RDz Cobol Tutorial 01
RDz Einführung

Dank an Frau Isabel Arnold, die in Hamburg im August 2006 eine IBM Training Session veranstaltete. Hier wurde das Material in dieser und in den beiden folgenden Tutorials vorgeführt. Die ursprüngliche Version des Materials wurde von Reginaldo W. Barosa, IBM Executive IT Specialist, erstellt.

„Rational Developer für System z“ (RDz) ist eine auf Eclipse basierende moderne Entwicklungsumgebung, welche unter anderem ein universelles Werkzeug für die Anwendungsentwicklung auf Mainframes darstellt. RDz wird in immer mehr Unternehmen für die Anwendungsentwicklung eingesetzt.

Nahezu alles, was unter Nutzung eines 3270-Emulators auf einem Mainframe möglich ist, ist auch unter RDz möglich; z.B. Datasets anlegen (allocate), Datasets mit einem Editor bearbeiten, Jobs in Form von JCL-Scripten mittels "SUB" ausführen, Output-Jobs ansehen, Member kopieren etc.

Dieses Tutorial demonstriert anhand von Beispielen den grundlegenden Umgang mit RDz. Details in den RDz Tutorials 2 – 4.

Voraussetzungen für die Durchführung dieses Tutorials:

- Internetfähiger Windows PC
- PRAK(xxx)-Login auf Leia oder Hobbit
- Login auf unseren virtuellem RDz-Server mit installiertem RDz.

Wir verwenden für die verschiedenen Tutorial Texte 3 unterschiedliche RDz Versionen, die über unterschiedliche IP Adressen gestartet werden:

- 139.18.8.211 für Version 6.0 (WSED)
- 139.18.8.212 für Version 7.0 (WDz)
- 139.18.8.214 für Version 7.5 (RDz)

} Der Einfachheit halber bezeichnen wir alle Versionen als RDz, obwohl dies nicht den IBM Bezeichnungen entspricht.

Der Text für dieses Tutorial basiert auf 139.18.8.212 für Version 7.0 (WDz), kann aber von Ihnen auch unter 139.18.8.211 für Version 6.0 oder 139.18.8.214 für Version 7.5 (RDz) durchgeführt werden. Die ebenfalls verfügbaren Tutorials in der Programmiersprache PL/1 verwenden 139.18.8.214 für Version 7.5 (RDz). Zu beachten ist lediglich, dass unterschiedliche Portnummern für JES und MVS Files angegeben werden müssen. Wir planen alle Tutorials nach RDz 8.0 zu konvertieren – really soon now -. Die Unterschiede sind jedoch vernachlässigbar.

Inhalt

Dieses Tutorial und die folgenden Tutorials sind installiert auf dem:
z/OS 1.8 System hobbit.cs.informatik.uni-tuebingen.de, oder 134.2.205.54, der Abteilung Technische Informatik der Universität Tübingen, und
z/OS 1.8 System leia.informatik.uni-leipzig.de, oder 139.18.4.30, der Abteilung Technische Informatik der Universität Leipzig

Alle RDz Installationen wurden durchgeführt von Isabel Arnold, Uwe Denneler, Elisabeth Puritscher und Mr. Martin Benjamin Storz, und unterstützt von Andreas Hermelink (alle IBM). Lokale Unterstützung ist verfügbar von Andreas Nagel (Universität Tübingen) sowie (Frank Güttler (Universität Leipzig).

Dieses Tutorial verwendet die folgenden Konventionen

1k bedeutet 1 Klick mit der linken Maustaste
2k bedeutet 2 Klick mit der linken Maustaste
1kr bedeutet 1 Klick mit der rechten Maustaste

1. Übersicht

Software wird normalerweise in einer Entwicklungsumgebung erzeugt und anschließend in einer unterschiedlichen Produktionsumgebung ausgeführt. Traditionelle z/OS Entwicklungsumgebungen sind TSO mit ISPF, sowie CMS unter dem z/VM Betriebssystem. Typische z/OS Produktionsumgebungen sind JES, CICS, IMS, DB2 Stored Procedures und der WebSphere Web Application Server (WAS).

Ein Mainframe Rechner verfügt normalerweise über mehreren Logische Partitionen (LPARs), von denen eine als Entwicklungsumgebung und eine oder mehrere parallel als Produktionsumgebungen dienen.

Die Übernahme einer neu entwickelten Anwendung von der Entwicklungsumgebung in die Produktionsumgebung ist in der Regel ein sehr komplexer und aufwendiger Prozess. Normalerweise wird eine Test Umgebung zwischengeschaltet, die in einer weiteren getrennten LPAR untergebracht ist.. Während die Entwicklungsumgebung sehr viel anders als die Produktionsumgebung sein kann (schließlich soll Code nur entwickelt und debugged, nicht aber unter praxisnahen Bedingungen ausgeführt werden), versucht eine Testumgebung eine Produktionsumgebung möglichst naturgetreu abzubilden. Eine neu entwickelte Mainframe Anwendung wird zunächst in der Testumgebung auf Herz und Nieren überprüft, ehe sie in der Produktionsumgebung installiert und für den täglichen Einsatz freigegeben wird.

Häufig besteht eine neue z/OS Anwendung aus einem Teil, der auf dem Mainframe, und einem weiteren Teil, der auf einem Linux oder Windows Vorrechner läuft. Sehr typisch ist z.B. eine CICS Anwendung auf dem Mainframe, deren graphische Präsentation auf einem Linux/Windows Server mit Hilfe des WebSphere Application Servers implementiert wird.

Bis etwa zum Jahre 2000 wurden neue Mainframe Anwendungen fast ausschließlich mit Hilfe einer Entwicklungsumgebung implementiert, die ebenfalls auf dem Mainframe lief. Seitdem gewinnt zunehmend eine Alternative an Bedeutung, bei der die Entwicklungsumgebung auf einem Windows Server läuft. Diese Entwicklungsumgebung wurde seit 2000 von IBM unter unterschiedlichen Namen vertrieben, z.B. Websphere Studio Enterprise Developer (WSED) oder WebSphere Developer for System z (WDz). Die derzeitige Produktbezeichnung ist „Rational Developer for System z“ (RDz). In der Praxis ist RDz eine sehr evolutionäre Weiterentwicklung von WSED, und RDz eine ebenso evolutionäre Weiterentwicklung von WDz. Die Versionen bedingen unterschiedliche Installationen auf der Workstation und der z/OS Host Seite. Bei den vorliegenden Tutorials werden die Funktionsunterschiede jedoch nicht sichtbar.

Bei der Benutzung von RDz wird der Quellcode (COBOL, PL/I, C/C++, Java oder Assembler).auf einem Windows Rechner erzeugt, kompiliert und ausgetestet. Der gleiche Code kann auch auf dem Mainframe kompiliert und ausgeführt werden. Dieses parallele Vorgehen wird von RDz hervorragend unterstützt.

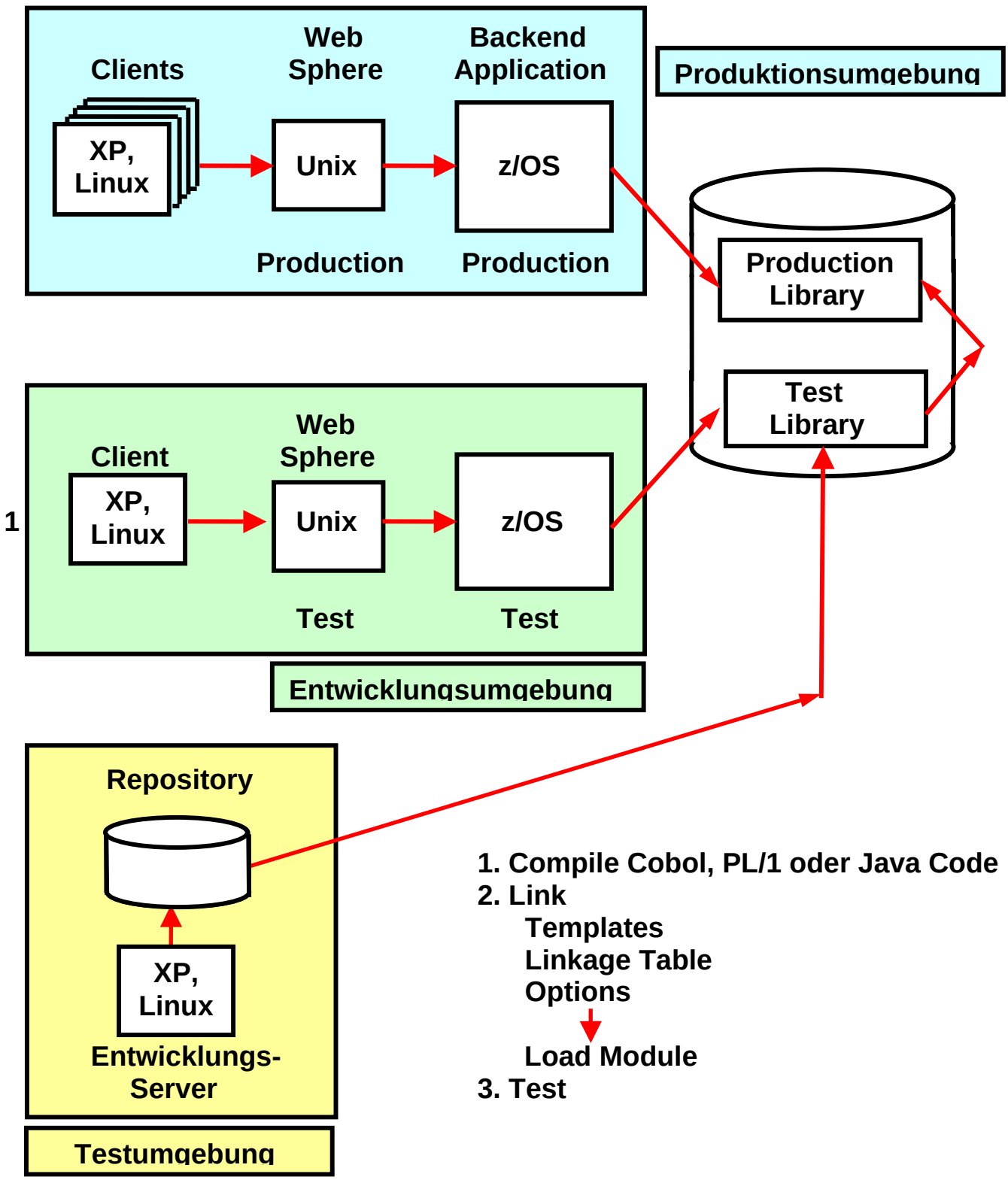


Abb. 1.1 Überführung von der Entwicklung in die Produktion

2. Integrierte Entwicklungsumgebung

Eine integrierte Entwicklungsumgebung (Abkürzung IDE, von engl. integrated Development Environment) ist eine Sammlung von Anwendungsprogrammen, mit denen die Aufgaben der Softwareentwicklung möglichst ohne Medienbrüche bearbeitet werden können.

Integrierte Entwicklungsumgebungen verfügen in der Regel über folgende Komponenten:

- **Texteditor**
- **Compiler bzw. Interpreter**
- **Linker**
- **Debugger**
- **Quelltextformatierungsfunktion**

Umfangreichere integrierte Entwicklungsumgebungen enthalten oft weitere hilfreiche Komponenten wie Versionsverwaltung, Projektmanagement, UML-Modellierung oder die Möglichkeit der einfachen Erstellung von grafischen Benutzeroberflächen (GUI).

In erster Linie sind integrierte Entwicklungsumgebungen hilfreiche Werkzeuge, die dem Softwareentwickler häufig wiederkehrende Aufgaben abnehmen, einen schnellen Zugriff auf wichtige Funktionen bieten, mit denen die Arbeits-(zwischen)-ergebnisse verwaltet und in spätere Bearbeitungsfunktionen direkt überführt werden können. Der Entwickler wird dadurch von formalen Arbeiten entlastet und kann sich ganz auf seine eigentliche Aufgabe, die Softwareentwicklung/Programmierung konzentrieren.

IDEs gibt es für nahezu alle Programmiersprachen und Plattformen. Oft wird nur eine Programmiersprache unterstützt. Es gibt aber auch IDE's, die mehrere Programmiersprachen unter einer gemeinsamen Benutzeroberfläche zusammenfassen.

Integrierte Entwicklungsumgebungen kamen in der ersten Hälfte der 1980er Jahre auf und lösten die damals übliche Praxis ab, Editor, Compiler, Linker und Debugger als vier getrennte Produkte anzubieten, die vom Benutzer über die Kommandozeile ausgeführt wurden. Eine der ersten erfolgreichen IDEs war Turbo Pascal.

Während die ersten IDEs noch textbasiert arbeiteten, ging der Trend vor allem bei den großen Anbietern ab 1990 zunehmend hin zu visuellen Programmierumgebungen.

http://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung

2.1. Die wichtigsten IDE's

Der Oracle Developer enthält als wichtigste Komponente den JDeveloper. Der JDeveloper ist eine kostenlose Integrierte Entwicklungsumgebung (IDE) von Oracle, und unterstützt Sprachen wie Java, XML, SQL and PL/SQL, HTML, JavaScript, BPEL and PHP.

Visual Studio ist eine von dem Unternehmen Microsoft angebotene, integrierte Entwicklungsumgebung. Die wichtigsten unterstützten Sprachen sind VB.NET (Visual Basic .NET), C, C++ und C# .

NetBeans ist eine Open-Source Entwicklungsumgebung, die komplett in der Programmiersprache Java geschrieben wurde und auf der NetBeans Plattform läuft. Die NetBeans IDE wurde hauptsächlich für die Programmiersprache Java entwickelt, unterstützt jedoch auch C, C++ und dynamische Programmiersprachen.

