,PREFIX=X
*
         PDFDIR MF=D,PREFIX=X
*
         END
```

2.3 C-Beispiel

Die Layout-Parameter werden in diesem Beispiel direkt angegeben.

```
// TESTAPI.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include "FHDR.H"
#include "PDFCVT.H"
#include "PDFDIR.H"
#include "PDFSPO.H"
#include <stdlib.h>
#include <string.h>

int main(int argc, char* argv[])
{
struct PDFCVT_pl_mdl pl;
struct PDFDIR_pl_mdl dirpl;
struct PDFSPO_pl_mdl spopl;
char fn[54];
char text1[11];
char bkmk1[11];
char line1[4096];
int j = 1;
int rc = 0;
char *pspopl = new char[1000];
memcpy(fn,"T-6000.PDF",10);

rc = 0;
memset((char*)&pl,'\0',sizeof(pl));
pl.action = PDFCVTaction_open;
pl.handle_ptr = 0;
pl.outname_ptr = fn;
pl.outname_len = 10;
pl.wrmode = PDFCVTwrmode_any;
pl.specified1.spec1_action = 1;
pl.specified1.spec1_outname = 1;
pl.specified1.spec1_wrmode = 1;
pl.specified1.spec1_spopar = 0;
pl.specified1.spec1_dirpar = 1;
pl.dirpar = &dirpl;
pl.spopar = 0;
memset ((char*)&dirpl,'\0',sizeof(dirpl));
dirpl.pagesz.media = PDFDIRpagesz_a4;
dirpl.specified1.spec1_custom = 0;
dirpl.specified1.spec1_media = 1;
dirpl.font.name = PDFDIRfn_courier;
```

```
dirpl.font.style = PDFDIRfn_normal;
dirpl.font.size = 8;
dirpl.specified1.spec1_font = 1;
dirpl.linesp.spacing = PDFDIRspace_1;
dirpl.linesp.cc_pos = 1;
dirpl.specified1.spec1_linesp = 1;
SPDFMOD(pl);
printf("open return code = %x-%x-%x\n",
pl.hdr.returncode.rc.structured_rc.subcode.subcode2,
pl.hdr.returncode.rc.structured_rc.subcode.subcode1,
pl.hdr.returncode.rc.structured_rc.mc.maincode);

if (pl.hdr.FHDR_RC_MAINCODE != PDFCVTok)
     printf("TEST 6000: open NOK\n");
else
{
       /* Write 1 bookmark and 100 lines */

     memcpy(bkmk1,"BOOKMARK 1",10);
     pl.action = PDFCVTaction_addb;
     pl.specified1.spec1_bkmrk = 1;
     pl.bookmark_ptr = bkmk1;
     pl.bookmark_len = 10;
     SPDFMOD(pl);
        if (pl.hdr.FHDR_RC_MAINCODE != PDFCVTok)
        {
          printf("TEST 6000: addbookmark NOK\n");
          rc = 1;
     }
     j = 1;
     while ((j <= 100) && (rc == 0))
     {
          pl.action = PDFCVTaction_addt;
          sprintf(text1,"LINE   %03d", j);
          pl.text_ptr = text1;
          pl.text_len = 10;
          pl.specified1.spec1_text = 1;
          SPDFMOD(pl);
               if (pl.hdr.FHDR_RC_MAINCODE != PDFCVTok)
               {
printf("return code = %x-%x-%x\n",
pl.hdr.returncode.rc.structured_rc.subcode.subcode2,
pl.hdr.returncode.rc.structured_rc.subcode.subcode1,
pl.hdr.returncode.rc.structured_rc.mc.maincode);
               printf("TEST 6000: addtext NOK\n");
               rc = 1;
          }
               j++;
```

```
     }
/* Write 2nd bookmark and 100 lines */

memcpy(bkmk1,"BOOKMARK 2",10);
     pl.action = PDFCVTaction_addb;
     pl.specified1.spec1_bkmrk = 1;
     pl.bookmark_ptr = bkmk1;
     pl.bookmark_len = 10;
     SPDFMOD(pl);
       if (pl.hdr.FHDR_RC_MAINCODE != PDFCVTok)
       {
          printf("TEST 6000: addbookmark NOK\n");
          rc = 1;
     }
     j = 1;
     while ((j <= 100) && (rc == 0))
     {
          pl.action = PDFCVTaction_addt;
          sprintf(text1,"LINE   %03d", j);
          pl.text_ptr = text1;
          pl.text_len = 10;
          pl.specified1.spec1_text = 1;
          SPDFMOD(pl);
               if (pl.hdr.FHDR_RC_MAINCODE != PDFCVTok)
               {
printf("return code = %x-%x-%x\n",
pl.hdr.returncode.rc.structured_rc.subcode.subcode2,
pl.hdr.returncode.rc.structured_rc.subcode.subcode1,
pl.hdr.returncode.rc.structured_rc.mc.maincode);
               printf("TEST 6000: addtext NOK\n");
               rc = 1;
          }
          j++;
     }
       /* Generate the PDF */

