

# Tutorial 1

## WDz 6.0 - Grundlagen

Copyright © Institut für Informatik, Universität Leipzig

WDz ist eine auf Eclipse basierende moderne Entwicklungsumgebung, welche unter anderem ein universelles Werkzeug für die Anwendungsentwicklung auf Mainframes darstellt. WDz wird in immer mehr Unternehmen in der Anwendungsentwicklung eingesetzt.

Nahezu alles, was unter Nutzung eines 3270-Emulators auf einem Mainframe möglich ist, ist auch unter WDz möglich; z.B. Datasets anlegen (allocate), Datasets mit 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 WDz.

Voraussetzungen für die Durchführung dieses Tutorials:

- Internetfähiger Windows XP-PC
- PRAK(T)-Login auf Mainframe
- IBMP-Login auf virtuellem XP-Server mit installiertem WDz 6.0

*Aufgabe: Beschäftigen Sie sich mit diesem Tutorial und lösen Sie vollständig und gewissenhaft alle kursiv geschriebenen und eingerahmten Aufgaben.*

Über die kursiv geschriebenen und eingerahmten Pflichtaufgaben hinausgehend wird empfohlen, sich mit den in diesem Tutorial vorgestellten Wdz-Funktionen näher zu beschäftigen, indem einfach selbständig Modifikationen zu den vorgestellten Wdz-Funktionen erdacht und ausprobiert werden.

### 1 Einloggen in entfernten Windows XP-Server

Verwenden der Windows Remote Desktop-Verbindung

Start → Alle Programme → Zubehör → Kommunikation → Remotedesktopverbindung

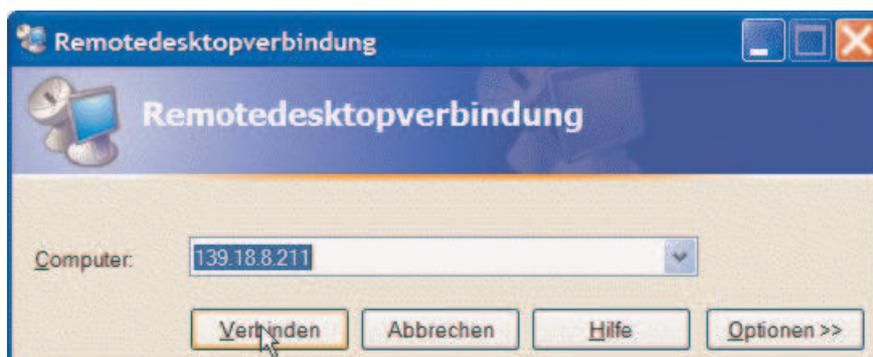


Abbildung 1

"Verbinden" anklicken

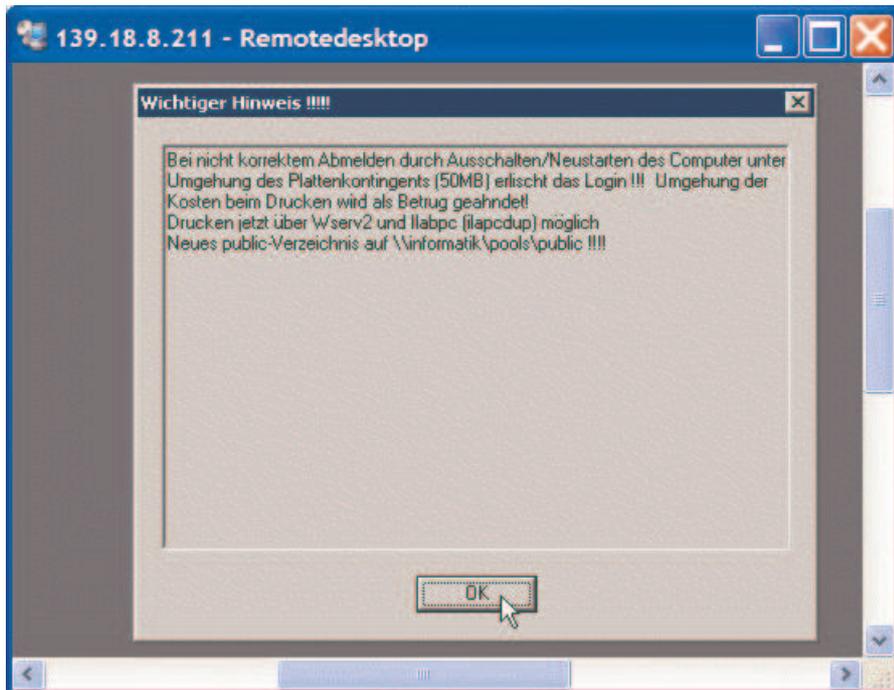


Abbildung 2

"OK" anklicken

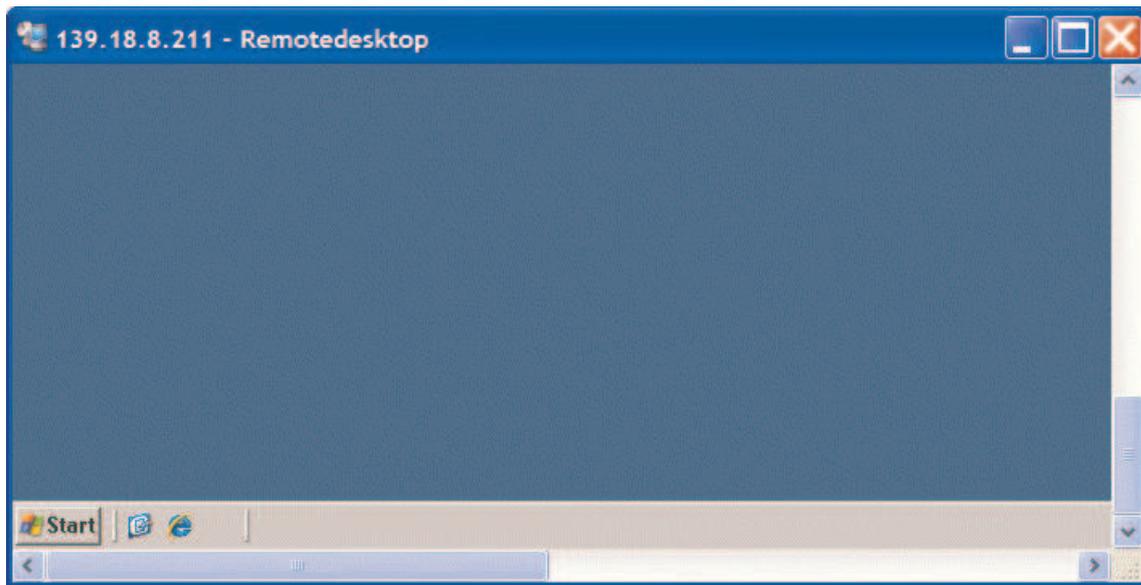
Benutzername "IBMP<Nr>" und Passwort eingeben

