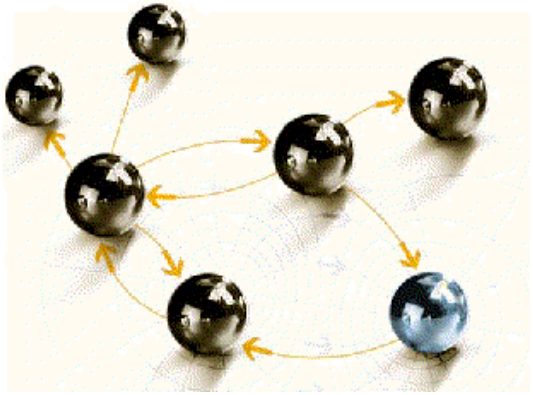


cps4it

File-AID for MVS – Batch

cps4it

consulting, projektmanagement und seminare für die informationstechnologie

Ralf Seidler, Stromberger Straße 36A, 55411 Bingen

Fon: +49-6721-992611, Fax: +49-6721-992613, Mail: ralf.seidler@cps4it.de

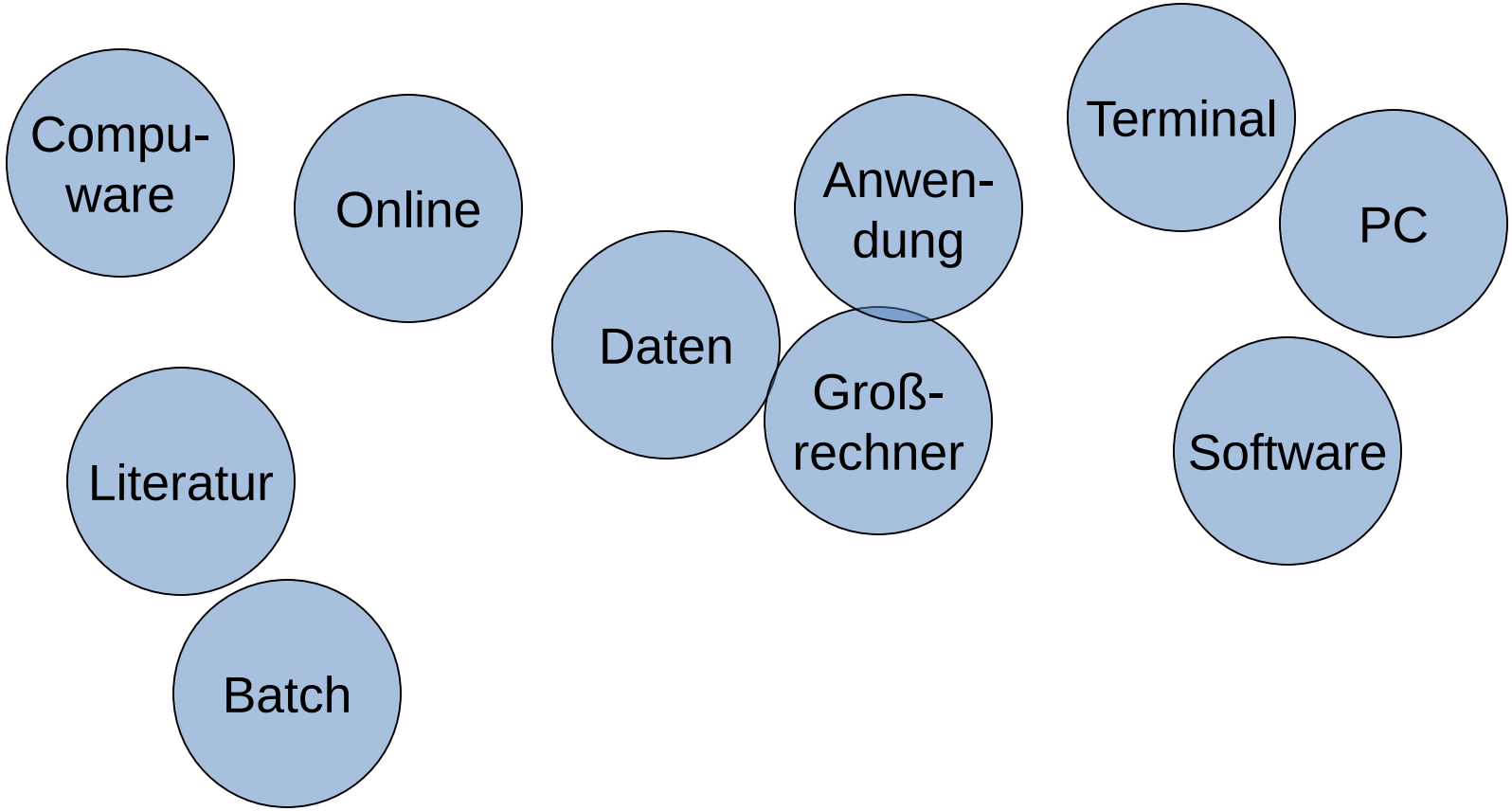
Internet: http://www.cps4it.de

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

Einführung

Begriffe



Einführung

Produkte von Compuware

- Xpediter CICS / Xpediter TSO/IMS
- Abend Aid / CICS Abend Aid
- File-AID for DB2 / File-AID for IMS / File-AID MVS
- Strobe
- Licence Management System
- QA Center
- Vantage
- etc.

Einführung

Literatur

- File-AID for MVS Reference Summary
- File-AID for MVS Online Reference Manual
- File-AID for MVS User’s Guide
- File-AID for MVS Batch Reference Manual
- Bookmanager im Hause
- http://frontline.compuware.com
  - > File-AID-MVS > Documentation
    - Books
  - > File-AID-MVS > Technical Reference
    - Tipps und Tricks

Einführung

Historie und Technik

- Seit vielen Jahren auf dem Markt
- eine *der* Standard-Software von Compuware
- Zielumgebung Großrechner
- Zielgruppe Anwendungsentwicklung
- Neuerungen: wenig und Kunden getrieben
- Zusammenspiel Online und Batch
- „Teil“ des Produkts FileAid MVS

Einführung

wesentliche Neuerungen (1)

- Rel 8.0 – GA 1998 Juli
  - bessere Tapeunterstützung, extended VSAM, Y2K, online Copy
- Rel 8.5 – GA 1999 Februar
  - compare, große Files, multi-volume, XREF-Erweiterung,
- Rel 8.6 – GA 1999 August
  - nix großartiges
- Rel 8.7 – GA 2000 Juli
  - compare auch mit Toleranzen, Batchparameter

Einführung

wesentliche Neuerungen (2)

- Rel 8.8 – GA 2001 Februar
  - compare Loads, HFS-Files, XML-Generierung, vieles im Online
- Rel 8.9 – GA 2006 März
  - viele Zwischenreleases, riesengroße Files, bessere Onlineunterstützung
- Rel 9.0 – GA 2007 August
  - ODO-Felder (COB), REFER-Felder(PL1), VPRINT, Tapeunterstützung
- Rel 9.1 – GA 2008 Februar
  - Unicode, neue Commands im Online

Einführung

wesentliche Neuerungen (3)

- Rel 9.2 – GA 2009 September
  - Feld-Feld-Vergleich, User-Variablen, z/OS 1.10 (EAV), segmentierte Records besser unterstützt
- Rel 9.3 – GA 2011 September
  - Integration Compuware Workbench
- Rel 9.4 – GA 2012 Oktober
  - Compare Option Case insensitve
- Rel 10.1 – GA 2013 Oktober
  - eine Menge „Kleinigkeiten“ siehe http://frontline.compuware.com/Doc/FA/FA101/HTML/cwfca10a.htm

Einführung

Ziele des Produkts (1)

- Hilfe für Anwendungsentwicklung
- Dateien auf Basis bestehender definieren
- Dateien / Testdateien definieren
- Testtabellen erzeugen
- Daten modifizieren
- Daten vergleichen
- Daten selektieren
- Testzeiten reduzieren

Einführung

Ziele des Produkts (2)

- "Ablösen" von Standardutilities wie
  - IDCAMS
  - IEBGENER
  - IEBPTPCH
  - IEBISAM
  - IEBCOPY
  - IEBUPDTE
  - IEHMOVE
  - IEHPROGM

Einführung

Ziele des Produkts (3) – ein Auszug

- "Erweitern" der Funktionen von Standard-Utilities wie
  - Massenupdates von JCL-Bibliotheken
  - Erzeugen von Steuerinformationen
  - Splitten von Dateien
  - formatiertes Ausdrucken
  - Vergleiche von Feldinhalten
  - Daten zählen
  - Feldinhalte kumulieren
  - Dateien vorwärts und rückwärts lesen
  - Update in Record

