



IBM eServer zSeries

Multi-Plattform Entwicklungsumgebung WSAD, RAD, VAGen, EGL

Überblick, Beispielanwendungen, Tools sowie ...

Bau von VSE back-end Programmen via 4th Generation Language (4GL)



ON DEMAND BUSINESS™



Wolfgang Bosch, IBM Lab Böblingen, z/VSE Service and Development

E-Mail: wbosch@de.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and / or other countries.

AIX*	IBM logo*	SQL/DS
CICS*	IMS	Virtual Image Facility
CICS/VSE*	Intelligent	VisualAge*
C/370	Language Environment*	VisualGen*
DB2*	Miner	VM/ESA*
DB2 Connect	MQSeries*	VSE/ESA
DB2 Universal Database	Multiprise*	VTAM*
DFSORT	MVS	WebSphere*
e-business logo*	OS/2*	xSeries*
eServer	OS/390*	z/Architecture
Enterprise Storage Server*	OS/400*	z/OS*
HiperSockets	Rational*	z/VM
IBM*	S/390*	z/VSE
	SNAP/SHOT*	zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

LINUX is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Tivoli is a trademark of Tivoli Systems Inc.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows the Windows 95 logo, and Windows NT, are registered trademarks of Microsoft Corporation.

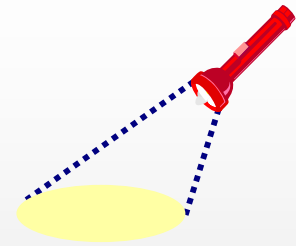
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

Intel is a registered trademark of Intel Corporation.

Other company, product, and service names, may be trademarks or service marks of others.

© 2005 IBM Corporation

Inhalt



- Die VSE Laufzeitumgebung (Language Environment for VSE/ESA) ◆ 4
- “Klassische” Anwendungs-Entwicklung (AE) für VSE ◆ 8
- „Java-Basierende” AE + Nutzung der „VSE Connectors“ ◆ 14
- “Moderner”, 4th Generation Language (4GL) Ansatz mit EGL ◆ 27
- Zusammenfassung ◆ 42

© 2005 IBM Corporation

© 2005 IBM Corporation

Die VSE Laufzeitumgebung (LE/VSE)

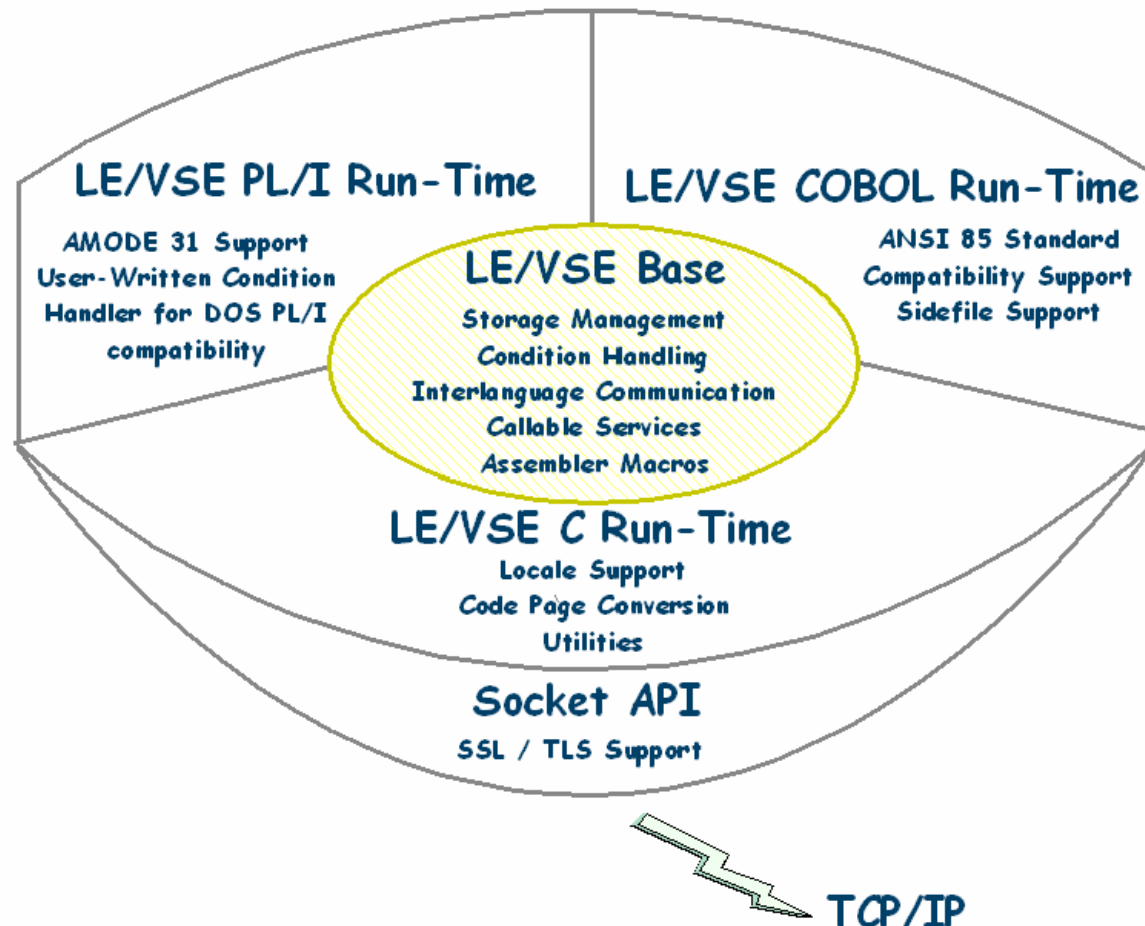


- Struktur und Komponenten
- Was bietet LE/VSE ?
- Überblick zu LE/VSE 1.4.4 in z/VSE 3.1

© 2005 IBM Corporation

Struktur und Komponenten von LE/VSE

* Die Laufzeitumgebung für VSE Anwendungen *



© 2005 IBM Corporation

Was bietet LE/VSE ?

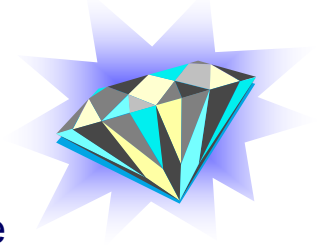
□ Eine gemeinsame Laufzeitumgebung für Anwendungen die

- In LE/VSE-konformen Hochsprachen (HLL) geschrieben sind
 - IBM COBOL for VSE/ESA
 - IBM PLI for VSE/VSE
 - IBM C for VSE/ESA
- Für sich allein oder gemischt vorliegen (batch + CICS)

□ Weiterhin werden u.a. folgende Aspekte unterstützt ...

<http://www.ibm.com/servers/eserver/zseries/zvse/products/languages.html#le>

- Einbindung der Assemblersprache (Makros usw.)
- Interfaces zu Produkten wie CICS, DB2, DL/I, SORT, Debug Tool usw.
- Interlanguage Kommunikation
- Speichermanagement
- Fehlerbehandlung, Laufzeitmeldungen
- Callable Services für zahlreiche Standard-Programmiersmuster
(➔ Datumsformate, mathematische Funktionen usw.)
- C run-time socket API (Programmier-Interface)



© 2005 IBM Corporation

* Überblick LE/VSE 1.4.4 in z/VSE 3.1 *

- Ergänzungen zur Internationalen Programmierung (neue Locales, UCS-2)
- Dumps in die **VSE/POWER LSTQ** (bisher für CICS, neu für **Batch-Umgebung**)
- Neuer „Callable Service“ für sicheren Speicherzugriff: **CEE5TSTG**
- Status an der VSE Konsole (neue VSE AR Kommandos “**D CEE, ...**”)
- Monitoring der **GETVIS** Nutzung (neue sub pools / mittels AR Kommando)
- Erweiterung der **CEEFETCH** and **CEERELES** Makros für COBOL/VSE
- COBOL Side-File Exits** (reduzieren die Phasengröße mit Debug Tool)
- VSE OLPD Support**

Mehr Details via: <http://www.ibm.com/servers/eserver/zseries/zvse/>
Products & components → VSE Central Functions → Language Environment (LE)

Recent enhancements for LE/VSE » (400KB)

© 2005 IBM Corporation

„Klassische“ Anwendungs-Entwicklung (AE) für VSE

- wie bisher ... (VM, ICCF usw.)
- Interaktionsumfeld von VSE Applikationen, IUI Support
- Programmieretechniken ... (Assembler, Makros, TCPIP)
- Interlanguage Kommunikation im Überblick
- Hilfen für Host-Based Debugging

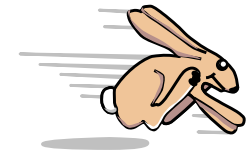
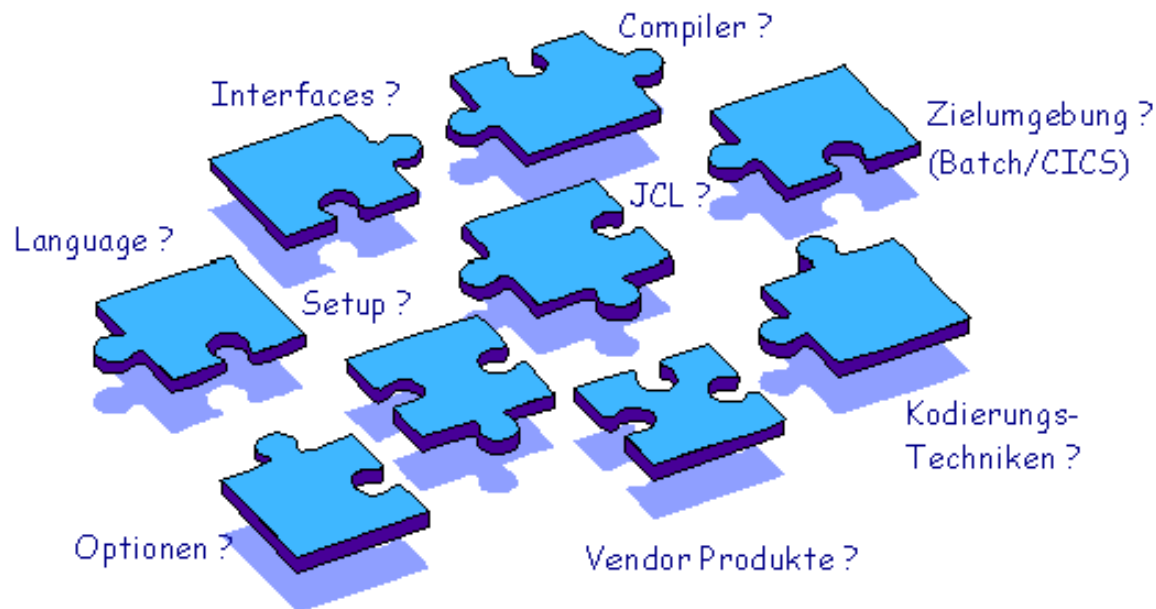
- **VSE Anwendungen, kodiert in Assembler, COBOL/VSE, PLI/VSE, C/VSE**
- **LE/VSE Run-Time Services & Interfaces zu Produkten wie CICS, DB2/VSE, DL/I, TCP/IP usw. werden direkt kodiert**

© 2005 IBM Corporation

Interaktionsumfeld von VSE Anwendungen

□ Typische Aspekte von ...