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



Abbildung 3

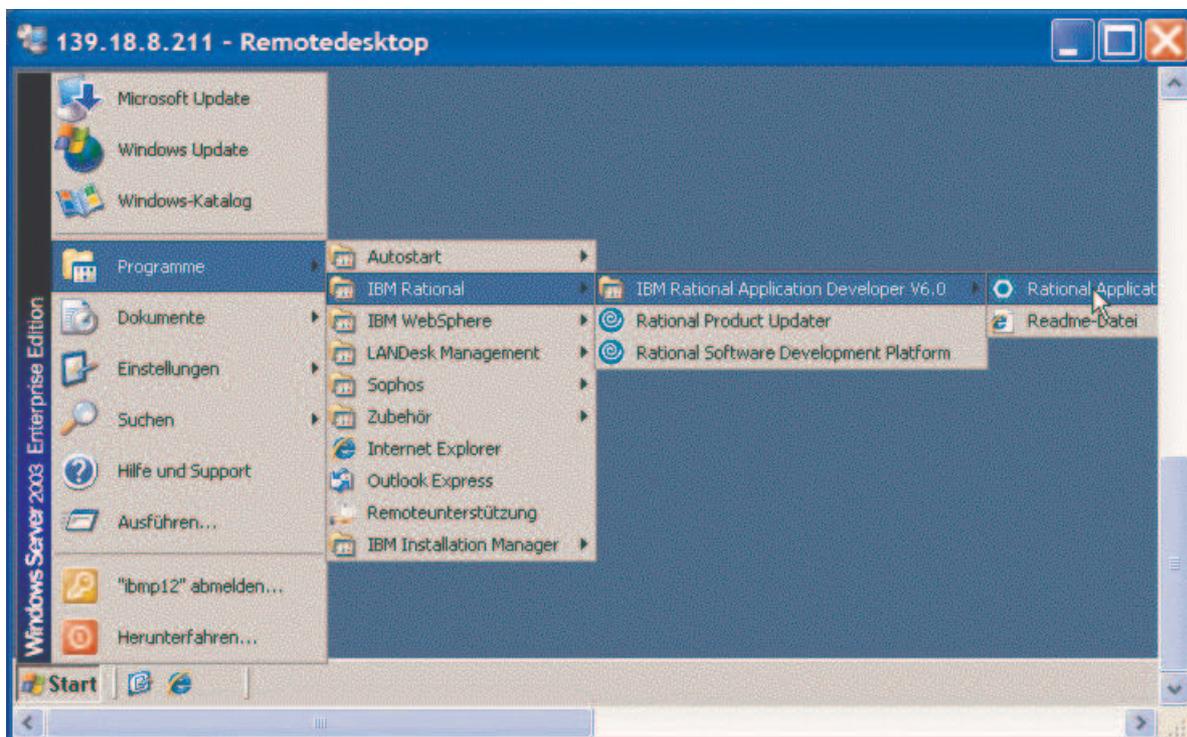
Nach ca. ½ Minute erscheint die Windows-Oberfläche der entfernten Maschine.



**Abbildung 4**

WDz hier starten

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



**Abbildung 5**

Der Ladevorgang des auf Eclipse basierenden Tools dauert ca. 20 Sekunden

Anschließend soll ein Defaultwert für den Arbeitsbereich festgelegt werden:  
"C:\Dokumente und Einstellungen\ibmp<Nr>\WDz\Workspace"  
Im Verzeichnisnamen Ihren persönlichen Login "ibmp<Nr>" verwenden

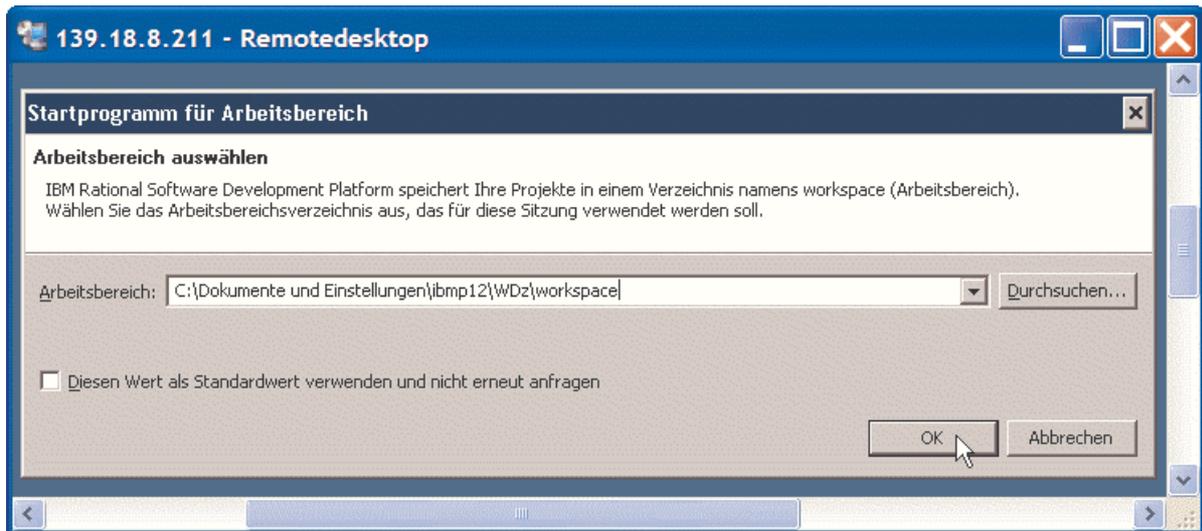


Abbildung 6

Beim erstmaligen Start erscheint

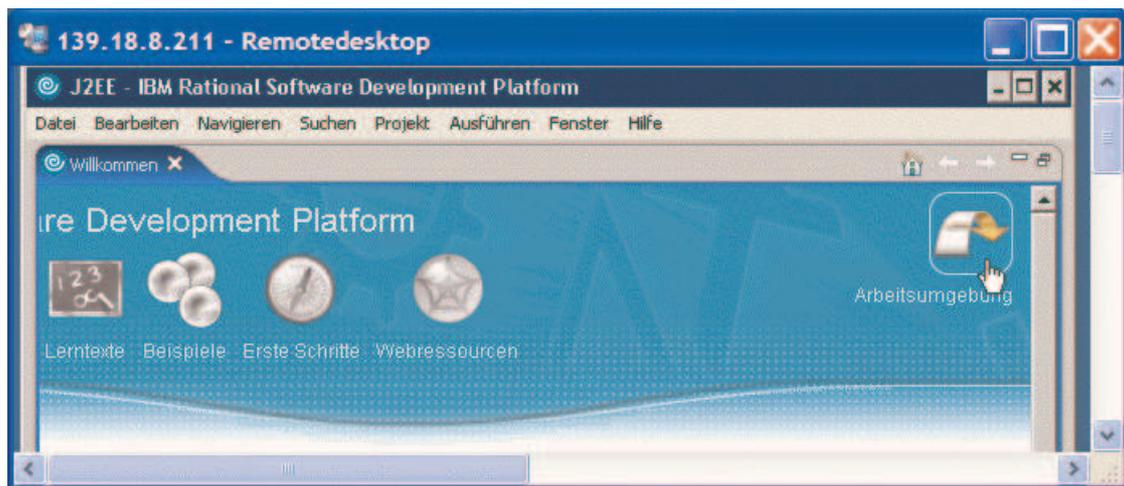
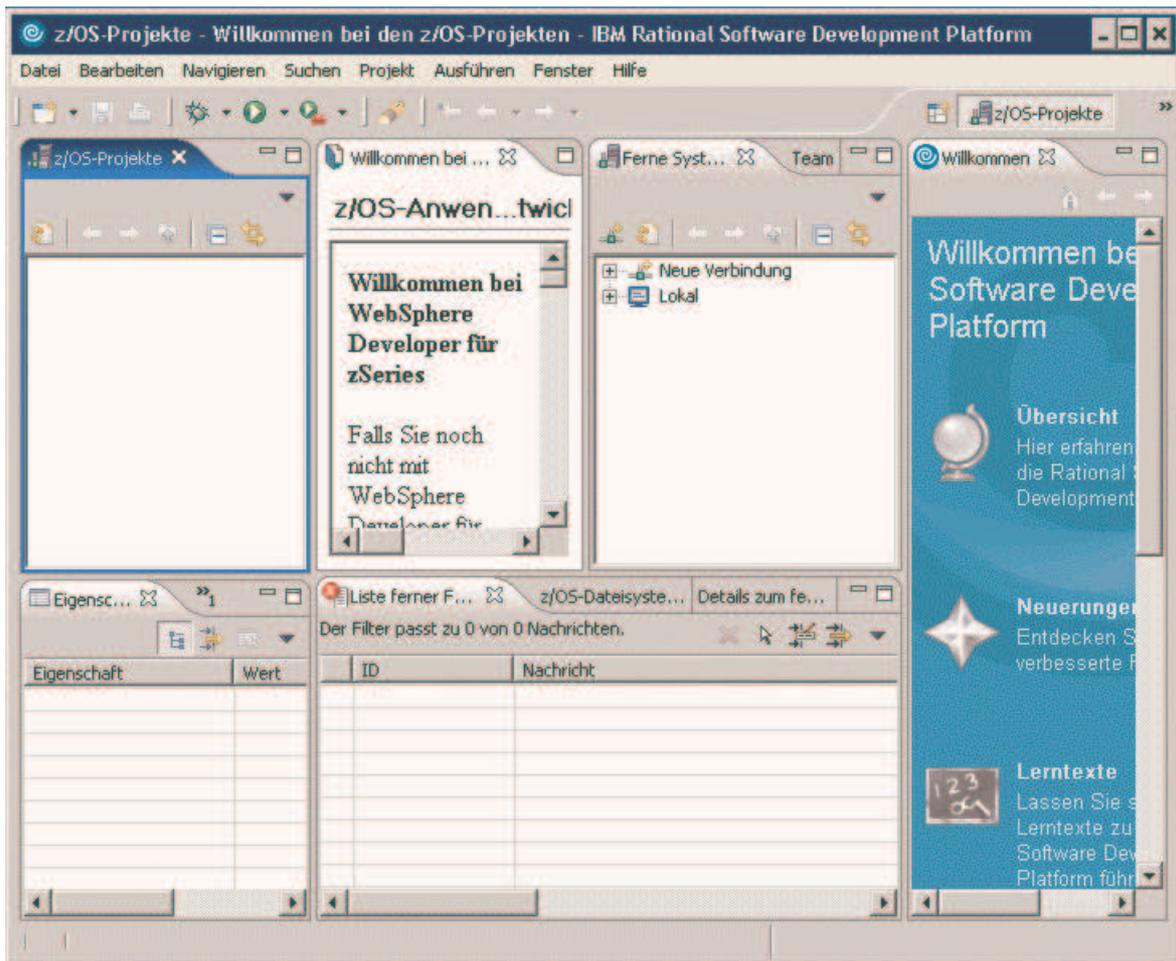


