

Xpediter/CICS

Grundlagen

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

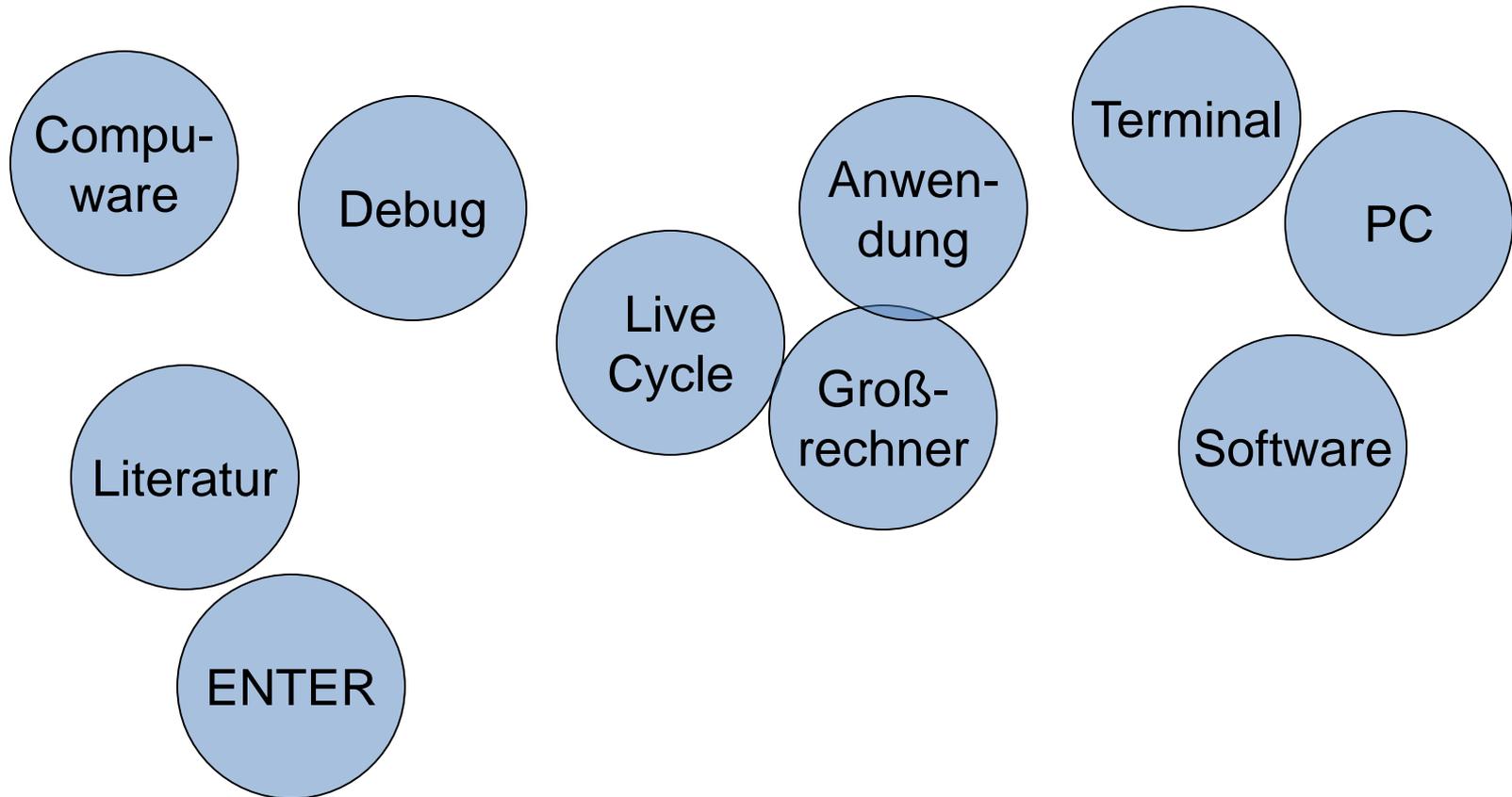
Internet: <http://www.cps4it.de>

-
- Funktionen von Xpediter/CICS beherrschen
 - Fehler mit Xpediter/CICS finden
 - Befehle in Xpediter/CICS beherrschen

 - Files und Datenbanken ansehen und bearbeiten
 - Besonderheiten

-
- Seite 5: Einführung und Überblick
 - Seite 17: Einstieg und Umwandlungen
 - Seite 29: Menüs in Xpediter – Überblick
 - Seite 37: COBOL-Abbruch ohne und mit Xped
 - Seite 49: zusätzliche Debugging-Menüs
 - Seite 81: File-Utility
 - Seite 109: Execution Coverage
 - Seite 117: Unterprogramme
 - Seite 125: Storage Protection
 - Seite 133: Scripting Facility

-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-



- 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.

- XPEDITER/CICS Quick Reference
- XPEDITER/CICS Reference Manual
- XPEDITER/CICS COBOL User's Guide
- XPEDITER/CICS Messages and Codes
- Bookmanager im Hause
- <http://frontline.compuware.com>
 - > XPEDITER/CICS
 - > Technical Reference
 - > Tips and Tricks
- <http://wiki.cps4it.de>



- Seit vielen Jahren auf dem Markt
- Software wurde gekauft
- Angleichung Xped/CICS an Xped/TSO/IMS
- Zielumgebung Großrechner
- Zielgruppe Anwendungsentwicklung

wesentliche Neuerungen (1)

- Rel 7.4
 - Unterstützung neue CICS / IMS / DB2 Releases
- Rel 7.5
 - CICS TS 2.2, WS MQ in FILE, Register, ASM, IMS 8.1
- Rel 7.6
 - Autokeep, CICS TS 2.3, WHEN, Browser, IMS 9.1
- Rel 7.7 (GA: Januar 2005)
 - CICS TS 3.1, OTE-Support, IEEE, 64-bit (Beginn), Code Coverage
- Rel 8.0 (GA: März 2006)
 - C, 64-bit, TRACE, INTERCEPT

wesentliche Neuerungen (2)

- Rel 8.1 (GA: August 2007)
 - Debug ohne Source, Intercept via Menü, File mit PL1, Profil, DLL, Anzeige Tabellen einfacher,
 - DUP, DDOWN, DTOP, DBOTTOM, KEEP
 - WHEN
 - GO nnn TRACE
- Rel 8.2 (GA: März 2008)
 - RUN TO (mit Cursor setzen)
 - Längenangabe bis 31 Stellen

wesentliche Neuerungen (3)

- Rel 8.3 (GA: Oktober 2008)
 - Cross-LPAR
 - newcopy
 - Wildcards auf 1.N erlaubt
 - Anzeige „konkurrierender“ User
 - im Batch aufrufbar
 - MQ in File-Utility
 - MONITOR heißt jetzt PROTECT
- Rel 8.3.1 (GA: März 2009)
 - GUI: neue Features
 - UTF-16 Unterstützung

wesentliche Neuerungen (4)

- Rel 9.0 (GA: 2010)
 - Onetime Breakpoint
 - Newcopy Interface
 - Hexanzeige verbessert
- Rel 9.1 (GA: Juni 2011)
 - Monitor/Reverse wie Xped TSO
- Rel 9.2 (GA: Juni 2012)
 - Xped/Eclipse

wesentliche Neuerungen (5)

- Rel 9.3 (GA: März 2014)
 - COBOL 5.1
 - LPAR-Einschränkung in Eclipse aufgehoben
- Rel 9.4 (GA: November 2014)
 - Watch Variable ohne Variablen suchen