VSE Applikationen



□ Unterstützung durch das Interactive User Interface (IUI)

- Primary Library, OPTION 8, generiert Skeletons für die Kompilierung (Fokus auf LE/VSE Sprachen)
- Verschiedene Call-Strukturen sind möglich, Produkteinbindungen z.B. CICS, DB2/VSE
- Dialoge für Security-Aspekte von CICS Transaktionen

© 2005 IBM Corporation

Beispiele für Programmieretechniken (Teil 1)

* Assembler + Hochsprachen (HLL) im LE/VSE Umfeld *

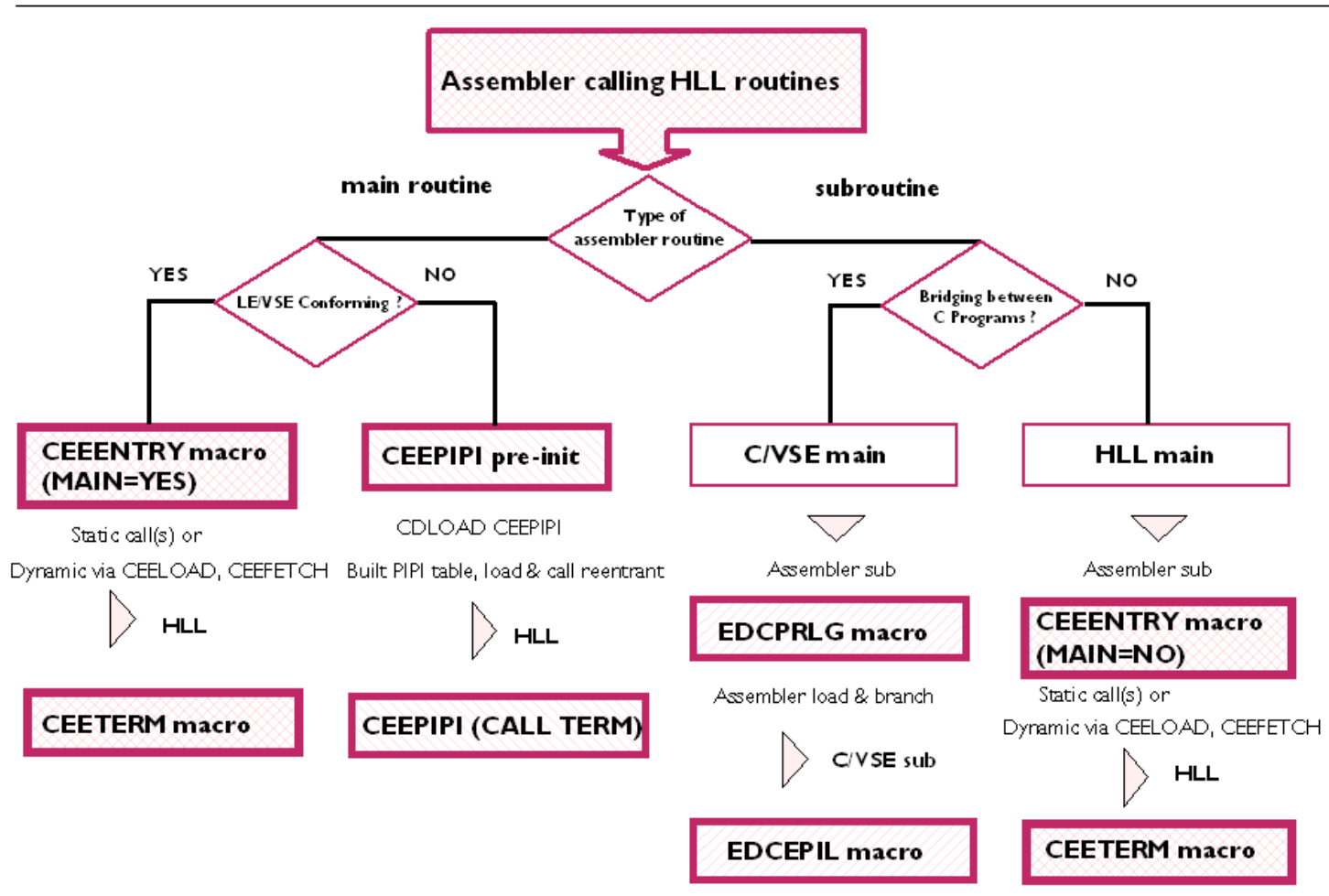


Fig. Assembler Calling High Level Language (HLL) Routines

Beispiele für Programmieretechniken (Teil 2)

* LE/VSE-C und TCP/IP *

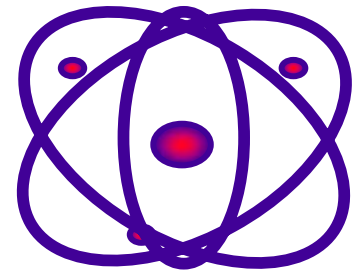
□ Empfohlene Kodierungsoptionen für das LE/VSE-C Interface

- EZASMI Makro Interface (HLASM Programme)
 - > weitgehend OS/390 source kompatibel
- EZASOKET Call Interface (für HLASM, COBOL/VSE und PLI/VSE Programme)
 - > weitgehend OS/390 source kompatibel
- TCP/IP Callable C-Funktionen

Dokumentation:

"TCP/IP for VSE/ESA - IBM Program Setup and Supplementary Information" (SC33-6601-05)

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/iestce30/2.1.6



□ EZASMI und EZASOKET eignen sich für Batch und CICS/TS Umgebungen

- CICS Anwendungen müssen LE-enabled sein. CICS CSD Definitionen (für die Interface Module: EZASOH00, EZASOH01, EZASOH03) existieren seit VSE/ESA 2.5
- 31-bit Addressing Mode wird vorausgesetzt
- Die Interface Module EZASOH00 und EZASOH03 sind SVA-31 eligible

□ Alternativen für die Kodierung

- CSI TCP/IP SOCKET Makro (effizient für Assembler, jedoch nicht portabel)
- CSI EXEC TCP Prozessor (effizient, jedoch nicht portabel)

© 2005 IBM Corporation

Interlanguage Kommunikation (ILC)

* Im Überblick *

❑ LE/VSE V1R4 Writing Interlanguage Communication Applications SC33-6686

❑ Kommunikation zwischen C und PL/I

- Statische + dynamische Aufrufe sind unterstützt. Weitere Details:
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE00/3.0
- Beispiel: eine PL/I Routine fokussiert auf Applikationslogik, die C Routine regelt den Datentransfer ...

❑ Analog für C-COBOL und PL/I-COBOL (ILC)

- C-COBOL (ILC):
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE00/2.0
- COBOL-PL/I (ILC):
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE00/4.0

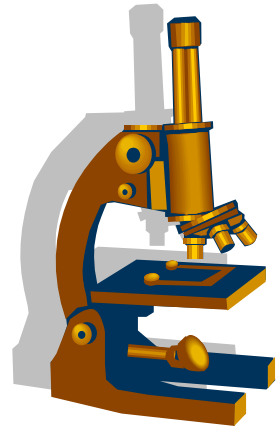
❑ Ebenso für die CICS Umgebung ...

- Die Kompilierung muss mit LE/VSE-konformen Compilern erfolgen
- Das CALL Interface kann benutzt werden, zu bevorzugen sind jedoch EXEC CICS XCTL/LINK Konstrukte
- Überblick:
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/FL2WIE00/7.0

© 2005 IBM Corporation

LE/VSE – Hilfe für Host-Based Debugging

- ❑ **LE/VSE V1R4 Debugging and Run-Time Messages SC33-6681**
- ❑ **Debugging von PL/I Routinen**
 - Fehlerursachen in PL/I Routinen identifizieren, LE/VSE Speicherauszüge
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre06/2.3
- ❑ **Debugging von C Routinen**
 - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre06/2.1
- ❑ **Debugging von COBOL Routinen**
 - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre06/2.2
- ❑ **Debugging in CICS**
 - LE/VSE & CICS bezogene Informationen:
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/fl2dre06/2.4
- ❑ **Debug Tool for VSE/ESA (Home Page + Produkt Dokumentation)**
 - <http://www.ibm.com/software/awdtools/debugtoolvse/>
 - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/EQAVI000
 - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/EQAVU000



© 2005 IBM Corporation

„Java-Basierende“ AE Ansätze + „VSE Connectors“

■ Konzepte

- Einfache Anbindung an den z/VSE Server
- Real time Zugriff mit Java Based Connector

■ Integrierte Entwicklungsumgebung (IDE)

- WebSphere Application Developer (WSAD) V5

■ LE spezifische Java Tools + Beispiele

- Beispiel 1: „JCalc_LEVSE-Tool“
- Beispiel 2: „JLink_LEVSE-Tool“
- Beispiel 3: „JDisplay_AR_Commands“

z/VSE Home → Downloads → Tools

■ VSE Connector spezifische Java Tools ...

- „VSE Navigator“: Zugriff auf VSE Ressourcen aller Art, generiert compile jobs, output usw.

z/VSE Home → Solutions

<http://www.ibm.com/servers/eserver/zseries/zvse/products/connectors.html#navi>

● Non-VSE Laufzeitumgebung

● Server- Daten, Ausgabeprotokolle, Statistiken, Interfaces, Skeletons, VSE Connectors usw. werden genutzt !

● Plattform-unabhängige Ausführung ist möglich (z.B. auf Linux)

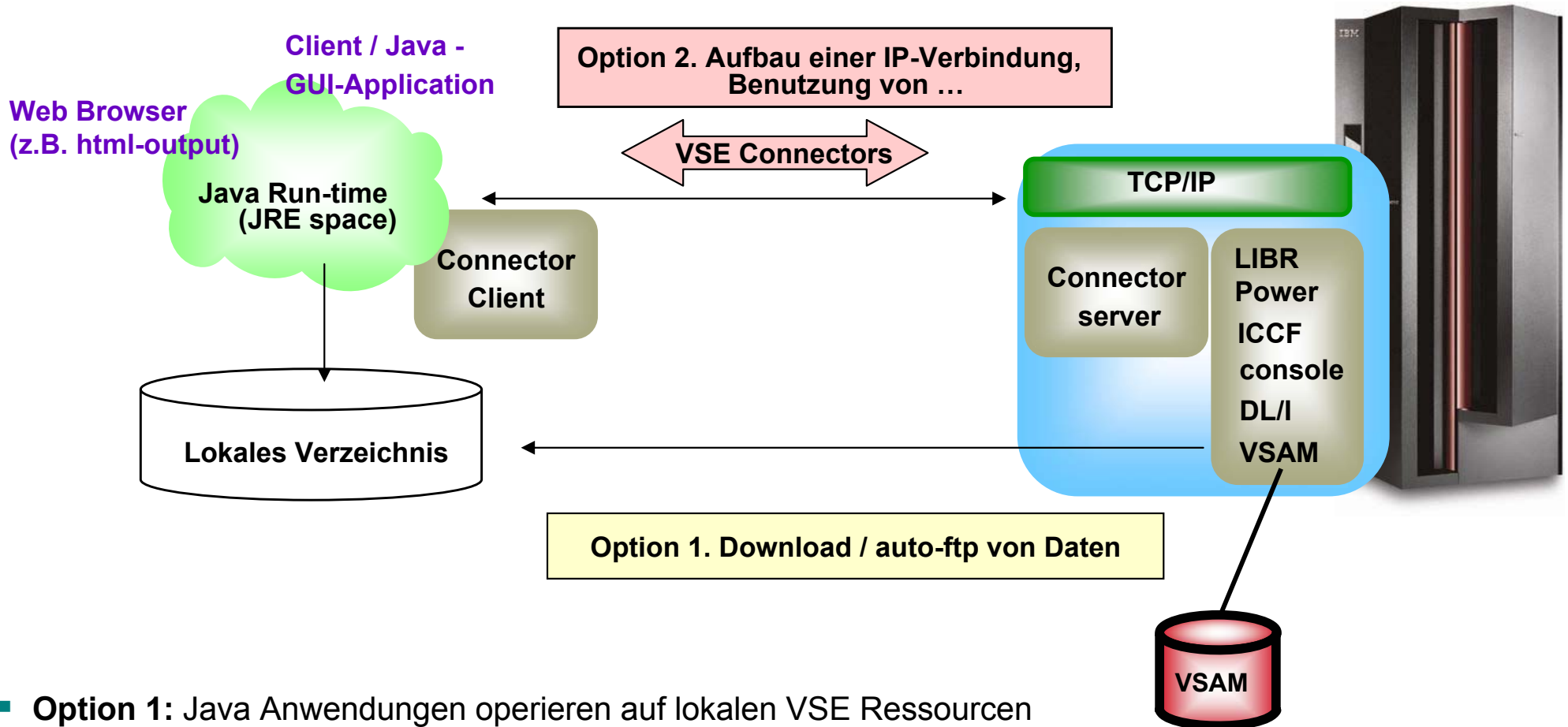
© 2005 IBM Corporation

Einfache Form der Anbindung an den z/VSE Server

* Ohne Web Application Server *

PC/Workstation

z/VSE Server



- **Option 1:** Java Anwendungen operieren auf lokalen VSE Ressourcen
- **Option 2:** Etablieren eine IP-Verbindung zum z/VSE server um direkt zu interagieren