Eclipse ist ein quelloffenes Programmierwerkzeug zur Entwicklung von Software verschiedenster Art. Ursprünglich wurde Eclipse als integrierte Entwicklungsumgebung für die Programmiersprache Java genutzt, aber mittlerweile wird es wegen seiner Erweiterbarkeit auch für viele andere Entwicklungsaufgaben eingesetzt. Für Eclipse gibt es eine Vielzahl sowohl quelloffener als auch kommerzieller Erweiterungen.

Das SAP NetWeaver Developer Studio (NWDS) von SAP ist eine auf Eclipse basierende integrierte Entwicklungsumgebung. Die IDE integriert sowohl Java-Technologien (JSE, JEE, XML, ...) als auch SAP-Technologien (Web Dynpro, Java Dictionary, ...). Es kommen viele Plug-ins des Web Toolkit Projekts zum Einsatz und die IDE wurde mit proprietären Erweiterungen von SAP ergänzt. Damit können Web-Dynpro- und JEE-Applikationen als Java-EE-konforme Anwendungen erstellt werden, welche auf SAPs Java EE NetWeaver Application Server zum Einsatz kommen.

Eclipse wurde ursprünglich von IBM entwickelt, dann aber als Open Source der Allgemeinheit zur Verfügung gestellt. Heute bieten zahlreiche Unternehmen proprietäre Erweiterungen (plugins) für Eclipse an, darunter Borland, HP, IBM und SAP.

IBM vermarktet proprietäre Eclipse Erweiterungen unter dem Namen Rational, spezifisch den Rational Application Developer (RAD). Ähnlich wie das SAP NetWeaver Developer Studio in erster Linie für Neuentwicklungen für den SAP Java EE NetWeaver Application Server eingesetzt wird, benutzt man RAD für Neuentwicklungen für den IBM WebSphere Application Server.

Eine Sonderstellung nimmt der IBM Rational Developer for System z (RDz) ein. RDz hat viele Gemeinsamkeiten mit RAD und basiert wie dieses auf Eclipse. RDz ist heute die führende IDE für die Entwicklung neuer z/OS Anwendungen und löst zunehmend TSO und ISPF in dieser Rolle ab. Allerdings wird die Benutzung von RDz wesentlich erleichtert, wenn Grundkenntnisse in TSO und ISPF vorhanden sind.

Selbst-Test

- **Muss man für die Entwicklung neuer z/OS Anwendung eine IDE benutzen ?**
- **Kann man an stelle von RDz auch Eclipse oder NetBeans oder NWDS benutzen ?**

2.2. Eclipse

Eclipse ist ein Open-Source-Framework zur Entwicklung von Software nahezu aller Art. Die bekannteste Verwendung ist die Nutzung als Entwicklungsumgebung (IDE) für die Programmiersprache Java. Aber auch für die Entwicklung von Rich-Client-Applikationen auf Basis der Eclipse Rich Client Platform (RCP) wird es zunehmend häufiger eingesetzt. Eclipse ist nicht auf Java festgelegt und wird aufgrund seiner offenen Plug-in-basierten Struktur mittlerweile für sehr unterschiedliche Entwicklungsaufgaben eingesetzt. Für Eclipse existieren eine Vielzahl von Plug-ins sowohl von Opensource-Projekten als auch kommerziellen Anbietern. Es existieren Plugins für weitere Programmiersprachen, unter anderem für C, C++, Perl, PHP, Ruby und Python.

Eclipse basiert auf der IDE „Visual Age for Java 4.0“, die von IBM entwickelt wurde. Im November 2001 wurde der Quellcode für das Programm freigegeben und seitdem kontinuierlich als Open Source Projekt erweitert.

Das von IBM geleitete Eclipse Konsortium umfasst über 80 Mitglieder neben IBM sind unter anderem Borland, MERANT, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft und Webgain vertreten. Das Konsortium gründete im Januar 2004 die rechtlich unabhängige Eclipse Foundation, die seitdem für die Entwicklung von Eclipse verantwortlich ist. Etwa die Hälfte der derzeitig am Eclipse-Basisframework arbeitenden Entwickler werden weiterhin von IBM bezahlt.

Eclipse basiert auf einem Prinzip, bei dem sich alles als Plugin integrieren lässt.

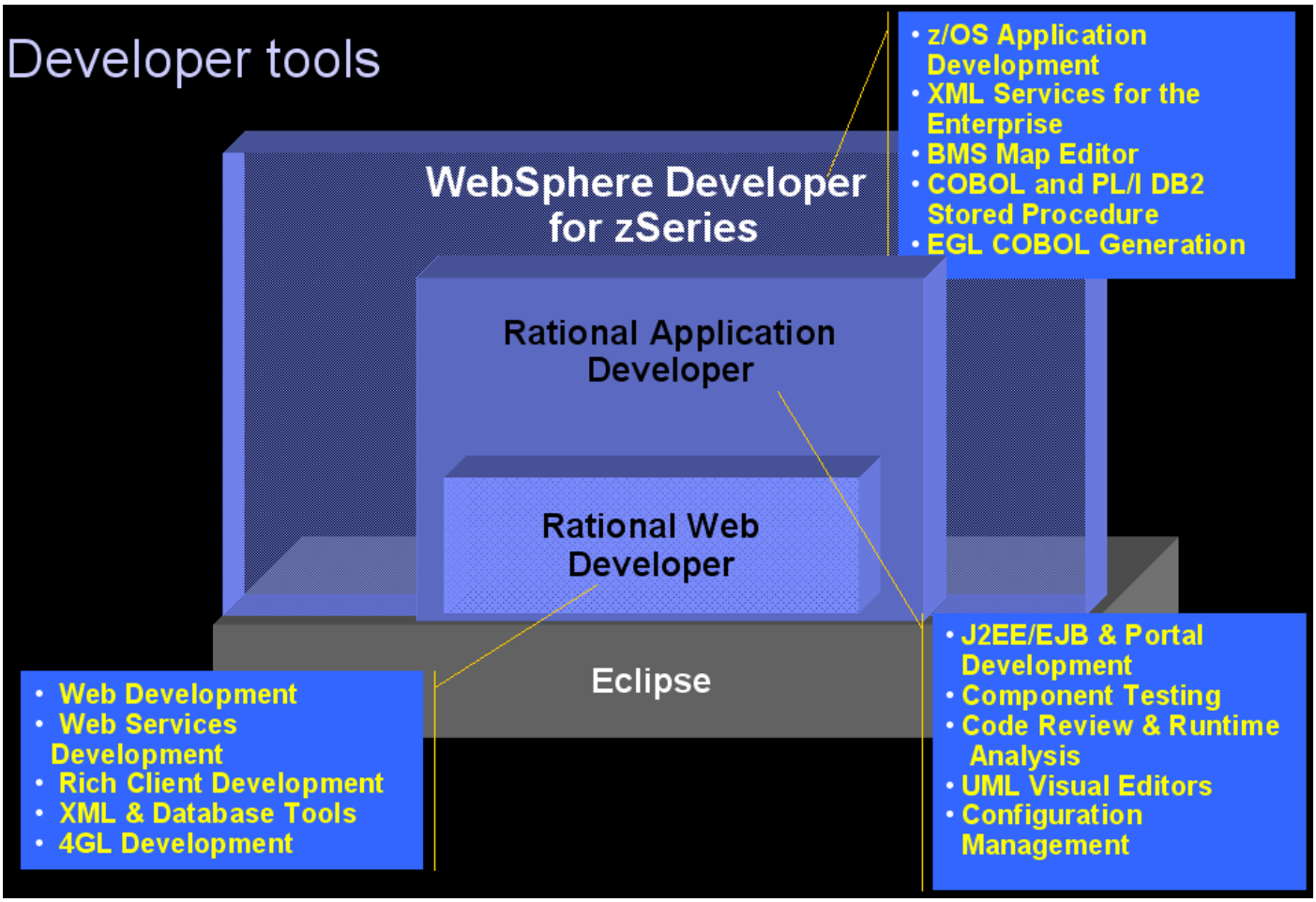
Seit Version 3.0 ist Eclipse selbst nur der Kern, der die einzelnen Plug-ins lädt, die dann die eigentliche Funktionalität zur Verfügung stellen. Diese Funktionalität basiert auf dem OSGi-Standard. Sowohl Eclipse als auch die Plugins sind vollständig in Java implementiert. Als GUI-Framework zur Erstellung der grafischen Oberfläche wurde SWT verwendet. Zur Darstellung der GUI-Komponenten basiert SWT ähnlich wie AWT auf den nativen GUI-Komponenten des jeweiligen Betriebssystems. Eclipse ist daher nicht platformunabhängig, wird aber für 14 verschiedene Systeme und Architekturen bereitgestellt. Die Plug-ins lassen sich durch den Download direkt in Eclipse von einem Update-Server oder durch einfaches Entpacken installieren.

Das frei verfügbare Eclipse SDK umfasst die Eclipse Platform, Werkzeuge zur Java-Entwicklung (Java Development Tools JDT) und die Umgebung zur Entwicklung von Eclipse-Plug-ins (Plug-in Development Environment PDE).

Die Eclipse Open Source Community versteht sich als Gemeinschaft, die eine kostenlose, erweiterbare Entwicklungsumgebung und ein Framework für die Laufzeit und Anwendungsentwicklung zur Verfügung stellt, welches die Entwicklung, Betreuung und Wartung eines Softwareprojekts über dessen gesamte Lebensdauer begleitet. Die Community stellt über 60 verschiedene Open Source Projekte bereit, die in sieben Kategorien unterteilt werden:

- **Enterprise Development**
- **Embedded und Device Development**
- **Rich Client Platform**
- **Rich Internet Applications**
- **Application Framework**
- **Application Lifecycle Management (ALM)**
- **Service Oriented Architecture (SOA)**

2.3 Rational Developer



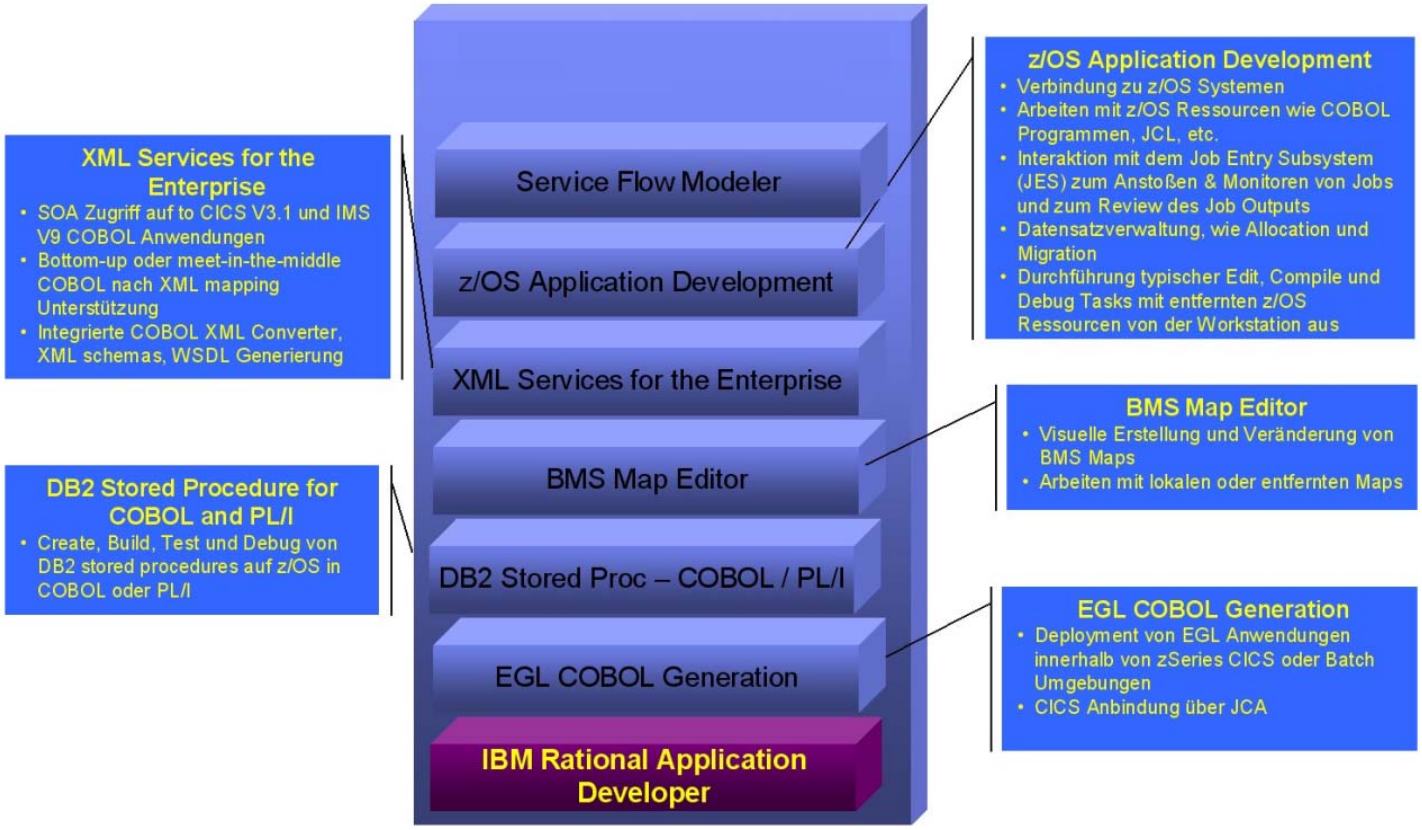
Von IBM existieren eine ganze Reihe von Eclipse Erweiterungen, die als Plugins in Eclipse integriert werden können. Die wichtigsten sind:

- **Rational Web Developer Plugin**
- **Rational Application Developer Plugin, welches das Rational Web Developer Plugin enthält.**
- **WebSphere Developer for System z Plugin, heute als Rational Developer for System z bezeichnet. WDz bzw. RDz enthalten das Rational Application Developer Plugin.**

Das von IBM bereitgestellte Plugin Rational Application Developer (RAD) unterstützt

- **die Entwicklung von HTML, XML, Java, JSP, Servlets, Enterprise JavaBeans und Webservices.**
- **eine integrierter Testumgebung, Debugger, Codegeneratoren für Tests**
- **integrierte Build-Prozesse.**

2.4 Rational Developer for System z (RDz)



Das von IBM bereitgestellte Plugin Rational Developer for System z (RDz) bietet zusätzlich zu der RAD Funktionalität Einrichtungen

- **zur integrierten Entwicklung von Großrechneranwendungen, z.B. die Entwicklung von COBOL, PL/I, C/C++, Java und Assembler Programmen**
- **für Entwicklungen für die Transaktionsmonitore IMS und CICS,**
- **für die Entwicklung von DB2 Stored Procedures.**

Sie können mit RDz auch Anwendungen entwickeln, die später unter Windows oder anderen Betriebssystemen laufen. Im Prinzip würde dafür das Basis Eclipse System ausreichen. RDz bietet den Vorteil, dass z.B. für die Kompilierung einer Windows Anwendung der IBM Enterprise Cobol oder Enterprise PL/I Compiler verwendet wird. Eclipse stellt hier Compiler zur Verfügung, die nicht vollständig kompatibel mit den IBM Enterprise Compilern sind.