Abbildung 7

Beim wiederholten Einloggen erscheint dieses Fenster eventuell nicht mehr.

Klick auf "Arbeitsumgebung"



**Abbildung 8**

Dies ist die z/OS-Projekte-Perspektive, die sich standardmäßig beim allerersten Einloggen öffnet.

## 2 Perspektiven und Views

Die sich nach dem Starten von WDz geöffnete Arbeitsumgebung kombiniert mehrere verschiedenartige Registerkarten. Diese heißen *Views*.

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

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

## 2.1 Schließen einer View

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

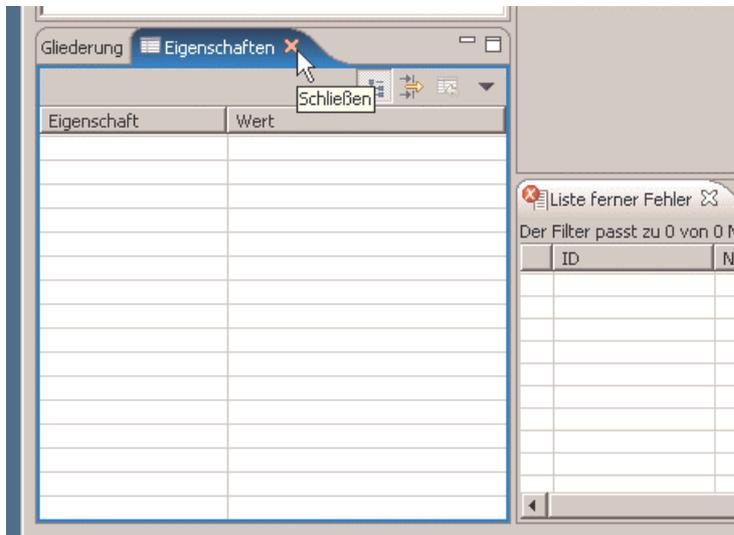


Abbildung 9

## 2.2 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 (siehe nachstehende Abbildung).

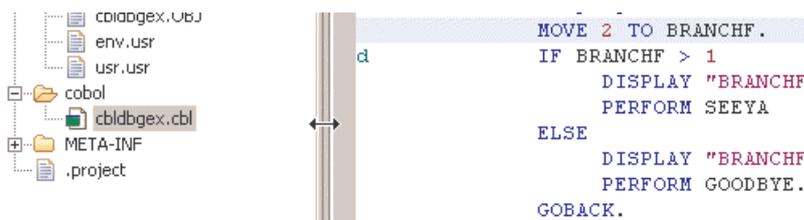


Abbildung 10

## 2.3 Vergrößern der Cobol-Programm-View

Werden die Views "Ferne Systeme" 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.

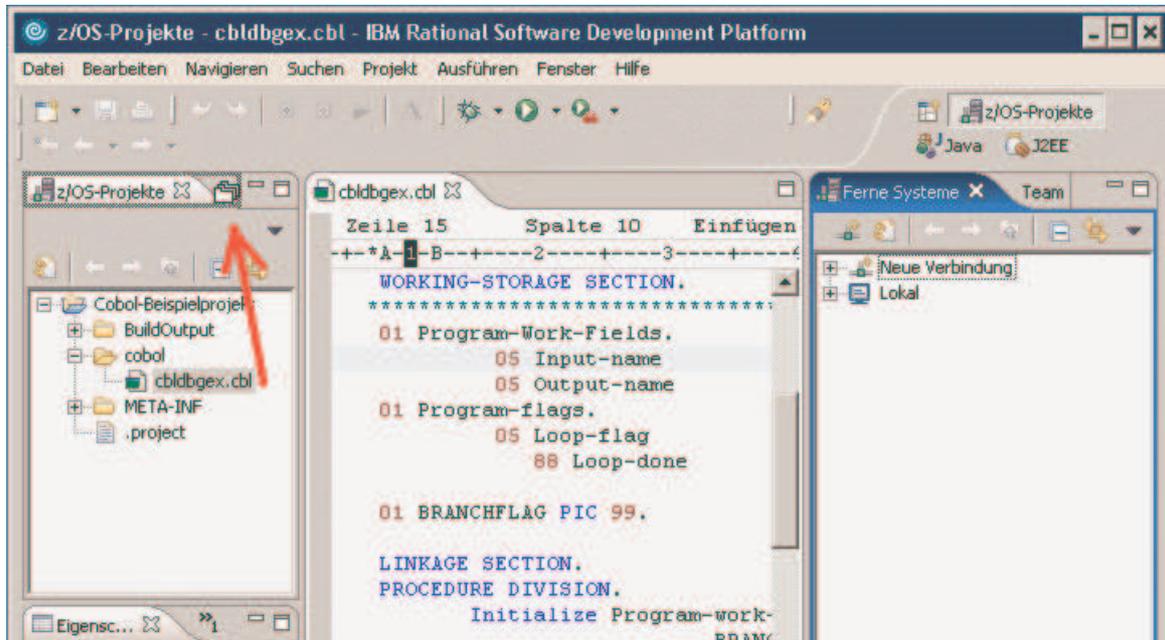


Abbildung 11

Die Views können nur einzeln hinübergezogen werden. Um z.B. die View "Ferne Systeme" 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 die View eingefügt.

Die nachstehende Abbildung demonstriert – im Gegensatz zu der vorstehenden Abbildung – den nun größer gewordenen Platz für das Editieren des Cobol-Programms.