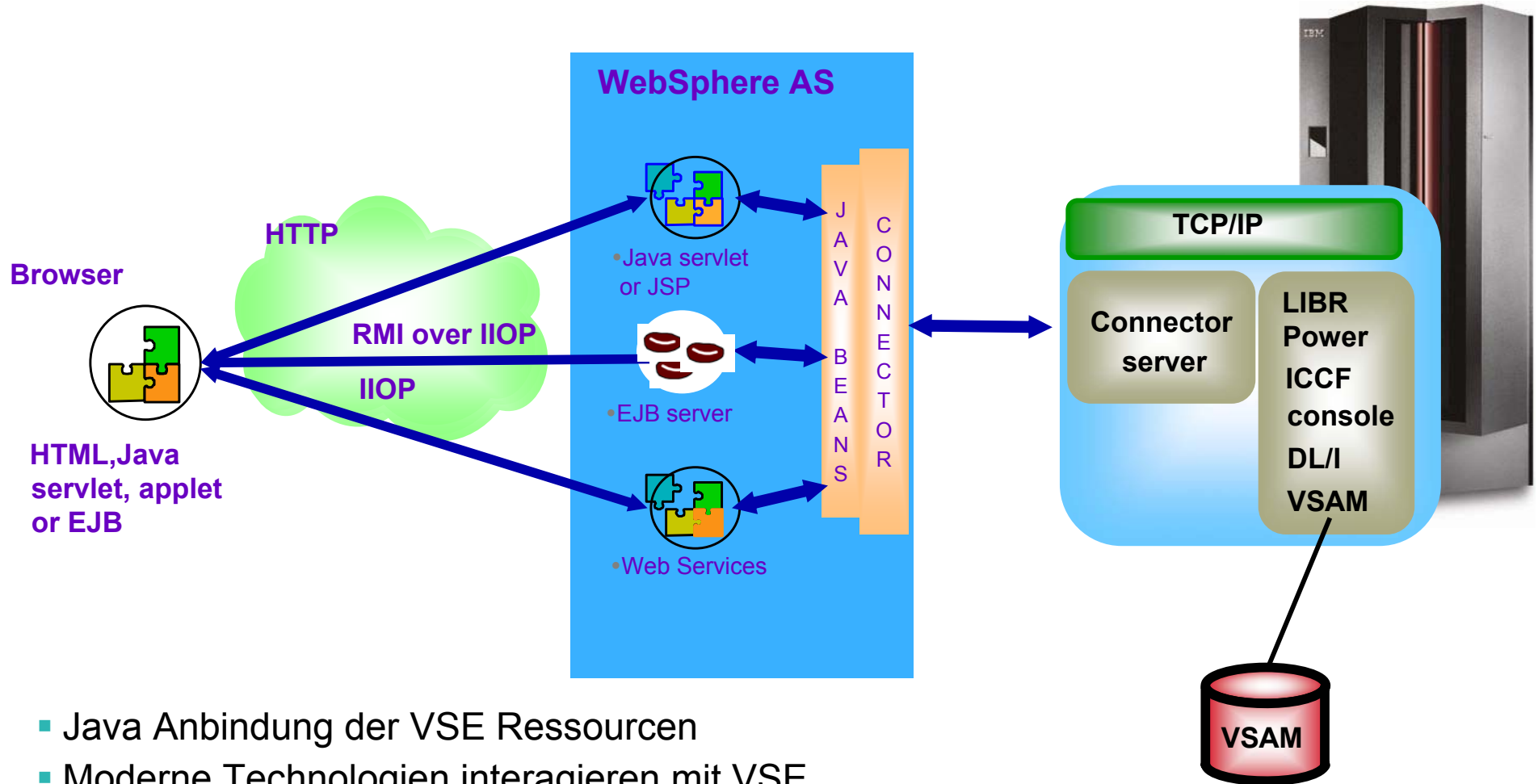
© 2005 IBM Corporation

Real time Zugriff mit Java-Based Connector

* Mit Web Application Server *

Web Client

z/VSE Server

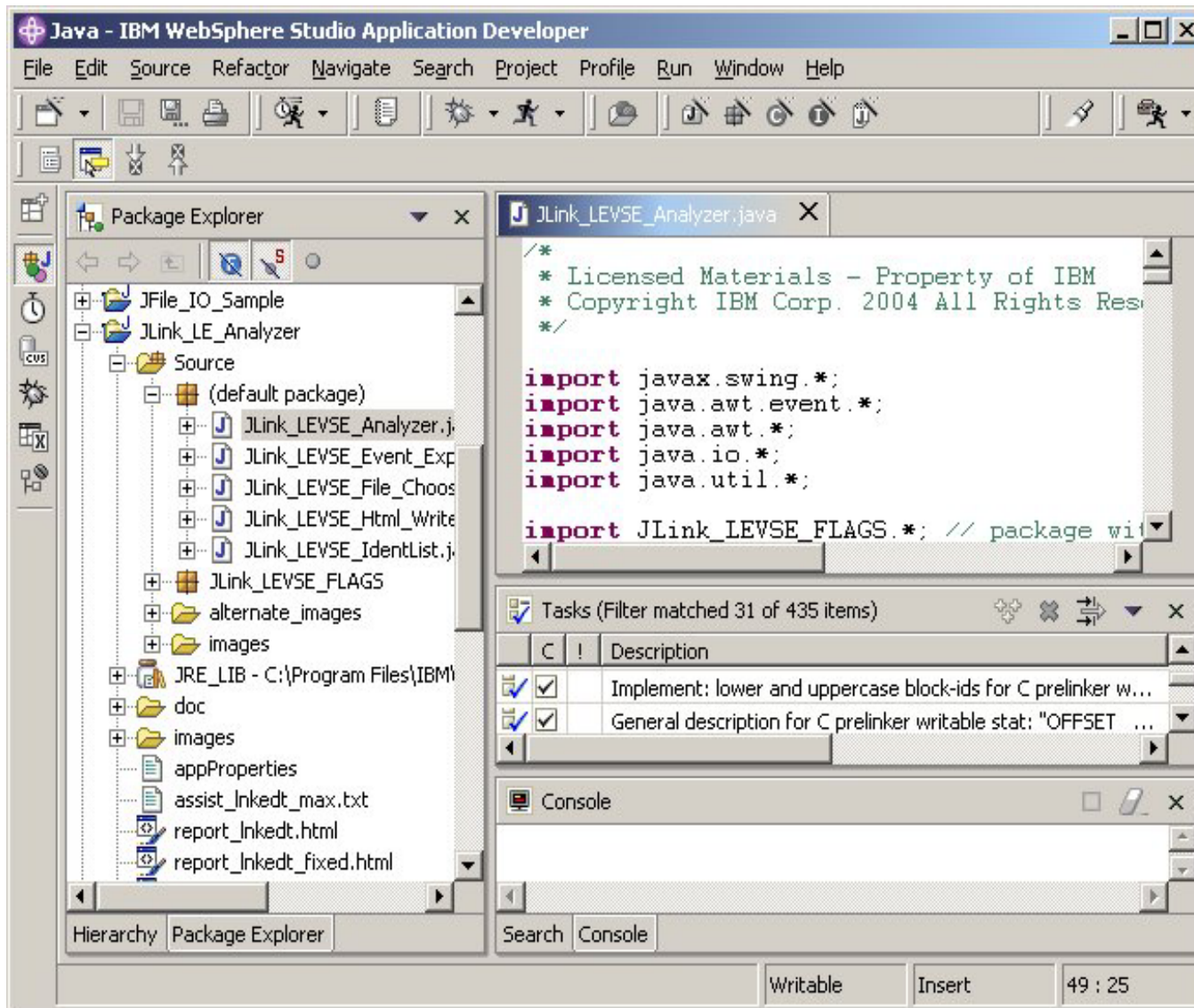
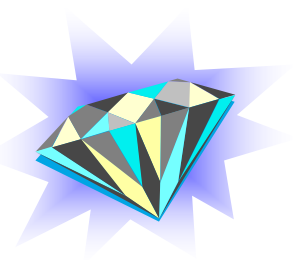


- Java Anbindung der VSE Ressourcen
- Moderne Technologien interagieren mit VSE

© 2005 IBM Corporation

WebSphere Application Developer (WSAD)

Blick auf die Integrierte Entwicklungsumgebung (IDE)



Basiert auf der Eclipse Plattform

<http://www.eclipse.org>

Offene Architektur für integrierte Web- & Applikationsentwicklung.
Unterstützt das plug-in model

- **Verschiedene Perspektiven**
Java-, Profiling-, Debugging-, CVS Repository-, J2EE-, XML ...
- **Basisfunktionalität wie:**
Projekt-, Class-, Interface- Erzeugung, Code-Formatierung
- **Kontrollfenster**
Quellcode, Tasks, Konsol- Meldungen, Package Explorer ...
- **Developer Tools wie:**
Profil-Agenten für die Analyse, javadoc Generator
- **Tools für Ausführung, Test**

© 2005 IBM Corporation

Beispiel 1: JCalc_LEVSE Tool (Teil 1)

- **Rechner für die LE/VSE Nutzung der Shared Virtual Area**
 - Java GUI Interface für die Auswahl einer oder mehrerer LE/VSE Komponenten



- **Auszug der VSE LIBR Download-Daten, die analysiert werden**

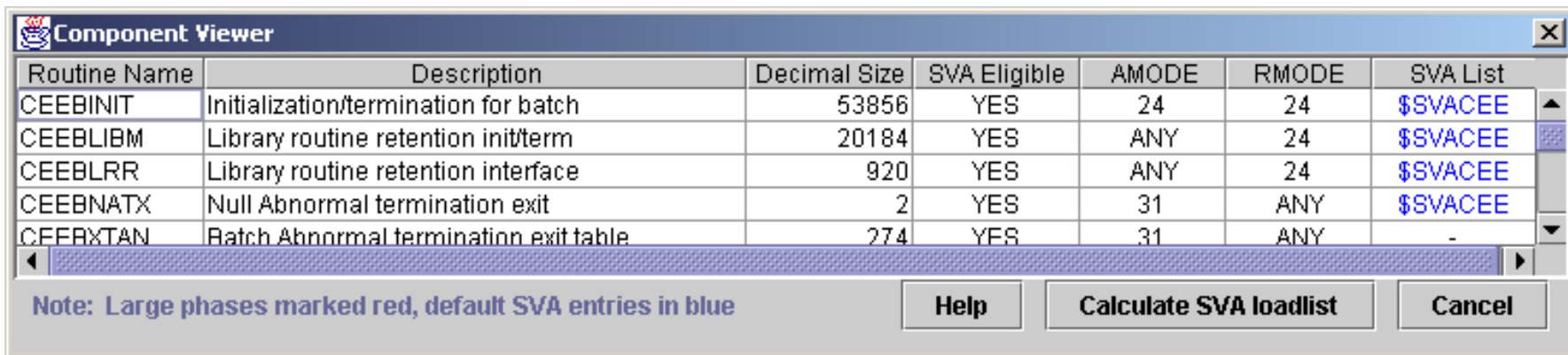
M E M B E R NAME	T Y P E	C R E A T I O N DATE	L A S T UPDATE	B Y T E S RECORDS	L I B R C O N T BLKS	S V A STOR	A - R - ELIG	M O D E
CEEFMON	PHASE	01-10-29	03-01-08	20 B	1	YES	YES	31 ANY
CEEFMTM	PHASE	01-10-29	03-01-08	20 B	1	YES	YES	31 ANY

© 2005 IBM Corporation

Beispiel 1: JCalc_LEVSE Tool (Teil 2)

□ Basis-Funktionalität

- Beschreibungen für identifizierte Module (Fokus auf SVA-eligible)
- **Aufbereitung des Ergebnisses in einer JTable (Übersicht Speichernutzung)**