     pl.action = PDFCVTaction_clos;
     SPDFMOD(pl);
printf("return code = %x-%x-%x\n",
pl.hdr.returncode.rc.structured_rc.subcode.subcode2,
pl.hdr.returncode.rc.structured_rc.subcode.subcode1,
pl.hdr.returncode.rc.structured_rc.mc.maincode);
     if (pl.hdr.FHDR_RC_MAINCODE != PDFCVTok)
          printf("TEST 6000: NOK\n");
     else
          printf("TEST 6000: OK\n");
}
}
```

2.4 COBOL-Beispiel

Die Layout-Parameter werden in diesem Beispiel über SPOOL-Parameter angegeben.

```
000100 IDENTIFICATION DIVISION.
000200*-----------------------*
000300*
000400 PROGRAM-ID.
000500*----------*
000600     TESTCOB.
000700/
000800 ENVIRONMENT DIVISION.
000900*--------------------*
001000*
001100 CONFIGURATION SECTION.
001200*---------------------*
001300*
001400 SPECIAL-NAMES.
001500*-------------*
001600*
001700     TERMINAL IS v-terminal,
001800     SYMBOLIC CHARACTERS
001900     COPY esmhexay. .
002000/
002100 DATA DIVISION.
002200*-------------*
002300*
002400 WORKING-STORAGE SECTION.
002500*----------------------*
002600*
002700 01  hexa-chars              PIC X(16) VALUE "0123456789ABCDEF".
002800 01  maincode-edit.
002900     02  maincode-dec            PIC S9(9) COMP.
003000     02  FILLER                  REDEFINES maincode-dec.
003100       03  maincode-byte         PIC X(01) OCCURS 4 TIMES.
003200     02  maincode-hex            PIC X(08).
003300     02  FILLER                  REDEFINES maincode-hex.
003400       03  maincode-char2                OCCURS 4 TIMES.
003500         04  maincode-char       PIC X(01) OCCURS 2 TIMES.
003600 01  work-fields.
003700     02  work-counters.
003800       03  i                     PIC S9(04) COMP.
003900       03  work-hw               PIC S9(04) COMP.
004000       03  FILLER                REDEFINES work-hw.
004100         04  work-hw-1           PIC X(01).
004200         04  work-hw-2           PIC X(01).
004300     02  fn                      PIC  X(54).
```

```
004400     02  fnl                      PIC  9(04) COMP.
004500     02  hdl                      PIC  9(09) COMP.
004600     02  line1                    PIC  X(80).
004700     02  lnl                      PIC  9(04) COMP.
004800     02  book1                    PIC  X(80).
004900     02  bkl                      PIC  9(04) COMP.
005000/
005100     COPY pdfcvty .
005200/
005300     COPY pdfspoy .
005400/
005500     COPY pdfdiry .
005600/
005700 PROCEDURE DIVISION.
005800*------------------*
005900*
006000 s-main SECTION.
006100*--------------*
006200*
006300 p-main.
006400*------*
006500*
006600     PERFORM s-test000.
006700     PERFORM s-test001.
006800     PERFORM s-test002.
006900     PERFORM s-test003.
007000     PERFORM s-test004.
007100*
007200 p-exit.
007300*------*
007400*
007500     STOP RUN.
007600/----+----+----+----+----+----+----+----+----+----+----+----+----*
007700*                                                                 *
007800*    Initialize PLs, and call entry                               *
007900*                                                                 *
008000*----+----+----+----+----+----+----+----+----+----+----+----+----*
008100*
008200 s-test000 SECTION.
008300*-----------------*
008400*
008500 p-test000-strt.
008600*--------------*
008700*
008800     DISPLAY "TEST0 OF TESTCOB"     UPON v-terminal.
008900     MOVE PDFCVT-I-pl               TO PDFCVT-pl.
009000     MOVE PDFSPO-I-pl               TO PDFSPO-pl.
009100     MOVE PDFDIR-I-pl               TO PDFDIR-pl.
```