Einführung

Ziele des Produkts (4)

und …

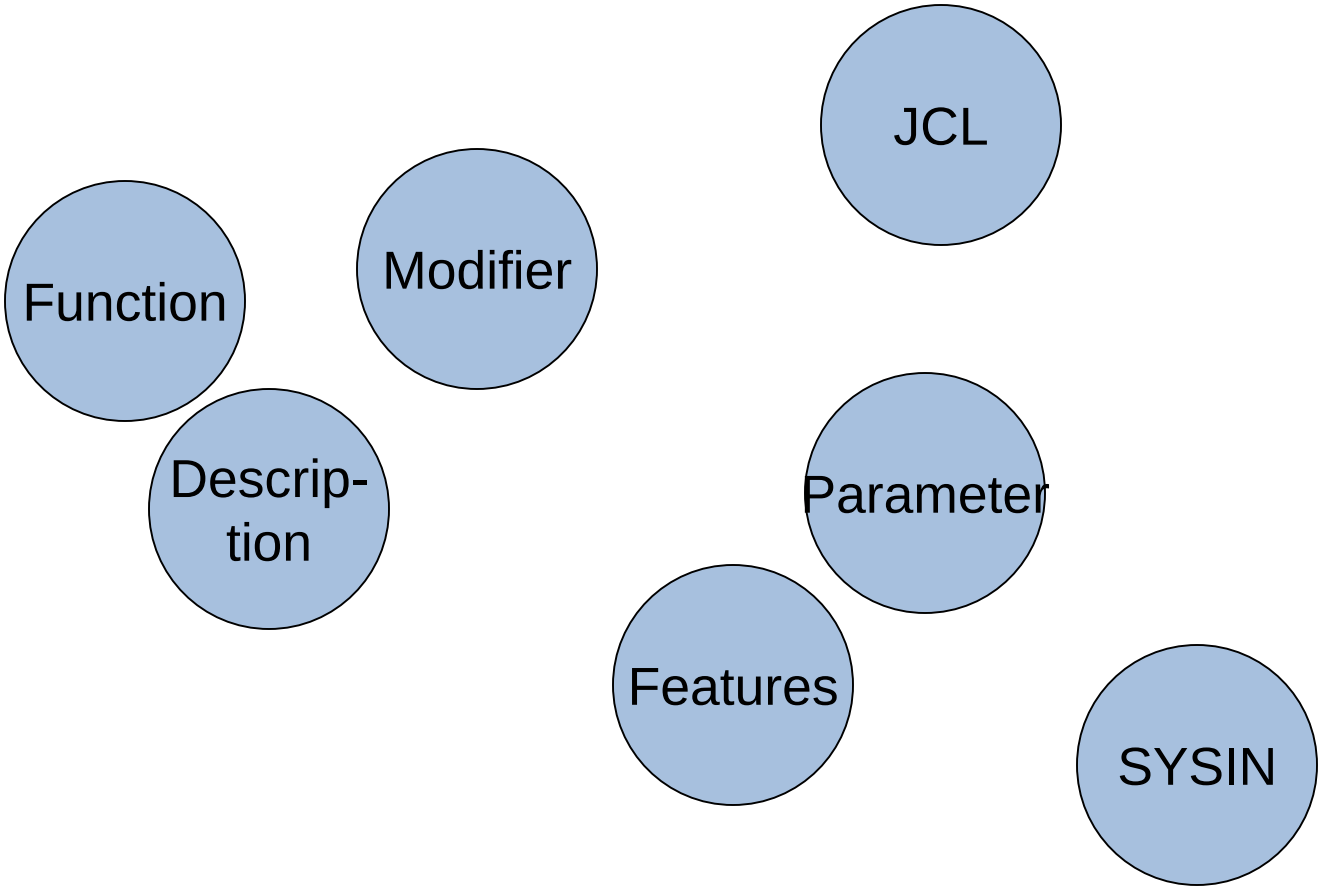
- Unterstützung für erzeugen Batchjob aus dem Online heraus

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

Produktelemente

Begriffe



Produktelemente

Umgebung

- File-AID/Batch ist ein MVS Batchprogramm
- JCL mit üblichem Standard
- Control statements für das Produkt
  via SYSIN DD

- File-AID/MVS (Online-Komponente) kann bei JCL-Erstellung (leider nur) grob unterstützen

Produktelemente

Funktionen / functions

- ... bearbeiten ein File
- ... (oder) produzieren Output („Hardcopy“)
- ... sind parametrisierbar – einschränkend
- ... sind parametrisierbar – weiterführend
  - d.h. weitere Prozesse können gestartet werden
- ... können tw. als Parameter genutzt werden

Produktelemente

Funktionen – Arten – 1

- Funktionen für "hardcopy" erstellen wie
  - DUMP Zeilen in vertikalem Hexaformat
  - FPRINT formatierte Ausgabe (Layout)
  - PRINT Druck alpha, Zeilennummer, Länge
  - VPRINT vertikaler Druck – Basis Layout
- Funktionen für bearbeiten Datei(en) wie
  - COPY kopieren mit Ausgabe Report
  - DROP Zeilen bei kopieren unterdrücken
  - TALLY Datei lesen und Felder summieren
  - UPDATE Änderung „in place“

Produktelemente

Funktionen – Arten – 2

- Funktionen für PO-Dateien / Lademodule wie
  - LMODDIR Member eines PDS anzeigen
  - LMODMAPA CSECT eines PDS-Load anzeigen
- Funktionen für Platten wie
  - VTOCDSN Platten- und Dateiinformationen Reihenfolge: Dateiname
  - VTOCINFO Platteninformationen
- sonstige Funktionen wie
  - SPACE Zeilenpointer verschieben

Produktelemente

function und "function modifier" – Beschreibung

- Funktion ist ein Codewort, das die Operation auf einer Datei beschreibt.
- function modifier ist ein Codewort, das die Funktion kontrolliert oder modifiziert.
- function modifier
  - ALL die Funktion operiert auf der gesamten Datei
  - BACK Funktion wird "rückwärts" ausgeführt (nur PS und VS egal welcher Art)
  - MEM Auswahl vom Members im PDS basierend auf Inhalt / Name des Members

Produktelemente

Parameter – Beschreibung und Arten

- Parameter sind Codeworte, die die Funktion kontrollieren oder beschränken.
- Parametertypen
  - Action
  - Control
  - Limit
  - Print
  - Selection

Produktelemente

Funktionen – Arten – hardcopy – Beispiele

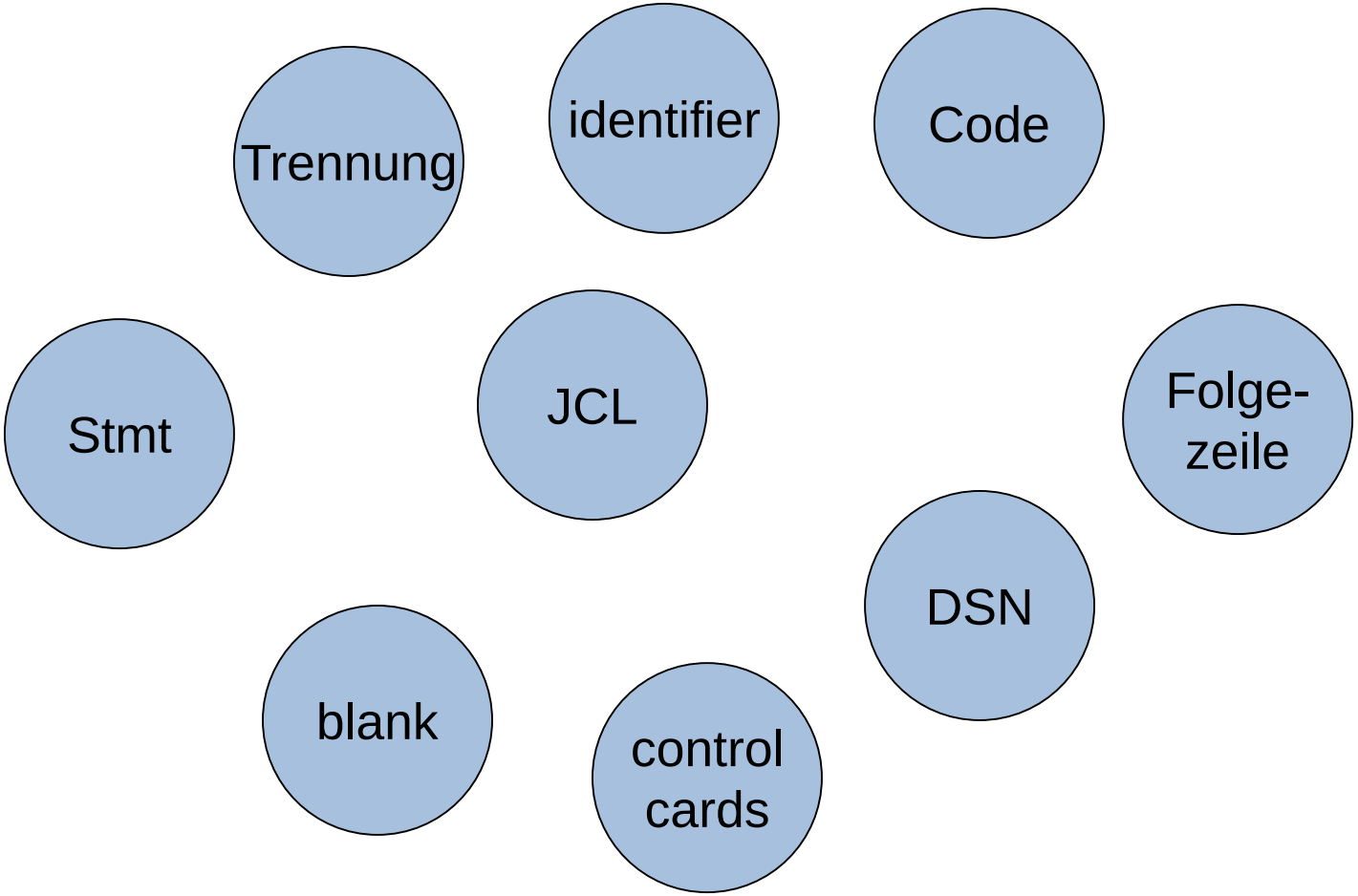
- Siehe Unterlagen

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

Konventionen und Funktionen (1)