Mit RDz ist es möglich, sich auf ein z/OS System einzuloggen und mit den auf dem z/OS System liegenden Daten zu arbeiten, als ob es Dateien der Workstation wären. Dies wird durch das sog. z/OS File System Mapping ermöglicht, bei dem auf der Workstation zusätzlich zu einzelnen Dateien auch spezielle Member dieser z/OS Datasets gemappt werden können. Das gestattet es mit Datasets zu arbeiten, die Member unterschiedlichen Typs beinhalten. Die Daten werden von der Workstation gemäß ihrer Mappingkriterien behandelt. Von der Workstation aus können auch neue Datasets und Member angelegt und bearbeitet werden.

Der bei der Bearbeitung der Dateien verwendete Editor hat volle ISPF (Interactive System Productivity Facility) Funktionalität und ist durch eine Vielzahl weiterer Module auf dem Stand aktueller Entwicklungsumgebungen. Hervorzuheben sind hierbei ein Content Assist für COBOL und PL/I, der bei Syntaxvervollständigung automatisch alle Ressourcen integriert, auf die ein Entwickler Zugriff hat. Weitere Funktionen sind lokale und remote Syntaxprüfung sowie die in Eclipse integrierten Werkzeuge Compare With... und Replace with Local History.

Ein weiterer Teil von RDz ist die Interaktion mit dem Job Entry System (JES), bei dem Jobs an den Großrechner übermittelt, überwacht und deren Ergebnisse betrachtet werden können. Die dabei verwendeten Job Control Language-Files (JCL-Files) für compile, link-edit und run können automatisiert aus dem vorhandenen Quellcode generiert und an das entsprechende z/OS System zur Ausführung gesandt werden.

Generell sind alle typischen Editier-, Kompilier- und Debugfunktionen lokal und auf dem remote z/OS System von der Workstation aus verfügbar.

Die meisten Funktionen können auch im Offline-Modus verwendet werden. Hier ist als Voraussetzung nötig, entsprechende Projekte und Daten offline verfügbar zu machen, wofür eine Routine in RDz bereit steht.

Um Änderungen lokal und offline testen zu können, ist für die Testumgebung der Workstation ein CICS Transaktionsserver in RDz integriert, der CICS-Anweisungen lokal übersetzt und dadurch ein Testen ermöglicht.

Ein weiterer Teil von WebSphere Developer for System z ist die Erstellung und Bearbeitung des IBM Basic Mapping Supports (BMS). Der dabei verwendete Editor ermöglicht die Bearbeitung der BMS Maps über ein Drag & Drop-Verfahren. Dargestellt und bearbeitet werden können die BMS Maps in einem Design View, bei dem direkt in eine angezeigte BMS Map Teile eingefügt werden können. Es besteht die Möglichkeit, neue Map Sets zu erstellen oder Bestehende zu importieren. Alle Arbeiten können lokal oder remote ausgeführt, sowie fertige Maps exportiert werden.

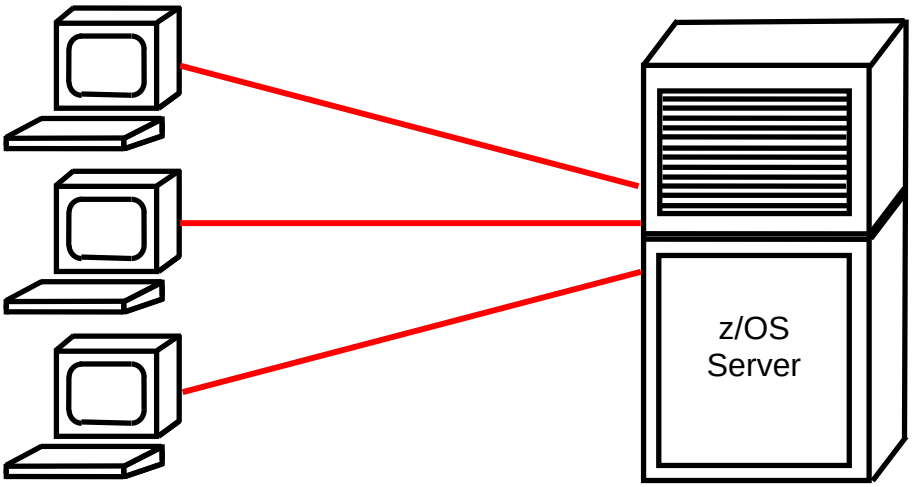
Mit RDz können COBOL und PL/1 Stored Procedures unter z/OS erstellt, getestet und im Bedarfsfall auf der Workstation debugged werden. Für die automatisierte Generierung der SQL Definitionen und der COBOL und PL/1 Stored Procedure Programme steht ein Wizard als Teil von RDz zur Verfügung.

Selbst-Test

- **Ist RDz Open Source ?**
- **Kann man mit RDz auch CICS Anwendungen entwickeln ?**
- **Kann man mit RDz auch Windows Anwendungen entwickeln ?**

3. Einloggen in den remote Windows XP-Server

3.1 Benutzung des Remote RDz Servers

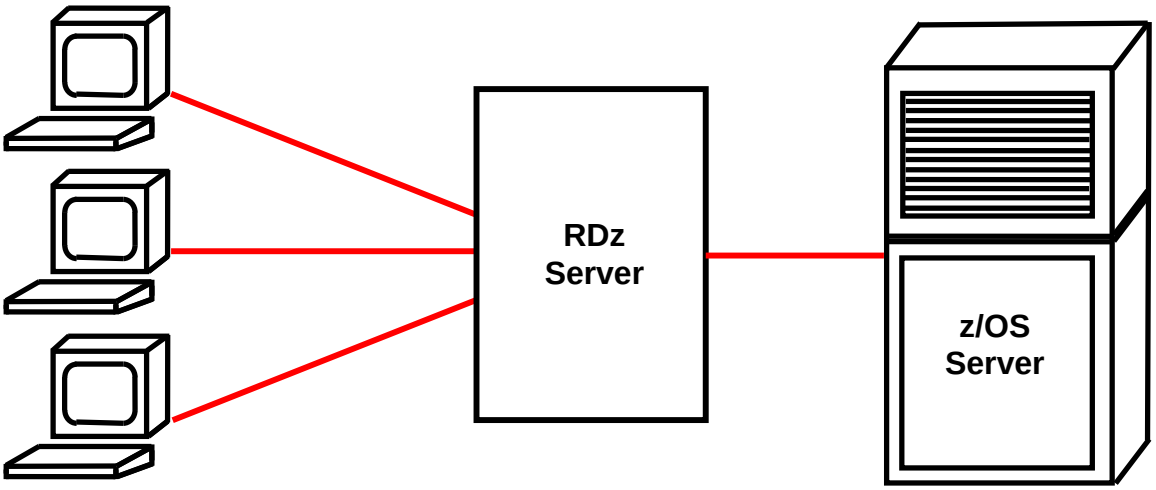


PC's mit RDz installiert

Abbildung 3.1

Abb. 1.1 zeigt eine Standard RDz Konfiguration. Auf allen teilnehmenden Arbeitsplatzrechnern ist Eclipse mit dem RDz Plugin installiert.

Eclipse belegt auf dem Plattenspeicher Ihres Arbeitsplatzrechners wenige 100 MByte Speicherplatz, Das RDz Plugin benötigt weitere 8 GByte.



PC's mit Remote Desktop

Abbildung 3.2

Um Ihnen die Arbeit der Installation zu ersparen verwenden wir die in Abb. 3.2 gezeigte Konfiguration. Jeder Arbeitsplatzrechner enthält lediglich die normale Windows Remote Desktop-Verbindung. Mit dieser erstellen wir eine Remotedesktop-Verbindung zu dem RDz Server des Institutes für Informatik. RDz läuft für alle Benutzer auf diesem Server.

Der RDz Server ist natürlich ein Performance Bottleneck. Für unsere praktischen Übungen hat er sich bisher aber als ausreichend bewährt.