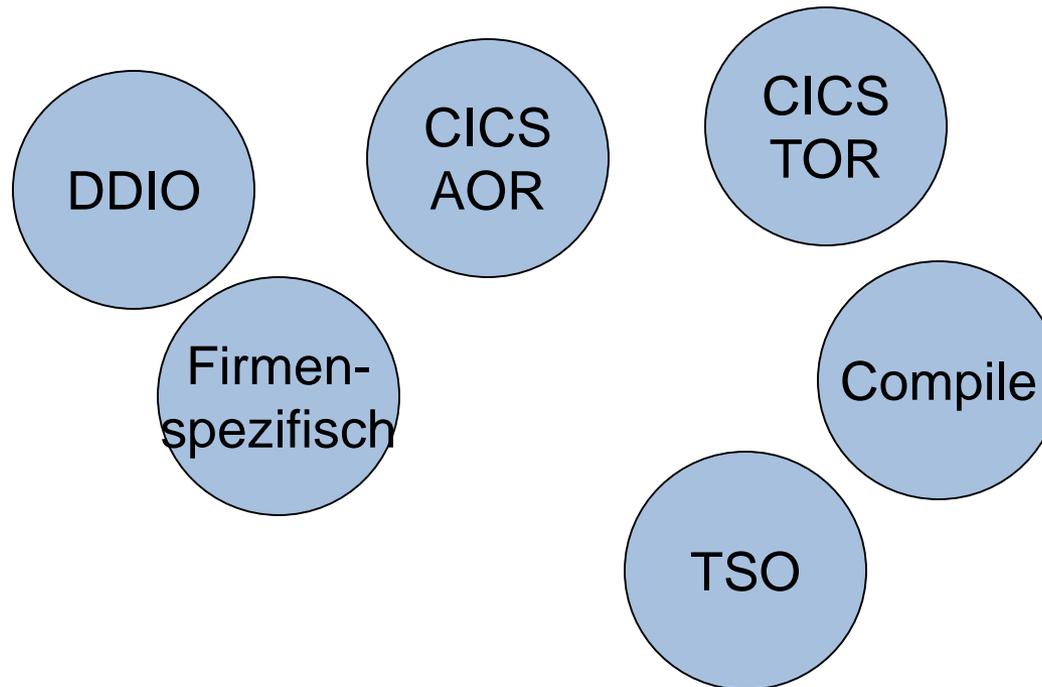
- Brainstorming / Diskussion
- Ausgangslage 1: neues Projekt - Wasserfallmodell
 - Frage: Wann ist Debugging sinnvoll?
- Ausgangslage 2: Wartungsprojekt
 - Frage: Wann ist Debugging sinnvoll?
- Ausgangslage 3: Fehlersituation
 - Frage: Wann ist Debugging sinnvoll?



- 01-01 - keine

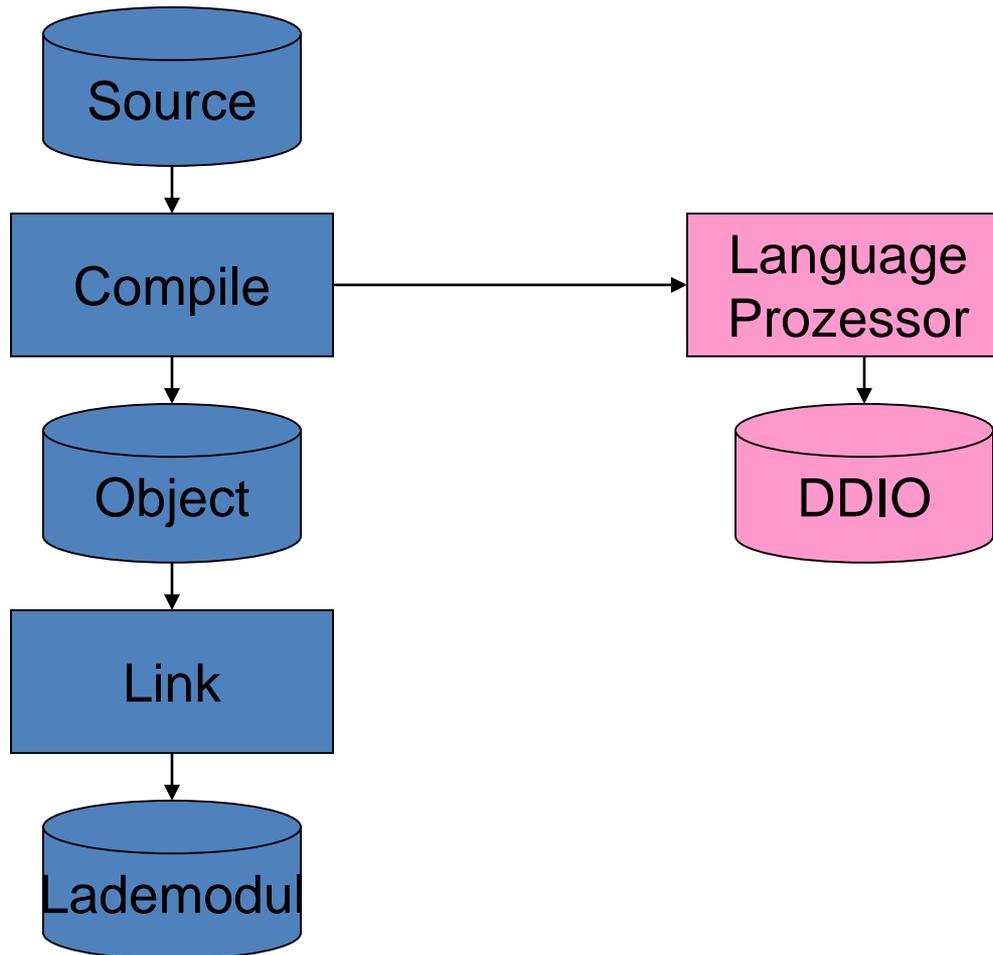


-
- Einführung und Überblick
 - • Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-