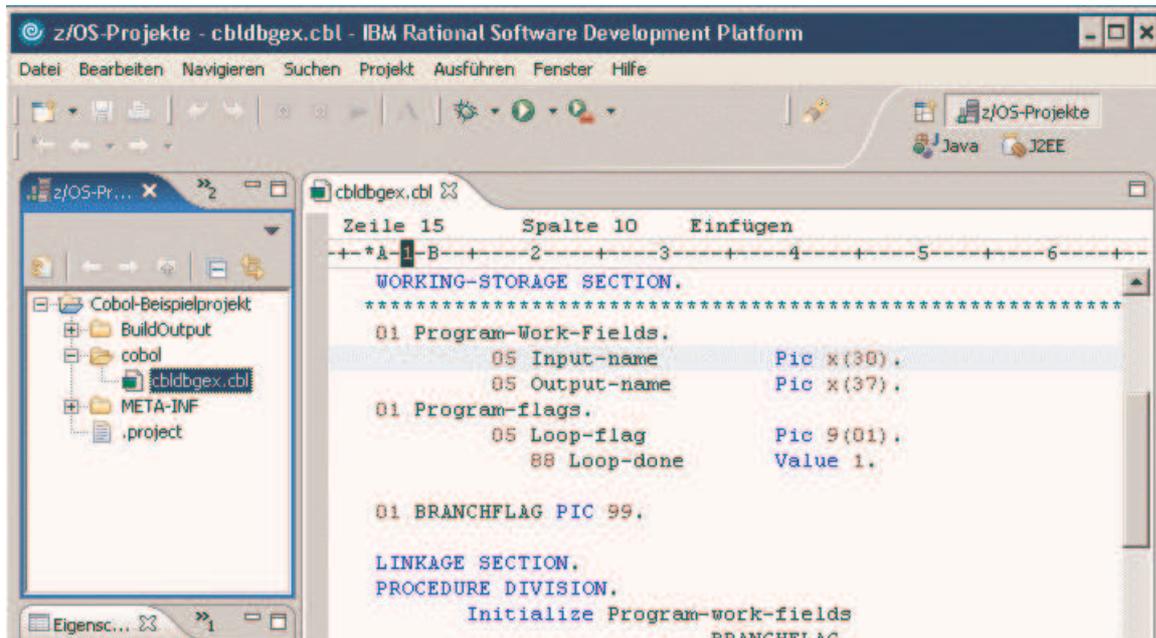


Abbildung 12

## 2.4 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:

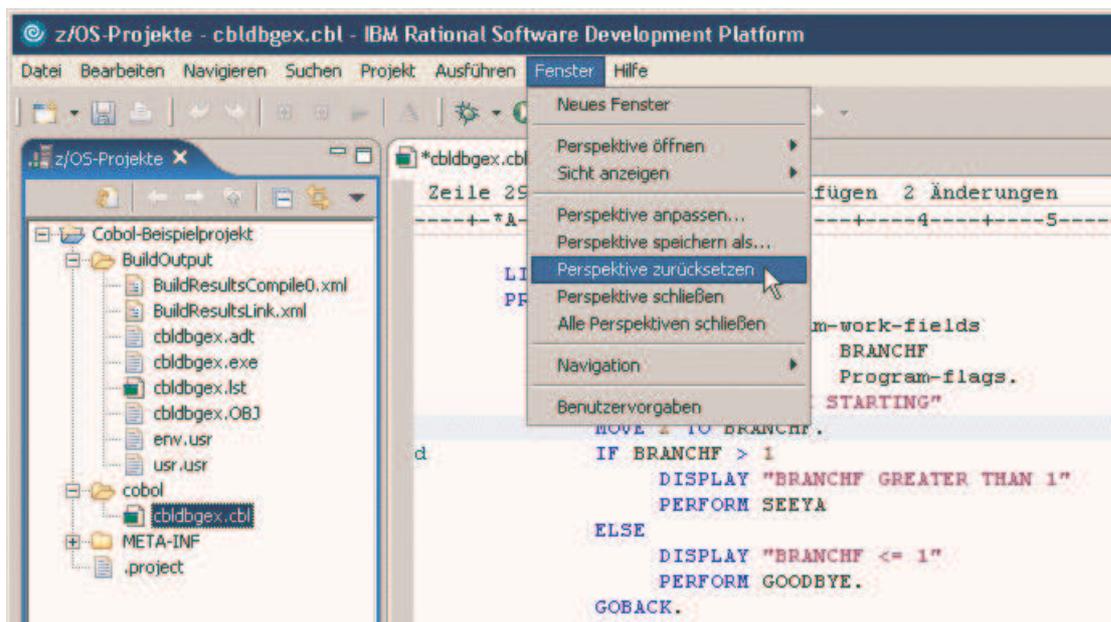


Abbildung 13

## 2.5 Wechseln zwischen verschiedenen geöffneten Perspektiven

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

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 auf folgendem Weg ebenfalls geöffnet werden:

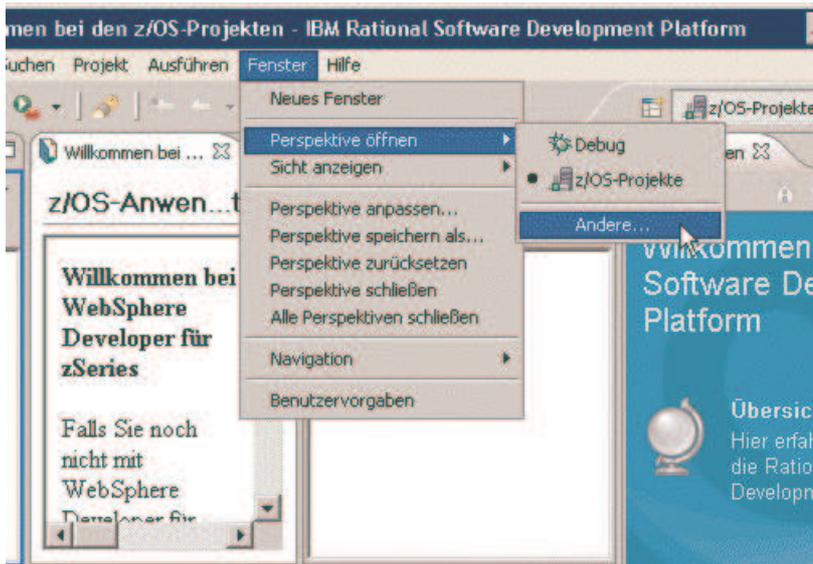
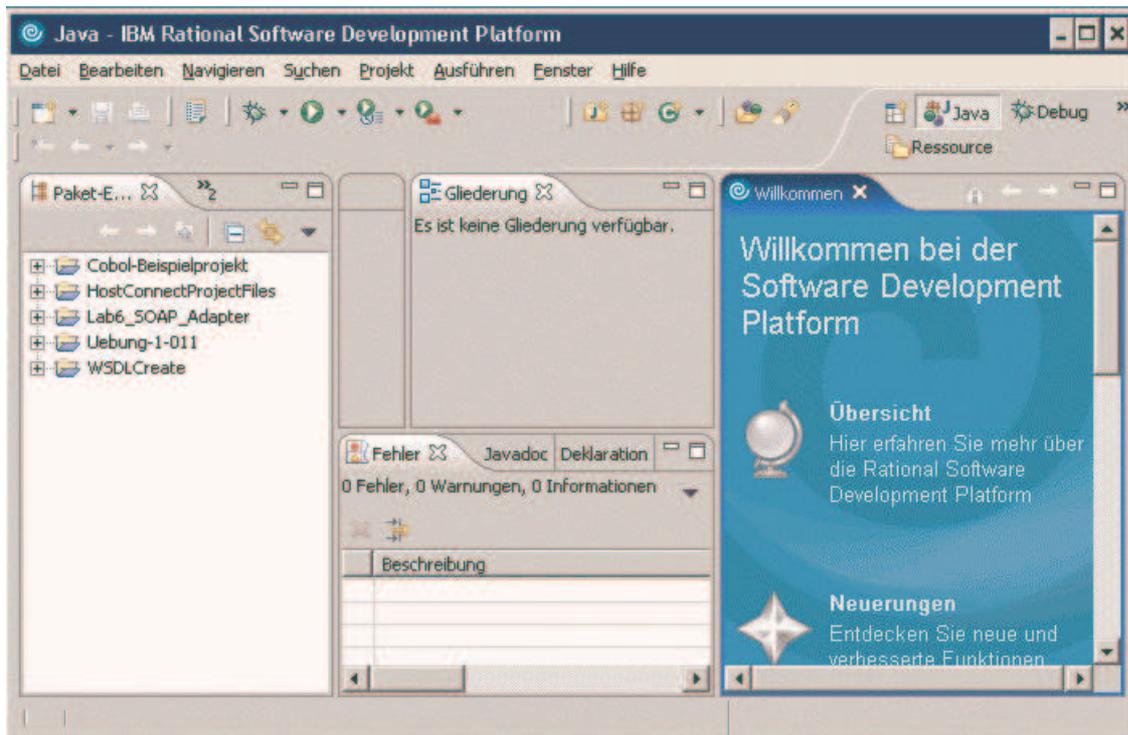