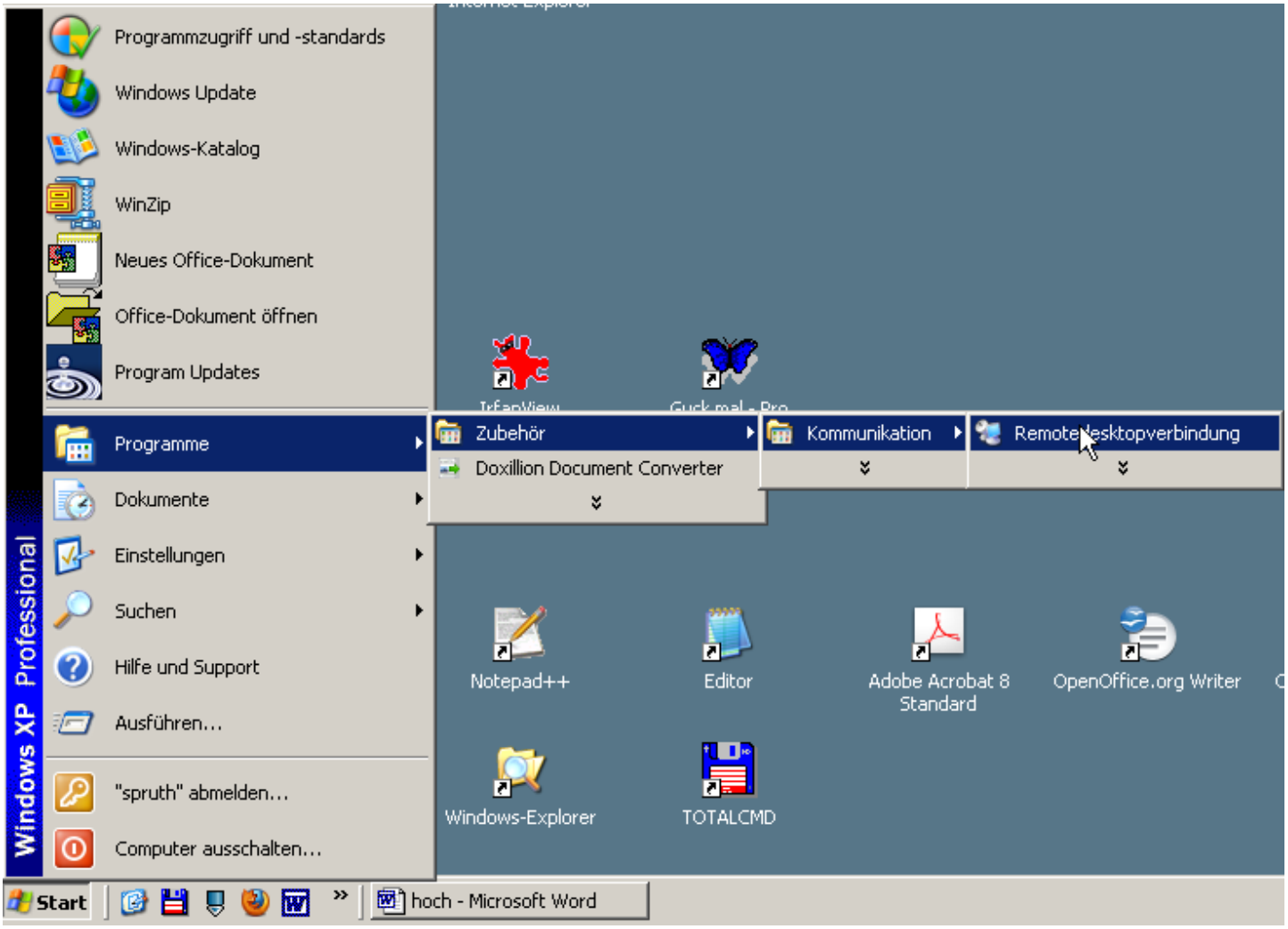


Abbildung 3.3

Start → Alle Programme → Zubehör → Kommunikation → Remotedesktopverbindung

Abbildung 3.4

"Verbinden" anklicken

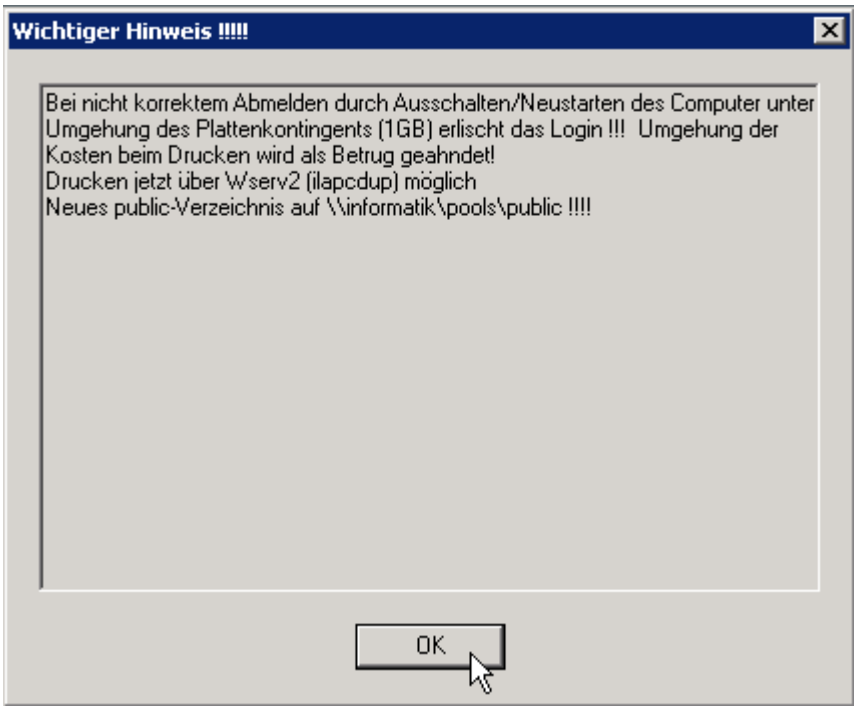


Abbildung 3.5
"OK" anklicken

Abbildung 3.6

Benutzername "IBMP<Nr>" und Passwort eingeben. Die von uns vergebenen Benutzernahmen haben typischerweise das Format "IBMP<Nr>", wobei <Nr> eine zweistellige Ziffer ist.

Dieser Benutzername ist ein Login auf einem entfernten Windows RDz Server.
Das Passwort lässt sich nicht ändern.

Nach ca. ½ Minute erscheint die Windows-Oberfläche der entfernten Maschine.

Selbst-Test

- **Was ist der Vor- und/oder Nachteil, einen RDz Server zu benutzen ?**
- **Würde man in praktischen Entwicklungsprojekten auch einen RDz Server benutzen ?**

3.2 RDz starten

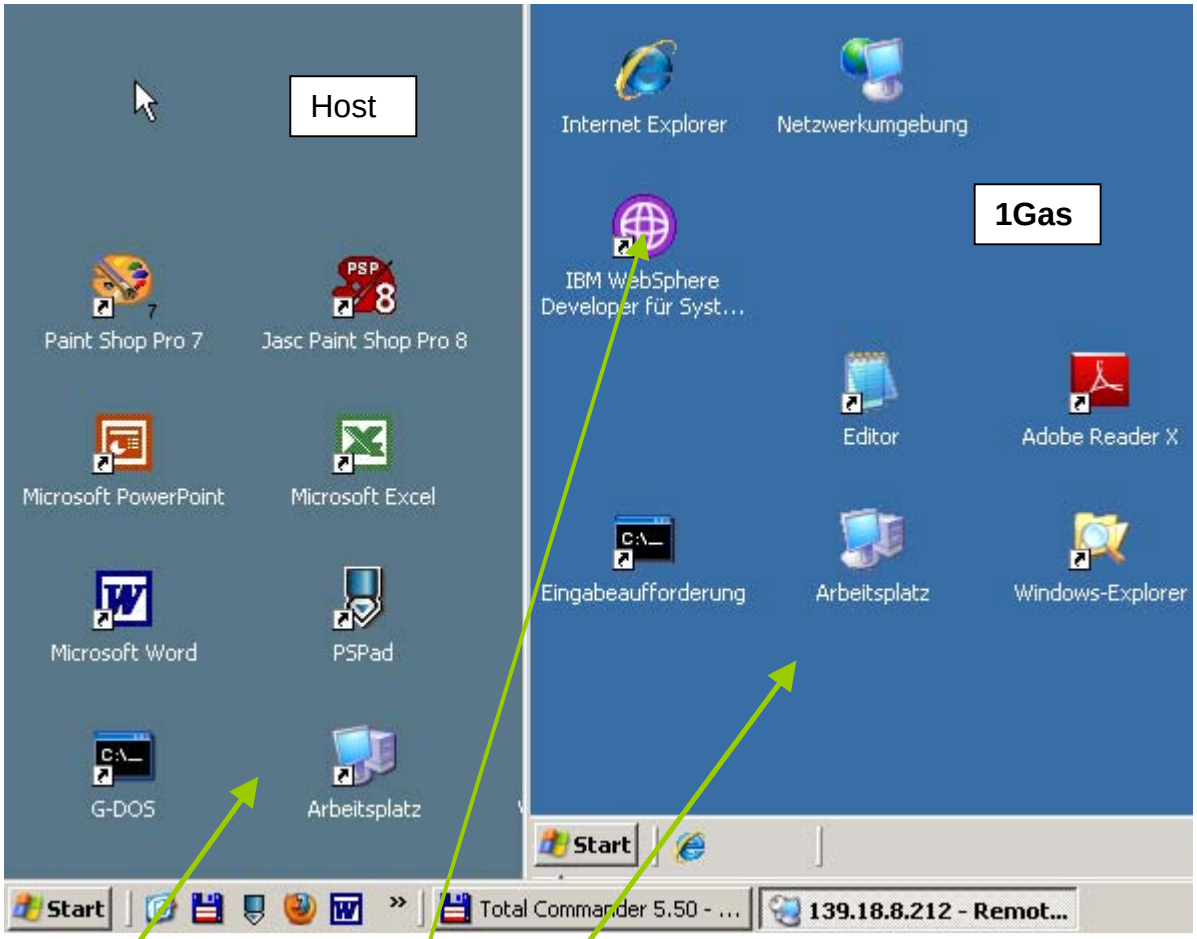


Abbildung 3.7

Es öffnet sich ein Prozess in einem getrennten Fenster, welches den entfernten Windows XP Server 139.18.8.212 repräsentiert, in Zukunft als „Gast“ bezeichnet. Dieses Fenster ist auf dem Desktop des PCs abgebildet, auf dem Sie die Remote Desktopverbindung aufgebaut haben (in Zukunft als Host bezeichnet). Durch anklicken mit der Maus auf dem Hostfenster oder dem Gastfenster können wir beide Fenster bedienen.

2k auf das „WebSphere Developer for System z“ Icon (RDz). Wenn es sich dort noch nicht befindet, ...

Start

→ Alle Programme → IBM Rational → IBM Rational Application Developer V6.0 → Rational Application Developer

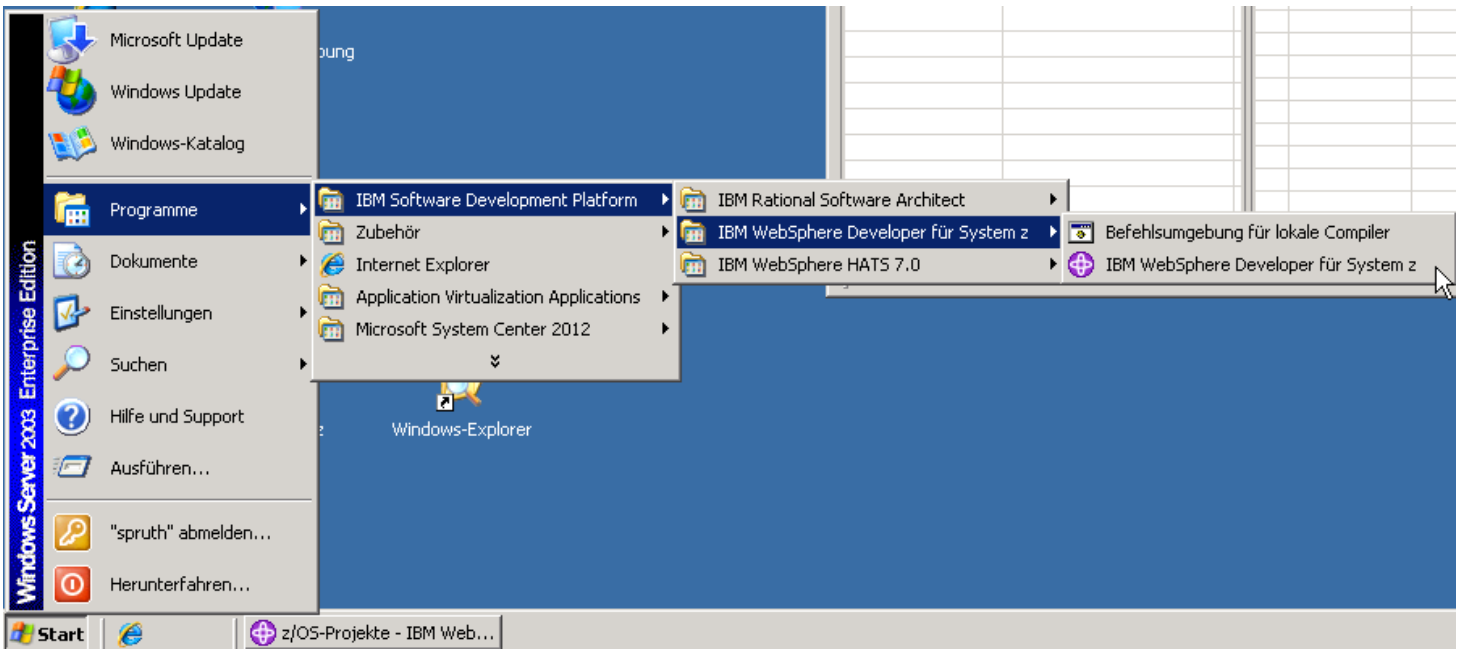


Abbildung 3.8

Der Ladevorgang des auf Eclipse basierenden Tools dauert ca. 1 Minute.

Anschließend soll ein Defaultwert für den Arbeitsbereich festgelegt werden:

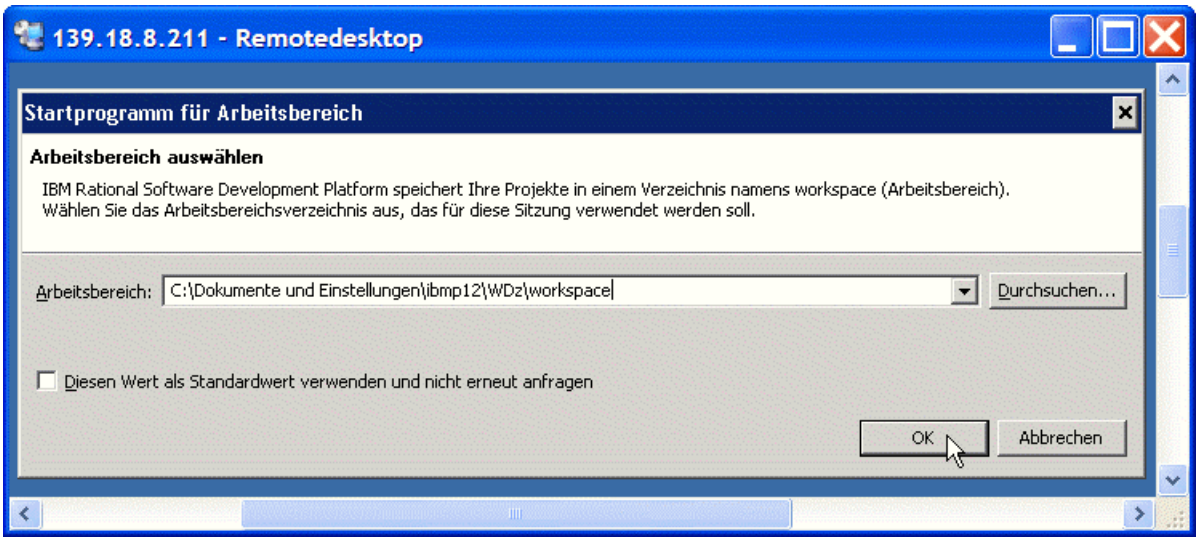


Abbildung 3.9

Es wird das Verzeichnis
C:\Dokumente und Einstellungen\ibmp12\WDz\ibmp12\WDz\Workspace
vorgeschlagen. Da auf dem remote Windows Server für jeden Benutzer eine eigene getrennte Festplatte mit der Bezeichnung z:\ eingerichtet wird, schlagen wir vor, dass Sie statt „C:\Dokumente und Einstellungen\ibmp12\WDz\ibmp12\WDz\Workspace“ den Wert „Z:\Workspace01“ benutzen.