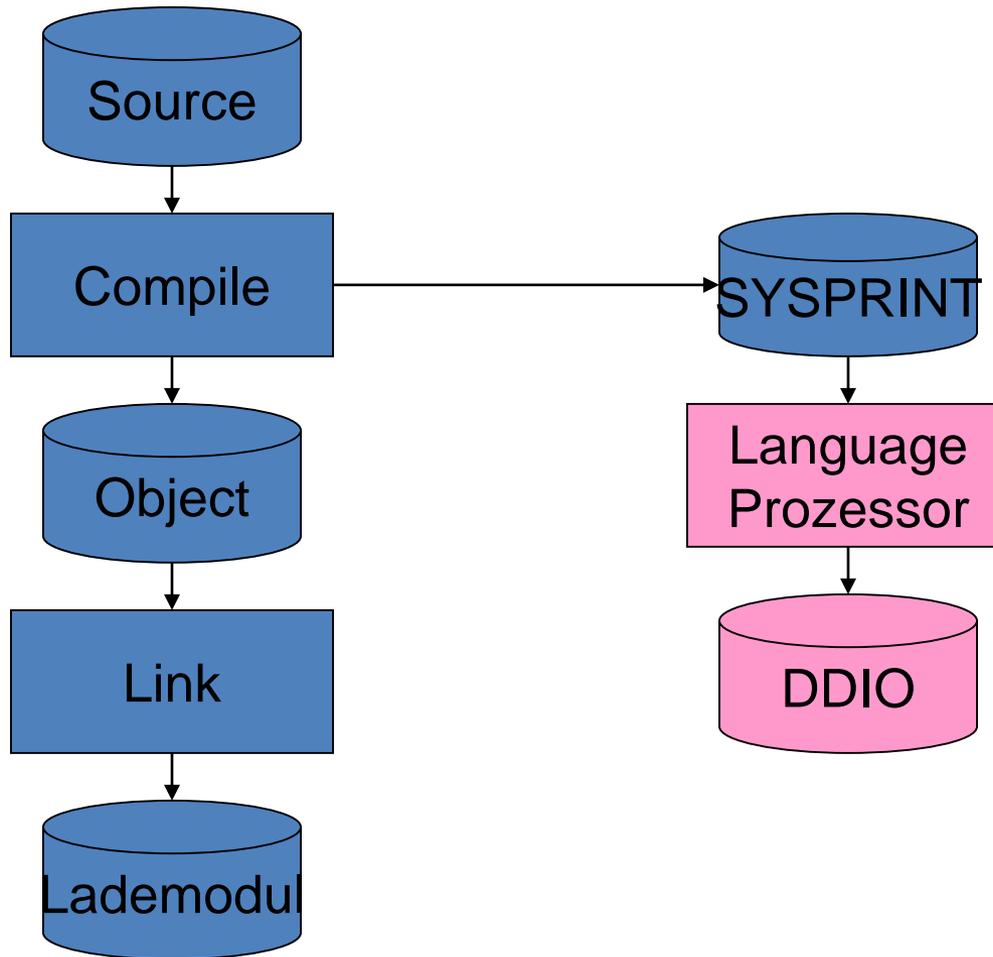
Einstieg und Umwandlungen

Pre-Compile Prozess



Einstieg und Umwandlungen

Post-Compile Prozess



wo tut man was?

- TSO / Client
 - Erstellen Source
 - Umwandlung
- CICS
 - Test der Anwendung
 - Debug
- Eclipse
 - Debug

Einstieg und Umwandlungen

die CICS-Umgebung im Überblick

Link



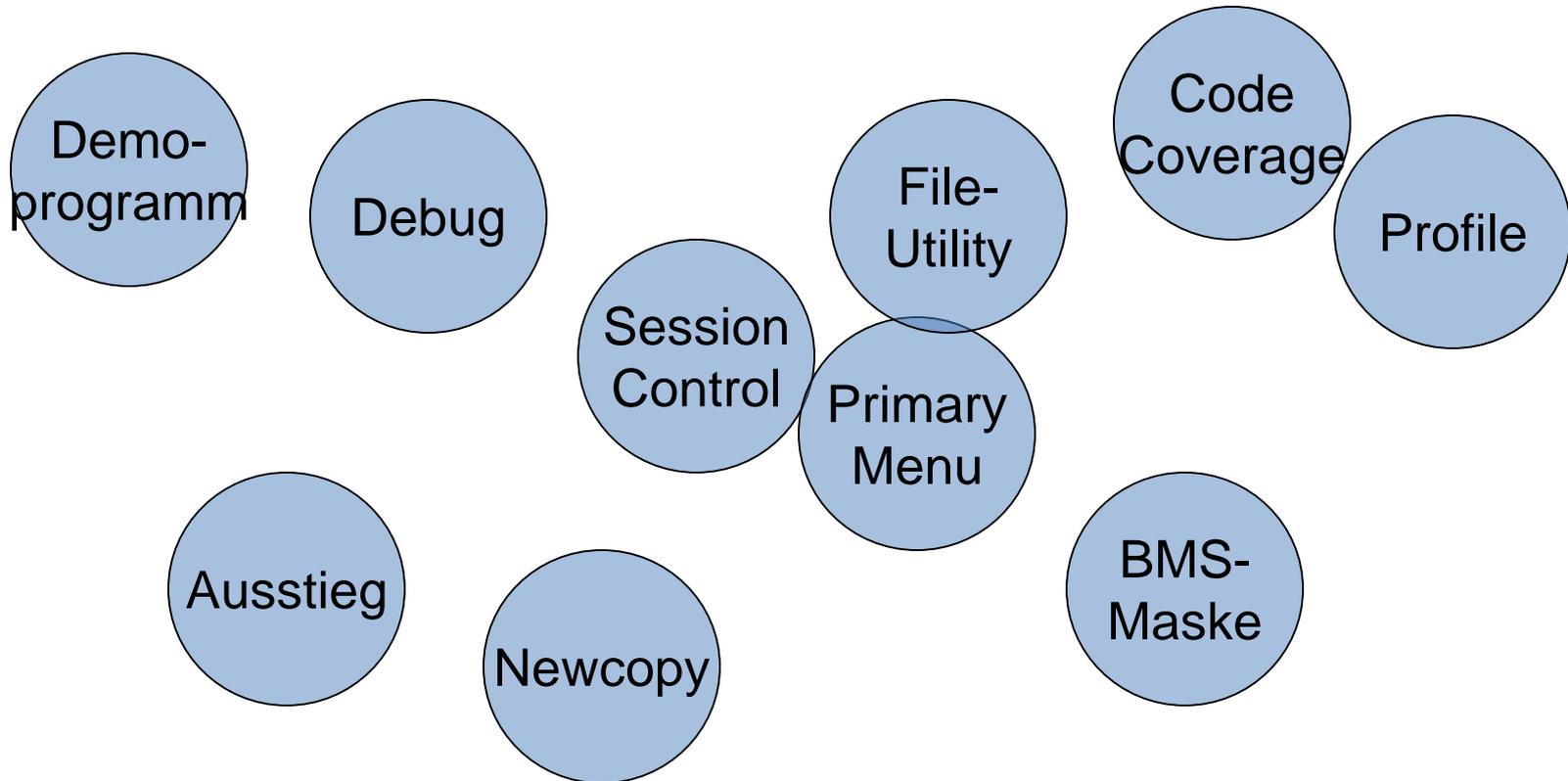
Übung(en)

- 02-01 – Test der Schulungs-ID-en
 - Einstieg in Netz, Netmaster, TSO, CICS
- 02-02 – Test der eigenen User-ID-en
 - Einstieg in Netz, Netmaster, TSO, CICS
- 02-03 – Umwandeln eines Moduls mit Xpediter
- 02-04 – Aufruf von Xpediter



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - • Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-

Begriffe



- Session Profile
- Session Control
- Debugging Facilities
- File Utility
- CICS Abend-Aid/FX
- Code Coverage

Einzelheiten später

wichtige Transaktionen

- XPED (Firma rrr im Normalfall: nXPE)
 - Aufruf von Xpediter
- NEWC (Firma rrr im Normalfall: nNEW)
 - neue Kopie des Programms zur Verfügung stellen
- DMAP
 - Anzeige der letzten Maske
- XPND (Firma rrr im Normalfall: nXPN)
 - Ausstieg aus Xpediter
- XCB2
 - Demotransaktion



Übung(en)