Begriffe



Konventionen und Funktionen (1)

Arten

- Code
- Statements (control cards)
- Dateien und Zugriffsmethoden
- JCL

Konventionen und Funktionen (1)

cps4it

Code – 1

- Stellen 1 bis 80
- Code muss vor Spalte 26 beginnen
- Fortsetzungszeilen möglich
  - Komma hinter dem letzten vollständigen Parameter auf der Zeile
  - Blank auf Spalte 1
- Trennung von Funktion und Parameter mit mindestens 1 Blank
- einzelne Parameterelemente nicht trennen

Konventionen und Funktionen (1)

cps4it

einfache Funktionen

- Funktionen für "hardcopy" erstellen wie
  - DUMP Zeilen in vertikalem Hexaformat
  - FPRINT formatierte Ausgabe (Layout)
  - PRINT Druck alpha, Zeilennummer, Länge
  - VPRINT vertikaler Druck – Basis Layout
- Funktionen für bearbeiten Datei(en) wie
  - COPY kopieren mit Ausgabe Report
  - DROP Zeilen bei kopieren unterdrücken

Konventionen und Funktionen (1)

cps4it

JCL –DD-Statements – 1

- STEPLIB klar
- STEPCAT klar
- SYSIN Steuerkarten – „control cards“
- SYSPRINT Protokoll (SYSOUT=*)
- SYSLIST „hardcopy“-Output; je nach Anforderung: auch 183 Stellen
- SYSTOTAL Anzeige Kommentare
  Anzeige Akkumulierungen
- Ausgaben auch auf DSN möglich; dann 80 Byte oder 133 Byte mit FBM/FBA

Konventionen und Funktionen (1)

JCL –DD-Statements – 2

- DDnn
  - Inputdatei; nn = 00-99
- DDnnO
  - Outputdatei erzeugt durch
    COPY, CONVERT, DROP, REFORMAT
- DDnnRL
  - Datei mit Record Layout COBOL, PL1
    PDS, Panvalet, Librarian als Source;

Konventionen und Funktionen (1)

Übungen

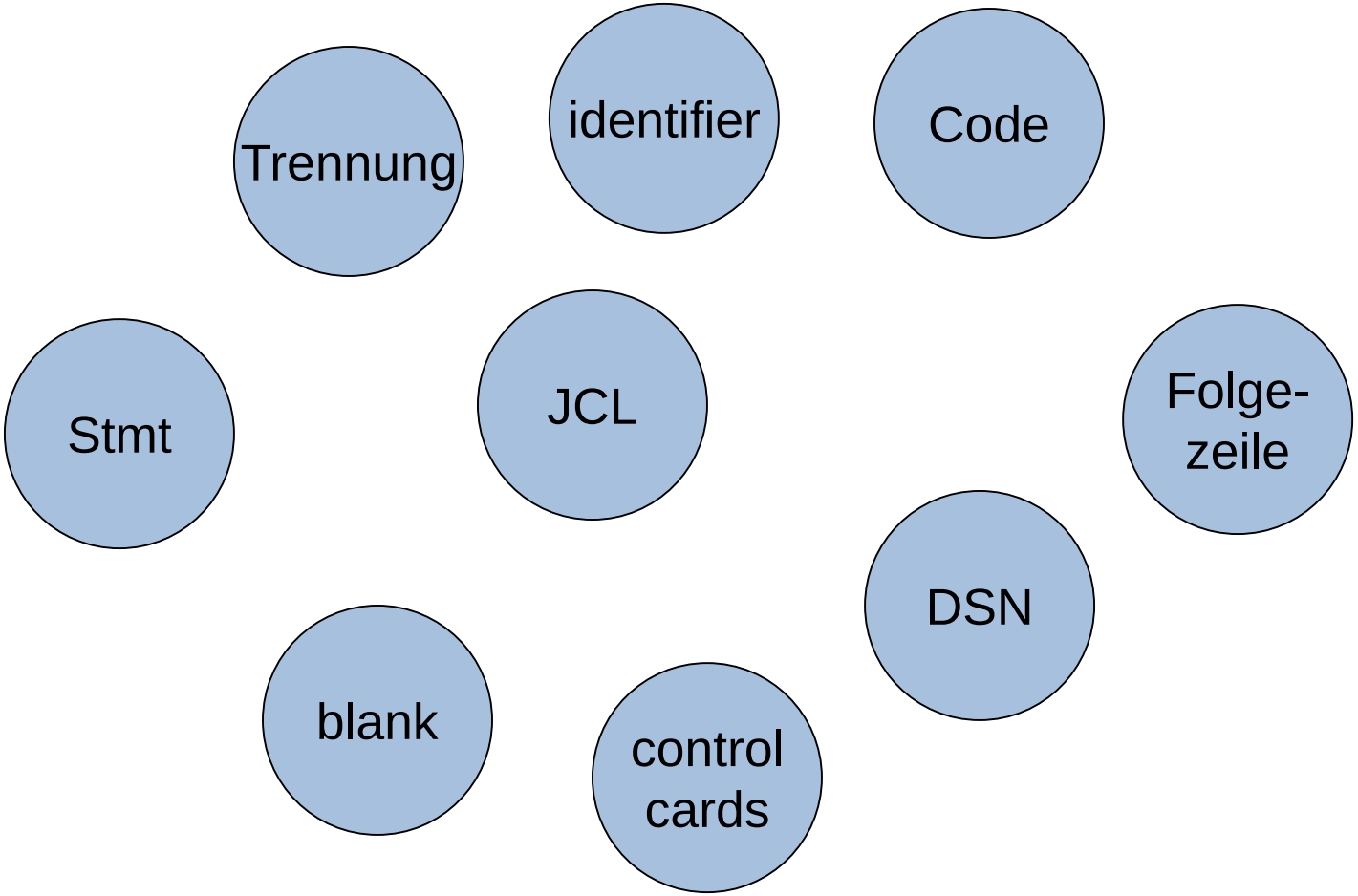
- siehe separate Unterlagen
  - Kapitel 3: vorbereitende Maßnahmen
  - Kapitel 4.1: einfache Ausgabefunktionen

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

Konventionen und Funktionen (2)

Begriffe



Konventionen und Funktionen (2)

Code – 1

- Mehrere Parameter pro Funktion sind möglich, die Reihenfolge entscheidet über die Logik.
- Gleiche Parameter hintereinander müssen mit einem Komma voneinander getrennt werden.
- Auf die Fortsetzungszeile können mehrere Parameter geschrieben werden.
- Abkürzungen für Funktionen und Parameter sind möglich.
  - Beispiele: COPYALL = CA, REPL=R

Konventionen und Funktionen (2)

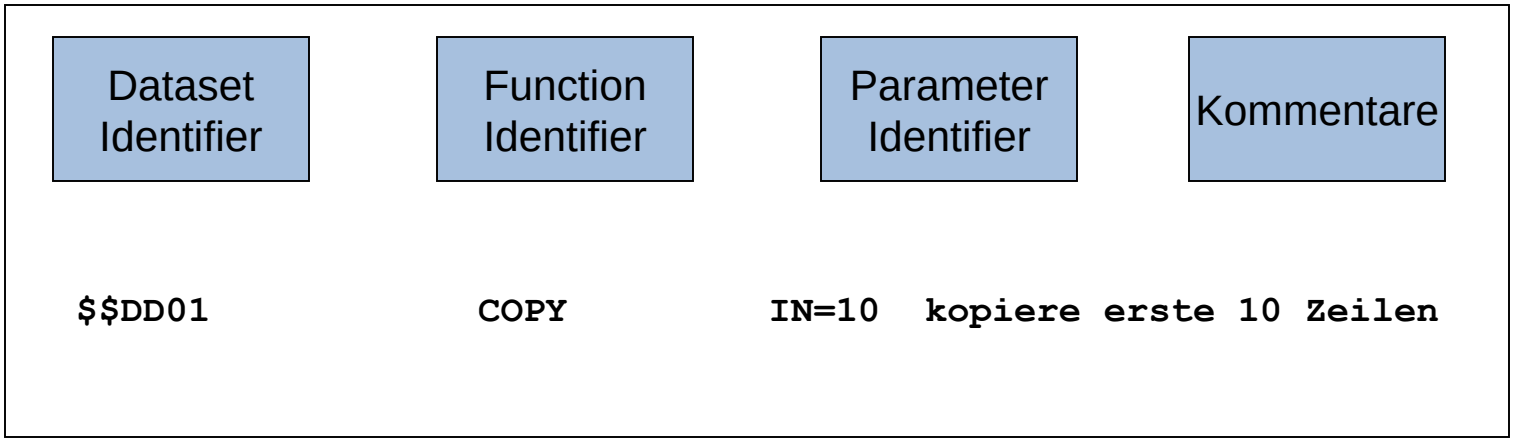
Code – Beispiele

- $$DD01 LIST IF=(1,0,C'XXX')
- $$DD01 LIST IF=(1,0,C'XXX'),IF=(1,0
  ,C'YYY')
- $$DD01 LIST IF=(1,0,C'XXX'),
  IF=(1,0,C'YYY')
- $$DD01 COPYALL REPL=(6,50,C’TE’,C’PR’)
- $$DD01 CA R=(6,50,C’TE’,C’PR’)

Konventionen und Funktionen (2)

control cards

- Über control cards spreche ich mit FileAid. ☺
- 4 Elementtypen
  - Dataset identifier
  - Function / Dataset organization identifier
  - Parameter identifier
  - Kommentare



Konventionen und Funktionen (2)

cps4it

control cards – dataset identifier

- erstes Element der Anweisungen
- Angabe ist obligatorisch
- fester Teil: $$DD
- beginnt auf Spalte 1
- $$DDnn
  - nn ist eine Zahl von 00 bis 99
  - exakte Zuordnung zu //DDnn DD in der JCL
  - exakte Zuordnung zu anderen DD-Anweisungen wie //DDnnRL DD ..., //DDnnO DD etc.