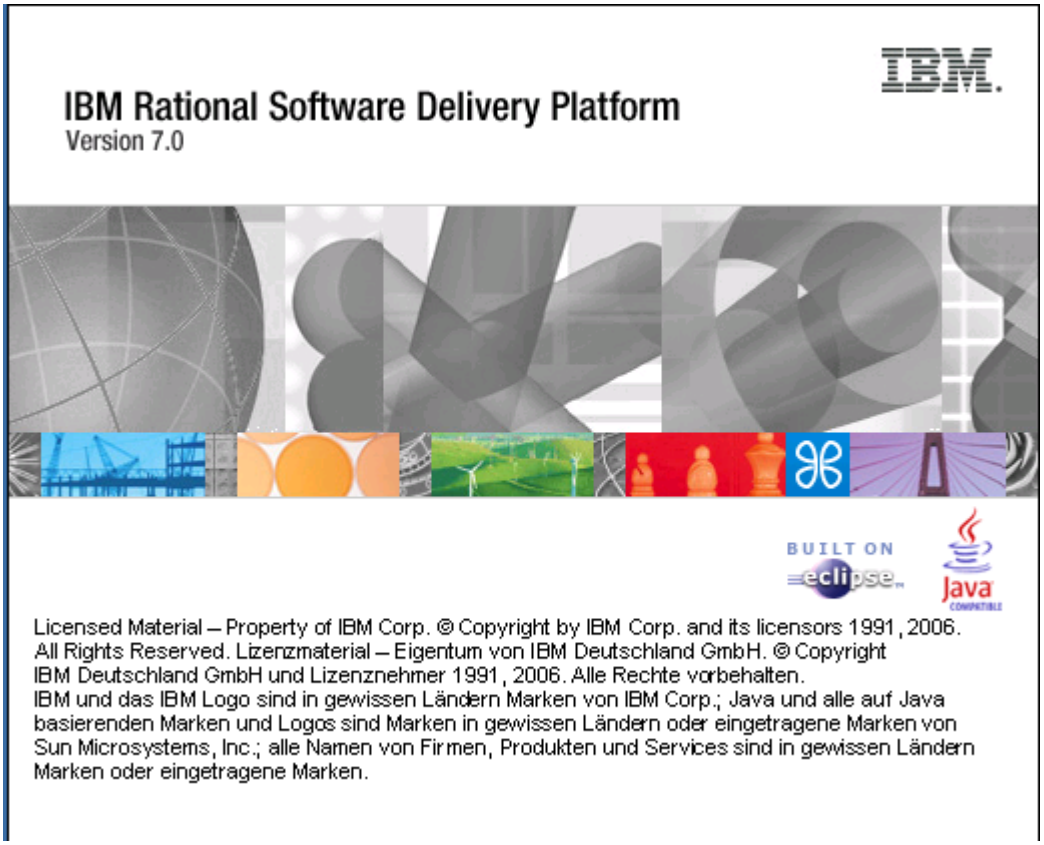


Abbildung 3.10

Der Welcome Screen erscheint. Bitte warten, dauert ca. 1 Minute. Dann erscheint

Evtl. erscheint ein spezielles Fenster.

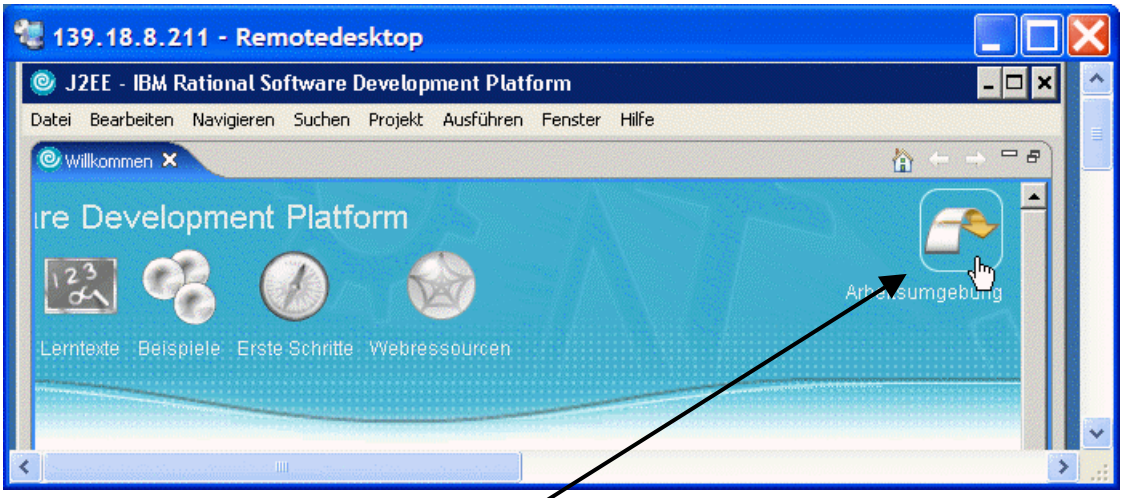


Abbildung 3.11

Beim wiederholten Einloggen erscheint dieses Fenster wahrscheinlich nicht mehr.

Klick auf "Arbeitsumgebung". Es erscheint der RDz Desktop.

3.3 RDz Desktop

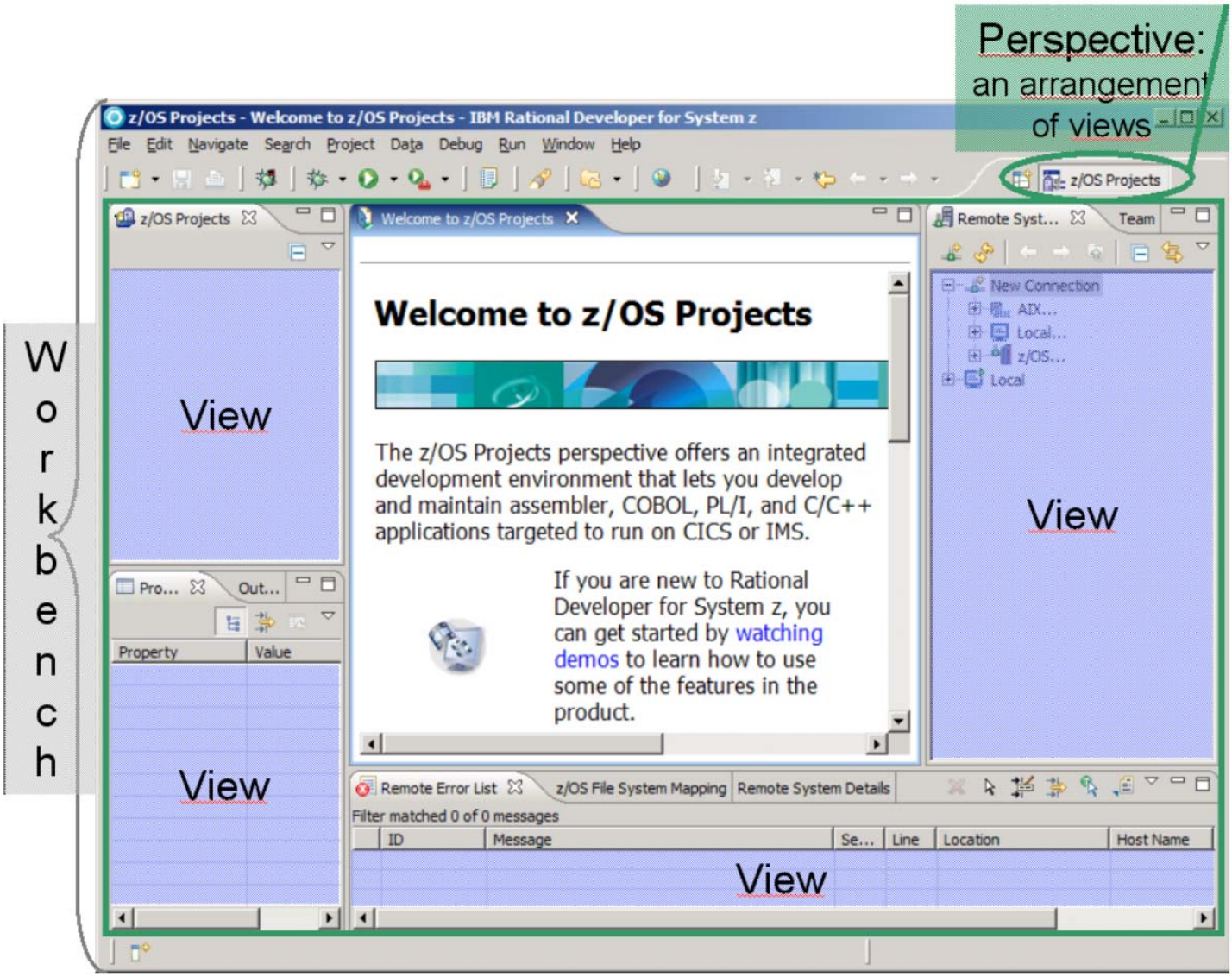


Abbildung 3.12

Eine RDz Oberfläche ist eine spezielle Art einer Eclipse Oberfläche. Programmierer mit Eclipse Vorkenntnissen haben in der Regel nur geringe Anlaufschwierigkeiten, sich in der RDz Oberfläche zurechtzufinden.

Eine Eclipse Oberfläche wird als Workbench bezeichnet. Sie besteht aus einer Reihe von Fenstern, von Eclipse als „Views“ bezeichnet. Für unterschiedliche Entwicklungsaufgaben existieren unterschiedliche Anordnungen der Views. Diese unterschiedlichen Anordnungen werden von Eclipse als „Perspectives“ bezeichnet. Es existieren spezielle Perspectives beispielsweise für XML, Java oder HTML Entwicklungen. Es ist einfach, in einer Eclipse Umgebung zwischen unterschiedlichen Perspectives hin und her zu schalten.

Für uns von besonderer Wichtigkeit ist die „z/OS Projects“. Perspective

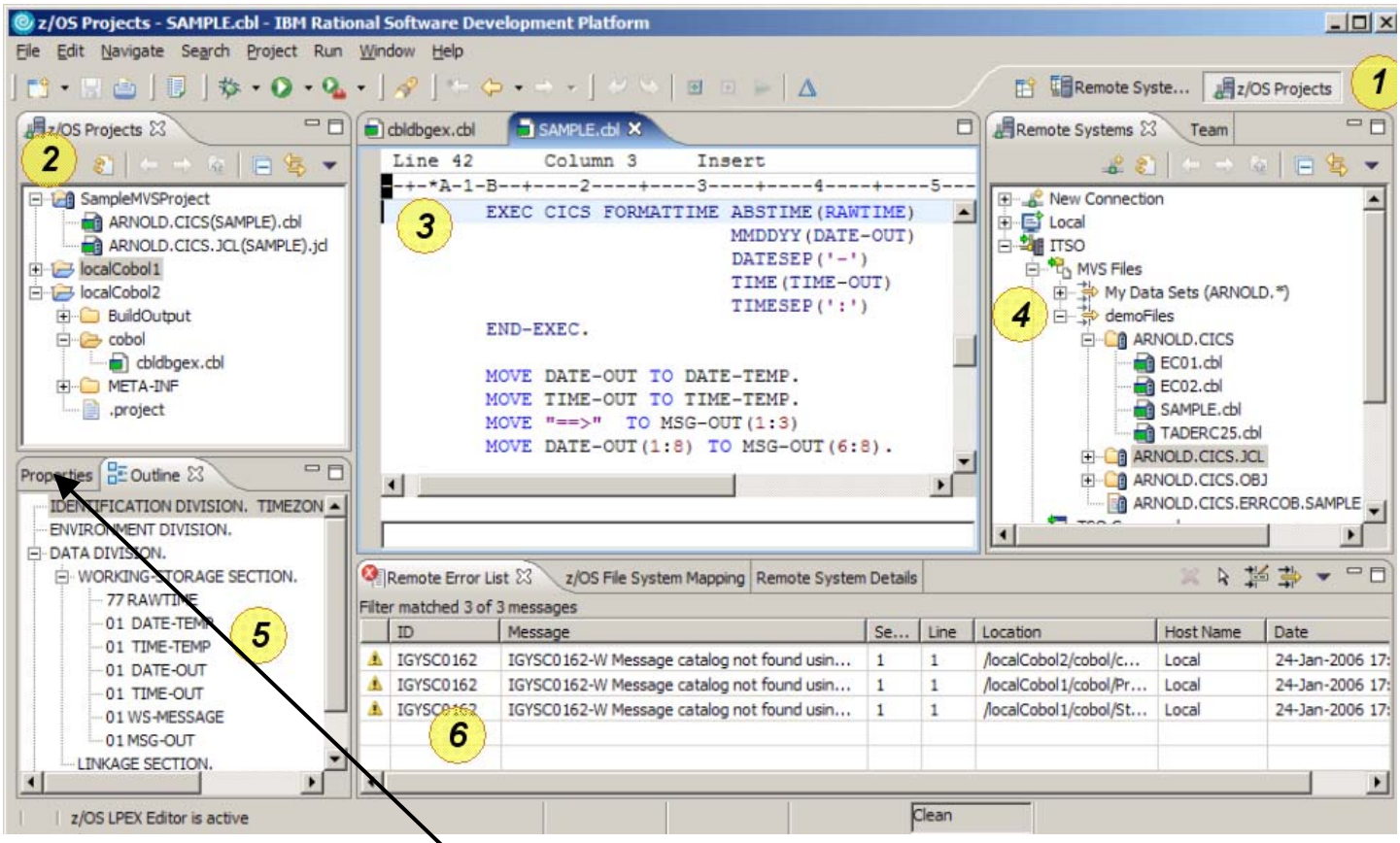


Abbildung 3.13

Dargestellt ist die exemplarische z/OS Projects Perspective. Sie besteht aus 6 Views:

1. **Short-cut Icons zu anderen Perspektiven,**
2. **z/OS Projects view (local view),**
3. **Editor,**
4. **Remote System view,**
5. **Outline view,**
6. **Tasks view**

Weil der Bildschirm nie groß genug ist, werden einige der Views wie Registrierkarten von anderen Views überlagert und verdeckt. Mit Hilfe von Tabs kann man einen verdeckten View sichtbar machen.