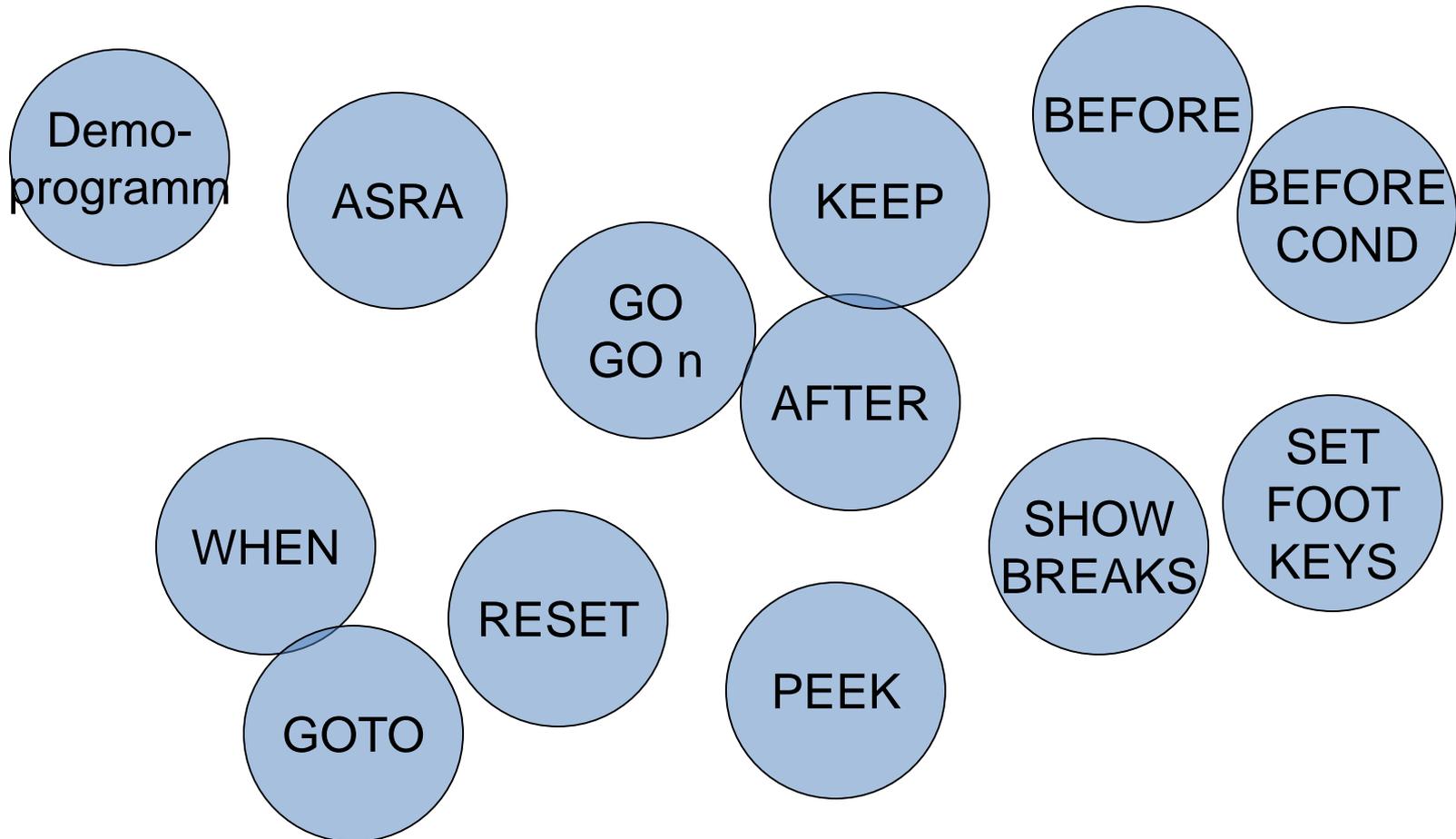
- 03-01 – einmaliger Aufruf von
 - allen Menüs im Primary Panel
- 03-02 – Aufruf der Transaktion NEWC
- 03-03 – Aufruf der Transaktion DMAP
- 03-04 – Aufruf der Transaktion XPND
- 03-05 – Aufruf der Transaktion XCB2
 - Ausstieg mit 00099



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - • COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-

COBOL-Abbruch mit und ohne Xpediter

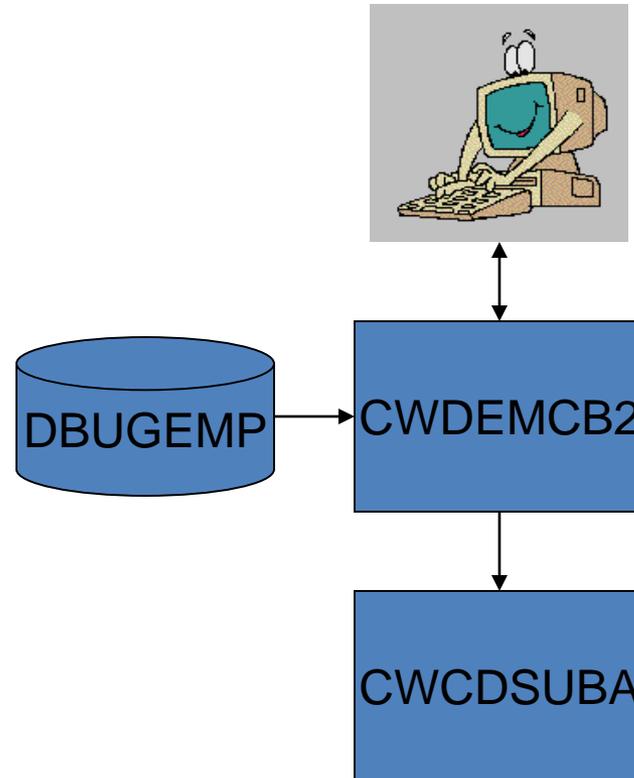
Begriffe



COBOL-Abbruch mit und ohne Xpediter

das Demoprogramm

- Aufruf XCB2



- **GO n**
 - n Schritte weiter
- **KEEP**
 - ständige Anzeige einer Variablen
- **BEFORE / BEFORE CONDITIONAL**
 - Breakpoint vor einem Befehl
- **AFTER**
 - Breakpoint nach einem Befehl
- **SHOW BREAKS**
 - Anzeige aller Zeile mit Breakpoints

- **RESET**
 - Display zurück setzen
- **WHEN**
 - Stop Execution wenn Bedingung oder wenn Variablenwert verändert
- **GOTO**
 - Execution Pointer versetzen
- **PEEK**
 - Variable / Struktur anzeigen

- SET FOOT KEYS
 - Anzeige der PF-Tasten
- TRACE
 - Anzeige Program Flow
- SOURCE
 - “normale” Anzeige Source

was wollen wir nun tun?

- ASRA erzeugen
- Xpediter benutzen, um ABEND zu lokalisieren
- Xpediter benutzen, um Grund des ABENDs zu erkennen
- Programmkontrolle übernehmen, um (temporär) den ABEND zu lösen
- Programm ohne ABEND beenden lassen



- Sinnvolle Optionen für Testumfeld:
 - NOOPTIMIZE
 - SSRANGE
- Sinnvolle / notwendige Optionen für Produktion
 - OPTIMIZE(FULL)
 - NOSSRANGE
 - NUMPROC(PFD/NOPFD)
 - TRUNC(OPT)
- und
 - DATA(31), AMODE(31), RMODE(ANY)



Übung(en)

- 04-01 – Führen Sie die vorgeführten Schritte an der Demotransaktion durch.
 - ASRA erzeugen
 - Xpediter benutzen, um ABEND zu lokalisieren
 - Xpediter benutzen, um Grund des ABENDs zu erkennen
 - Programmkontrolle übernehmen, um (temporär) den ABEND zu lösen
 - Programm ohne ABEND beenden lassen
 - Nutzen Sie die vorgeführten Befehle
 - “Spielen” Sie mit den Befehlen
 - Verändern Sie die Anzeige Ihres Bildschirms



Übung(en)