```
009200     CALL "SCPADDR"                USING PDFCVT-spo-par,
009300                                        PDFSPO-pl.
009400     CALL "SCPADDR"                USING PDFCVT-dir-par,
009500                                        PDFDIR-pl.
009600*
009700 p-test000-call.
009800*--------------*
009900*
010000     CALL "SPDFMOD"                USING PDFCVT-pl.
010100*
010200 p-test000-retc.
010300*--------------*
010400*
010500     IF esmfhdr-rc-nbr IN PDFCVT-pl = ZERO
010600     THEN
010700       DISPLAY "MAINCODE = X'00000000'" UPON v-terminal,
010800     ELSE
010900       MOVE esmfhdr-rc-nbr IN PDFCVT-pl TO maincode-dec,
011000       PERFORM s-edit-maincode,
011100       DISPLAY "MAINCODE = X'", maincode-hex, "'" UPON v-terminal,
011200     END-IF.
011300*
011400 p-test000-exit.
011500*--------------*
011600     EXIT.
011700/----+----+----+----+----+----+----+----+----+----+----+----+----*
011800*                                                                 *
011900*    Test operand SPOPAR                                          *
012000*                                                                 *
012100*----+----+----+----+----+----+----+----+----+----+----+----+----*
012200*
012300 s-test001 SECTION.
012400*-----------------*
012500*
012600 p-test001-strt.
012700*--------------*
012800*
012900     DISPLAY "TEST1 OF TESTCOB"    UPON v-terminal.
013000     MOVE PDFCVT-I-pl              TO PDFCVT-pl.
013100     MOVE PDFSPO-I-pl              TO PDFSPO-pl.
013200     MOVE PDFDIR-I-pl              TO PDFDIR-pl.
013300     CALL "SCPADDR"                USING PDFCVT-spo-par,
013400                                        PDFSPO-pl.
013500     CALL "SCPADDR"                USING PDFCVT-dir-par,
013600                                        PDFDIR-pl.
013700*
013800     MOVE 2                        TO PDFSPO-first-ch.
013900     MOVE 85                       TO PDFSPO-last-ch.
```

```
014000     MOVE 80                          TO PDFSPO-linepp.
014100     MOVE "STD001"                    TO PDFSPO-form.
014200     MOVE "ABC"                       TO PDFSPO-loop.
014300     MOVE "101"                       TO PDFSPO-charset.
014400     SET PDFSPO-hp                    TO TRUE.
014500     MOVE 25                          TO PDFSPO-leftmarg.
014600     SET PDFSPO-rotation-yes          TO TRUE.
014700*
014800     SET PDFSPO-specified1-recpart    TO TRUE.
014900     CALL "SCPSETB1"                  USING PDFSPO-specified1,
015000                                            PDFSPO-specified1-set.
015100     SET PDFSPO-specified1-linepp     TO TRUE.
015200     CALL "SCPSETB1"                  USING PDFSPO-specified1,
015300                                            PDFSPO-specified1-set.
015400     SET PDFSPO-specified1-form       TO TRUE.
015500     CALL "SCPSETB1"                  USING PDFSPO-specified1,
015600                                            PDFSPO-specified1-set.
015700     SET PDFSPO-specified1-loop       TO TRUE.
015800     CALL "SCPSETB1"                  USING PDFSPO-specified1,
015900                                            PDFSPO-specified1-set.
016000     SET PDFSPO-specified2-charset    TO TRUE.
016100     CALL "SCPSETB1"                  USING PDFSPO-specified2,
016200                                            PDFSPO-specified2-set.
016300     SET PDFSPO-specified1-prtype     TO TRUE.
016400     CALL "SCPSETB1"                  USING PDFSPO-specified1,
016500                                            PDFSPO-specified1-set.
016600     SET PDFSPO-specified1-leftmar    TO TRUE.
016700     CALL "SCPSETB1"                  USING PDFSPO-specified1,
016800                                            PDFSPO-specified1-set.
016900     SET PDFSPO-specified1-rot        TO TRUE.
017000     CALL "SCPSETB1"                  USING PDFSPO-specified1,
017100                                            PDFSPO-specified1-set.
017200*
017300     SET PDFCVT-action-open           TO TRUE.
017400     MOVE "MYFILE.PDF"                TO fn.
017500     MOVE 10                          TO fnl.
017600     CALL "SCPADDR"                   USING PDFCVT-outname-ptr,
017700                                            fn.
017800     MOVE fnl                         TO PDFCVT-outname-len.
017900     SET PDFCVT-wrmode-create         TO TRUE.
018000     CALL "SCPADDR"                   USING PDFCVT-handle-ptr,
018100                                            hdl.
018200*
018300     SET PDFCVT-specified1-action     TO TRUE.
018400     CALL "SCPSETB1"                  USING PDFCVT-specified1,
018500                                            PDFCVT-specified1-set.
018600     SET PDFCVT-specified1-outname    TO TRUE.
018700     CALL "SCPSETB1"                  USING PDFCVT-specified1,
```