Konventionen und Funktionen (2)

control cards – function / DSORG identifier

- zweites Element der Anweisungen
- Angabe ist obligatorisch
- DSORG-Angabe zwingt FileAid, genau die angegebene Zugriffsart zu benutzen.
  - COPYDA auf einer PS Datei führt zum Öffnen der Eingabedatei als BDAM-Datei.
  - Saubere Zuordnung ist wichtig, da sonst "unpredictable results" auftreten.
  - Gültig: PS/QSAM, DA/BDAM, VS/VSAM, PO/BPAM
- Beispiel: $$DD01 COPYDA

Konventionen und Funktionen (2)

control cards – Parameter identifier

- Angabe ist optional
- definiert, wie Sätze selektiert und verändert werden
- besteht aus dem Namen und 1-n Elementen
- Elemente definieren Eingabedaten, Ausgabedaten, verändernde Eigenschaften der Parameter
- wichtigste Elemente sind Stelle im Satz, Länge, Operator und Inhalt

Detaillierung später

Konventionen und Funktionen (2)

Arten – hardcopy

- DUMP * Zeilen in vertikalem Hexaformat
- FPRINT * formatierte Ausgabe (Layout)
- LIST * Druck alpha, bin/pack als blank(?)
- PRINT * Druck alpha, Zeilennummer, Länge
- RLPRINT Druck recordlayout
- VPRINT * vertikaler Druck – Basis Layout

*) kann auch als Parameter benutzt werden

Konventionen und Funktionen (2)

Arten – hardcopy – Ausgabe

- Output geht immer auf SYSLIST
  - Normalfall SYSOUT=*
  - Ausgabe als Datei ist möglich
    80-Byte-Ausgabe erzwingt 80-stellige Ausgabe !

Konventionen und Funktionen (2)

Arten – Dateibearbeitung – 1

- COPY kopieren mit Ausgabe Report
- DROP Zeilen bei kopieren unterdrücken
- SPACE Zeilenpointer verschieben
- UPDATE Änderung „in place“

Konventionen und Funktionen (2)

function und "function modifier" – Beschreibung

- Funktion ist ein Codewort, das die Operation auf einer Datei beschreibt.
- function modifier ist ein Codewort, das die Funktion kontrolliert oder modifiziert.
- function modifier
  - ALL die Funktion operiert auf der gesamten Datei
  - BACK Funktion wird "rückwärts" ausgeführt (nur PS und VS egal welcher Art)
  - MEM Auswahl vom Members im PDS basierend auf Inhalt / Name des Members

Konventionen und Funktionen (2)

Funktion und "function modifier" – Zuordnung (vollständige Liste)

- COPY ALL, BACK, MEM
- DUMP ALL, BACK, MEM
- FPRINT ALL, BACK, MEM
- LIST ALL, BACK, MEM
- PRINT ALL, BACK, MEM
- SPACE BACK
- UPDATE ALL
- VPRINT ALL, BACK, MEM

Konventionen und Funktionen (2)

Parameter zur Selektion von Zeilen

- AND logisches „und“ innerhalb IF
- ELSE else-Zweig innerhalb IF
- IF Angabe Selektionskriterium
- ORIF logisches „oder“ innerhalb IF
- OR logisches „oder“ innerhalb IF

Konventionen und Funktionen (2)

cps4it

Syntax: IF – 1

```
$$DD01 DUMP IF=(23,EQ,C'TEST FILE')

beides gleich (oder-Angaben):
$$DD01 PRINT IF=(1,EQ,C'A',17,EQ,C'1,2,3')
$$DD01 PRINT,
  IF=(1,EQ,C'A',17,EQ,C'1',17,EQ,C'2',17,EQ,C'3')

und-Anweisung:
$$DD01 PRINT IF=(1,EQ,C'A'),IF=(17,EQ,C'1,2,3')

$$DD01 DUMP IF=(23,0,EQ,C'TEST FILE')
```

Konventionen und Funktionen (2)

cps4it

Syntax: IF – 2

```
ab Stelle 6 10 Byte gepackt:
IF=(6,10,EQP)

ab Stelle 20 10 gepackte Felder mit jeweils 5 Bytes:
IF=(20,5,10EQP)

ab Stelle 20 5 gepackte Felder beliebiger Länge
IF=(20,0,5EQP)

Umkehrung von vorher:
IF=(20,0,5NEP)
```

Konventionen und Funktionen (2)

Parameter zur Beschränkung (Limit)

- DROP Anzahl zu überlesenden Sätze
- IN Anzahl zu lesenden Sätze
- OUT Anzahl zu schreibenden Sätze
- SELECT wählt den jeweils n-ten Satz für Verarbeitung

Konventionen und Funktionen (2)

Syntax: IN / OUT / SELECT

```
IN=n
n={1,999999999}
erste 200 Zeilen aus Eingabe kopieren:
$$DD01 COPY IN=200

OUT=n
hexaprint der ersten 25 Sätze
$$DD01 DUMP OUT=25

SELECT=n
jeden dritten Satz nehmen, der Kriterien erfüllt:
$$DD01 PRINT OUT=10,IF=(1,EQ,P'50'),SELECT=3
```

Konventionen und Funktionen (2)

Parameter zur Änderung – REPL

- REPL Inhalte ändern
  - by location
  - by condition
  - at alternate location depending on condition

- Syntax / Beispiele

```
$$DD01 COPY REPL=(4,C'6')
$$DD01 COPY REPL=(4,EQ,C'2',C'6')
$$DD01 COPY REPL=(4,EQ,C'222',16,C'400')

REPL=(6,EQ,C"634,21",C'634521')
```

Konventionen und Funktionen (2)

Parameter zur Änderung – EDIT

- EDIT Inhalte ändern

- Syntax / Beispiel

```
EDIT=(1,5,C'1234',C'ABCDE')
vorher:
 ----+----1----+----2----+----3
 1234 ABCD999999999999STUVWXYZ
 12346ABCD999999999999STUVWXYZ
nachher:
 ----+----1----+----2----+----3
 ABCDE ABCD999999999999STUVWXY
 ABCDE6ABCD999999999999STUVWXY
```

Konventionen und Funktionen (2)

Übungen

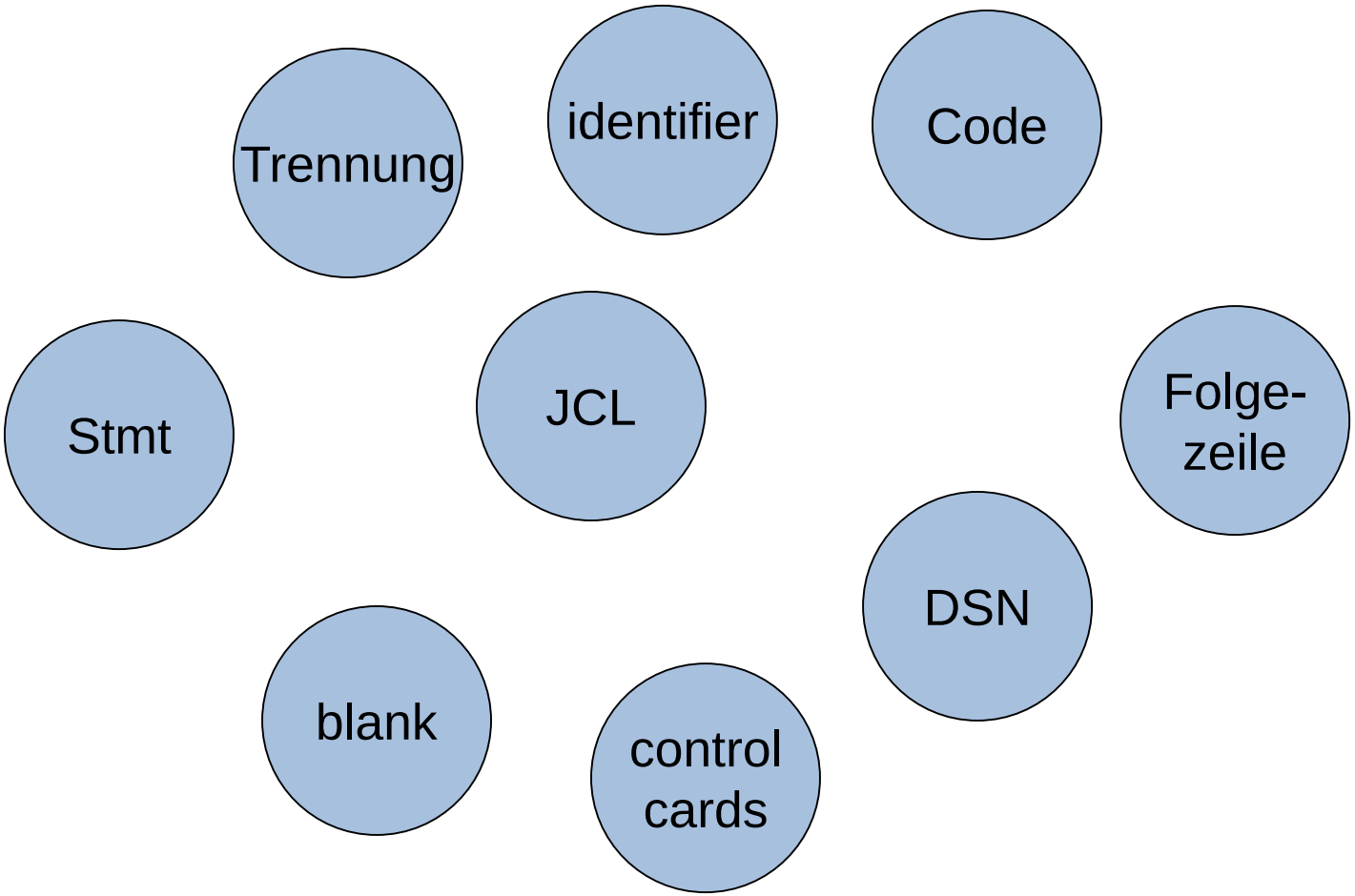
- siehe separate Unterlagen
  - Kapitel 4.2: kopieren von Zeilen
  - Kapitel 4.3: entfernen von Zeilen