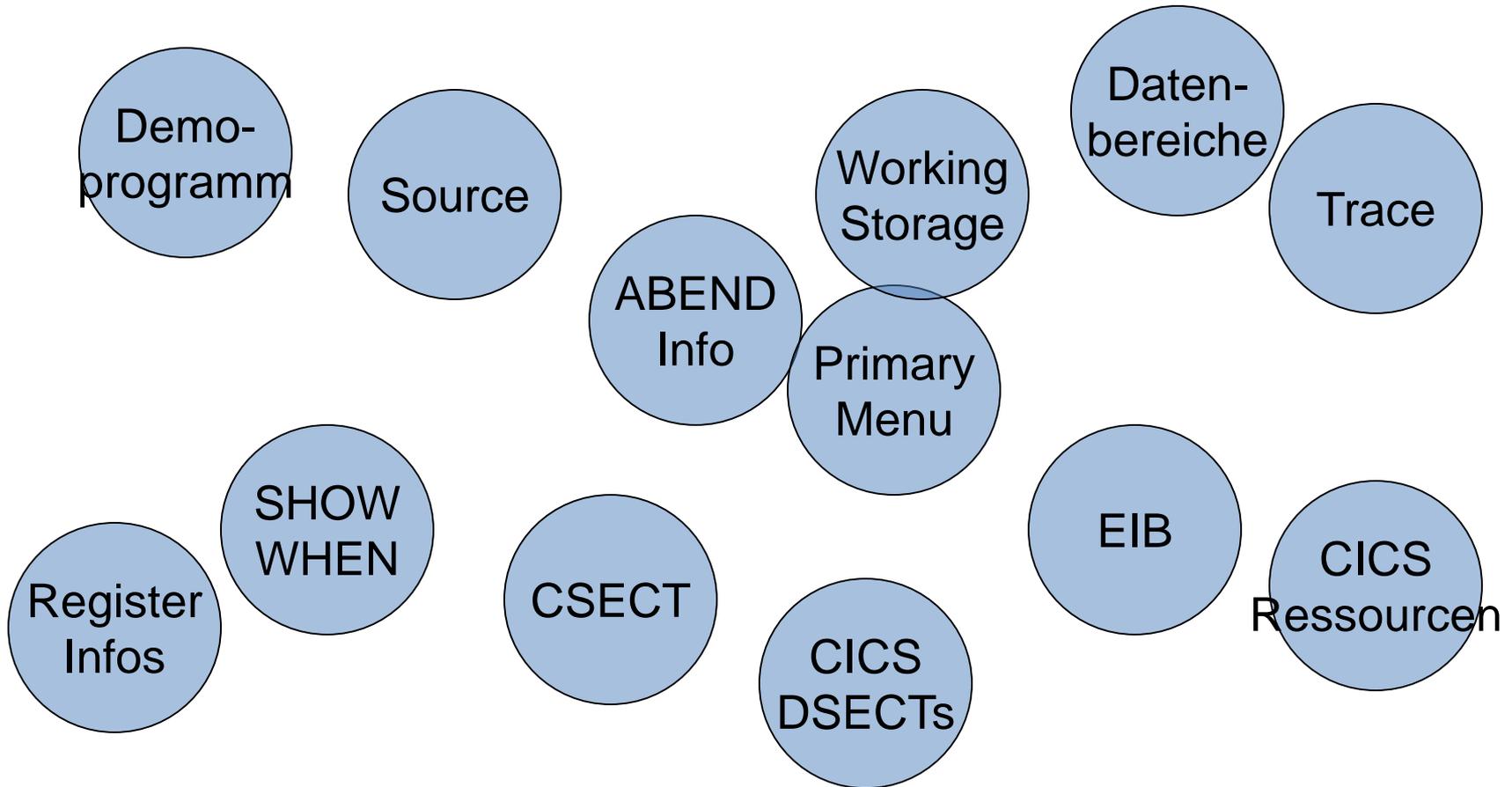
- 04-02 – Führen Sie die vorgeführten Schritte an einer eigenen Transaktion durch.
 - Erzeugen Sie einen Fehler (oder auch nicht).
 - Xpediter benutzen, um ABEND zu lokalisieren
 - Xpediter benutzen, um Grund des ABENDs zu erkennen
 - Programmkontrolle übernehmen, um (temporär) den ABEND zu lösen
 - Programm ohne ABEND beenden lassen
 - Nutzen Sie die vorgeführten Befehle
 - “Spielen” Sie mit den Befehlen
 - Korrigieren Sie den Fehler im Code.



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - • zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-

Zusätzliche Debugging Menüs

Begriffe



Auswahlmöglichkeiten – 1

- L – Source Listing – schon bekannt
- 4 – Program Trace – schon bekannt
- 7 – SHOW WHEN – schon bekannt
 - (und wird später noch einmal angerissen)
- 2 – Memory Display
 - Lademodul
 - Commarea
 - CSA (Common System Area)
 - Möglichkeiten der Auswahl siehe HELP

Auswahlmöglichkeiten – 2

- D – CICS DSECTs
 - CSA
 - PCT
 - Möglichkeiten der Auswahl siehe HELP

- R – CICS Ressourcen d.h Informationen über
 - Programm
 - Transaction
 - DB2-Verbindungen im aktuellen Programm



Übung(en)

- 05-01 – Testen Sie im Debugging Menü die Optionen
 - L - Source Listing
 - 4 - Program Trace
 - 7 - SHOW WHENPrüfen Sie, ob Sie bei allen Optionen die erwarteten Resultate erhalten.
- 05-02 – Testen Sie im Debugging Menü die Optionen
 - 2 - Memory Display
 - D - CICS DSECTs
 - R - CICS Ressourcen



- 3 – Program Data Area – d.h. Working Storage
 - Befehle möglich wie
 - f(ind)
 - hex / hex off / hex dump
 - show dataonly / show attr / show qual / show offset / show data
 - weitere Befehle siehe HELP
 - Anzeige und Überschreiben von Occurs-Stufen



Übung(en)

- 05-03 – Testen Sie im Debugging Menü die Option
 - 3 – Program Data Area
 - Testen Sie verschiedene Befehle, um zu positionieren.
 - Schauen Sie sich die Working Storage an und zeigen Sie sich die Variablen in den verschiedenen Möglichkeiten an.
 - Suchen Sie nach OCCURS-Feldern und zeigen Sie die verschiedenen Occurences an.
 - Nutzen Sie den Help-Befehl.



- 5 – CICS Exec Interface Block
- 6 – CSECT Display (Teile des Lademoduls)
 - Liste der CSECTs
 - Auswahl einer CSECT
- 8 – letzter 3270 Bildschirm



Übung(en)

- 05-04 – Sehen Sie sich nachfolgende Optionen an
 - 5 – CICS Exec Interface Block
 - 6 – CSECT Display (Teile des Lademoduls)
 - 8 – letzter 3270 Bildschirm
 - Prüfen Sie, ob Sie das erwartete Ergebnis erhalten.
 - Nutzen Sie den Help-Befehl soweit vorhanden.
 - Ist es möglich, verschiedene Funktionstasten zu nutzen?
Sehen Sie diese im EIB?
 - Sehen Sie in der Funktion 8 die verschiedenen Eingaben, die Sie gemacht haben?



- 20 – Registerstand bei derzeitigem Befehl
 - Anzeige des ASM-Befehls
 - Auch verfügbar ohne Xpediter-Umwandlung!
 - Befehle sind verfügbar wie
 - GO (ASM-Schritte!), SET FOOT SOURCE, SET FOOT DATA, PEEK (für Register, für PSW), SET FOOT OFF
- 1 – Break bzw. ABEND Informationen
- nur im Abbruch-Fall die “bekannte Anzeige”



Übung(en)

- 05-05 - Sehen Sie sich die folgende Option an
 - 20 – ASSEMBLER BREAK/ABEND
 - Prüfen Sie, ob Sie das erwartete Ergebnis erhalten.
 - Nutzen Sie den Help-Befehl soweit vorhanden.
 - Steppen Sie durch einige Zeilen des Codes.