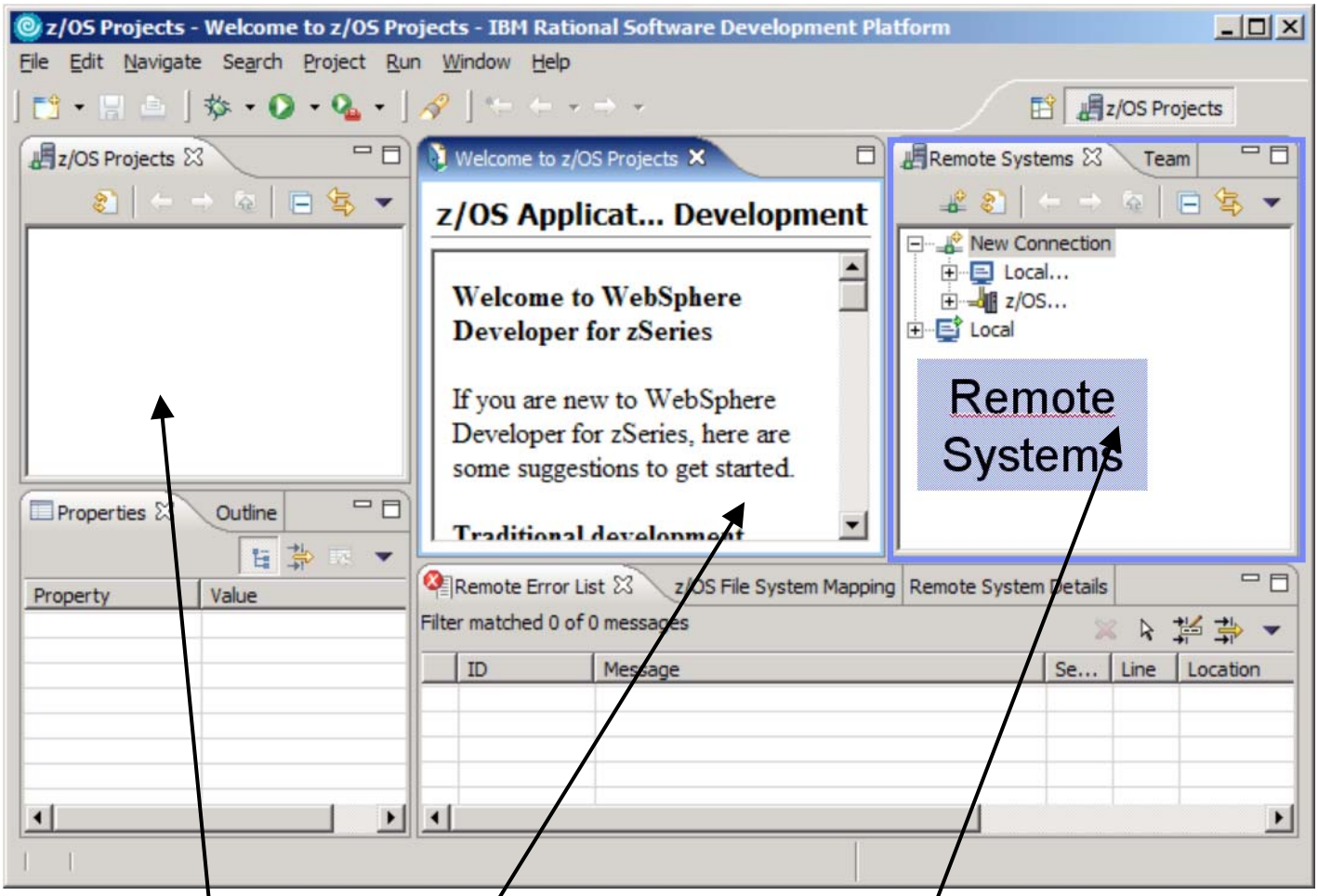


Abbildung 3.14

Für die Entwicklung von Mainframe Anwendungen ist der Remote Systems View (in der Standard „z/OS Projects“ Perspektive immer rechts oben angeordnet) von besonderem Interesse. Der Remote Systems Viel enthält typischerweise etwas ähnliches wie das ISPF DLIST Panel, und ermölicht einen Zugriff auf alle Data Sets, welche Sie unter Ihrer Prakxxx User ID auf unserem Mainframe Rechner angelegt haben.

Das z/OS Projects View enthält normalerweise Cobol Files , die Sie unter z/OS gerade bearbeiten. Diese Files sind unter RDz als Windows Files gespeichert. Sie können derartige Files mit der Maus und Drag und Drop zwischen den z/OS Projects View und dem Remote Systems View hin- und herziehen und auf diese Art kopieren.

Das mittlere Fenster enthält normalerweise einen Editor, mit dem Sie Ihren Cobol Quelltext (oder Ihr JCL Script) bearbeiten.

Insgesamt besteht ein überraschend großes Maß an Flexibilität.

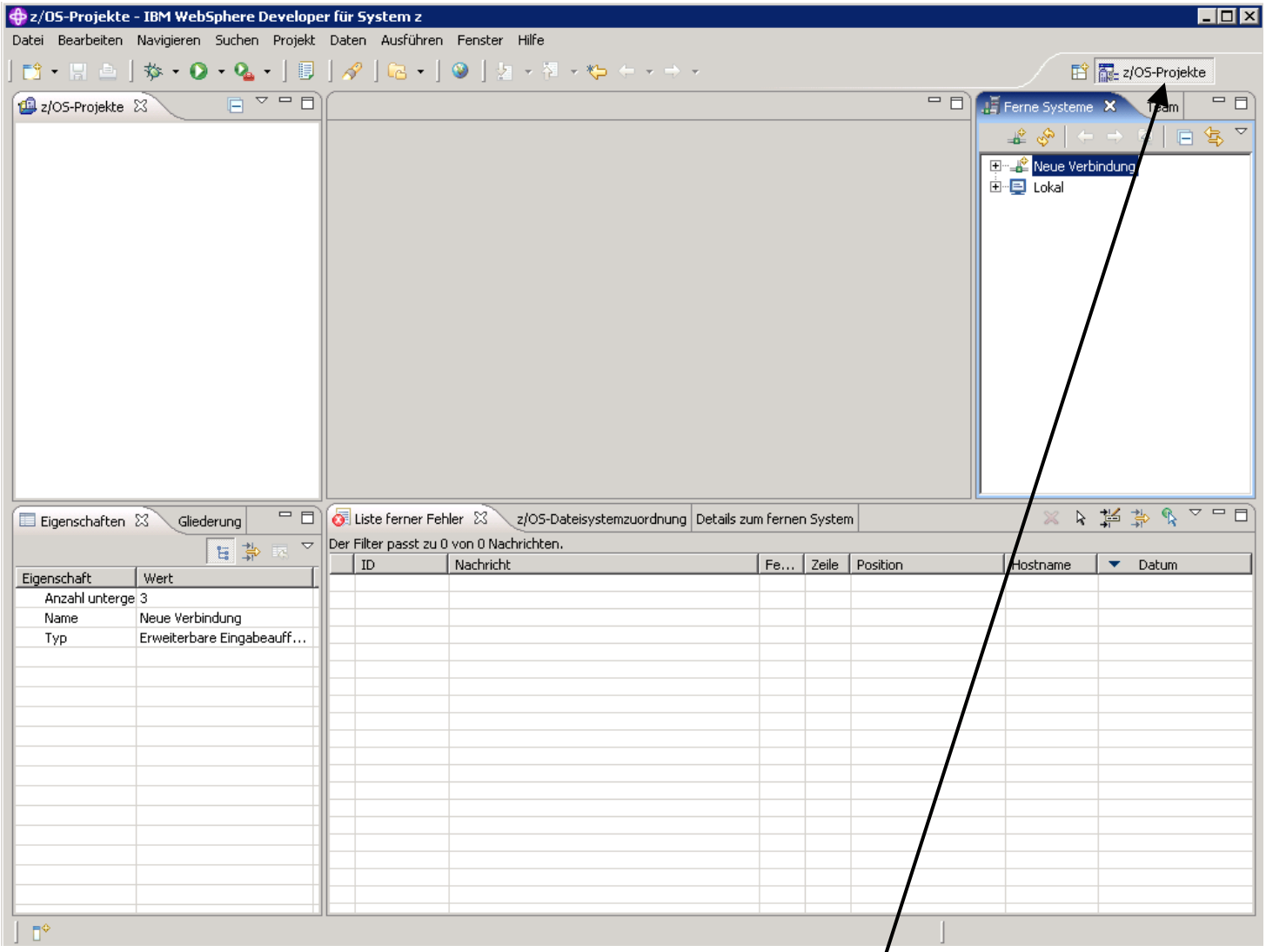


Abbildung 3.15

Normalerweise ist die z/OS Projects Perspective beim Starten von RDz bereits enabled. Falls nicht, müssen Sie das Enable nachholen. Hierzu .....

Selbst-Test

- **Was ist der Unterschied zwischen dem z/OS Projects View und dem Remote Systems View ?**

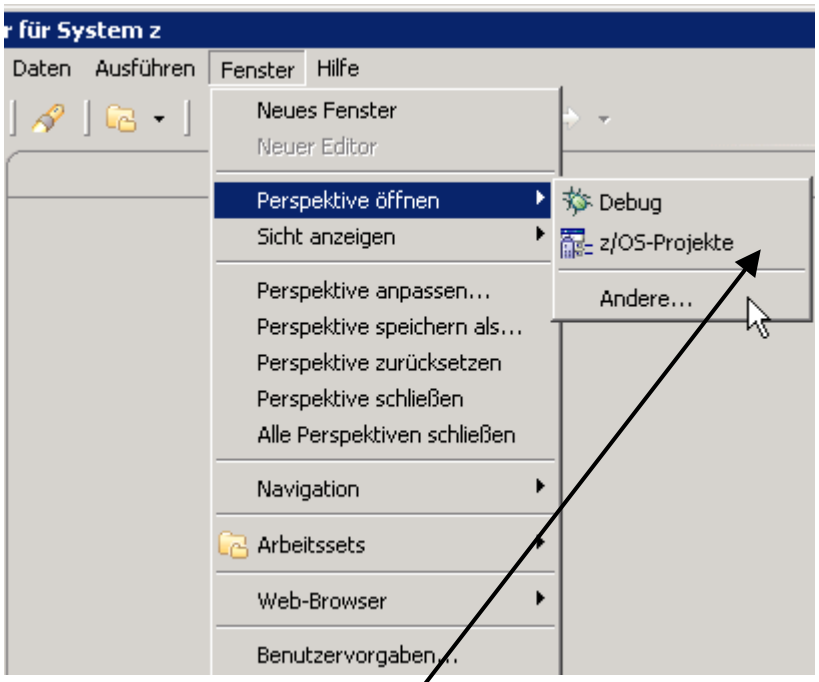


Abbildung 3.16

Fenster → Perspektive öffnen → z/OS Projekte, 1k .

4. Perspektiven und Views

4.1 Views

Workbench Tools sind in kleinen Bereichen angeordnet, die sich wie Registrierkarten verhalten und als“Views” bezeichnet werden. Views sind kleine Fenster, welche Datei oder Projekt Information wiedergeben oder einen Zugriff auf RDz Funktionen ermöglichen.
Die sich nach dem Starten von RDz geöffnete Arbeitsumgebung kombiniert mehrere verschiedenartige Views. Eine Überlagerung mehrerer Views ist möglich; nur der oberste View ist dann zu sehen. Durch 1k auf den Tab einer Registrierkarte (View) kann diese sichtbar gemacht werden.

Die Kombination aller angezeigten Views (Art und Position der Views) heißt Perspektive.

Es können sogar mehrere Perspektiven gleichzeitig geöffnet sein. In einem geöffneten RDz-Fenster wird jedoch genau eine der geöffneten Perspektiven angezeigt.

Die z/OS Projects Perspective besteht aus einer großen Anzahl von Views. Von diesen müssen Sie nur eine Untermenge kennen, um mit RDz produktiv arbeiten zu können. Sies sind spezifisch:

- **Remote Systems View**
- **z/OS Projects View**
- **COBOL Source Editor**
- **Properties View**
- **Outline View**
- **Remote Error List View**
- **Perform Hierarchy View**

14.1.1 Schließen eines Views

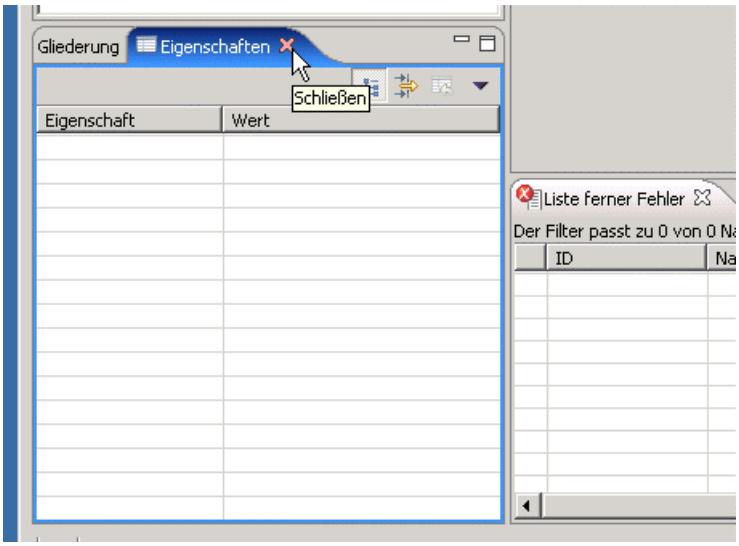


Abbildung 4.1

Eine View kann durch Klick auf das Kreuz der jeweiligen Registrierkarte geschlossen werden:

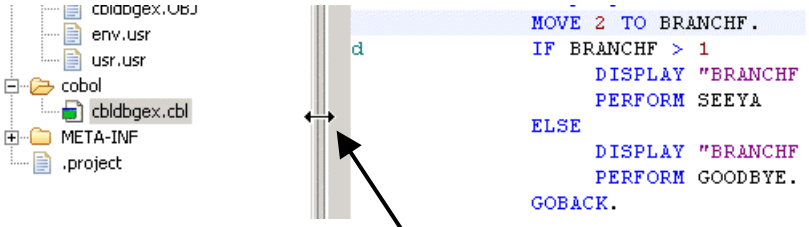


Abbildung 4.2

4.1.2 Verändern der Breite und Höhe von Views

Views kann man auch in Breite und Höhe seinen Bedürfnissen anpassen. Dazu sind einfach die jeweiligen Ränder entsprechend zu ziehen.