```
018800                                          PDFCVT-specified1-set.
018900     SET PDFCVT-specified1-wrmode   TO TRUE.
019000     CALL "SCPSETB1"                USING PDFCVT-specified1,
019100                                          PDFCVT-specified1-set.
019200     SET PDFCVT-specified1-handle   TO TRUE.
019300     CALL "SCPSETB1"                USING PDFCVT-specified1,
019400                                          PDFCVT-specified1-set.
019500     SET PDFCVT-specified1-spopar   TO TRUE.
019600     CALL "SCPSETB1"                USING PDFCVT-specified1,
019700                                          PDFCVT-specified1-set.
019800*
019900 p-test001-call.
020000*--------------*
020100*
020200     CALL "SPDFMOD"                 USING PDFCVT-pl.
020300*
020400 p-test001-retc.
020500*--------------*
020600*
020700     IF esmfhdr-rc-nbr IN PDFCVT-pl = ZERO
020800     THEN
020900       DISPLAY "MAINCODE = X'00000000'" UPON v-terminal,
021000     ELSE
021100       MOVE esmfhdr-rc-nbr IN PDFCVT-pl TO maincode-dec,
021200       PERFORM s-edit-maincode,
021300       DISPLAY "MAINCODE = X'", maincode-hex, "'" UPON v-terminal,
021400     END-IF.
021500*
021600 p-test001-exit.
021700*--------------*
021800     EXIT.
021900/----+----+----+----+----+----+----+----+----+----+----+----+----*
022000*                                                                 *
022100*    Test operand BOOKMRK=(ADDR(BOOK1),SPACE(BOOK1))              *
022200*                                                                 *
022300*----+----+----+----+----+----+----+----+----+----+----+----+----*
022400*
022500 s-test002 SECTION.
022600*-----------------*
022700*
022800 p-test002-strt.
022900*--------------*
023000*
023100     DISPLAY "TEST2 OF TESTCOB"     UPON v-terminal.
023200     SET PDFCVT-action-addbookmark  TO TRUE.
023300     MOVE "BOOKMARK 0001"           TO book1.
023400     MOVE 13                        TO bk1.
023500     CALL "SCPADDR"                 USING PDFCVT-bookmark-ptr,
```

```
023600                                          book1.
023700     MOVE bkl                          TO PDFCVT-bookmark-len.
023800*
023900     SET PDFCVT-specified1-action      TO TRUE.
024000     CALL "SCPSETB1"                   USING PDFCVT-specified1,
024100                                             PDFCVT-specified1-set.
024200     SET PDFCVT-specified1-bkmrk       TO TRUE.
024300     CALL "SCPSETB1"                   USING PDFCVT-specified1,
024400                                             PDFCVT-specified1-set.
024500*
024600 p-test002-call.
024700*--------------*
024800*
024900     CALL "SPDFMOD"                    USING PDFCVT-pl.
025000*
025100 p-test002-retc.
025200*--------------*
025300*
025400     IF esmfhdr-rc-nbr IN PDFCVT-pl = ZERO
025500     THEN
025600       DISPLAY "MAINCODE = X'00000000'" UPON v-terminal,
025700     ELSE
025800       MOVE esmfhdr-rc-nbr IN PDFCVT-pl TO maincode-dec,
025900       PERFORM s-edit-maincode,
026000       DISPLAY "MAINCODE = X'", maincode-hex, "'" UPON v-terminal,
026100     END-IF.
026200*
026300 p-test002-exit.
026400*--------------*
026500     EXIT.
026600/----+----+----+----+----+----+----+----+----+----+----+----+----+----*
026700*                                                                       *
026800*    Test operand TEXT=(ADDR(LINE1),SPACE(LINE1))                       *
026900*                                                                       *
027000*----+----+----+----+----+----+----+----+----+----+----+----+----+----*
027100*
027200 s-test003 SECTION.
027300*-----------------*
027400*
027500 p-test003-strt.
027600*--------------*
027700*
027800     DISPLAY "TEST3 OF TESTCOB"        UPON v-terminal.
027900     SET PDFCVT-action-addtext         TO TRUE.
028000     MOVE "LINE 0001"                  TO line1.
028100     MOVE 9                            TO lnl.
028200     CALL "SCPADDR"                    USING PDFCVT-text-ptr,
028300                                             line1.
```