Auswahlmöglichkeiten – 1

- 1 - Session Control
 - 1 – List Breakpoints
 - 3 – List ABENDs
 - 4 – Trace Summary
 - 6 – Trap Summary
 - 8 – Storage Protection
 - 9 – User Labels
 - P – Ressource Summary

Auswahlmöglichkeiten – 2

- 0 – Session Profile
 - 1 – Defaults
 - 2 – Keys
 - 3 – Create Profile
 - 4 – Load Profile
 - 5 – Save Profile
 - 6 – Script DSN



Übung(en)

- 05-06 – Sehen Sie sich die folgende Optionen an
 - 0 – Session Profile
 - 1 – Session Control
- “Spielen” Sie ein wenig mit den Einstellungen.
- Speichern Sie ein Profil und lassen Sie eine/n Kolleg/i/en diese aufrufen.



Ausstieg

- Beenden der Xpediter-Session
 - =x in allen Panels
 - XPND auf leerem Schirm

Übung(en)

- 05-07 – Beenden Sie die “ASRA-Transaktion”
- 05-08 – Beenden Sie die Xpediter-Session
 - 0 – Session Profile
 - 1 – Session Control
- 05-09 – Bitte überlegen/prüfen Sie,
 - welche Befehle Ihnen für Ihre Debugging-Session noch fehlen.
 - ob Ihnen für Ihre Debugging-Session noch ein Hilfsmittel fehlt.



Zusätzliche Debugging Menüs

alle Line Commands auf einen Blick

A	Breakpoint after	L	anzeigen letzte x-Zeile
AC	Breakpoint after cond	M	verschiebe Zeile
B	Breakpoint before	P	gehe zur Working
BC	Breakpoint before cond	R	lösche alle Breakpoints
C, CC	setze execution count	S, SS	skip Zeile(n)
D, DD	lösche Befehle auf Zeilen	SC	skip Zeile cond
DA	lösche breakpoint after	U	unlock entry
DB	lösche Breakpoint before	V, VV	ändern ASM-Code
DC	lösche count	X, XX	excluden Zeile(n)
DK	lösche keep		
DS	lösche skip		
DV	lösche verify		
F	anzeigen erste x-Zeile		
GT	springe hierher		
K	zeige Variable an		

Zusätzliche Debugging Menüs

alle Primary Commands auf einen Blick

=	DLEFT	HELP	PFSHOW	SET
ABEND	DOWN	HEX	PREV	SHOW
ACCESS	DRIGHT	INCLUDE	PRINT	SKIP
AFTER	DS	INTERCEPT	PSEUDOCODE	SOURCE
ALLOW	DTOP	KEEP	REDISP	TOP
BEFORE	DUP	KEYS	RELEASE	UNLOCK
BOTTOM	ELEMENT	LEFT	REPEAT	UP
CALC	END	LIBRARY	RESAVE	VERIFY
CAPTURE	EXCLUDE	LOAD	RESET	VS
CHAIN	EXIT	LOCATE	RESTORE	WHEN
COLS	FILE	LOCK	RESUME	WS
COUNT	FIND	MEMORY	RETRIEVE	
DBOTTOM	GO	MENU	RFIND	
DDOWN	GOTO	MOVE	RIGHT	
DELETE	GPREGS	NEXT	SAVE	
DISPLAYP	GROUP	PEEK	SELECT	

- Nutzen Sie die HELP-Funktion.
- Schauen Sie in die Broschüren.

- Probieren Sie immer wieder mal etwas Neues aus.

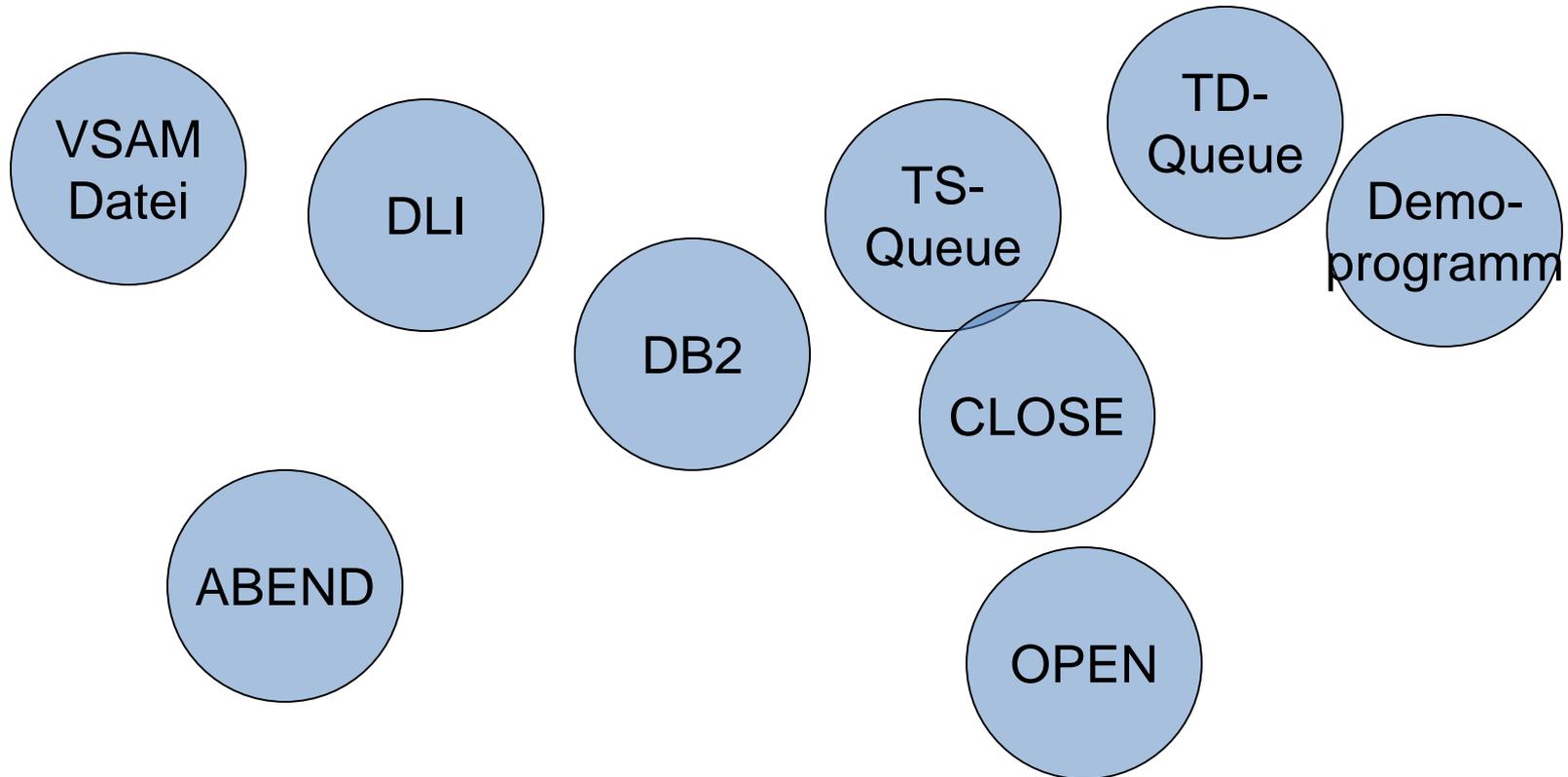
- Seien Sie mutig; Sie können nichts kaputt machen.

Übung(en)

- 05-10 – Testen Sie Ihren Zugang zum Bookmanager.
- 05-11 – Sehen Sie sich an, welche Broschüren vorhanden sind.
- 05-12 – Schauen Sie in der Quick Reference die Syntax für den Befehl BEFORE an. Testen Sie verschiedene Einstellungen an Hand des Demoprogramms.



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - • File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-



- VSAM-Dateien anzeigen und ändern
- Zugriff auf DL/I
- Zugriff auf DB2
- TS-Queue anzeigen und ändern
- TD-Queue anzeigen

AEIM – was wollen wir nun tun?