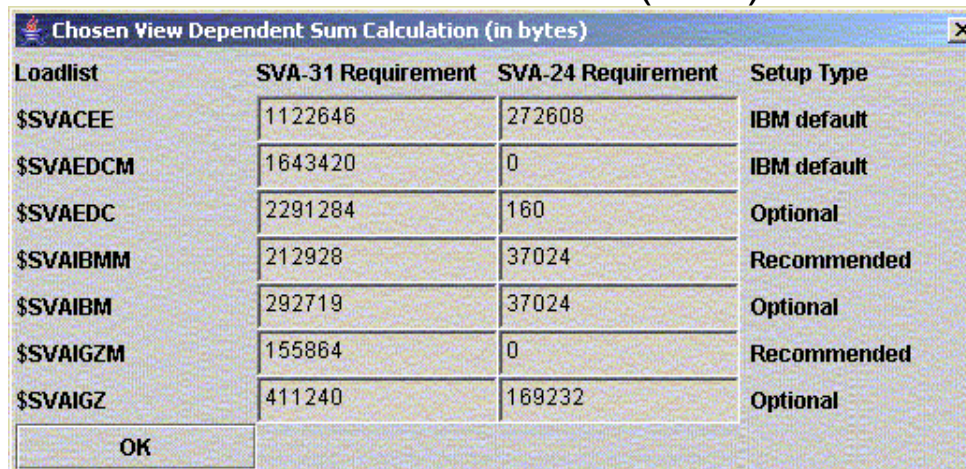


Routine Name	Description	Decimal Size	SVA Eligible	AMODE	RMODE	SVA List
CEEBINIT	Initialization/termination for batch	53856	YES	24	24	\$SVACEE
CEEBLIBM	Library routine retention init/term	20184	YES	ANY	24	\$SVACEE
CEEBLRR	Library routine retention interface	920	YES	ANY	24	\$SVACEE
CEEBNATX	Null Abnormal termination exit	2	YES	31	ANY	\$SVACEE
CEEBXTAN	Batch Abnormal termination exit table	274	YES	31	ANY	-

Note: Large phases marked red, default SVA entries in blue

Buttons: Help, Calculate SVA loadlist, Cancel

- Generieren von VSE shared virtual area (SVA) Statistiken/Reports



Loadlist	SVA-31 Requirement	SVA-24 Requirement	Setup Type
\$SVACEE	1122646	272608	IBM default
\$SVAEDCM	1643420	0	IBM default
\$SVAEDC	2291284	160	Optional
\$SVAIBMM	212928	37024	Recommended
\$SVAIBM	292719	37024	Optional
\$SVAIGZM	155864	0	Recommended
\$SVAIGZ	411240	169232	Optional

OK

© 2005 IBM Corporation

Beispiel 2: JLink_LEVSE Tool (Teil 1)

❑ Überprüfung von VSE Compile/link Listen

- GUI Anwendung (3 stufiger Interaktionsprozess: Auswahl Datei, Analyse, Report)
- Überblick zu konkreten VSE Anwendungen (Ressourcen, Programmier Techniken ...)
- Zusammenfassung von hits und deren Kontext (z.B. compile- options/messages ...)
- Spezifische Informationen ... (z.B. Indikatoren für ungeeignete stub Verwendung ...)



© 2005 IBM Corporation

Beispiel 2: JLink_LEVSE Tool (Teil 2)

□ Ausgabe-Report im html-Format (Web Browser Ansicht)

C Prelinker - Object Resolution events

CEESETL	Unresolved prelink module reference. Also check linkage editor resolution
WARNING EDC4015:	Unresolved references are detected. Verify if the reference(s) given in this section are later resolved by the linkage editor

C Prelinker - Writable Static events

INFORMATIONAL EDC4013:	No map contents displayed as no writable static was found
-------------------------------	-----------------------------------------------------------

Linkage editor - Autolink events

CEESG003	Signature module for C/VSE has been linked into application phase	Library reference not visible in link list (typical for C)
CEESTART	Compiler generated default entry point available	Library reference not visible in link list (typical for C)
EDCINPL	LE/C initialization triggered (might involve an assembler program with EDCPRLG and EDCEPIL macros coded)	Library reference not visible in link list (typical for C)
PRINTF	C Run-Time library function - printf (Format and Write Data)	Library reference not visible in link list (typical for C)

Linkage editor - Unresolved Reference events

© 2005 IBM Corporation

EXTRN CE\$SETL	Do not code the CEESETL callable service. For C routines the setlocale() function should be used instead
WXTRN @@DLL	Internal LE/C reference, fine to be present

Beispiel 2: JLink_LEVSE Tool (Teil 3)

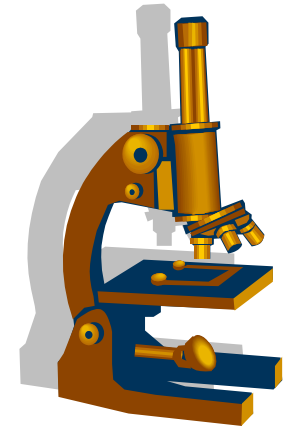
*** Profiling: Ansicht – Methoden/Klassen/Verweilzeit ***

Profiling - IBM WebSphere Studio Application Developer

Method Statistics [<terminated> JLink_LEVSE_Analyzer [PID:1928]]

Filter: Case-sensitive

>Method Names	Class Names	Calls	Base Time	Average Base T...	Cumulative Time
-clint-	JLink_LEVSE_Analyzer	1	0,000008	0,000008	0,000008
actionPerformed(ActionEvent) void	JLink_LEVSE_Analyzer	3	0,161124	0,053708	29,473198
actionPerformed(ActionEvent) void	JLink_LEVSE_File_Chooser_Dialog	2	25,720928	12,860464	25,720928
actionPerformed(ActionEvent) void	AbstractButton\$ForwardActionEv...	2	0,000029	0,000014	25,721062
analyze_vector_for_multiples(Vecto...	JLink_LEVSE_Html_Writer	2	0,296247	0,148124	0,296247
asm_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000003	0,000003	0,000003
autolink_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,001535	0,001535	0,298894
c_comp_opts_html_reporting(Vecto...	JLink_LEVSE_Html_Writer	1	0,000003	0,000003	0,000003
c_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000004	0,000004	0,000004
c_prk_opts_html_reporting(Vecto...	JLink_LEVSE_Html_Writer	1	0,000003	0,000003	0,000003
cics_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000004	0,000004	0,000004
cob_comp_opts_html_reporting(V...	JLink_LEVSE_Html_Writer	1	0,000004	0,000004	0,000004
cob_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000003	0,000003	0,000003
compare(Object, Object) int	JLink_LEVSE_Event_Expert\$Ident...	5605	0,014901	0,000003	0,015095
createImageIcon(String) ImageIcon	JLink_LEVSE_Analyzer	1	0,319107	0,319107	0,319107
createImageIcon(String) ImageIcon	JLink_LEVSE_File_Chooser_Dialog	1	0,019184	0,019184	0,019184
dispatchEvent(AWTEvent) boolean	LightweightDispatcher	2	0,000032	0,000016	25,722278
dispatchEvent(AWTEvent) void	Component	3	0,000038	0,000013	25,722384
dispatchEvent(AWTEvent) void	EventQueue	2	0,000029	0,000014	25,722413
dispatchEventImpl(AWTEvent) void	Component	2	0,000299	0,000150	25,721946
dispatchEventImpl(AWTEvent) void	Container	4	0,000076	0,000019	25,722328
dispatchEventImpl(AWTEvent) void	Window	2	0,000029	0,000014	25,722357
final_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000004	0,000004	0,000004
fireActionPerformed(ActionEvent)...	AbstractButton	2	0,000105	0,000052	25,721033
fireActionPerformed(ActionEvent)...	DefaultButtonModel	2	0,000042	0,000021	25,721104
handle_asm_events(String, Vecto...	JLink_LEVSE_Event_Expert	2	0,000135	0,000068	0,000140
handle_autolink_events(String, Ve...	JLink_LEVSE_Event_Expert	107	0,432021	0,004038	0,513584
handle_c_events(String, Vector) ...	JLink_LEVSE_Event_Expert	2	0,000392	0,000196	0,000397
handle_cics_events(String, Vecto...	JLink_LEVSE_Event_Expert	2	0,000102	0,000051	0,000105
handle_final_events(String, Vecto...	JLink_LEVSE_Event_Expert	1	0,000053	0,000053	0,000056
handle_jcl_events(String, Vector)...	JLink_LEVSE_Event_Expert	8	0,000696	0,000087	0,000710
handle_lang_events(String, Vecto...	JLink_LEVSE_Event_Expert	10	0,016113	0,001611	0,016141
handle_pli_events(String, Vector)...	JLink_LEVSE_Event_Expert	2	0,000113	0,000056	0,000118
handle_prod_events(String, Vecto...	JLink_LEVSE_Event_Expert	3	0,000168	0,000056	0,000176
handle_unresolved_link_events(St...	JLink_LEVSE_Event_Expert	54	0,006424	0,000119	0,028268
ilc_if_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000008	0,000008	0,000008
jcl_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000013	0,000013	0,000013
JLink_LEVSE_Analyzer(String)	JLink_LEVSE_Analyzer	1	39,177398	39,177398	39,592319
JLink_LEVSE_Event_Expert\$IdentL...	JLink_LEVSE_Event_Expert\$Ident...	126	0,000289	0,000002	0,000289
JLink_LEVSE_Event_Expert()	JLink_LEVSE_Event_Expert	1	0,001690	0,001690	0,001690
JLink_LEVSE_File_Chooser_Dialog(...	JLink_LEVSE_File_Chooser_Dialog	1	1,008189	1,008189	1,027373
JLink_LEVSE_Html_Writer(String)	JLink_LEVSE_Html_Writer	1	0,006332	0,006332	0,006332
JLink_LEVSE_IdentList()	JLink_LEVSE_IdentList	96	0,000244	0,000003	0,000244
lang_html_reporting(Vector) void	JLink_LEVSE_Html_Writer	1	0,000004	0,000004	0,000004
loadProperties() void	JLink_LEVSE_Analyzer	1	0,095814	0,095814	0,095814
main(String[]) void	JLink_LEVSE_Analyzer	1	0,000049	0,000049	39,592368
mergeSort(Object[], Object[], int,...	Arrays	1228	0,075910	0,000062	0,090811
mouseReleased(MouseEvent) void	AWTEventMulticaster	2	0,000060	0,000030	25,721567
mouseReleased(MouseEvent) void	BasicButtonListener	2	0,000103	0,000052	25,721507
parse_Link(String) boolean	JLink_LEVSE_Analyzer	1	0,065089	0,065089	0,642884
pli_comp_opts_html_reporting(Vecto...	JLink_LEVSE_Html_Writer	1	0,000003	0,000003	0,000003

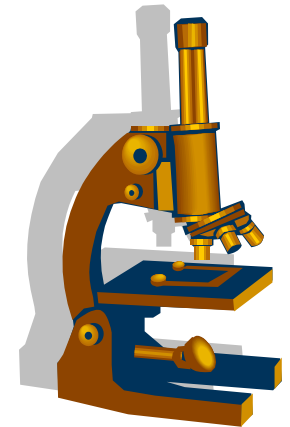
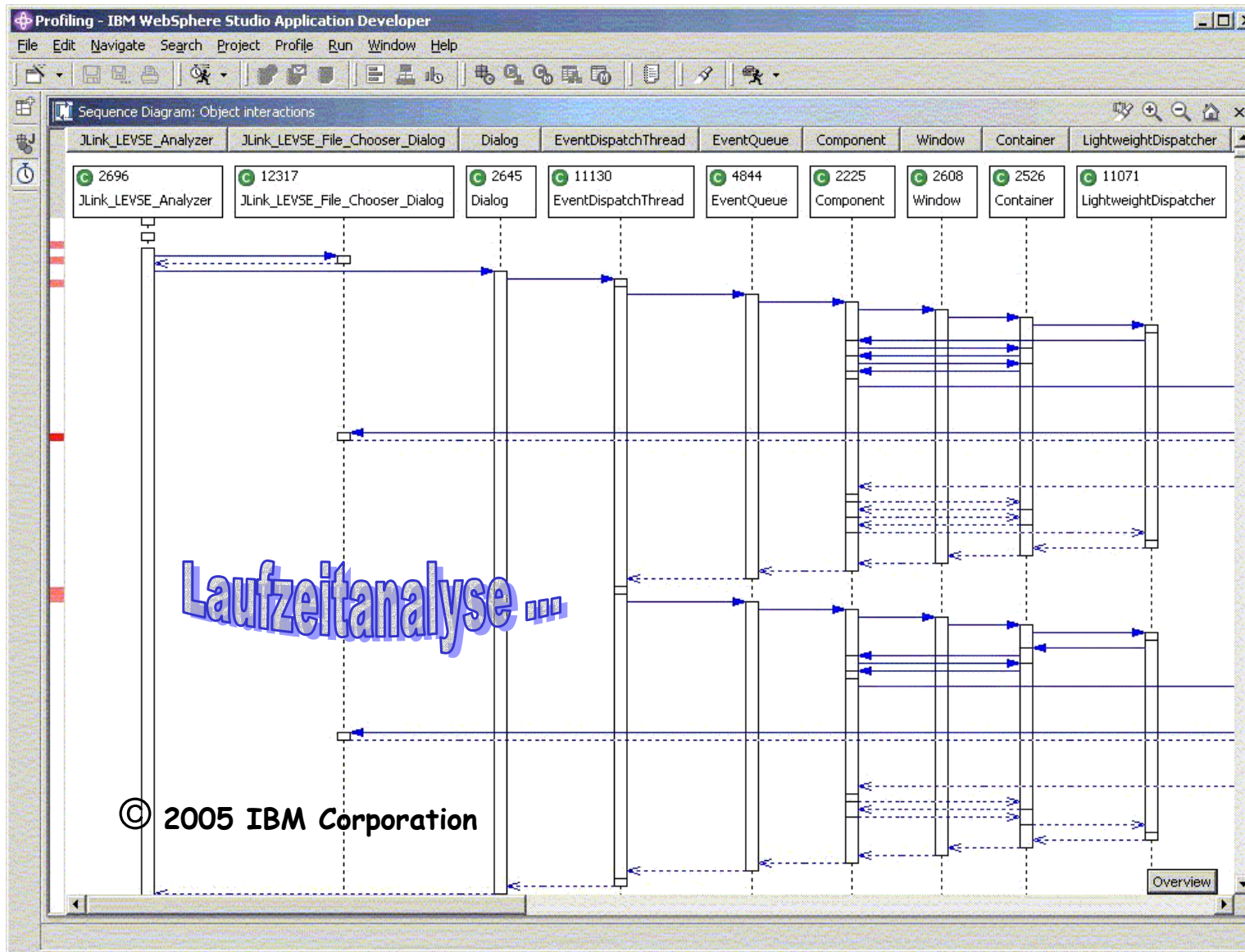