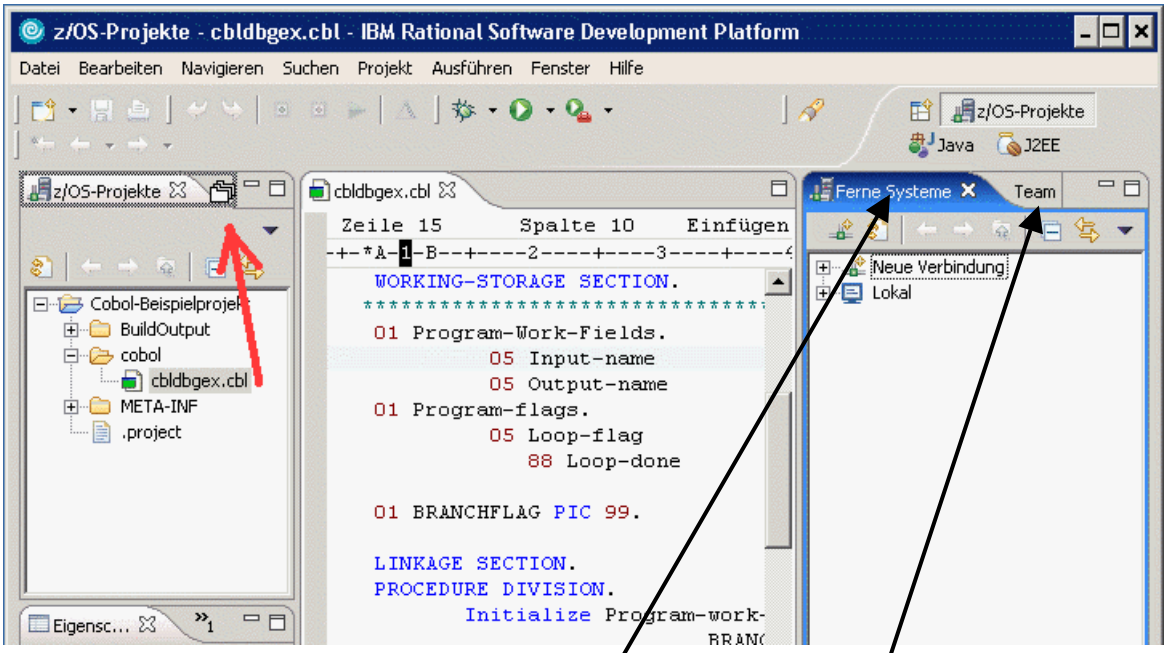


Abbildung 4.3

4.2. Vergrößern des Cobol-Programm-View

Werden die Views "Ferne Systeme" (remote Systems) und "Team" gerade nicht benötigt, können diese per drag and drop auf die linke Seite – unter die "z/OS-Projekte"-View – gezogen werden. So erhält man mehr Platz für das Editieren des Cobol-Programm-Codes.

Die Views können nur einzeln hinübergezogen werden. Um z.B. die View "Ferne Systeme" (remote Systems) hinüberzuziehen, klickt man auf "Ferne Systeme", hält den Klick und zieht. Der Cursor verwandelt sich in einen kurzen dicken schwarzen Pfeil und an einer möglichen Ablagestelle weiter zu . Wird die Maustaste losgelassen, wird der View eingefügt.

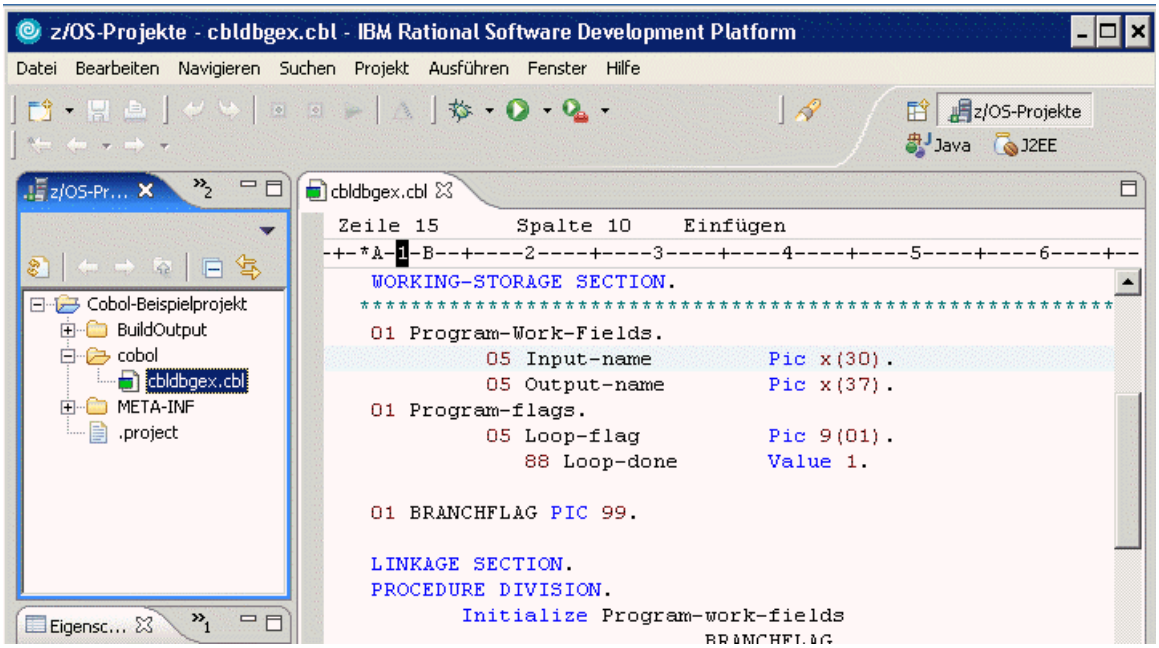


Abbildung 4.4

Dies ist das Ergebnis: Im Gegensatz zu der Abbildung 2.3 ist der Platz für das Editieren des Cobol-Programms in Abb. 2.4 größer geworden.

4.3. Zurücksetzen einer Perspektive in ihren Default-Zustand

Möchte man zum Standard-Aussehen einer Perspektive zurück, also u.a.

- **Manuell geschlossene Views wieder öffnen**
- **Views auf ihre Standardbreiten und -höhen zurücksetzen,**

dann ist das wie folgt möglich:

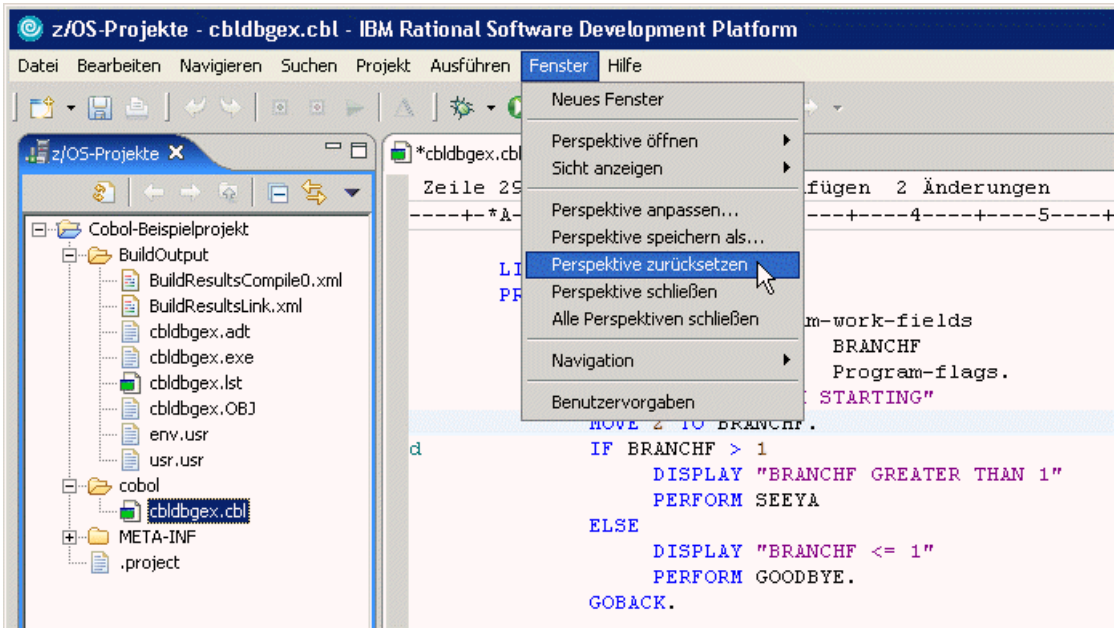


Abbildung 4.5

4.4. Wechseln zwischen verschiedenen geöffneten Perspektiven

Es ist möglich, mehrere Perspektiven zu öffnen. Doch nur eine wird jeweils in einem RDz-Fenster angezeigt.

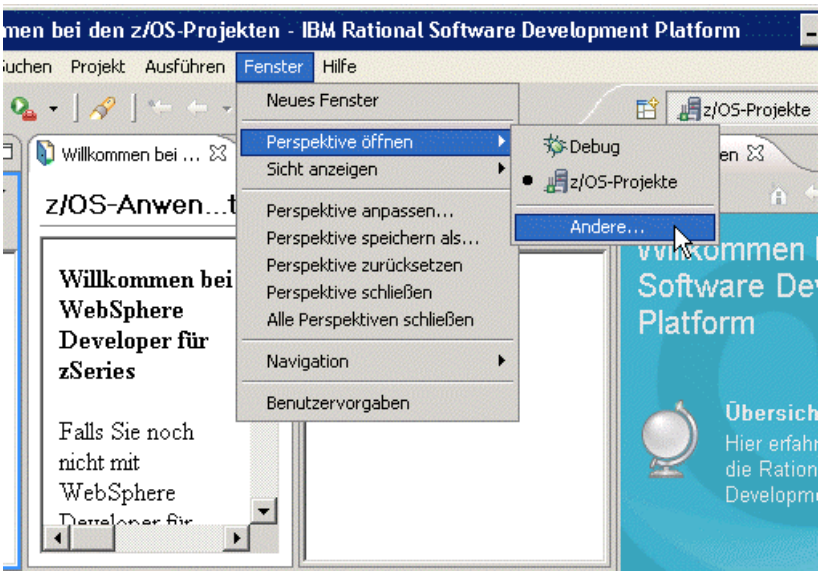


Abbildung 4.6

Neben der Perspektive "z/OS-Projekte" gibt es eine Reihe weiterer Perspektiven.
Benötigt man neben dieser Perspektive z.B. noch die "Java"-Perspektive, so kann diese ebenfalls geöffnet werden:

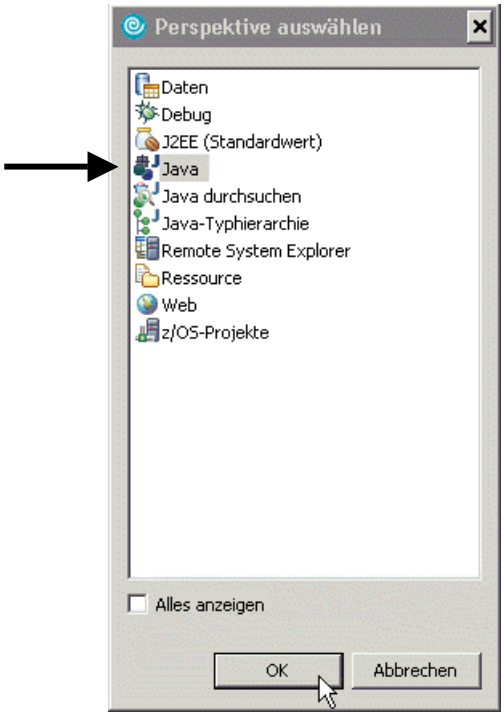


Abbildung 4.7

Klick auf Java

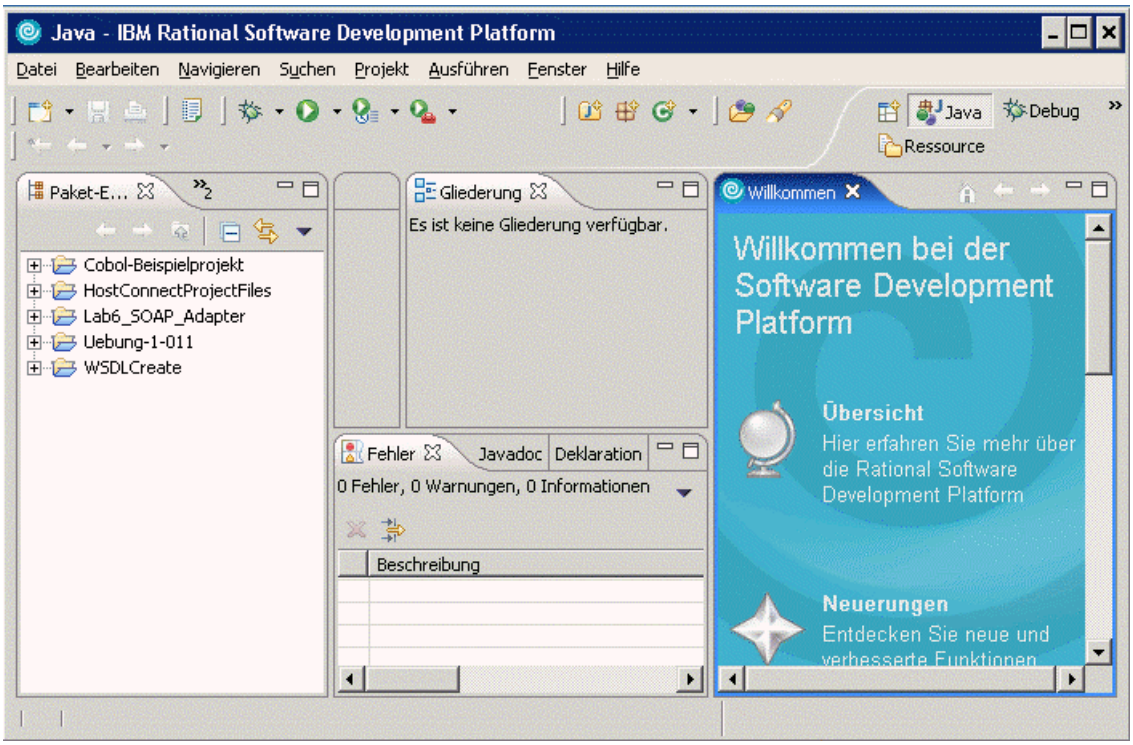


Abbildung 4.8

Nach einigen Sekunden erscheint die neue Perspektive, die "Java"-Perspektive.