- AEIM (Record not found) erzeugen
- Xpediter benutzen, um den ABEND zu lokalisieren
- Xpediter benutzen, um Grund des ABENDs zu erkennen
- Programmkontrolle übernehmen, um (temporär) den ABEND zu lösen
- Programm ohne ABEND beenden lassen



AEIM – was haben wir in Xpediter getan? – 1

- Abbruch lokalisiert
- File-Menü aufgerufen / Liste der Datasets
- locate auf DBUGEMP
- browse Dataset
- Key ansehen
- in Source Key ändern
- Programm weiterlaufen lassen

AEIM – was haben wir in Xpediter getan? – 2

- Abbruch lokalisiert
- File-Menü aufgerufen / Liste der Datasets
- locate auf DBUGEMP
- Dataset schließen / ADD und DEL zulassen / öffnen
- Dataset browsen
- einen Record auswählen
- Key ändern und speichern
- Programm weiterlaufen lassen

AEIP – was wollen wir nun tun?

- AEIP (invalid request) erzeugen
- Xpediter benutzen, um den ABEND zu lokalisieren
- Xpediter benutzen, um Grund des ABENDs zu erkennen
- Programmkontrolle übernehmen, um (temporär) den ABEND zu lösen
- Programm ohne ABEND beenden lassen



AEIP – was haben wir in Xpediter getan?

- Abbruch lokalisiert
- File-Menü aufgerufen / Liste der Datasets
- locate auf DBUGEMP
- Dataset schließen / UPD zulassen / öffnen
- browse Dataset
- Key ansehen
- in Source Key ändern
- Programm weiterlaufen lassen

Übung(en)

- 06-01 – Teilübung, da konkurrierender Zugriff!!
 - Erzeugen Sie den AEIM, ändern Sie den Zugriffskey im Programm entsprechend und beenden Sie die Transaktion ohne ABEND. Überspringen Sie dabei den AEIP.
- 06-02 – Alternative
 - Beispiel(e) aus den Projekten?



- Mappen Datensatz auf Struktur im Programm
 - USING Command
 - Anzeigeformate wie HEX ON, HEX DUMP
 - blättern in der Datei mit NEXT und PREV
 - Generisches Lesen mit READ (und blanks)



- 06-03 - Mappen Datensatz auf Struktur im Programm
 - Testen Sie das Mappen der Datei DBUGEMP auf die entsprechende Struktur im Testprogramm.
 - Probieren Sie die verschiedenen Einstellungen für die Anzeige aus.
- 06-04 - Generisches Lesen mit dem File-Utility
 - Lesen Sie die Datei DBUGEMP beliebig generisch.



TS-Queue / TD-Queue – was wollen wir tun?

- Liste der (eigenen) TS-Queues anzeigen
- TS-Queue browsen
- Satz selektieren und in weitere TS-Queue schreiben
- TS-Queues anlisten und erzeugte wieder löschen

- Liste der TD-Queues anzeigen lassen



- 06-05 - Arbeiten mit TS-Queue
 - Listen Sie die TS-Queues auf
 - Schreiben Sie einen Satz oder mehrere Sätze in eine weitere TS-Queue mit dem Namen ihrer User-ID.
 - Zeigen Sie die TS-Queue an, verändern Sie diese und löschen Sie sie später wieder.
- 06-06 - Arbeiten mit TD-Queue
 - Listen Sie die TD-Queues auf
 - Schauen Sie sich eine TD-Queue an.



- besondere Rechte sind im Allgemeinen erforderlich
- Möglichkeiten
 - Liste der PSBen anzeigen
 - zugehörnde DBDs auflisten
 - Segmentliste anzeigen
 - Inhalt der Seqmente anzeigen
 - SSA anzeigen
 - PSB schedulen
 - DLI-Code absetzen



- besondere Rechte sind im Allgemeinen erforderlich
- Möglichkeiten
 - Auflisten der Tables zu einem Creator
 - Table auswählen, Row auswählen und Inhalte anzeigen
 - Query absetzen mit WHERE-Angaben
 - generierte SQL anzeigen
 - in SQL-Transfer-Dataset schreiben



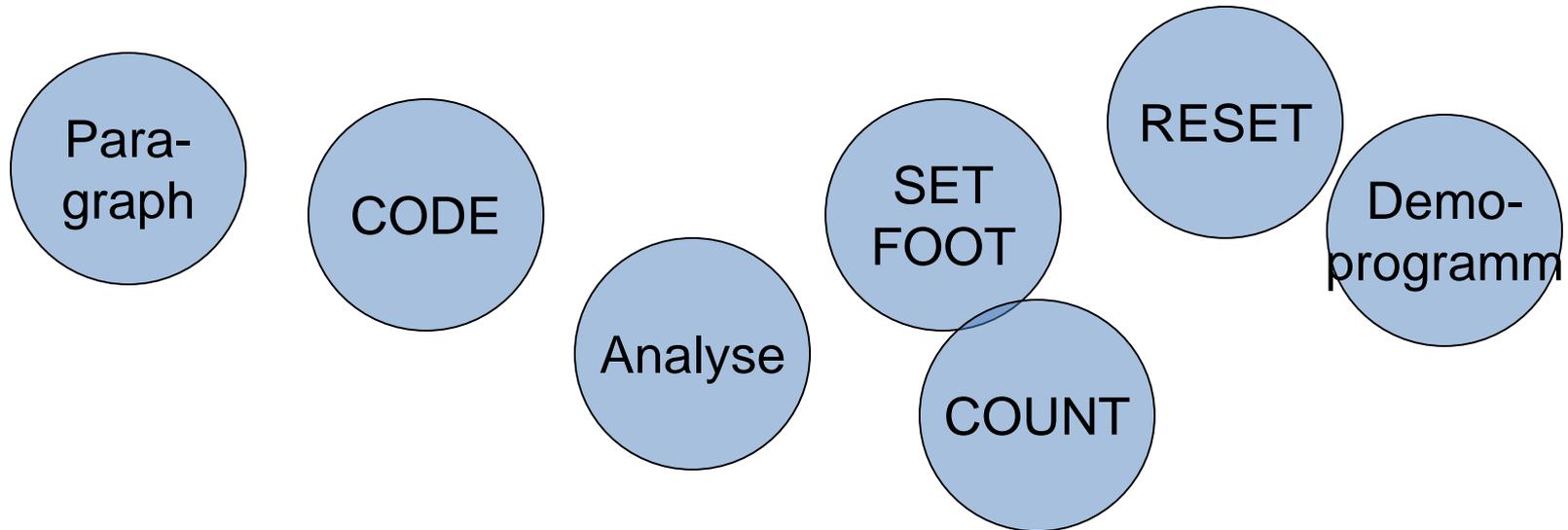
Übung(en)

- 06-07 - Arbeiten mit DL/I-Datenbanken
 - Probieren Sie – soweit möglich – die vorgeführten Funktionen aus.
- 06-08 - Arbeiten mit DB2-Tabellen
 - Probieren Sie – soweit möglich – die vorgeführten Funktionen aus.



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- ➔ • Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility
-

Begriffe



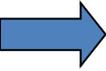
Überblick

- Xpediter kann benutzt werden, um den Programmdurchlauf zu überprüfen. Dadurch sollen nicht ausgeführter Code und logische Pfade erkannt werden.
- zugehöriger Command: COUNT



- 07-01 - Code Coverage
 - Testen Sie die Code Coverage mit dem Demoprogramm bei fehlerfreiem Durchlauf.
 - Testen Sie die Code-Coverage bei einem fehlerhaften Durchlauf. Ist das sinnvoll?