```
028400     MOVE lnl                          TO PDFCVT-text-len.
028500*
028600     SET PDFCVT-specified1-action      TO TRUE.
028700     CALL "SCPSETB1"                   USING PDFCVT-specified1,
028800                                             PDFCVT-specified1-set.
028900     SET PDFCVT-specified1-text        TO TRUE.
029000     CALL "SCPSETB1"                   USING PDFCVT-specified1,
029100                                             PDFCVT-specified1-set.
029200*
029300 p-test003-call.
029400*--------------*
029500*
029600     CALL "SPDFMOD"                    USING PDFCVT-pl.
029700*
029800 p-test003-retc.
029900*--------------*
030000*
030100     IF esmfhdr-rc-nbr IN PDFCVT-pl = ZERO
030200     THEN
030300       DISPLAY "MAINCODE = X'00000000'" UPON v-terminal,
030400     ELSE
030500       MOVE esmfhdr-rc-nbr IN PDFCVT-pl TO maincode-dec,
030600       PERFORM s-edit-maincode,
030700       DISPLAY "MAINCODE = X'", maincode-hex, "'" UPON v-terminal,
030800     END-IF.
030900*
031000 p-test003-exit.
031100*--------------*
031200     EXIT.
031300/----+----+----+----+----+----+----+----+----+----+----+----+----*
031400*                                                                 *
031500*    Close                                                        *
031600*                                                                 *
031700*----+----+----+----+----+----+----+----+----+----+----+----+----*
031800*
031900 s-test004 SECTION.
032000*-----------------*
032100*
032200 p-test004-strt.
032300*--------------*
032400*
032500     DISPLAY "TEST4 OF TESTCOB"        UPON v-terminal.
032600     SET PDFCVT-action-close           TO TRUE.
032700*
032800     SET PDFCVT-specified1-action      TO TRUE.
032900     CALL "SCPSETB1"                   USING PDFCVT-specified1,
033000                                             PDFCVT-specified1-set.
033100*
```

```
033200 p-test004-call.
033300*--------------*
033400*
033500     CALL "SPDFMOD"                 USING PDFCVT-pl.
033600*
033700 p-test004-retc.
033800*--------------*
033900*
034000     IF esmfhdr-rc-nbr IN PDFCVT-pl = ZERO
034100     THEN
034200       DISPLAY "MAINCODE = X'00000000'" UPON v-terminal,
034300     ELSE
034400       MOVE esmfhdr-rc-nbr IN PDFCVT-pl TO maincode-dec,
034500       PERFORM s-edit-maincode,
034600       DISPLAY "MAINCODE = X'", maincode-hex, "'" UPON v-terminal,
034700     END-IF.
034800*
034900 p-test004-exit.
035000*--------------*
035100     EXIT.
035200/
035300 s-edit-maincode SECTION.
035400*-----------------------*
035500*
035600 p-edit-maincode-strt.
035700*--------------------*
035800*
035900     PERFORM WITH TEST AFTER         VARYING i FROM 1 BY 1
036000       UNTIL i > FUNCTION LENGTH(maincode-dec)
036100       MOVE ZERO                     TO work-hw,
036200       MOVE maincode-byte(i)         TO work-hw-2,
036300       MOVE hexa-chars(work-hw / 16 + 1: 1)
036400                                     TO maincode-char(i, 1),
036500       MOVE hexa-chars(FUNCTION MOD(work-hw, 16) + 1: 1)
036600                                     TO maincode-char(i, 2),
036700     END-PERFORM.
036800*
036900 p-edit-maincode-exit.
037000*--------------------*
037100*
037200     EXIT.
```

Literatur

Die Handbücher finden Sie im Internet unter *http://manuals.ts.fujitsu.com*. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.

AID (BS2000)
Advanced Interactive Debugger
Testen von ASSEMBH-Programmen
Benutzerhandbuch

Assemblerbefehle (BS2000)
Sprachbeschreibung

ASSEMBH (BS2000)
Beschreibung

BINDER (BS2000)
Benutzerhandbuch

BS2ZIP
Zip-Archivierung in BS2000
Benutzerhandbuch

BS2000 OSD/BC
Makroaufrufe an den Ablaufteil
Benutzerhandbuch

Spool & Print - Kommandos (BS2000)
Benutzerhandbuch

Spool & Print - Makros und Exits (BS2000)
Benutzerhandbuch

SPOOL (BS2000)
Benutzerhandbuch

XHCS (BS2000)
8-bit-Code-Verarbeitung im BS2000
Benutzerhandbuch