Abbildung 14



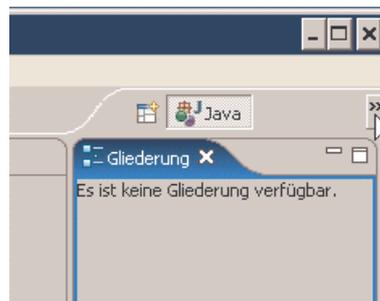
Abbildung 15

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



**Abbildung 16**

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



**Abbildung 17**

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

### 3 Erstellen eines Projektes mit einem Beispiel-Cobol-Programm

Wenn erforderlich: "z/OS-Projekte"-Perspektive öffnen  
(Fenster → Perspektive öffnen (→ andere...) → z/OS-Projekte)

Eventuell die View "Willkommen zu z/OS-Projekten" schließen.

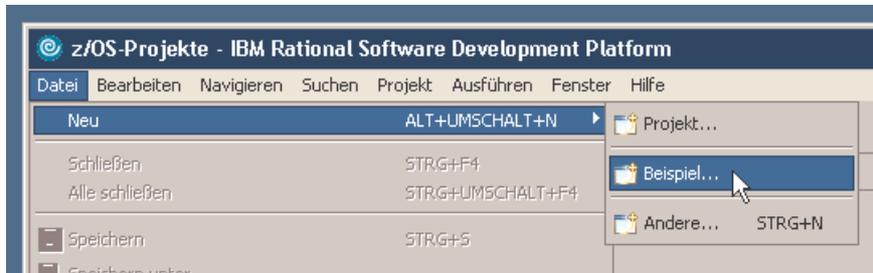


Abbildung 18

"Workstation COBOL" erweitern

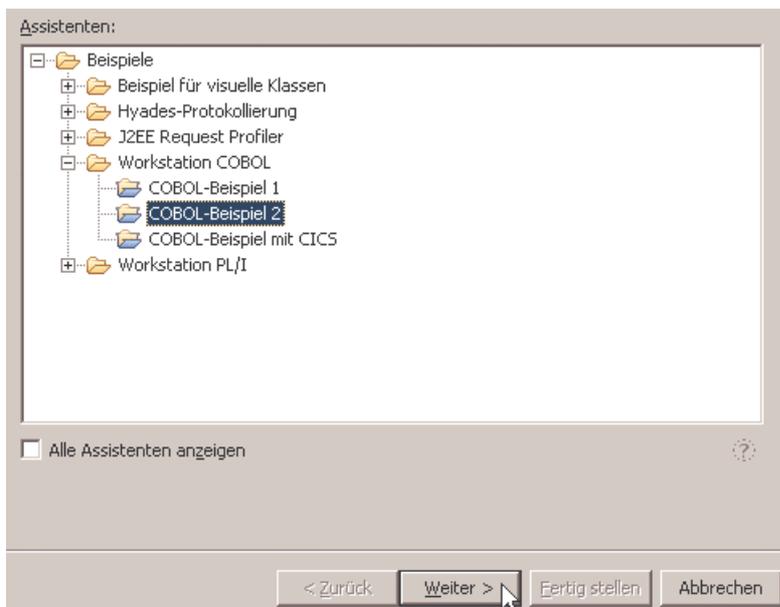


Abbildung 19

"Cobol-Beispielprojekt" als Projektnamen eingeben

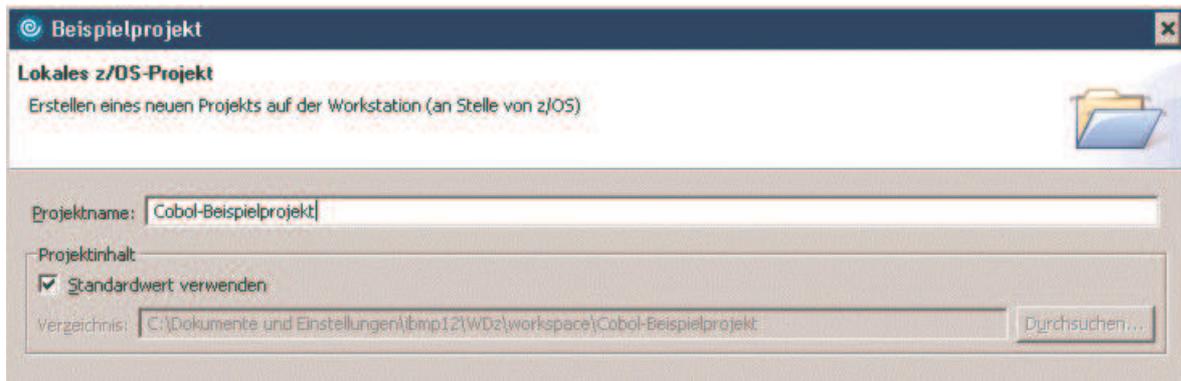


Abbildung 20

Klick auf "Weiter"

Im nächsten Fenster lassen sich Optionen für den lokalen Cobol-Compiler festlegen. Die Default-Werte können übernommen werden, vor allem muss die Compilierungsoption "TEST" hier festgelegt werden.