Laufzeitanalyse ...

© 2005 IBM Corporation

Beispiel 2: JLink_LEVSE Tool (Teil 4)

*** Profiling: Ansicht – Objekt Interaktionen ***



Beispiel 3: JDisplay_AR_Commands (Teil 1)

- ❑ GUI Kommunikation mit der VSE Konsole
- ❑ VSE Java Bean / Console API

The screenshot shows a Java Swing window titled "JDisplay_AR_Commands Sample - Copyright IBM Corp. 2005 - Version: 2...". The window has a title bar with a close button (X). The main content area is titled "Purpose: Run VSE console AR-commands". It is divided into two panes:

- Connection Pane:** Contains three input fields:
 - "Enter VSE host (machine name or IP address):" with the text "servername" entered.
 - "Please enter VSE user ID:" with an empty field.
 - "Please enter password:" with an empty field.
- Command Process Pane:** Contains one input field:
 - "Please enter AR command:" with the text "D CEE,CEEDOPT" entered.

At the bottom of the window, there are three buttons: "START", "HELP", and "EXIT".

© 2005 IBM Corporation

Beispiel 3: JDisplay_AR_Commands (Teil 2)

❑ VSE Konsolmeldungen in der WebSphere Entwicklungsumgebung

The screenshot shows the IBM WebSphere Studio Application Developer interface. The Package Explorer on the left shows the project structure for JDisplay_AR_Commands. The main editor displays the source code for JDisplay_AR_Commands.java. The Console window at the bottom right shows the output of a VSE command, including a list of installation options and their default values.

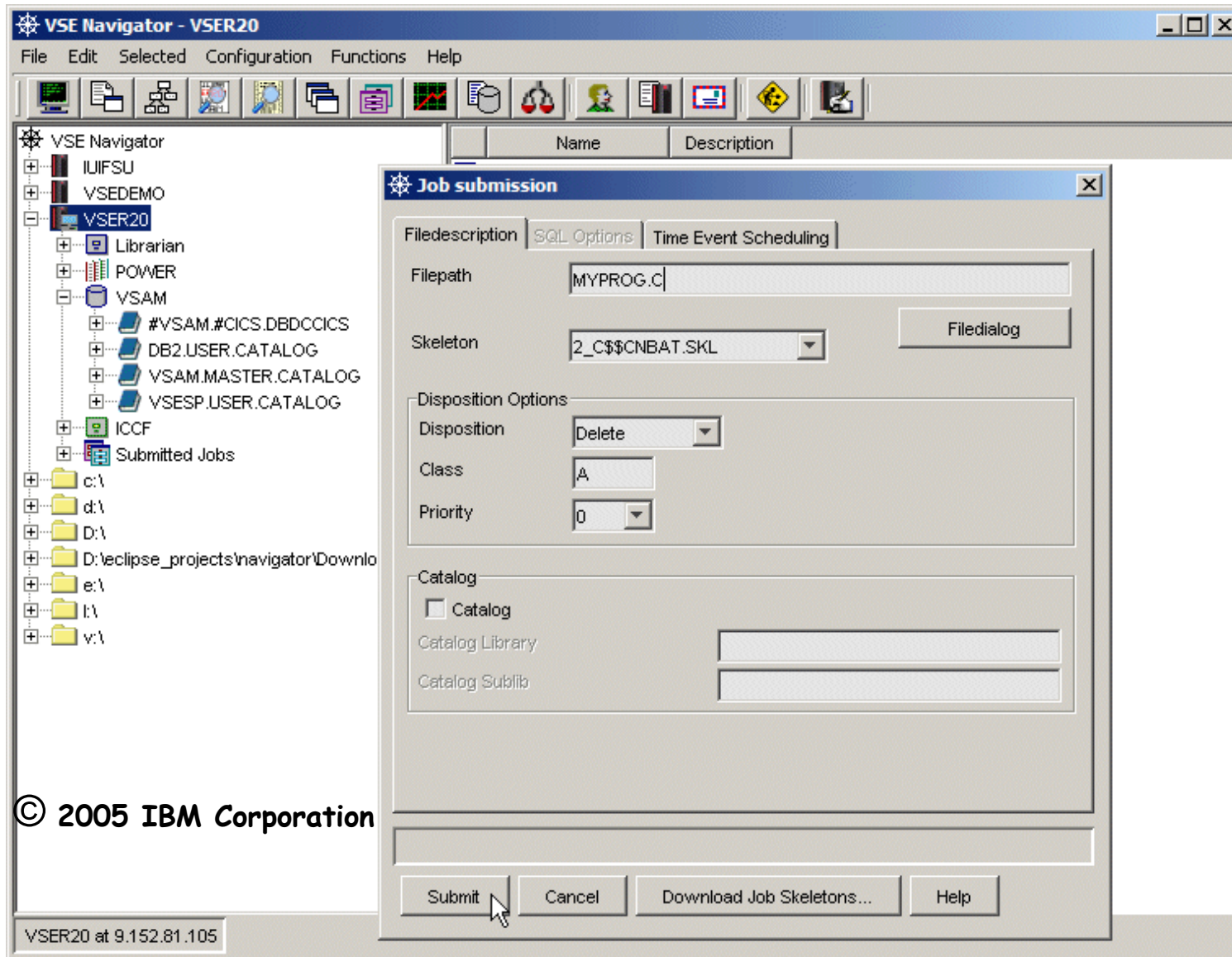
© 2005 IBM Corporation

```

VSE Console line by line response from AR command: D CEE,CEEDOPT
AR 0015 Lang. Env. for VSE command processing completed
AR 0015 1140I READY
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R88I OK : 1 ENTRY PROCESSED BY R RDR,CEL$OPT$
C1 0001 1Q47I C1 CEL$OPT$ 03910 FROM (SYSA) , TIME=16:58:36
C1 0046 // JOB CEL4OPTS - PRODUCE BATCH INSTALLATION-WIDE OPTIONS REPORT
DATE 03/02/2005, CLOCK 16/58/36
C1 0046 Options Report for Enclave CEL4OPTS 03/02/05 4:58:36 PM
C1 0046 Language Environment for VSE/ESA V1 R4.4
C1 0046
C1 0046 LAST WHERE SET OPTION
C1 0046
C1 0046 Installation default ABPERC(NONE)
C1 0046 Installation default ABTERMENC(ABEND)
C1 0046 Installation default NOAIXBLD
C1 0046 Installation default ALL31(OFF)
C1 0046 Installation default ANYHEAP(16384,8192,ANYWHERE,FREE)
C1 0046 Installation default BELOWHEAP(8192,4096,FREE)
C1 0046 Installation default CBLOPTS(ON)
C1 0046 Installation default CBLPSHPOP(OFF)
C1 0046 Installation default CHECK(OFF)
C1 0046 Installation default COUNTRY(US)
C1 0046 Installation default NODEBUG
C1 0046 Installation default DEPTHCONDLMT(10)
C1 0046 Installation default ENVAR("")
C1 0046 Installation default ERRRCOUNT(20)
C1 0046 Installation default HEAP(32768,32768,ANYWHERE,KEEP,8192,4)
C1 0046 Installation default HEAPCHK(OFF,1,0)
C1 0046 Installation default LIBSTACK(12288,4096,FREE)
C1 0046 Installation default MSGFILE(SYSLST)
C1 0046 Installation default MSGQ(15)
C1 0046 Installation default NATLANG(UEN)
C1 0046 Installation default RETZERO(OFF)
C1 0046 Installation default RPTOPTS(OFF)
  
```

VSE Navigator

❑ VSE compile/job submission ...



© 2005 IBM Corporation

“Moderner” 4th Generation Language Ansatz

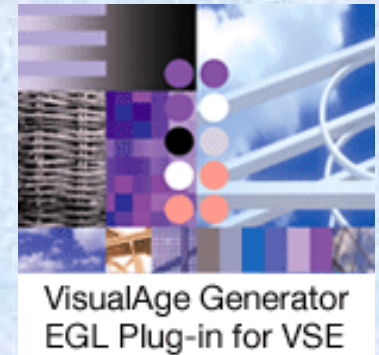
■ Überblick

- „VisualAge Generator“ (Vorgängerprodukt)
 - Ohne Enterprise Generation Language (EGL) Option
- „VisualAge Generator EGL Plug-in VSE“ (aktuelles Produkt)
 - Generierung von COBOL/VSE back-end Programmen mittels EGL front-end
- IBM Whitepaper: Exploiting Java with Enterprise Generation Language
<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/egl-cernosek-wp.pdf>



■ Integrierte Entwicklungsumgebung (IDE)

- Rational Application Developer (RAD) +
- Schließt Möglichkeiten des WebSphere Application Developers (WSAD) ein



■ Beispiel: „COBV6“ ... basierend auf

- **Generating COBOL using EGL and JSF with WebSphere Studio Enterprise Developer**
http://www.ibm.com/developerworks/websphere/library/techarticles/0502_barosa/0502_barosa.html
- Angepasst für **RAD V6** (Generierung, Entsendung + Ausführung auf **z/VSE 3.1** ...)