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

PO-Dateien und Platten

Begriffe



PO-Dateien und Platten

Member-Kontrolle

- MBRNAME Gruppe von PDS-Member definieren mit Basis Membername
- MEMBER PDS-Member (1) definieren
- MEMBERS Gruppe von PDS-Member definieren mit „Maske”
- NEWMEM neuen Membernamen definieren
- NEWMEMS Output-PDS-Member definieren mit „Maske”

PO-Dateien und Platten

Syntax: MEMBER

```
MEMBER={name                }
       {(name1,name2...)}

$$DD01 COPY MEMBER=PROG241

$$DD01 COPY MEMBER=(PROG241,PROG242,PROG243)
```

PO-Dateien und Platten

Dateien / Zugriffe – BPAM

- feste und variable Länge
- Input und Output
- Verkettung möglich
- Auswahl der Member in unterschiedlichen Variationen
  - exakte Angabe
  - Maske
  - Bereich von Membern
  - Nutzung der ISPF-Statistik

PO-Dateien und Platten

Funktionen – PO-Dateien / Ladebibliotheken

- LMODDIR Member eines PDS anzeigen
- LMODMAPA CSECT von PDS-Load anzeigen
- LMODMAPN CSECT von PDS-Load anzeigen

PO-Dateien und Platten

Dateien / Zugriffe – Ladebibliotheken Kopierregeln

- Inputdatei, Outputdatei UNDEF
- nicht unterstützt
  - kein Umblocken: BLKSIZE-O >= BLKSIZE-I
  - PDSE
  - Overlay Module
  - Scatter-loaded Module
  - note listed Module
- kopieren Alias möglich
  - Auswahl mit Basemember: Alias wird angelegt
  - Auswahl ohne Basemember: Alias wird echte Kopie

PO-Dateien und Platten

Funktionen – Platte

- VTOCDSN Platten- und Dateiinformationen
  Reihenfolge: Dateiname
- VTOCINFO Platteninformationen
- VTOCMAP Platten- und Dateiinformationen
  Reihenfolge: Plattenadresse

PO-Dateien und Platten

Übungen

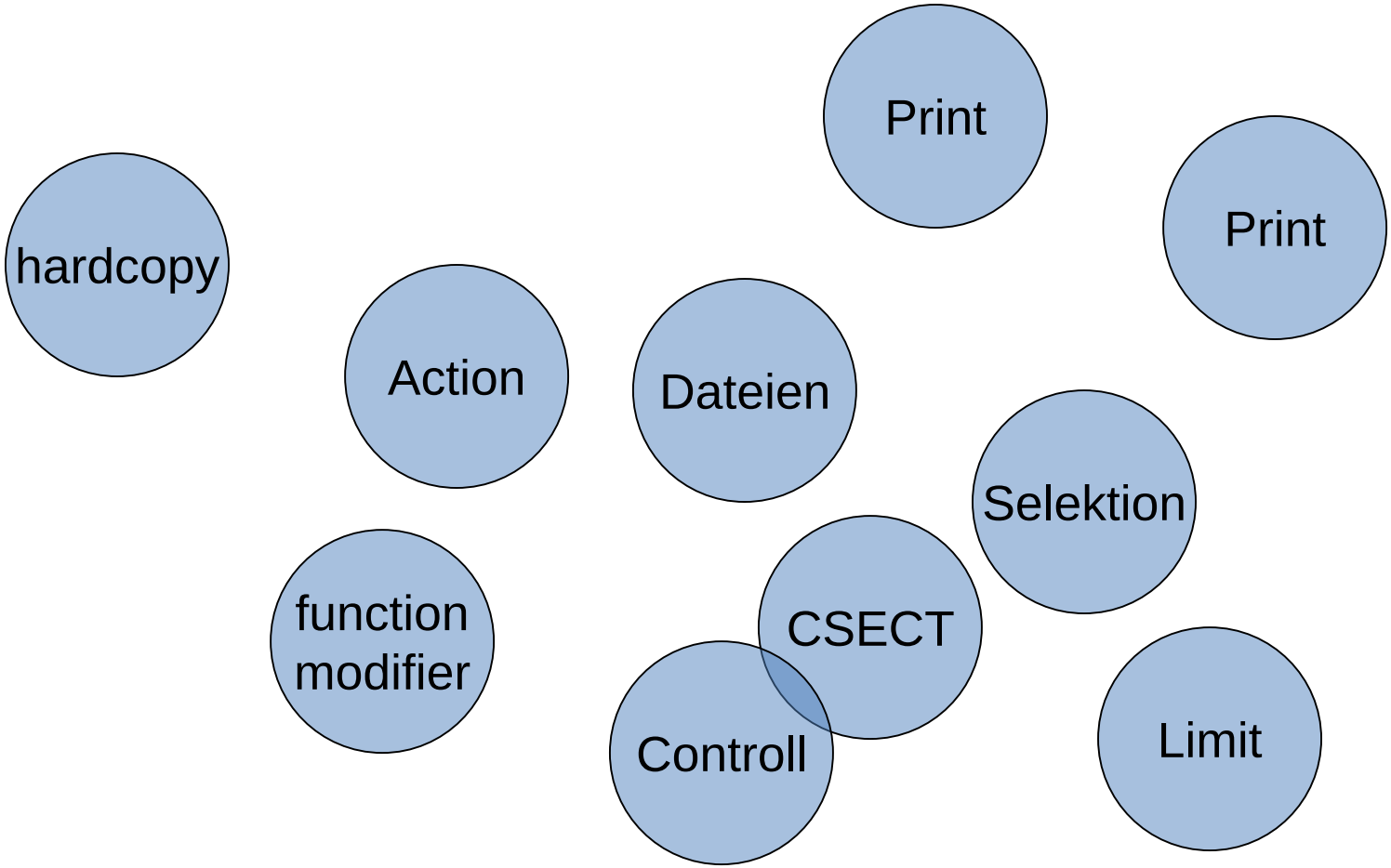
- siehe separate Unterlagen
  - Kapitel 4.4: arbeiten mit PO-Dateien

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

weitere Funktionen und Parameter

Begriffe



weitere Funktionen und Parameter

Funktionen – hardcopy

- DUMP * Zeilen in vertikalem Hexaformat
- FPRINT * formatierte Ausgabe (Layout)
- LIST * Druck alpha, bin/pack als blank(?)
- PRINT * Druck alpha, Zeilennummer, Länge
- RLPRINT Druck recordlayout
- VPRINT * vertikaler Druck – Basis Layout

- APRINT print audit trail
- SCPRINT Druck selection criteria
- XRPRINT XREF-Datei drucken

*) kann auch als Parameter benutzt werden

weitere Funktionen und Parameter

Funktionen – Dateibearbeitung – 1

- COPY kopieren mit Ausgabe Report
- DROP Zeilen bei kopieren unterdrücken
- SPACE Zeilenpointer verschieben
- UPDATE Änderung „in place“

- TALLY Datei lesen / Felder summieren

weitere Funktionen und Parameter

Funktionen – Dateibearbeitung – 3

- COMPARE 2 Dateien vergleichen
- REFORMAT Zeilen umformatieren
- USER kopieren user-defined in eine oder mehrere Outputdateien
- XMLGEN XML-Datei generieren