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 -  Unterprogramme
 - Storage Protection
 - Scripting Facility
-

PPT

statisch
gelinkt

dynamisch
gelinkt

CSECT

Demo-
programm

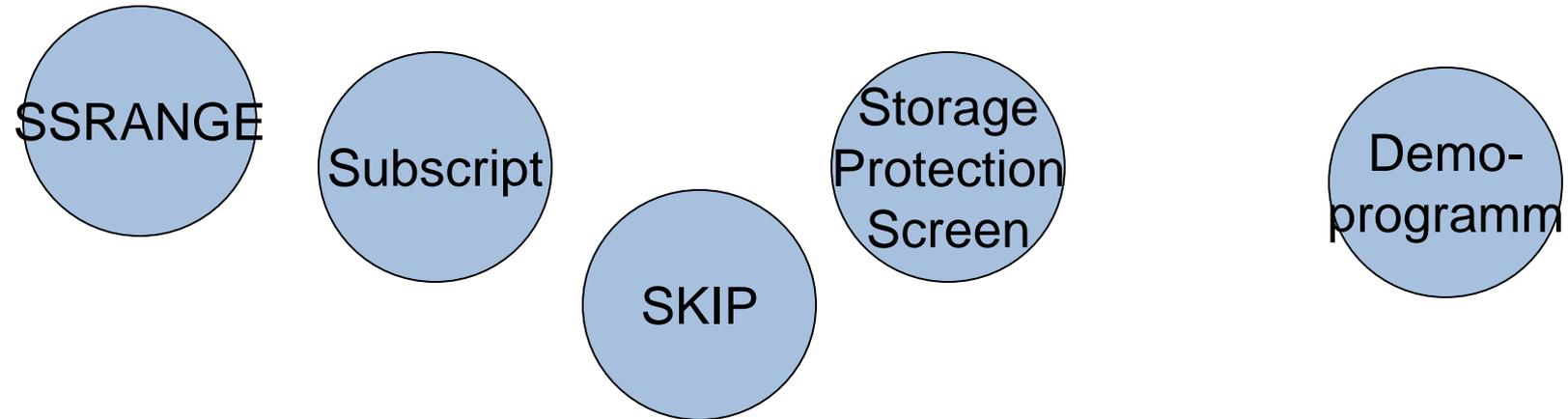
- Xpediter kann benutzt werden, um dynamisch gelinkte Programme zu debuggen.
 - Vorgehen wie bei Hauptprogramm.
- Xpediter kann benutzt werden, um statisch gelinkte Programme zu debuggen.
 - Vorgehen: Auswahl über 2.6.1
 - Danach: wie bekannt



- 08-01 – statisch gelinktes Unterprogramm
 - Setzen Sie einen Breakpoint auf dem statisch gelinkten Unterprogramm zur Demotransaktion.
 - Rufen Sie die Demotransaktion mit 00005 auf und testen Sie.
- 08-02 – dynamisch gelinktes Unterprogramm
 - Setzen Sie – soweit vorhanden – einen Breakpoint auf ein eigenes Unterprogramm.



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - • Storage Protection
 - Scripting Facility
-



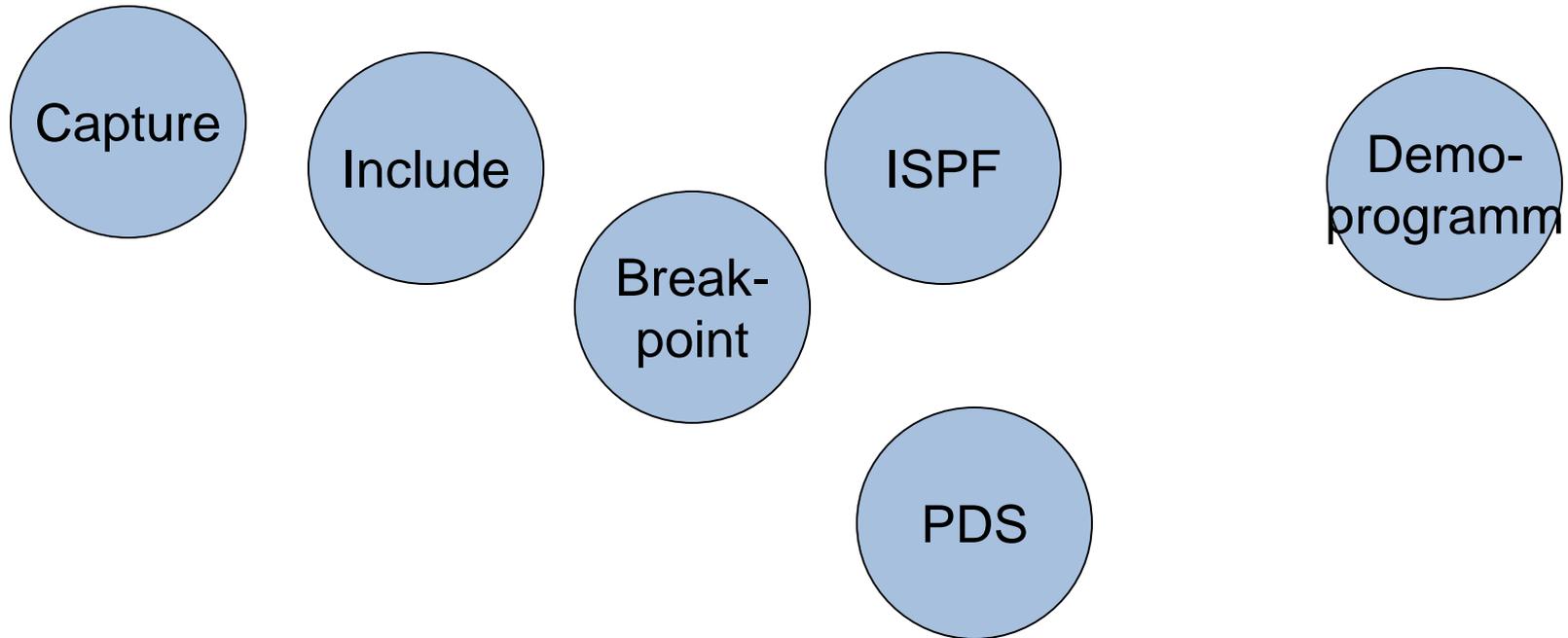
- Xpediter kann benutzt werden, um temporär eine Speicherverletzung anzuzeigen.
- Dies entspricht in etwa der COBOL Compile Option SSRANGE, die für Test empfohlen wird.
- Vorgehen:
 - Auswahl über 1.8
 - Storage Protection Option setzen
 - Transaktion laufen lassen
 - entsprechend reagieren
 - Storage Protection Option aufheben



- 09-01 - Storage Protection verhindern
 - Setzen Sie die Storage Protection Option für Ihren Bildschirm.
 - Rufen Sie die Demotransaktion mit 00333 auf und testen Sie.
 - Reagieren Sie entsprechend.



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility



- Wieder kehrende Aktionen können aufgezeichnet und wieder verwendbar in einer Datei abgelegt werden.
- Als Datei wird ein PDS angelegt, die über ISPF angesehen und auch verändert werden kann.



- 10-01 - Script anlegen
 - Zeichnen Sie eine beliebige Sequenz von Xpediter-Befehlen auf.
 - Legen Sie diese in einem Script ab.
 - Testen Sie den Script.
 - Testen Sie einen Script einer Kollegin / eines Kollegen.



-
- Einführung und Überblick
 - Einstieg und Umwandlungen mit Xpediter
 - Menüs in Xpediter – Überblick
 - COBOL-Abbruch ohne und mit Xpediter
 - zusätzliche Debugging-Menüs
 - File-Utility
-
- Execution Coverage
 - Unterprogramme
 - Storage Protection
 - Scripting Facility