● **Benötigt eine Laufzeitumgebung, die EGL unterstützt**

© 2005 IBM Corporation

Bisheriger 4GL Ansatz mit dem VisualAge Generator

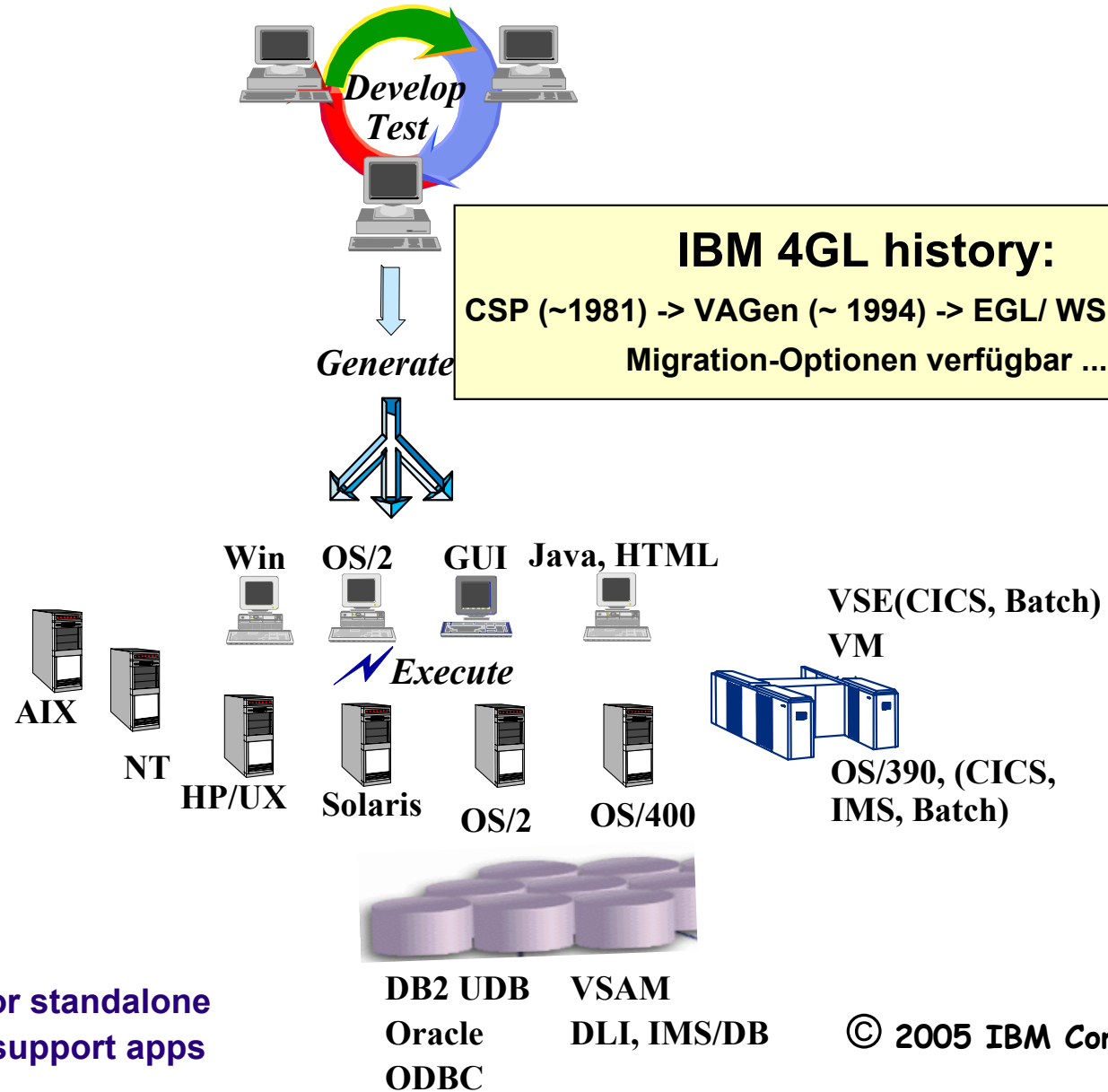
Rapid Development

Generation

- Cobol , C++, Java

Execution

- Multiple platforms
- Stand-alone or C/S
- Java applets/servlets or standalone
- Multiple database/file support apps

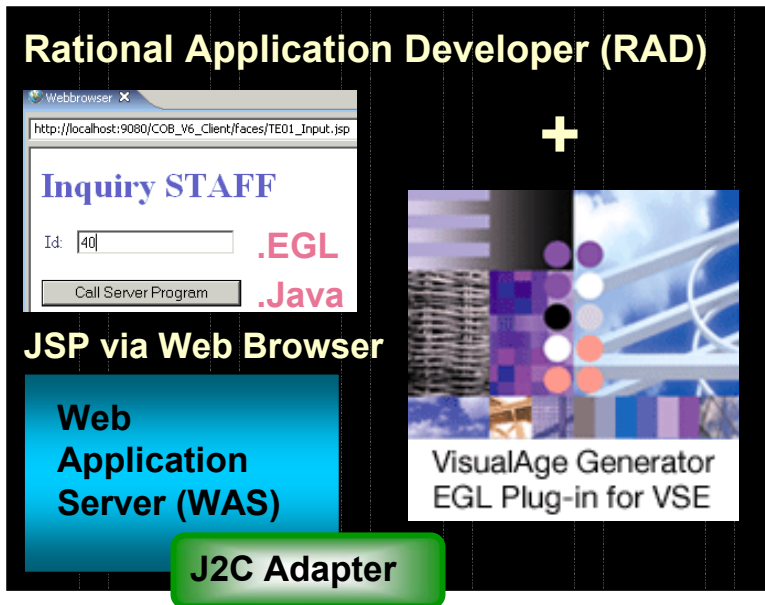


© 2005 IBM Corporation

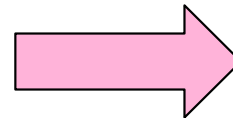
Entwicklung von COBOL/VSE back-end Programmen

* **Neu:** VisualAge Generator EGL Plug-in for VSE *

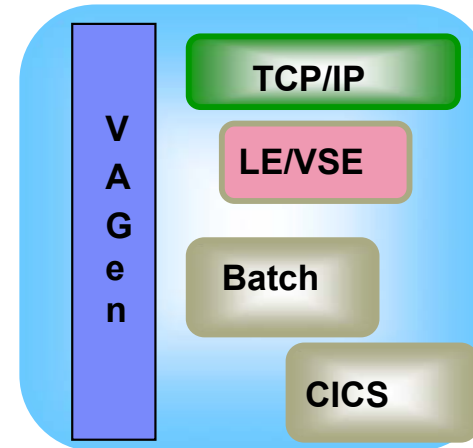
PC/Workstation



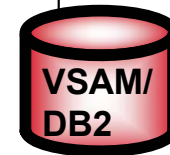
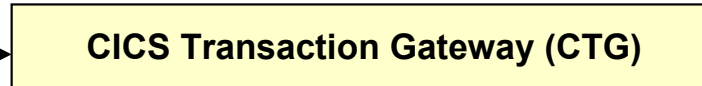
„Generieren/Verteilen von:“



.CBL
.PPT
.FTPResults
+ Buildplan.xml



COBOL/VSE/CICS



z/VSE Server

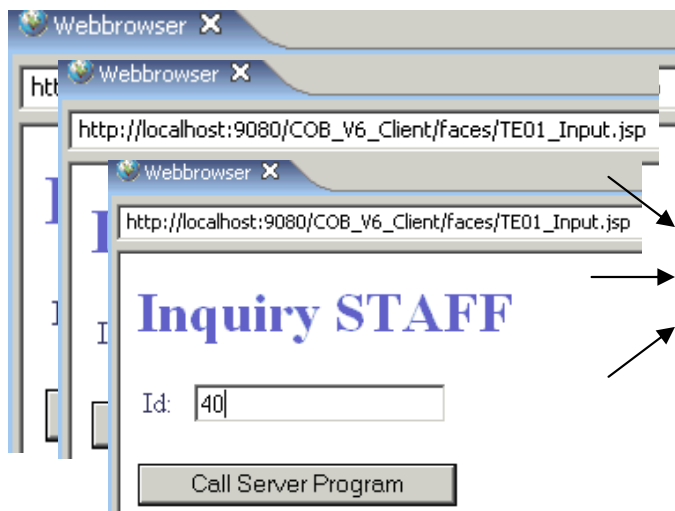


- Enterprise Generation Language (EGL)
- Java™ 2 Platform, Enterprise Edition (J2EE) connection Architektur (J2C/JCA)
- Java Server Pages (JSP), eine dem Web Application Server (WAS) bekannte Anwendung !

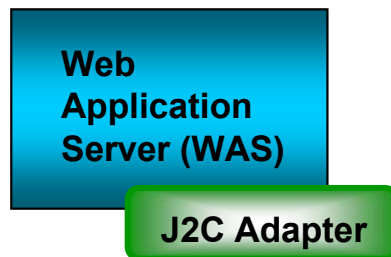
© 2005 IBM Corporation

Komponenten für die VSE back-end Ausführung

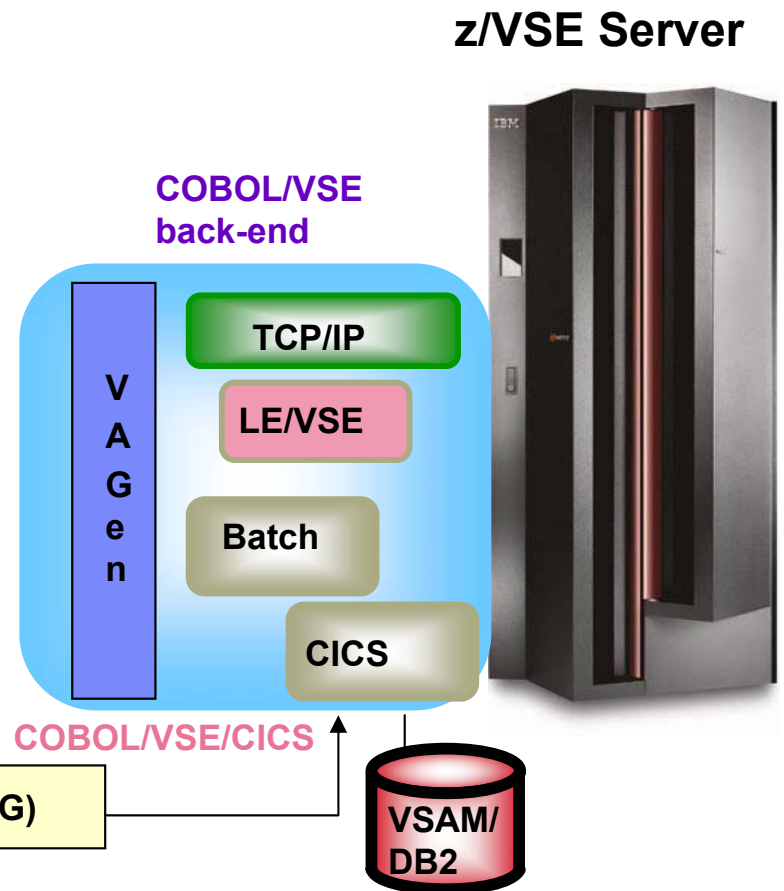
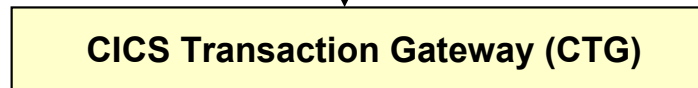
Web Browser



JSP front-end
(n-times)



(1-time)



- Enterprise Generation Language (EGL)
- Java™ 2 Platform, Enterprise Edition (J2EE) connection Architektur (J2C/JCA)
- Java Server Pages (JSP), eine dem Web Application Server (WAS) bekannte Anwendung !

© 2005 IBM Corporation

Referenzen zu Komponenten (Workstation)

❑ VisualAge Generator EGL Plug-in VSE

- Überblick: <http://www.ibm.com/software/awdtools/visgen/vse/index.html>

❑ Rational Application Developer (RAD)

- Überblick: <http://www.ibm.com/software/awdtools/developer/application/index.html>
- Download: <http://www.ibm.com/developerworks/downloads/r/rad>
- Plug-ins: <http://www.ibm.com/developerworks/websphere/downloads/plugin/>
- Bewährtes, Technische Bibliothek: <http://www.ibm.com/developerworks/rational/>

❑ Enterprise Generation Language (EGL)

- Überblick: <http://www.ibm.com/developerworks/rational/products/egl/>
- Technische Dokumentation: <http://www.ibm.com/developerworks/rational/library/egldoc.html>

❑ WebSphere Application Server (WAS)

- Überblick: <http://www.ibm.com/software/webservers/appserv/was/>

❑ CICS Transaction Gateway (CTG) und External Call Interface (ECI)

- Java™ 2 Platform, Enterprise Edition (J2EE) – Connector zu CICS Anwendungen
- Überblick: <http://www.ibm.com/software/htp/cics/ctg/>
- ECI Return Codes: <http://publib.boulder.ibm.com/infocenter/adiehelp/index.jsp?topic=/com.ibm.etools.ctc.cics.api.doc/com.ibm/ctg/client/ECIReturnCodes.html>

❑ DB2 Universal Database (UDB)

- Überblick: <http://www.ibm.com/software/data/db2/udb/>
- Problemlösung: <http://publib.boulder.ibm.com/infocenter/db2help>

© 2005 IBM Corporation

Referenzen zu Komponenten (z/VSE Server)

Language Environment for VSE/ESA

- Überblick: <http://www.ibm.com/servers/eserver/zseries/zvse/products/languages.html#le>
- Support: <http://www.ibm.com/servers/eserver/zseries/zvse/support/le.html>
- Beispiele: <http://www.ibm.com/servers/eserver/zseries/zvse/downloads/samples.html#le>
- Dokumentation: <http://www.ibm.com/servers/eserver/zseries/zvse/documentation/#le>