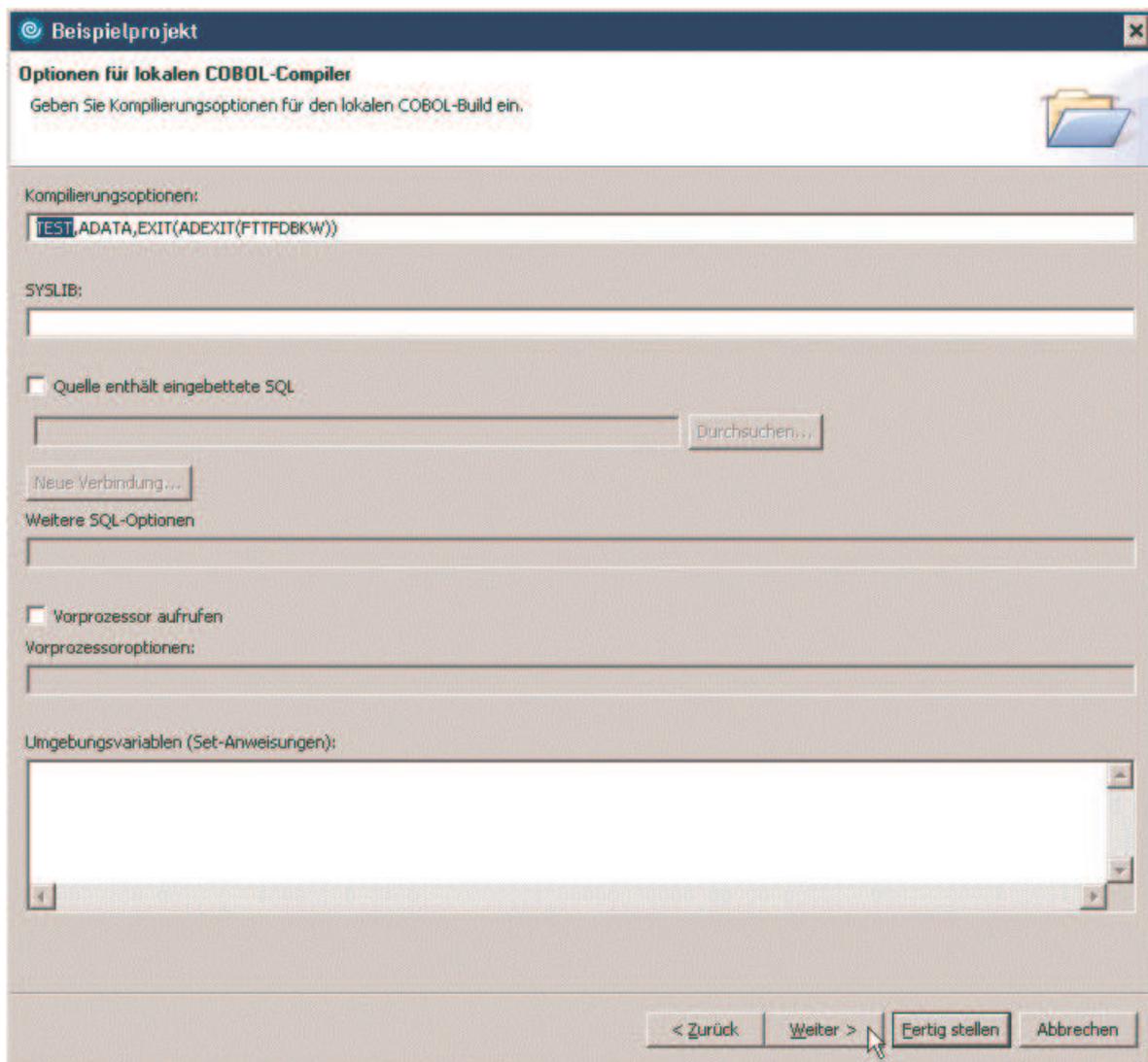
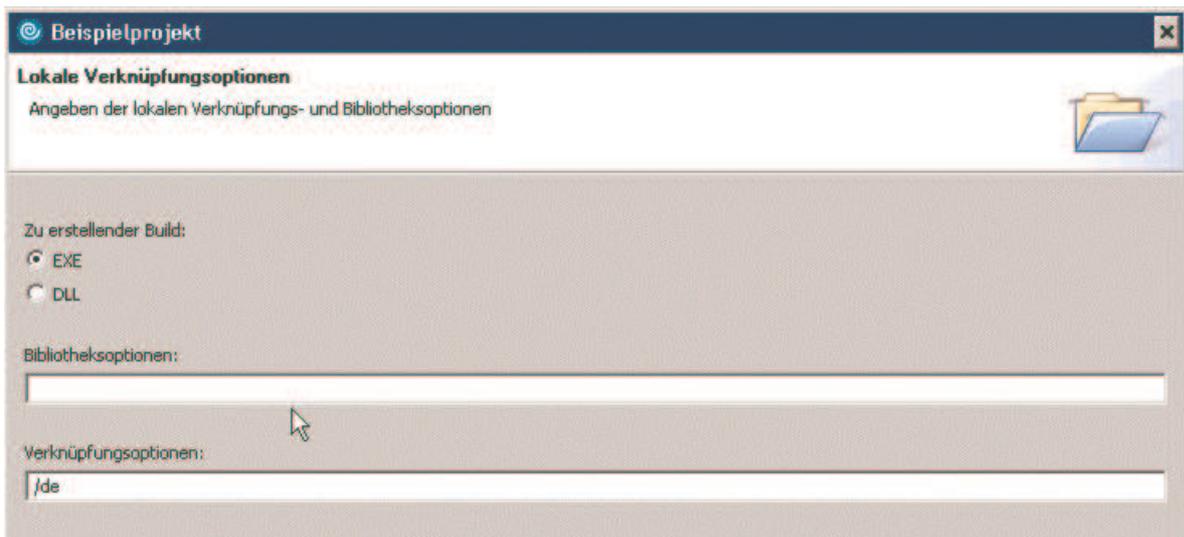


Abbildung 21

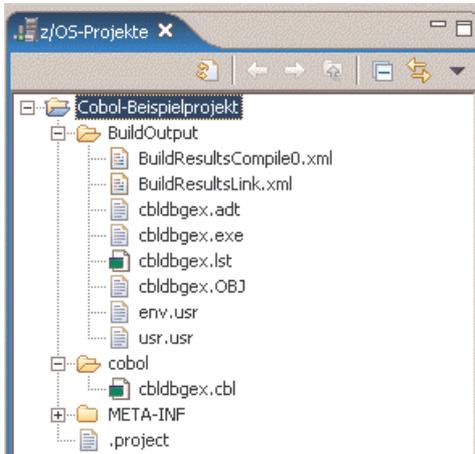
Im folgenden sind die Link-Parameter festzulegen: "EXE" und "/de"



**Abbildung 22**

Ein Klick auf "Fertig stellen" erstellt das Projekt mit dem Beispiel-Cobol-Programm.

Nach einigen Sekunden ist das Ergebnis links oben in der z/OS-Perspektive sichtbar. Nach Erweiterung von "BuildOutput" und "cobol" sieht das Ergebnis wie folgt aus:



**Abbildung 23**

Beispiel-Cobol-Programm-Code: "cbldbgex.cbl"  
Daraus erzeugte ausführbare EXE-Datei: "cbldbgex.exe"

## 4 Anlegen von Datasets (allocate)

WDz beherrscht auch das Anlegen von Datasets (allocate). Es lassen sich unter anderem die folgenden Typen von Datasets anlegen:

- Sequentielle Datasets
- Partitioned Data Sets (PDS)
- Partitioned Data Sets Extended (PDS/E)

Im Tutorial 1 a wurde gezeigt, wie man einen PDS "PRAKT20.TEST.DATASET" unter Nutzung eines 3270-Emulators anlegen lässt. Nachfolgend wird gezeigt, wie sich dieser Dataset unter Nutzung von WDz ebenfalls anlegen lässt:

Zuerst ist die Verbindung zu den Datasets zu aktivieren, hier Aktivierung von "Verbindung zu Padme" (siehe Kapitel 6.2).