- CONVERT konvertieren in neues Release

weitere Funktionen und Parameter

cps4it

Parameter – Arten

- Action
  - zeigt Änderung oder Verschiebung von Daten an
- Control
  - Bedingungen für die Ausführung der Funktion
- Begrenzung (Limit)
  - (Zähler auf) Ein- und Ausgaben
- Print
  - Art der Ausgabe
- Selection
  - Auswahl basierend auf Inhalt

weitere Funktionen und Parameter

Parameter – Action

- DFTL_WRITE definieren Default Outputfile
- EDIT ändern eines Satzes
- EDITALL ändern aller Sätze
- MOVE erzeugt Outputsatz
- READNEXT beenden der Verarbeitung des aktuellen Satzes
- REPL ändern eines Satzes (in place)
- REPLALL ändern aller Sätze (in place)
- TYPRUN validieren Vergleichskriterien
- WRITE neuen Satz schreiben (USER)

weitere Funktionen und Parameter

Parameter – Control – 1

- ABEND ändern ABEND-Verhalten
- AMODE adressing mode angeben
- CEM leere PO-Member kopieren
- CHANGED Memberauswahl mit Datum
- CHARSET Sprachauswahl
- COPTNS zusätzliche Optionen bei Compareausgabe
- CREATED Memberauswahl mit Datum
- DSNAME VTOC-Einschränkung Name
- ERRS Fehleranzahl erlauben

weitere Funktionen und Parameter

Parameter – Control – 2

- EXPAND — nested Includes von Panvalet oder Librarian auflösen
- FEOV — „forces end-of-volume” des Outputfiles, wenn Input bei EOV
- FIELDS — definieren Felder des Inputfiles bei VPRINT
- FORM — Kontrolle von JCL-Dateien mehrfache Verarbeitung
- IOEXIT — definieren Exit für In- und Output

weitere Funktionen und Parameter

Parameter – Control – 3

- KEY — verarbeiten Dateien mit Key
- KEYINFO — verarbeiten Key bei Konvert von von XREF-Dateien
- LANGTYP — Member bei Panvalet-Eingabe
- LAYOUT — Memberangabe für DDxxRL bei FPRINT (Funktion / Parameter)
- LINKDATE — Gruppe von PDS-Member definieren mit Basis Linkdatum
- LPI — „lines-per-inch” bei Druckausgabe
- MAP — Alias für „LAYOUT”

weitere Funktionen und Parameter

Parameter – Control – 4

- MAXENT „Extends area in which File-AID parameter information is stored beyond the default limit.”
- MAXOUT mehr als 8 user-controlled Ausgabedateien pro Ausführung
- MBRNAME Gruppe von PDS-Member definieren mit Basis Membername
- MEMBER PDS-Member (1) definieren
- MEMBERS Gruppe von PDS-Member definieren mit „Maske”

weitere Funktionen und Parameter

Parameter – Control – 5

- NEWMEM neuen Membernamen definieren
- NEWMEMS Output-PDS-Member definieren mit „Maske”
- PADCHAR definieren Füllcharacter bei Verlängerung des Satzes
- PANSTAT definieren Panvalet-Member mit Basis „Status“
- PDSSTAT PDS-Statistik aktualisieren
- PRTRECS Auswahl der Records bei COMPARE

weitere Funktionen und Parameter

cps4it

Parameter – Control – 6

- RBA — Funktion an RBA beginnen
- RDW — RDW ein- oder ausschließen
- REFOUT — definieren Satz, der bei Reformatierung kopiert werden soll
- RLM — Überschreiben der PDS-Member kontrollieren
- RMODE — RMODE angeben
- RRN — Angabe relative Satzadresse bei KSDS- und BDAM-Dateien
- SHOW — Reportformat bei VPRINT

weitere Funktionen und Parameter

Parameter – Control – 7

- STOP — Anhalten bei Bedingung
- TYPE — Angabe Art der Konvertierung (nur für Releasewechsel)
- UNIT — Abgabe bei VTOC-Funktionen
- USERID — Gruppe von PDS-Member definieren auf Basis Userid
- VOLSER — Angabe bei VTOC-Funktionen
- VOLSTAT — Angabe bei VTOC-Funktionen

weitere Funktionen und Parameter

Parameter – Beschränkung (Limit)

- DROP Anzahl zu überlesenden Sätze
- IN Anzahl zu lesenden Sätze
- OUT Anzahl zu schreibenden Sätze
- SELECT wählt den jeweils n-ten Satz für Verarbeitung

weitere Funktionen und Parameter

Parameter – Druck

- ACCUM addieren Feld(er)
- DUMP Ausgabe hexa-format
- FPRINT Ausgabe Basis Layout
- LIST Ausgabe alphanummerisch (ohne Satznummer / -länge)
- PRINT Ausgabe alphanummerisch (mit Satznummer / -länge)
- RLPRINT Ausgabe Layout
- VPRINT vertikale Ausgabe Basis Layout

weitere Funktionen und Parameter

Parameter – Auswahl (Selection)

- AND logisches „und“ innerhalb IF
- ELSE else-Zweig innerhalb IF
- IF Angabe Selektionskriterium
- ORIF logisches „oder“ innerhalb IF
- OR logisches „oder“ innerhalb IF

weitere Funktionen und Parameter

Übungen

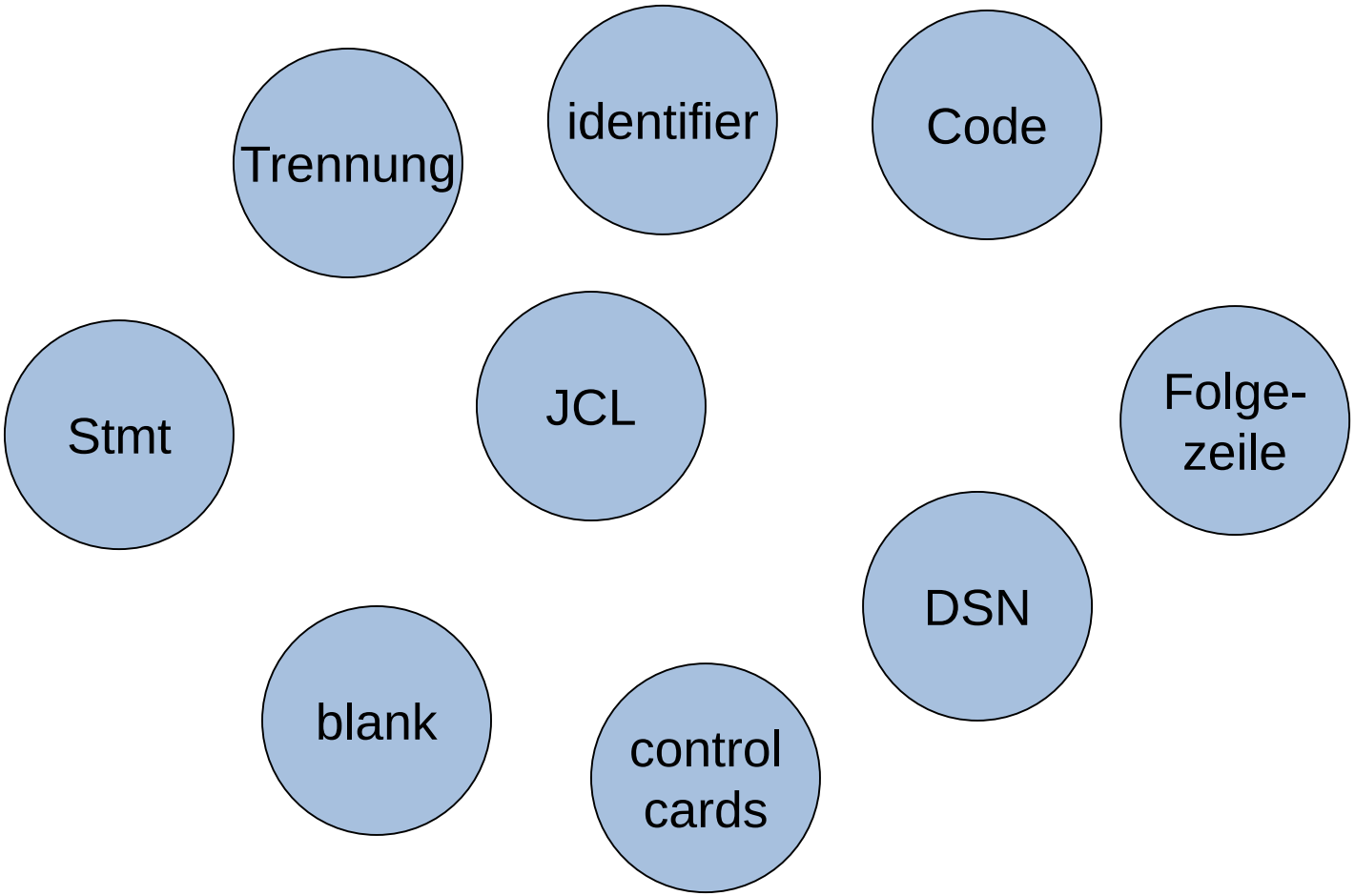
- siehe separate Unterlagen
  - Kapitel 4.5: Einfache Ausgabefunktionen
  - Kapitel 4.6: Vergleichen von Dateien
  - Kapitel 4.7: Überlesen von Zeilen und Änderung
  - Kapitel 4.8: weitere Funktionen und Parameter

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

JCL und Dateien

Begriffe



JCL und Dateien

Dateien / Zugriffe

- unterstützte Zugriffsmethoden sind
  - QSAM sequentielle Dateien
  - BDAM basic direct access Dateien (DA)
  - VSAM KSDS, ESDS, RRDS
  - BPAM PO-Dateien
  - HFS Unix-Dateien

JCL und Dateien

Dateien / Zugriffe – QSAM

- nur ohne Key (Key gilt als Dateninhalt)
- als Input und Output
- Verkettung nur bei analogen Eigenschaften
  - Protokoll zeigt einzelne Dateien an
- multi-volume möglich
  - Protokoll zeigt einzelne Volumes an
- vorwärts und rückwärts bearbeiten