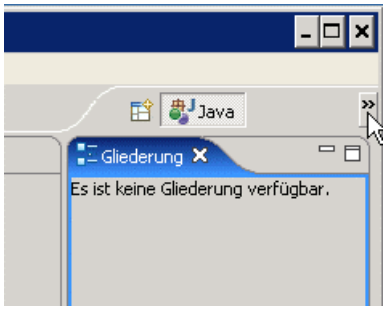


Abbildung 4.9

**Es ist nun möglich, zwischen beiden geöffneten Perspektiven zu wechseln.
Dazu ist, wie Abbildung 2.9 zeigt, auf das entsprechende Feld zu klicken.**

Geöffnete Perspektiven lassen sich auch wieder schließen:

**"Fenster → Perspektive schließen" schließt die gerade angezeigte Perspektive.
"Fenster → Alle Perspektiven schließen" schließt alle zur Zeit geöffneten Perspektiven.**

Selbst-Test

- **Müssen Sie unbedingt die z/OS Perspektive für die Entwicklung von z/OS Anwendungen benutzen ?**

5. RDz beenden und Ausloggen

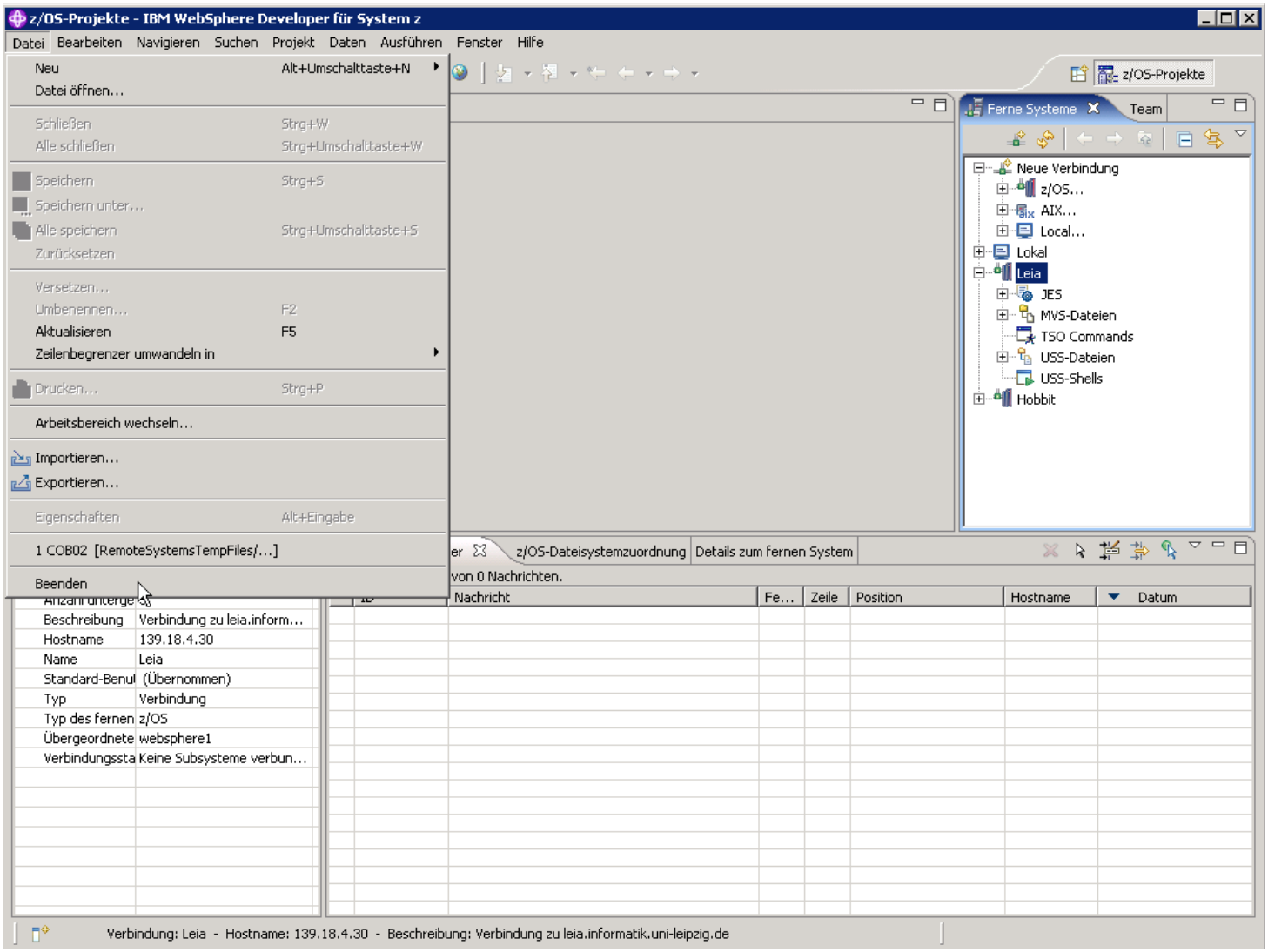
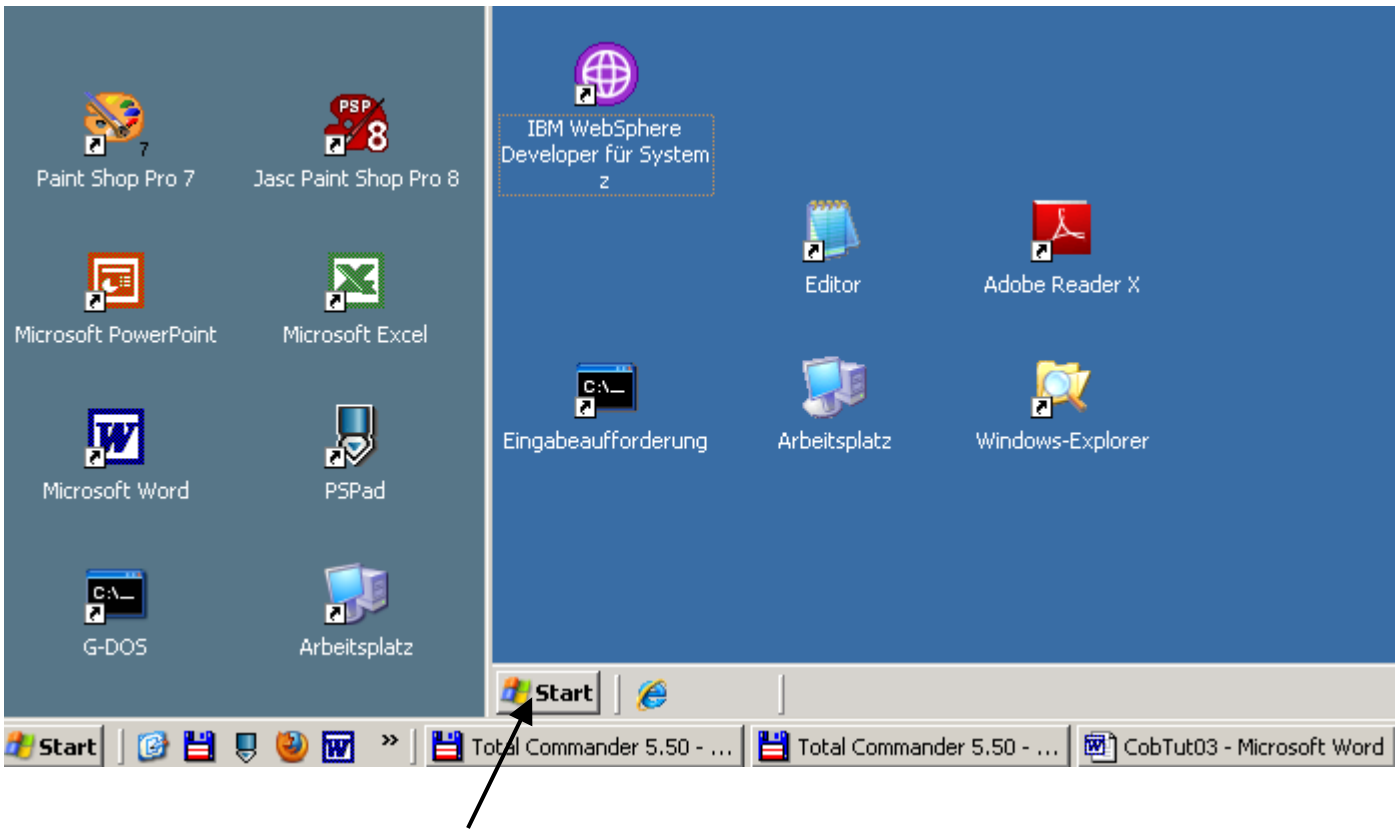


Abbildung 5.1

RDz wird wie folgt beendet: Datei → Beenden



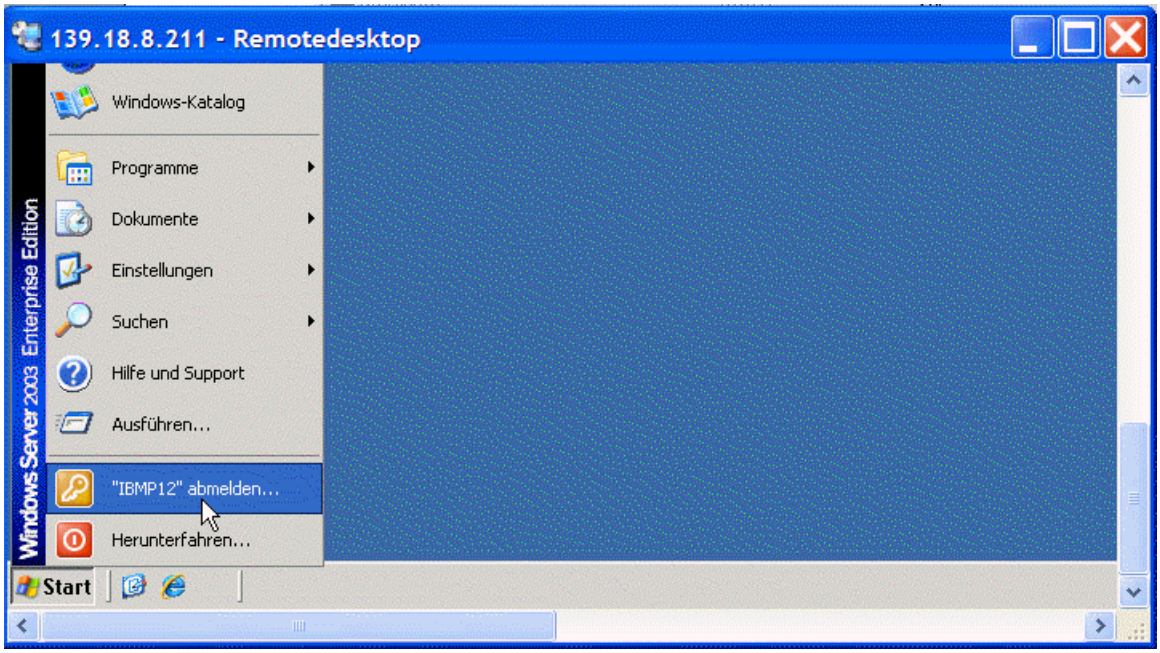


Abbildung 5.2

Nun ist noch ein Abmelden vom virtuellen entfernten Windows-Server erforderlich:

START → "IBMP<xx> abmelden..."
Es ist noch die Frage zu beantworten, ob man sich wirklich abmelden möchte. Hierzu ein Klick auf den Button "Abmelden".

Das Fenster mit der Remote Desktop Verbindung wird geschlossen.

Das war es, Sie haben RDz Tutorial 01 erfolgreich abgeschlossen. In dem nächsten Tutorial werden wir unter Windows ein Cobol Programm entwickeln und austesten.

Ende