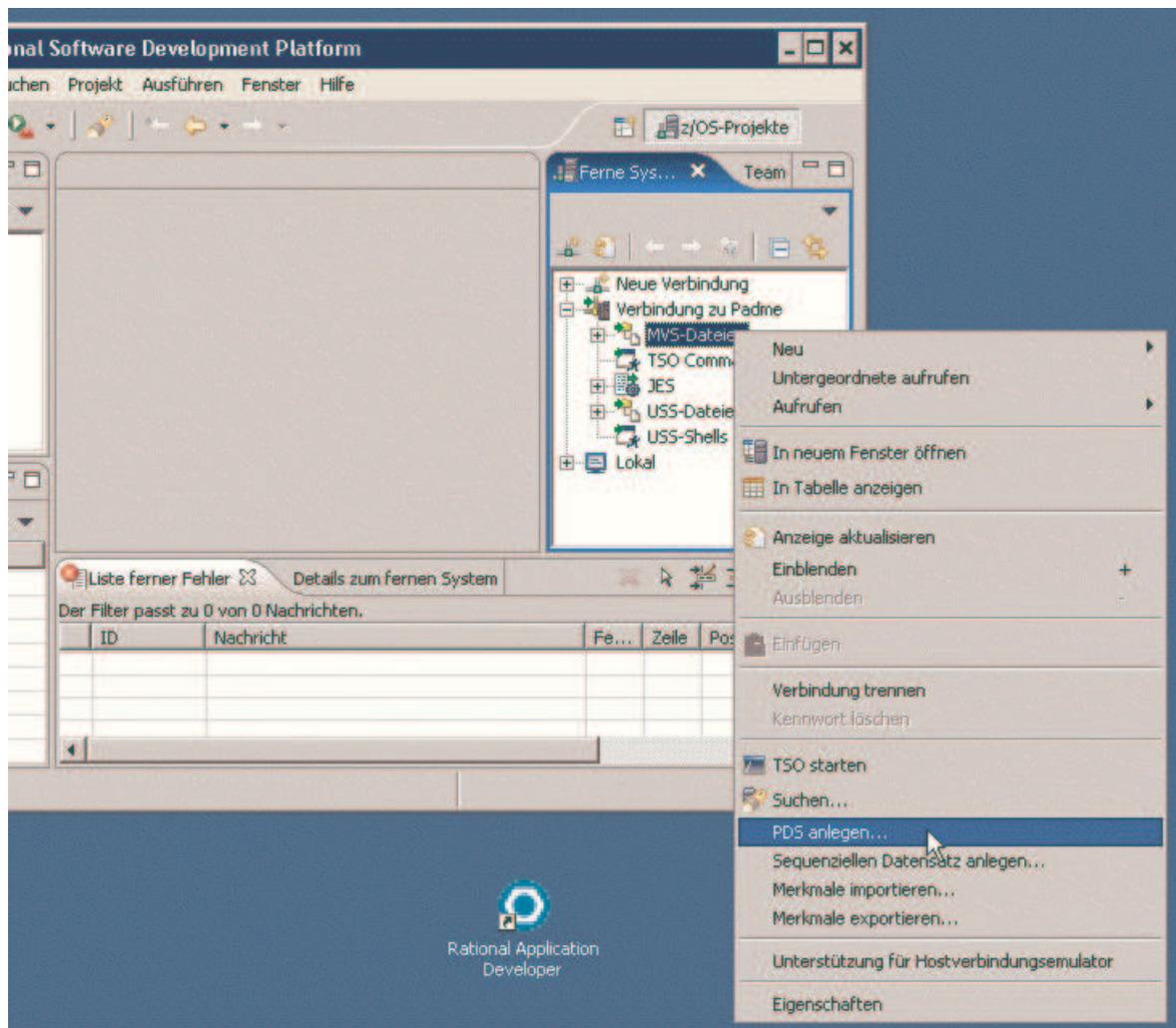
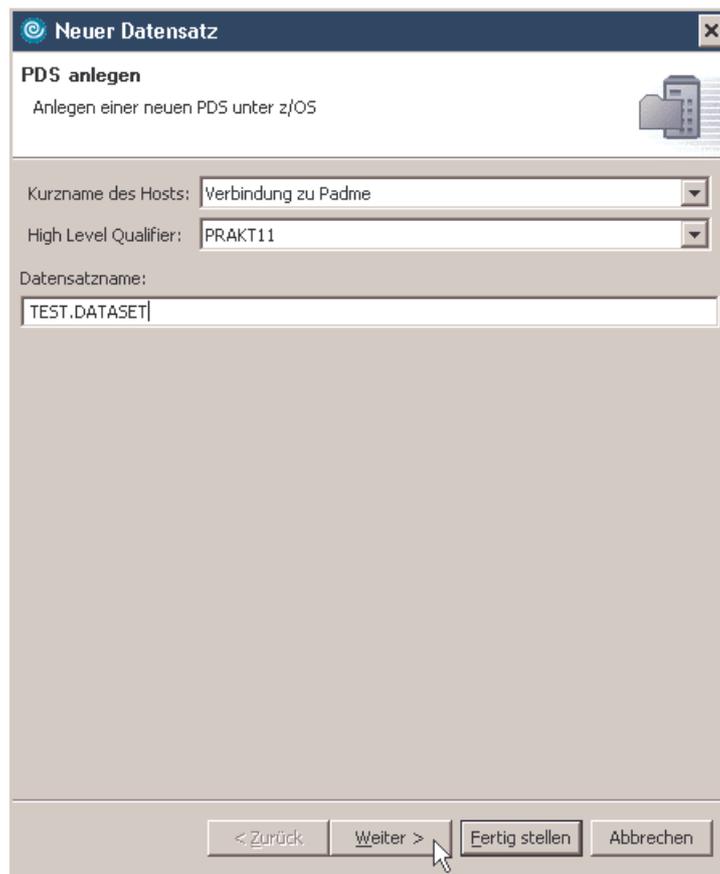


Abbildung 24

In der View "Ferne Systeme" nun Rechtsklick auf "MVS-Dateien" und im sich geöffneten Popup-Fenster Klick auf "PDS anlegen..."

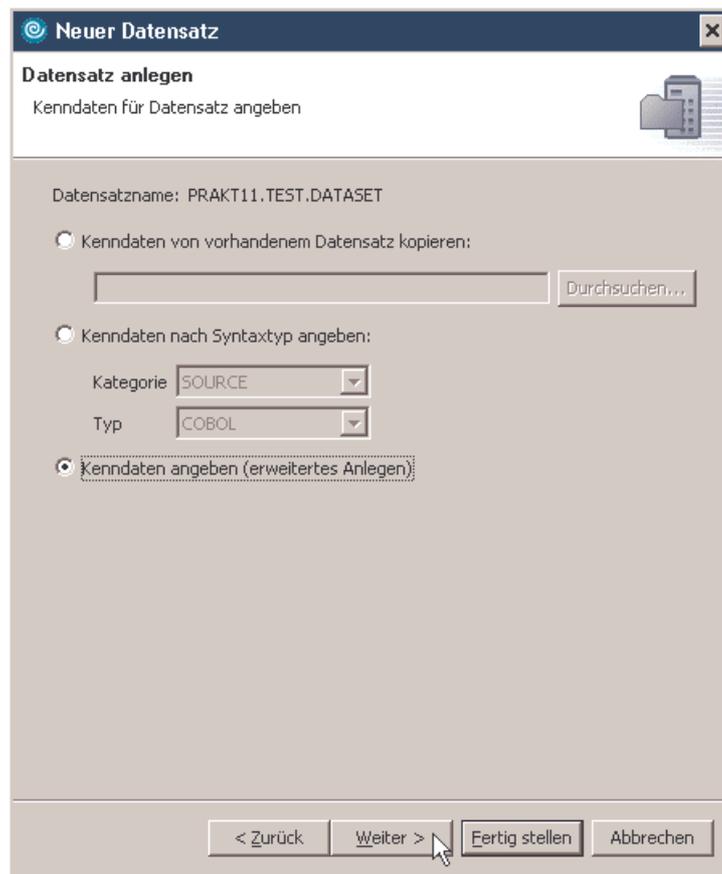


**Abbildung 25**

Zu spezifizieren sind

- der Name der verwendeten und aktivierten Verbindungsressource, hier "Verbindung zu Padme"
- Die High Level-Komponente des Namens des anzulegenden Datasets (mit dem Mainframe-Login-Namen identisch)
- alle weiteren Komponenten des Dataset-Namens, hier "TEST.DATASET"

Man kann den Vorgang jetzt schon durch Klick auf "Fertig stellen" beenden. Dann legt WDz den Dataset unter Verwendung von Default-Kenndaten an. Wir klicken auf "Weiter", um die Kenndaten selbst eintragen zu können.



**Abbildung 26**

Es lassen sich

- die Kenndaten eines schon existierenden Datasets verwenden,
- die Kenndaten abhängig von Kategorie und Typ von WDz festlegen lassen oder
- die Kenndaten selbst festlegen (dies wird hier getan)

**Abbildung 27**

Nach Klick auf "Fertig stellen" ist nach ein paar Sekunden der Dataset angelegt.

## **5 Umgang mit lokalen Cobol-Programmen, LPEX-Editor**

In diesem Kapitel wird der Einsatz des LPEX-Editors für Arbeiten am Programm-Code eines Cobol-Programms gezeigt (am Beispiel des im vorigen Kapitel angelegten Cobol-Beispielprojektes).

### **5.1 Öffnen eines Cobol-Programms unter Nutzung des LPEX-Editors**

In der "z/OS-Projekte"-View Doppelklick auf "cbldbgex.cbl".  
Nun wird in einer zentralen View der Programm-Code des Cobol-Programms angezeigt.