VisualAge Generator VSE

- Überblick: <http://www.ibm.com/software/awdtools/visgen/>
- Program Directory: <http://www.ibm.com/software/awdtools/visgen/library/v45docs.html>

CICS Web Support (CWS)

- Konfiguration: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/dfhwe705/2.1
- Web Bridge (Basis link, muss konfiguriert werden): <http://vse-ip:cws-port/cics/CWBA/dfhwbtta/cemt>
- Conversion Tabelle: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/dfhwtl06/2.3.2.7

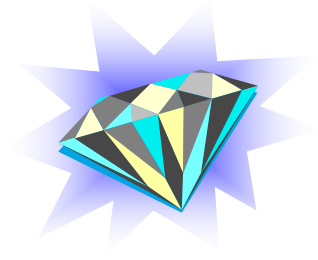
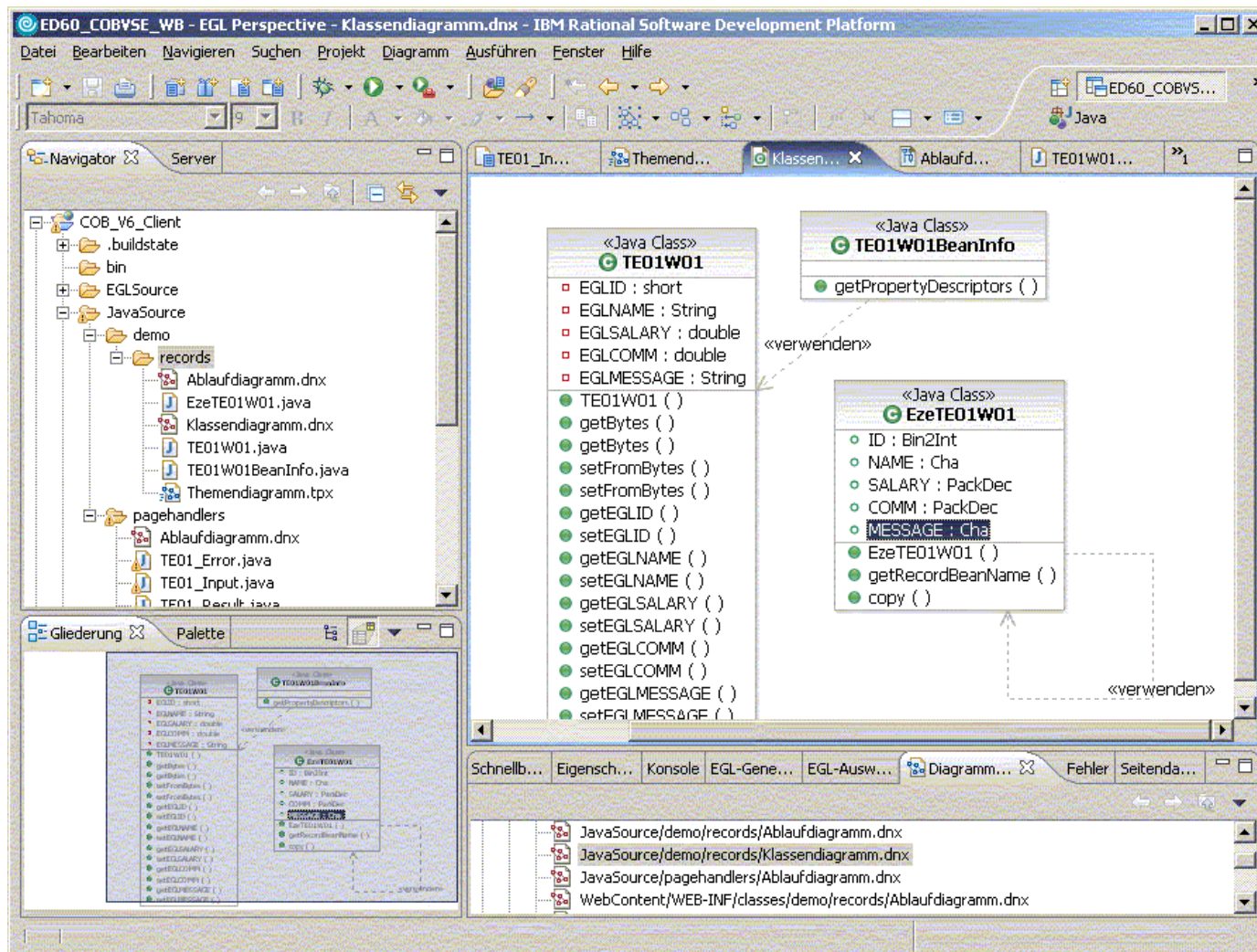
DB2/VSE (optional)

- Planung: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/iespie41/10.4.5
- Aktivierung: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/iespie41/10.4.4
- Anpassung: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IESWUE41/HDRINDB2BC

© 2005 IBM Corporation

Rational Application Developer (RAD)

Integrierte Entwicklungsumgebung (IDE) – UML Editor/Darstellung



**Funktionalität für
Enterprise Development**

- **Zusätzliche Perspektiven**
EGL ...
- **Migrationshilfen:**
EGL V6 Migration, VisualAge
Generator Migration
- **Kontrollfenster/Sichten**
EGL Generierung-/Auswertung,
SQL Fehler...
- **Weitere Developer Tools:**
z.B. Page Designer, Web Site
Designer, Portal Tools, Service
Data Objects, XML Tools, UML
Visual Editor usw.

Tech-Artikel: **IBM Rational Developer: Powerful support for rapid Java and J2EE development**

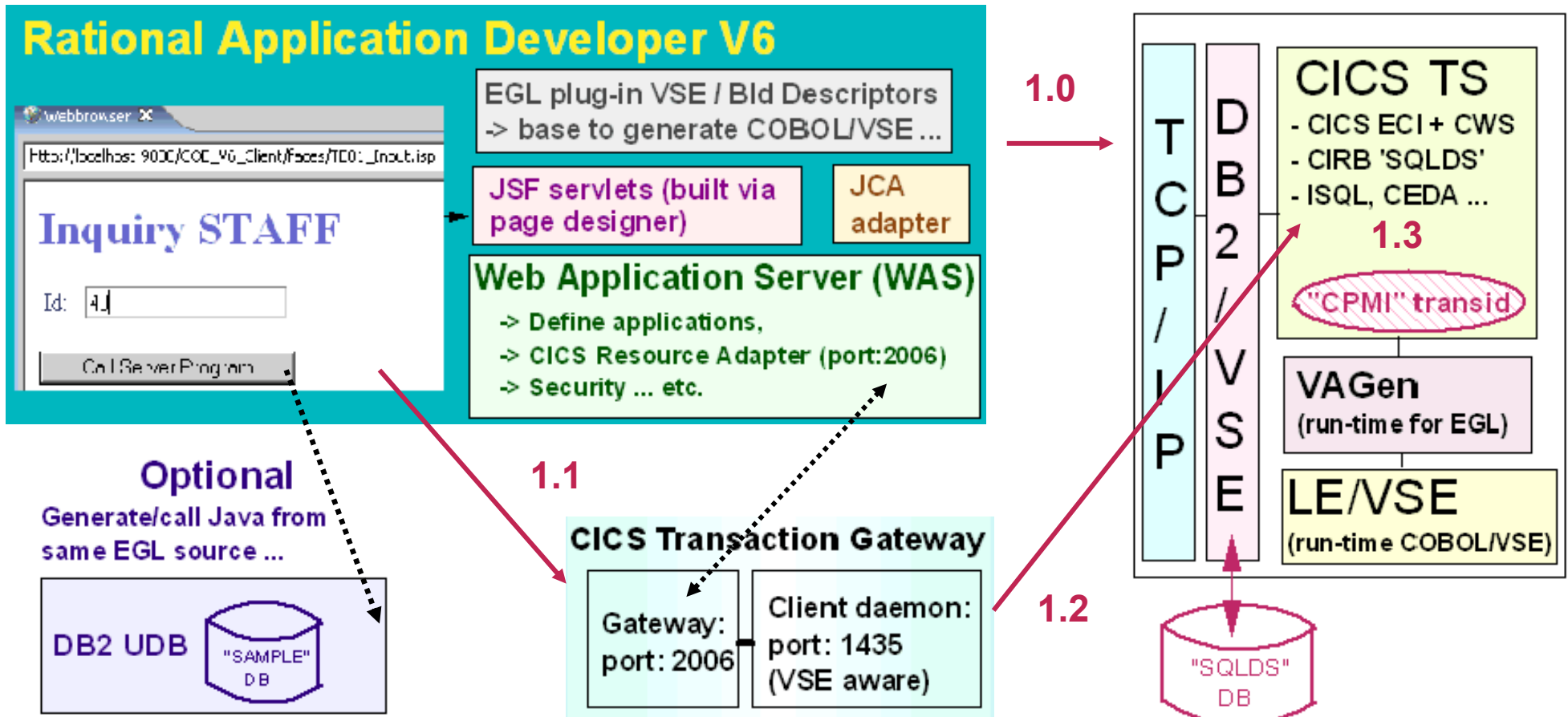
<http://www.ibm.com/developerworks/rational/library/dec04/parkin/>

© 2005 IBM Corporation

Beispiel: „COBV6“ - Entwicklungsumgebung

PC / Workstation

z/VSE 3.1 Server



- 1.0 – Generieren + ftp des COBOL/CICS/DB2 Programms
 - 1.1 – JSP Aufruf des COBOL Server Programms
 - 1.2 – CICS ECI (gerufene Programme via „CPMI“)
 - 1.3 – Ausführung auf dem Server
- Action Sequence*

© 2005 IBM Corporation

Software Details zur Testumgebung

❑ VisualAge Generator EGL Plug-in for VSE – Voraussetzungen:

- IBM Rational Application Developer (RAD) 6.0.0.1 + ifix level 001, 002 + 003
- [WebSphere Application Server \(WAS\) V5.1 Run-Time \(optional\)](#)
- WebSphere Application Server (WAS) V6 Run-Time
- [IBM Rational Agent Controller](#)

❑ z/VSE Server - Voraussetzungen:

- VSE operating system (vorzugsweise z/VSE 3.1)
- [VisualAge Generator Server for VSE Version 1.2](#)
- CICS Transaction Server 1.1.1
- [TCP/IP 1.5](#)
- COBOL/VSE compiler 1.1.0
- [LE/VSE 1.4.4 run-time](#)
- DB2/VSE 7.4 + „SQLDS“ database

❑ Workstation Voraussetzungen:

- DB2 UDB 8.1 Enterprise Server Edition + „SAMPLE“ database
- [IBM CICS Transaction Gateway V6.0.1](#)

© 2005 IBM Corporation

Überblick der wichtigsten Definitionen

❑ z/VSE Server seitig:

- DB2/VSE batch & online, Sample DB „SQLDS“, IVALPKEY + ISQL + CIRB
- VisualAge Generator, CICS Resource Definitionen (u.a. TD-queues, files, CPMI:TWASIZE=1024!)
- CICS SIT, JCT, startup Modifikationen für CICS CWS + CICS ECI
- TCP/IP 1.5 (ftp-Transfer zur Zielmaschine)
- LE/VSE 1.4.4 Laufzeit (+ GA PTFs, besser noch + PSP upgrade)

❑ Workstation:

- Windows: den Service „Server“ aktivieren
- CICS Transaction Gateway: Konfiguration der Zielmaschine
- Rational Application Developer - IDE:
 - Definieren der EGL Projekte/parts/Build Deskriptoren, benutzerspezifische- Arbeitsbereiche, Perspektiven
 - Sehr lange Projektnamen sollten vermieden werden (Windows Restriktionen / Pfadlängen)
 - Einsatz von Tools z.B. dem Page Designer um EGL & JSP Dateien zu erzeugen
 - Alternativ können existierende Projekt .zip Dateien importiert/angepasst werden (Projektaustausch/Refresh)
 - Überprüfen der EGL templates auf die gewünschte COBOL/VSE code Generierung z.B. Optionen
 - WAS V6 Server Konfiguration
 - ❑ Ressourcen, Adapter, Security ... am besten im gestarteten Modus mit der „Administrationskonsole“
 - ❑ Hinzufügen/löschen von Projekten kann im „down“ Modus erfolgen
 - ❑ Das setting „SOAP“ eignet sich für firewalls (anstatt „RMI“)
 - Benutzervorgaben -> Workbench -> Funktionen: das „EGL Entwickler“ Profil aktivieren (mit EGL Migration V6)
 - Benutzervorgaben -> EGL -> SQL Abruf: „EGL Typ char“ ist sinnvoll („string“ ungeeignet für COBOL gen)
 - Importieren von JAR files (z.B. für DB2 UDB Treiber, VSE Connectors), Namen können case-sensitive sein
- DB2 UDB Control Center: Verwaltung/Synchronisation der DB2 sample Datenbanken/Tabellen (workstation + VSE). Diese sind nicht identisch !