- nur „normale“ BDAM-Dateien
  - nicht spanned
  - ohne Track-Overflow
- wenn Outputbereich größer als Daten, wird Rest **nicht** mit x‘00‘ aufgefüllt
- FileAid nutzt BSAM zum Lesen von BDAM
- falls mit Key muss dieser aufsteigend sein
- Key muss auf Stelle 1 beginnen

Dateien / Zugriffe – VSAM

- KSDS, ESDS, RRDS sind unterstützt
- feste und variable Länge
- Input und Output
- RDW (record descriptor word) wird automatisch hinzugefügt wo erforderlich
- automatische Längenanpassung
- separate Kontrollblöcke für Input und Output
- AMP-Parameter wird nicht unterstützt

JCL und Dateien

Dateien / Zugriffe – UNIX HFS

- Begrenzung durch <newline> dann Behandlung wie variabel lange Sätze
- Keine Begrenzung durch <newline> dann Angabe LRECL in JCL (1-32756) notwendig
- keine Verkettung
- PATH-Informationen in JCL anzugeben
- kein Update „in place“
- kein rückwärts verarbeiten

JCL und Dateien

Dateien / Zugriffe – Inputdateien – 1

- 1-100 Dateien gleichzeitig möglich
- Nutzung des DCB
- für Überschreiben des DCB nimm JCL
- VBS-Dateien (spanned) voll unterstützt
- wenn VBS-Dateien segmentiert verarbeitet werden sollen, braucht JCL RECFM=VB
- VBS-BDAM werden nicht vollständig unterstützt
- CA-Librarian möglich (wenn installiert)
- CA-Panvalet möglich (wenn installiert)

Dateien / Zugriffe – Inputdateien – 2

- Bänder
  - Label wird benutzt
  - falls unlabeled: RECFM=U,BLKSIZE=32767 benutzt
  - JCL kann Label-Info überschreiben
  - maximale LRECL: 32760
  - maximale BLKSIZE: je nach Device (256k bei 3590)

Dateien / Zugriffe – Inputdateien – 3

- kopieren VB nach FB möglich
  - ohne Rücksicht auf eventuelles Abschneiden
- RDW wird sauber berücksichtigt
- Bei mehreren Dateien auf gleichem Medium kann Parameter AFF benutzt werden, was mehrfaches Allokieren unnötig macht.
  - //DD01 DD DSN=NAME1,UNIT=TAPE
  - //DD02 DD DSN=NAME2,UNIT=AFF=DD01

cps4it

Dateien / Zugriffe – Outputdateien

- maximal 100 DSN sind möglich
- keine Einschränkung bzgl. Device
- Bänder
  - Label wird benutzt
  - falls unlabeled: RECFM=U,BLKSIZE=32767 benutzt
  - JCL kann Label-Info überschreiben
  - maximale LRECL: 32760
  - maximale BLKSIZE: je nach Device (256k bei 3590)

JCL – REGION-Angabe

- REGION-Angabe 6M empfohlen oder
  - 1024k wenn Aufruf via Linkliste (üblich)
  - 2048k wenn Aufruf via STEPLIB
  - 3072k wenn große BLKSIZE oder viele DD-Statements
  - 6M bei APRINT, CONVERT, FPRINT, VPRINT

JCL und Dateien

JCL – DD-Statements – 1

- STEPLIB klar
- STEPCAT klar
- SYSIN Steuerkarten – „control cards“
- SYSPRINT Protokoll (SYSOUT=*)
- SYSLIST „hardcopy“-Output; je nach Anforderung: auch 183 Stellen
- SYSTOTAL Anzeige Kommentare
  Anzeige Akkumulierungen
- Ausgaben auch auf DSN möglich; dann 80 Byte oder 133 Byte mit FBM/FBA

JCL und Dateien

JCL – DD-Statements – 2

- DDnn
  - Inputdatei; nn = 00-99
- DDnnO
  - Outputdatei erzeugt durch
  - COPY, CONVERT, DROP, REFORMAT
- DDnnRF
  - Datei für Reformatierung; Datei wird im Online mit der Option 9 erzeugt, wenn Batch gewählt
- DDnnRL
  - Datei mit Record Layout COBOL, PL1 PDS, Panvalet, Librarian als Source;

JCL – DD-Statements – 3

- DDnnRLN
  - Datei mit Record Layout COBOL, PL1
    PDS, Panvalet, Librarian als Source;
    formatierter COMPARE mit dem COMPARE NEW;
    auch benötigt, wenn DDnnSC vorhanden
- DDnnXR
  - Datei mit Cross-reference (XREF)
    generiert mit Option 5.1
    auch benötigt, wenn DDnnSC vorhanden
- DDnnXRN
  - Analog DDnnXR mit „compare new“

JCL – DD-Statements – 4

- DDnnSC
  - Datei mit Selection Criteria
    generiert im Online bei verschiedenen Options
- DDnnSCN
  - Analog DDnnSC mit „compare new“
- DDnnCP
  - Datei mit Key- und Compareinformationen
    generiert im Online in Optionen 7 oder 10
- DDnnC
  - zu vergleichende Datei; aus Option 10

JCL – DD-Statements – 5

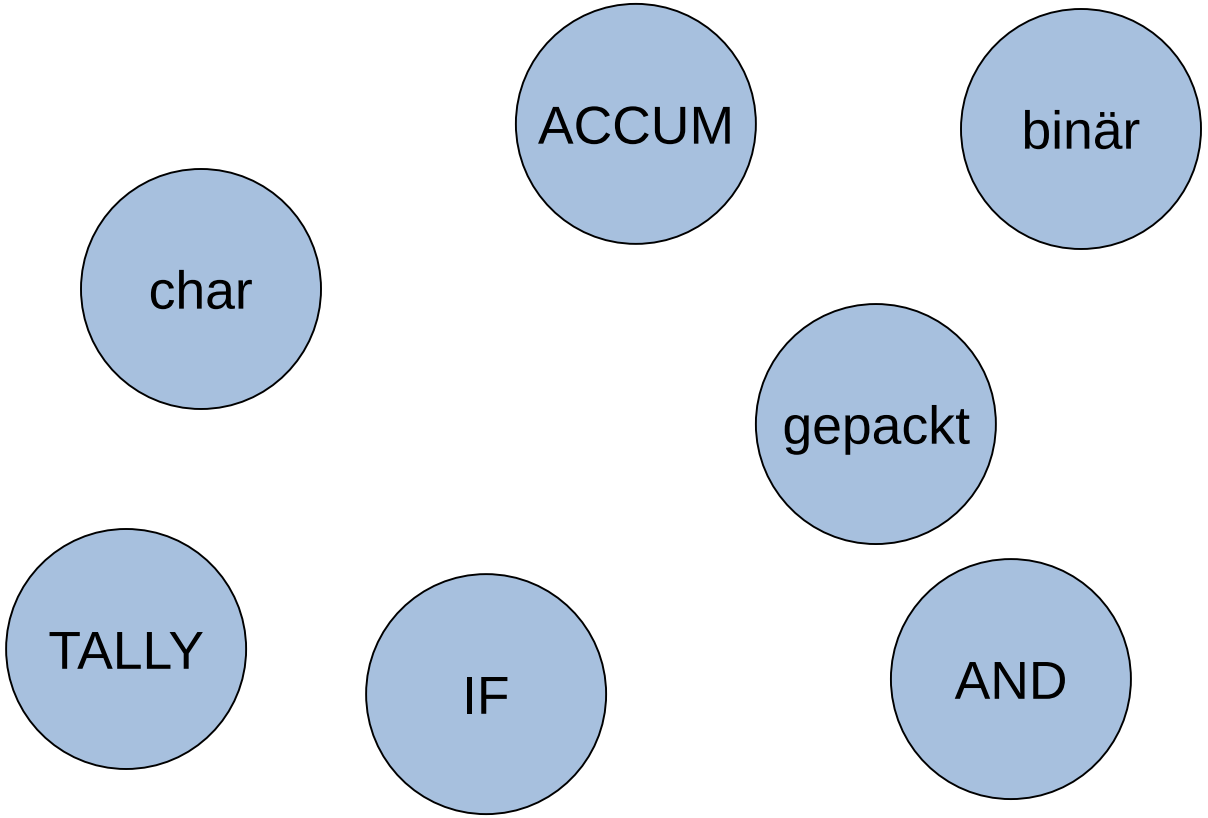
- DDnnCOm
  - Compare-Output (m={1,6}) aus Option 10
- DDnnCHG
  - Datei für Data-Solutions aus Option 10
- MSGLIB
  - Datei für Data-Solutions messages aus Option 10
- anyname
  - beliebiger DD-Name bei Nutzung der Funktion USER mit dem WRITE-Parameter

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch

Syntax

Begriffe



Syntax

ACCUM

- Bin-Felder oder char-Felder
  - ACCUM=(location,length,data-type[,'description'])
- gepackte Felder
  - ACCUM=(location[,'description'])
- mit data-type:
  - C signed or unsigned numeric characters
  - B binary
  - BS binary signed

Syntax

ACCUM - Beispiele

```
$$DD01 COPYALL IF=(43,EQ,C'100'),ACCUM=(46,'BIN-100-WGT'),
               IF=(43,EQ,C'101'),ACCUM=(46,'BIN-101-WGT')

$$DD01 TALLY IF=(10,EQ,C'91'),
             IF=(14,EQ,C'01'),
             ACCUM=(18,4,B,'JANUARY-1991'),
             IF=(10,EQ,C'90'),
             IF=(14,EQ,C'01'),
             ACCUM=(18,4,B,'JANUARY-1990')

$$DD01 TALLY IF=(1,EQ,C'C01'),AND=(178,NE,C'R'),
             OR=(1,EQ,C'D01'),AND=(120,NE,C'R'),
             ACCUM=(2,2,C,'XYZ TOTALS')
```