© 2005 IBM Corporation

Generieren des Client - JSP Interfaces ...

Client Java - front-end Programme / GUI interface

- Page Design für die JSP's ist abgeschlossen, Build Deskriptoren existieren ...
- Jede JSP hat korrespondierende „pagehandler“, EGL + Java Quell-Dateien ...

The screenshot shows the IBM Rational Software Development Platform interface. The main window is titled 'ED60_COBYSE_WB - EGL Perspective - IBM Rational Software Development Platform'. The 'Navigator' pane on the left shows a project structure for 'COB_V6_Client' with folders like '.buildstate', 'bin', 'EGLSource', 'pagehandlers', 'JavaSource', and 'WebContent'. The 'pagehandlers' folder contains files 'TE01_Error.egl', 'TE01_Input.egl', and 'TE01_Result.egl'. The 'Webbrowser' pane at the bottom shows the URL 'http://localhost:9080/COB_V6_Client/faces/TE01_Input.jsp' and a web page titled 'Inquiry STAFF' with an input field for 'Id: 40' and a 'Call Server Program' button.

The 'Generieren' (Generate) dialog box is open, prompting the user to select build descriptors for the target system. The dialog has the following content:

Build-Deskriptoren
Wählen Sie Build-Deskriptoren für das Zielsystem aus.

Für alle Abschnitte einen Build-Deskriptor verwenden:
 Für jeden Abschnitt einen Build-Deskriptor auswählen:

The selected descriptor is: `ics-client <COB_V6_Client/EGLSource/COB_V6_Client.eglbld>`

Abschnitt	Build-Deskriptor
TE01_Error	COB_V6_ClientWebBuildOptions <COB_V6_Client/EGLSource/COB_V6_Clie
TE01_Input	COB_V6_ClientWebBuildOptions <COB_V6_Client/EGLSource/COB_V6_Clie
TE01_Result	COB_V6_ClientWebBuildOptions <COB_V6_Client/EGLSource/COB_V6_Clie

Buttons at the bottom: < Zurück, Weiter >, Fertig stellen, Abbrechen.

© 2005 IBM Corporation

Generieren der COBOL/VSE Server Parts ...

- Das COBOL/VSE Server Programm wird vom JSP/EGL front-end gerufen
 - Build Deskriptoren sind angepasst, ... Die Verbindung zu VSE existiert usw.

The screenshot displays the IBM Rational Software Development Platform interface for generating COBOL/VSE server parts. The main window is titled "ED60_COBVSE_WB - EGL Perspective - IBM Rational Software Development Platform".

Build-Deskriptoren: The "Generieren" dialog shows the configuration for building the server parts. The "Benutzer-IDs und Kennwörter" section is filled with the following details:

- sqlID: (empty)
- sqlPassword: (empty)
- destUserID: bosh
- destPassword: *****

Terminal Window (Session C - [32 x 80]): The terminal shows the execution of the program, displaying system information and job details:

```

SYSTEM: z/VSE z/VSE 3.1 TURBO (01) USER: BOSH TIME: 09:17:20
VM USER ID: LEUSER
BG 0000 EOJ SQL MAX.RETURN CODE=0004
DATE 07/25/2005, CLOCK 08/45/38, DURATION 00/00/04
BG 0001 1Q47I BG TE01AJ 05449 FROM (SYSTCPIP), TIME= 8:45:38
BG 0000 // JOB CICS TRANSLATE TE01A
DATE 07/25/2005, CLOCK 08/45/38
BG 0000 1T20I SYSIPT HAS BEEN ASSIGNED TO X'FEC' (TEMP)
BG 0000 EOJ CICS MAX.RETURN CODE=0000
DATE 07/25/2005, CLOCK 08/45/39, DURATION 00/00/00
BG 0001 1Q47I BG TE01AJ 05450 FROM (SYSTCPIP), TIME= 8:45:39
BG 0000 // JOB COMPILE TE01A
DATE 07/25/2005, CLOCK 08/45/39
BG 0000 1T20I SYSIPT HAS BEEN ASSIGNED TO X'FEC' (TEMP)
BG 0000 EOJ COMPILE MAX.RETURN CODE=0000
DATE 07/25/2005, CLOCK 08/45/44, DURATION 00/00/05
BG 0001 1Q47I BG TE01AJ 05451 FROM (SYSTCPIP), TIME= 8:45:44
BG 0000 // JOB LINK EDIT TE01A
DATE 07/25/2005, CLOCK 08/45/44
BG 0000 1T20I SYSIPT HAS BEEN ASSIGNED TO X'FEC' (TEMP)
BG 0000 EOJ LINK MAX.RETURN CODE=0002
DATE 07/25/2005, CLOCK 08/45/46, DURATION 00/00/01
BG 0000 // JOB CLEANUP TEMPORARY PROCEDURE
DATE 07/25/2005, CLOCK 08/45/46
BG 0000 1S78I JOB TERMINATED DUE TO 'GOTO $EOJ'
BG 0000 EOJ CLEANUP MAX.RETURN CODE=0000
DATE 07/25/2005, CLOCK 08/45/46, DURATION 00/00/00
    
```

Server Explorer: The "Server" window shows the file structure of the generated server parts, including folders like "codegen" and "Server", and files like "TE01A.cbl", "TE01A.jcp", and "TE01A.ppt".

© 2005 IBM Corporation

Aufruf des COBOL/VSE Server Programms via EGL front-end

Das Java front-end Programm / „Application Driver“

- Die DB2/VSE Beispieldatenbank „SQLDS“ existiert & ist zugreifbar, die ...
- „STAFF“ Tabelle wurde in „SQLDS“ ergänzt (Quelle: DB2/UDB: „SAMPLE“)

The screenshot illustrates the development environment for the COBOL/VSE server program. The EGL Perspective window shows the project structure for COB_V6_Client, including folders like .buildstate, bin, EGLSource, JavaSource, WebContent, META-INF, theme, WEB-INF, and files like TE01_Error.jsp, TE01_Input.jsp, and TE01_Result.jsp. The Web browser window shows the URL http://localhost:9080/COB_V6_Client/faces/TE01_Input.jsp and displays the 'STAFF Query Result' page. The result page shows: Id: 30, Name: Marenghi, Salary: 17506,75, and Comm: (empty). The console window at the bottom shows system properties: res-resolution-control: 999 (undefined), isCMP1_x: false (not CMP1.x), and isJMS: false (not JMS).

© 2005 IBM Corporation

WebSphere Application Server V6

❑ Schnappschuss (Definition: Resource Adapter/Connection Factory)

Administrationskonsole

Willkommen wbosch | Abmelden | Support | Hilfe

Willkommen

Server

Anwendungen

Ressourcen

- JMS-Provider
 - JDBC-Provider
 - Ressourcenadapter
- Asynchronous Beans
 - Scheduler
- Cache-Instanzen
 - Object Pool Manager
 - Mail-Provider
 - URL-Provider
 - Ressourcenumgebungsprovider

Sicherheit

Umgebung

Systemverwaltung

Überwachung und Optimierung

Fehlerbehebung

Serviceintegration

UDDI

Ressourcenadapter

Ressourcenadapter

Nachrichten

⚠ Es wurden Änderungen an der lokalen Konfiguration vorgenommen. Klicken Sie auf [Speichern](#), um die an der Master-Konfiguration vorgenommenen Änderungen anzuwenden.

ℹ Der Server muss möglicherweise erneut gestartet werden, damit die Änderungen wirksam werden.

Ressourcenadapter > CICSECConnector > J2C-Verbindungs-Factories > LEUSER > Benutzerdefinierte Merkmale

Benutzerdefinierte Merkmale, die möglicherweise für Ressourcenprovider und Ressourcen-Factories erforderlich sind. Die meisten Datenbanklieferanten beispielsweise erfordern zusätzliche benutzerdefinierte Merkmale für Datenquellen, die auf die Datenbank zugreifen.

Einstellungen

Name	Wert	Beschreibung	Erforderlich
TPNName	CPMI		false
ClientSecurity			false
ConnectionURL	tcp://127.0.0.1		false
KeyRingClass			false
KeyRingPassword			false
Password			false
PortNumber	2006	PortNumber	false
ServerName	LEUSER		false
ServerSecurity			false
TraceLevel	1	TraceLevel	false
TranName			false
UserName			false

Insgesamt 12

- Applikationen benutzen die J2EE connection Architektur (J2C/JCA)
- Kein Direktzugriff auf den Resource Adapter !
- TPNName: via ECI gerufene Programme laufen unter der Server transid „CPMI“ (als Attribut muss TWASIZE=1024 definiert sein !)
- Username + Password sind zu ergänzen
- ServerName muss angepasst werden

🌟 Die ECI /J2C connection factory der WASV6 Administrationskonsole (u.a. wird hier der CTG port definiert !)

© 2005 IBM Corporation

Typen von EGL-Programmen ...

❑ Internet Applikationen:

- Standalone (via WebSphere Application Server)
- Remote calls zu legacy business logic (C.I.C.S.), EGL generiert oder native COBOL

❑ Aufrufbare Programme

- Java clients (java wrappers)

❑ Standalone Applikationen & herkömmliche „green-screen“ Anwendungen

- Generiert COBOL und deployed in VSE Batch
- ... sowie VSECICS Umgebungen

❑ Für wen kommt EGL in Betracht ?

- Alle die „Business Logik“ implementieren und auf Plattformunabhängigkeit Wert legen
- Im Besonderen für: Datenbank, Java-, Host-, VisualAge Generator- Entwickler

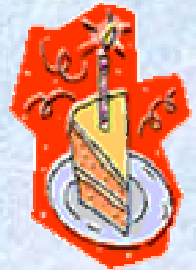
❑ Welche Veränderungen gibt es im Entwicklungsprozess ?

- Spezifisches Sprachwissen ist nachrangig (COBOL, Java ...)
- Stattdessen liegt der Fokus auf der EGL-Sprache und Erstellung von Build Deskriptoren
- Kodierung und Programmpflege erfolgt in EGL, nicht den generierten Programmen !
- Die Datenimplementierung wird ebenfalls in EGL vorgegeben und kann variiert werden z.B. „indexedRecord“ (VSAM KSDS) <-> „sqlRecord“ usw.

© 2005 IBM Corporation

Zusammenfassung

- **Nutzen Sie die von z/VSE angebotene e-business Infrastruktur für die AE**
 - VSE Connectors, VSE Redirector, Solution Scenarios“ (→ z/VSE home) usw.
- **Prüfen Sie die Möglichkeit einer plattformunabhängigen Implementierung (insbesondere für neue Anwendungen ... z.B. mit target: Linux)**
 - WSAD-/RAD-based development kann host-orientiert sein und diesen entlasten
 - EGL hat zahlreiche Ausprägungen zur Generierung von Server Programmen
 - Flexibilität und eine modernere Anwendungslandschaft sind der Gewinn
- **Vermeiden Sie unnötige Risiken durch Komplettumstellungen**
 - Das „Wagenrad“ neu zu erfinden kann wertvolle AE-Ressourcen binden, oft mit unsicherem Ausgang des Projekts - oder ohne „echten Gewinn“
 - Bewährte Funktionalität ist möglicherweise in Vergessenheit geraten, kann nicht ohne Aufwand neu abgebildet werden oder ungewollte Abhängigkeiten entstehen
 - Trends sollten kritisch im Hinblick auf zentrale Unternehmensprozesse überprüft werden. Oft kann man ihnen auf verhältnismäßig „sicheren“ Pfaden folgen !
- **Wo „klassische“ AE sinnvoll ist, sollte man sie beibehalten**
 - Der host bietet zahlreiche Services (z.B. Assembler Treiber, Makros, Programmier-Techniken usw.), die in anderen Umgebungen nicht sinnvoll genutzt werden können



© 2005 IBM Corporation