Syntax

CHANGED / CREATED

```
CHANGED=(from-date,to-date)
        (,to-date)
        (from-date)
        from-date

YY/MM/DD
YY/MM
YY

$$DD01 LIST CREATED=(94/07/19,94/07)
```

Syntax

DFLT_WRITE

```
DW=anyname


$$DD01 USER IF=(86,EQ,C'W'),READNEXT,
 IF=(86,EQ,C'S'),WRITE=OUTS,READNEXT,
 IF=(86,EQ,C'M'),WRITE=OUTM,
 DFLT_WRITE=OUTO

(OUTS, OUTM, OUTO sind DD-Namen)
```

Syntax

DROP

```
DROP=n

n={1,999999999}

$$DD01 DROP IF=(6,EQ,C'ABCDE'),DROP=10

(Es werden die ersten 10 Zeilen weggelassen,
  in denen ABCDE ab Stelle 6 steht.)
```

Syntax

EDIT – 1

```
EDIT=(location,{length},[dupl]compare-data,new-data)
                      {operator}

EDIT=(1,5,C'1234',C'ABCDE')

vorher:
 ----+----1----+----2----+----3
 1234 ABCD999999999999STUVWXYZ
 12346ABCD999999999999STUVWXYZ
nachher:
 ----+----1----+----2----+----3
 ABCDE ABCD999999999999STUVWXY
 ABCDE6ABCD999999999999STUVWXY
```

Syntax

EDIT – 2

```
EDIT=(1,5,C'1234',C'1')

vorher:
----+----1----+----2----+----3
1234 ABCD99999999999ZZZZ
12346ABCD99999999999ZZZZ
nachher:
----+----1----+----2----+----3
1    ABCD99999999999ZZZZ
16ABCD99999999999ZZZZ
```

Syntax

EDIT – 3

```
EDIT=(1,6,C'AAAA',C'')

----+----1----+----2----+----3
AAAAA    12  34  56  789000
AAAAAAAA123 456  78  900000

----+----1----+----2----+----3
A        12  34  56  789000
AAAA123     456  78  900000
```

Syntax

EDIT – 4

```
EDIT=(1,6,C'123',P'+00123')

CHAR 123ABC 123                CHAR    ABC 123
ZONE FFFCCC4FFF                ZONE 013CCC4FFF
NUMR 1231230123                NUMR 02C1230123
     1...5...10                     1...5...10

CHAR AB123C 123                CHAR AB   C 123
ZONE CCFFFC4FFF                ZONE CC013C4FFF
NUMR 1212330123                NUMR 1202C30123
     1...5...10                     1...5...10
```

Syntax

EDITALL

```
EDITALL=(1,50,C'ABC,GHI',C'')

 ----+----1----+----2----+----3
 ABC 999 ABC 999 GHI999 GHI
 ABCABC999GHIGHI999 ABC 999

 ----+----1----+----2----+----3
     999      999 999
 999999                   999
```

Syntax

IF – 1

```
$$DD01 DUMP IF=(23,EQ,C'TEST FILE')

beides gleich (oder-Angaben):
$$DD01 PRINT IF=(1,EQ,C'A',17,EQ,C'1,2,3')
$$DD01 PRINT,
  IF=(1,EQ,C'A',17,EQ,C'1',17,EQ,C'2',17,EQ,C'3')

und-Anweisung:
$$DD01 PRINT IF=(1,EQ,C'A'),IF=(17,EQ,C'1,2,3')

$$DD01 DUMP IF=(23,0,C'TEST FILE')
```

Syntax

IF – 2

```
ab Stelle 6 10 Byte gepackt:
IF=(6,10,EQN)

ab Stelle 20 10 gepackte Felder mit jeweils 5 Bytes:
IF=(20,5,10EQP)

ab Stelle 20 5 gepackte Felder beliebiger Länge
IF=(20,0,5EQP)

Umkehrung von vorher:
IF=(20,0,5NEP)
```

Syntax

IF / AND

```
$$DD01 COPY  IF=(1,0,C'ABC'),
             IF=(1,0,C'XYZ')

$$DD01 COPY  IF=(1,0,C'ABC'),
            AND=(1,0,C'XYZ')
sind gleich
```

Syntax

IF / OR

```
$$DD01 COPY  IF=(25,EQ,C'1'),
             AND=(30,EQ,C'2'),
           ORIF=(16,EQ,C'3'),
             AND=(50,EQ,C'7')

$$DD01 COPY  IF=(21,EQ,C'ABC'),
           ORIF=(41,EQ,C'XYZ'),
             REPL=(61,C'DEF')

$$DD01 TALLY IF=(1,EQ,C'C01'),AND=(178,NE,C'R'),
             OR=(1,EQ,C'D01'),AND=(120,NE,C'R'),
             ACCUM=(2,2,C,'XYZ TOTALS')
```

Syntax

ELSE

```
$$DD01 COPYALL IF=(1,0,C'ABC'),
                 REPL=(1,C'DEF'),
               ELSE,
                 REPL=(1,C'XYZ')
falsch:
$$DD01 COPYALL IF=(1,0,C'ABC'),
                 REPL=(1,C'DEF'),
               ELSE,
                 IF=(1,0,C'DEF')  (not an action)
```

Syntax

IN / OUT / SELECT

```
IN=n
n={1,999999999}
erste 200 Zeilen aus Eingabe kopieren:
$$DD01 COPY IN=200

OUT=n
hexaprint der ersten 25 Sätze
$$DD01 DUMP OUT=25

SELECT=n
jeden dritten Satz nehmen, der Kriterien erfüllt:
$$DD01 PRINT OUT=10,IF=(1,EQ,P'50'),SELECT=3
```

Syntax

MBRNAME

```
MBRNAME=(from-value,to-value
         (,to-value)
         (from-value)
         from-value

$$DD01 LIST MBRNAME=(EMPL,EMPL)

$$DD01 LIST MBRNAME=(,ACCTBZZY)

$$DD01 LIST MBRNAME=ACCTBZZY
```

Syntax

MEMBER

```
MEMBER={name                }
       {(name1,name2...)}

$$DD01 COPY MEMBER=PROG241

$$DD01 COPY MEMBER=(PROG241,PROG242,PROG243)
```

Syntax

MEMBERS

```
MEMBERS={ALL}
        {mask-name}

$$DD01 LIST MEMBERS=GE

$$DD01 PRINT MEMBERS=GE-----P
```

Syntax

PADCHAR

```
PADCHAR={C'c' }
        {X'nn'}

$$DD01 COPY PADCHAR=C'*'

Default: X'00'
```

Syntax

MOVE – 1

Aus Eingabedatei nach Ausgabedatei:

MOVE=(to-location,length,from-location)

```
$$DD01 LIST MOVE=(+0,10,+0)

$$DD01 COPY MOVE=(1,10,15)

$$DD01 USER MOVE=(1,0,10),WRITE=A

$$DD01 USER MOVE=(10,5,30),WRITE=A
```

Syntax

MOVE – 2

Nach Ausgabedatei:

```
MOVE=(to-location,[dupl]data)

$$DD01 COPY MOVE=(1,C'ABC')

$$DD01 COPY MOVE=(1,10C'ABC')

$$DD01 COPY MOVE=(+0,P'+00001')
```

Syntax

REPL

```
REPL=(location,[dupl]new-data)

$$DD01 COPY REPL=(4,C'6')


REPL=(location,{length},[dupl]compare-data,[dupl]new-data)
                {operator}

$$DD01 COPY REPL=(4,EQ,C'2',C'6')

$$DD01 COPY REPL=(4,EQ,C'222',16,C'400')
```

Syntax

WRITE

```
//OUTFILE     DD DSN=XX.XXX
//NEWFILE     DD DSN=YY.YYY
      .
      .
      .

WRITE=(NEWFILE,OUTFILE)
```

Beide DD-Namen werden beschrieben.

Syntax

Operatoren

- Elemente sind char, gepackt, hexa

Element	Condition
EQ	Equal to
NE	Not equal to
GT	Greater than
LT	Less than
GE	Greater than or equal to
LE	Less than or equal to

Syntax

Operatoren

- Elemente sind binär

Element	Condition
EQ	Bits are all ones
NE	Bits are all zeros
NO	Bits are not all binary ones (all zeros or mixed)
MX	Bits are mixed (ones and zeros)

Syntax

Datenelemente

Identifier	Data Type	Example
C	Character, alphanumeric	C'ABCD123'
T	Text, alphanumeric	T'ABC'
X	Hexadecimal	X'10CF00'
P	Packed	P'+1'
B	Binary mask	B'01001000'
B	Binary OR (REPL new-data only)	B'01001000'
BM	Binary minus (REPL new-data only)	BM'C8'
BS	Binary signed (ACCUM parameter only)	BS'01001000'
BX	Binary exclusive OR (REPL new-data only)	BX'C8'

Syntax

weitere Beispiele

- siehe separate Unterlagen

Inhalt

- Einführung und Überblick
- Produktelemente
- Konventionen und Funktionen (1)
- Konventionen und Funktionen (2)
- PO-Dateien und Platten
- weitere Funktionen und Parameter
- JCL und Dateien
- Syntax in Auswahl
- Diskussion und Austausch