## 5.2 Arbeiten mit der Gliederung des Cobol-Programm-Codes

Links unten befindet sich die View "Eigenschaften", unter ihr verdeckt die View "Gliederung". Um letztere View anzuzeigen, Klick auf "Gliederung".

Klick auf "ENVIRONMENT DIVISION" in der Gliederung bewirkt, dass genau dieser Teil des Cobol-Programm-Codes in der zentralen View angezeigt wird.

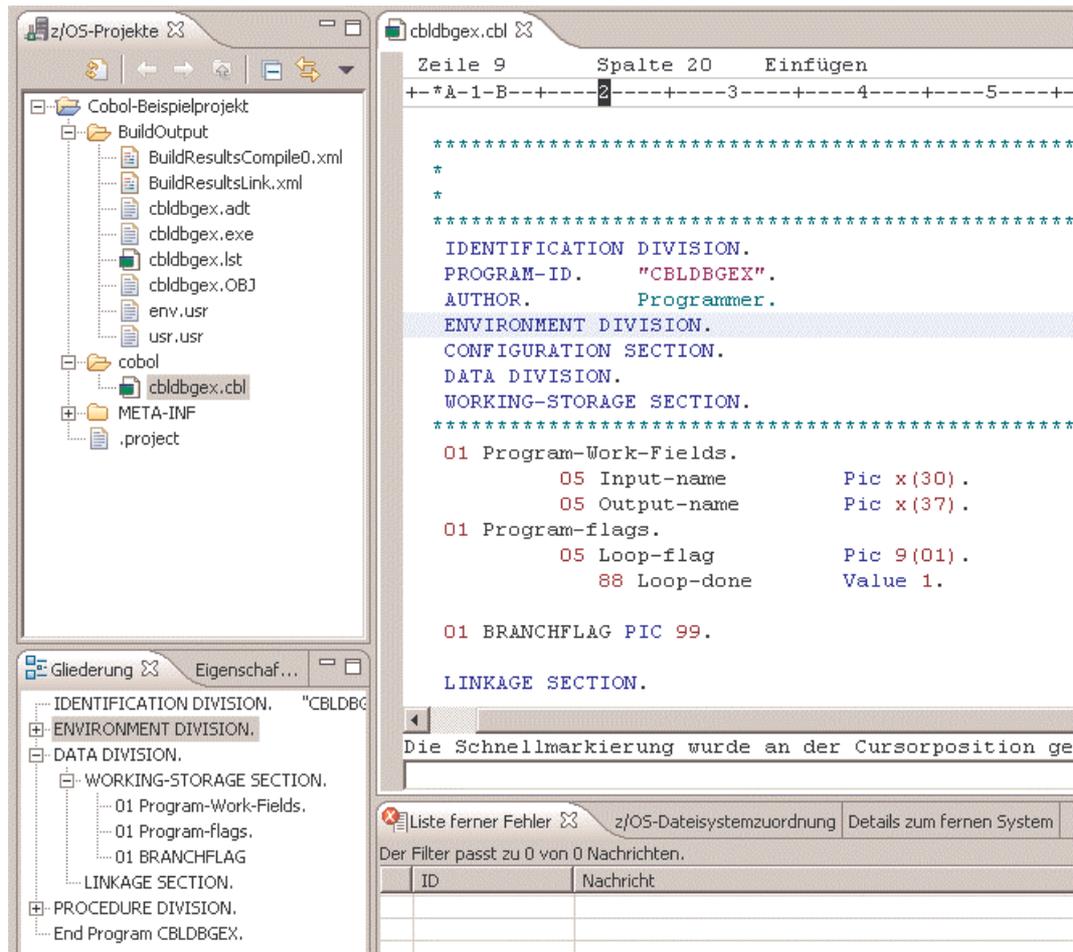


Abbildung 28

Analog dazu würde ein Klick auf "PROCEDURE DIVISION." genau den Teil "PROCEDURE DIVISION." in der zentralen Programm-Code-View anzeigen.

### 5.3 Emulations-Fähigkeiten des LPEX-Editors

Der ins WDz integrierte Editor heißt LPEX-Editor.

Dieser Editor kann verschiedene Editoren emulieren, u.a. den

- ISPF-Editor,
- Xedit-Editor,
- Emacs-Editor,
- Vi-Editor                      etc.

Standardmäßig ist der LPEX-Editor eingestellt. Ist man die Funktionsweise des ISPF-Editors gewöhnt, so lässt sich beispielsweise die ISPF-Editor-Emulation wie folgt aktivieren:

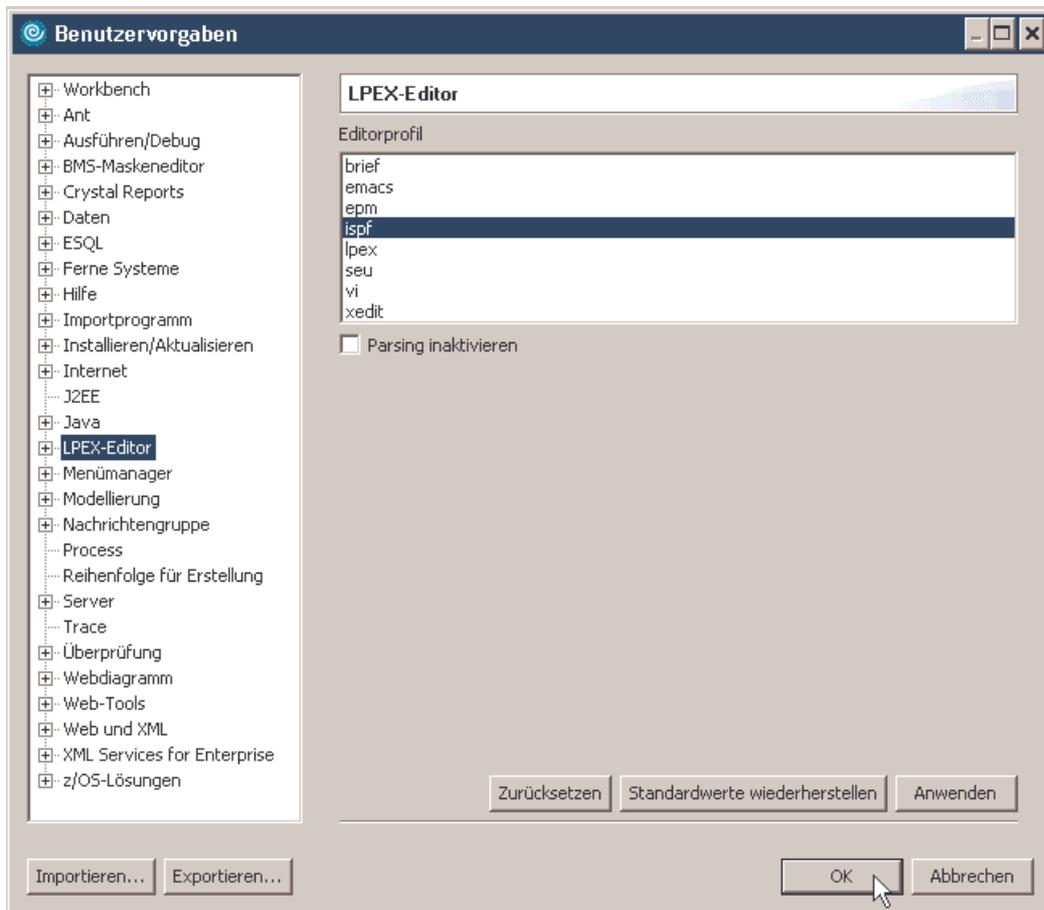


Abbildung 29

## 5.4 Zu einer bestimmten Zeilen-Nummer springen

Möchte man zu einer bestimmten Zeilen-Nummer im Cobol-Programm-Code springen, dann die Tastenkombination Strg+L benutzen. Es erscheint ein neues Feld "Zeilennummer:" unter dem Cobol-Programm-Code:

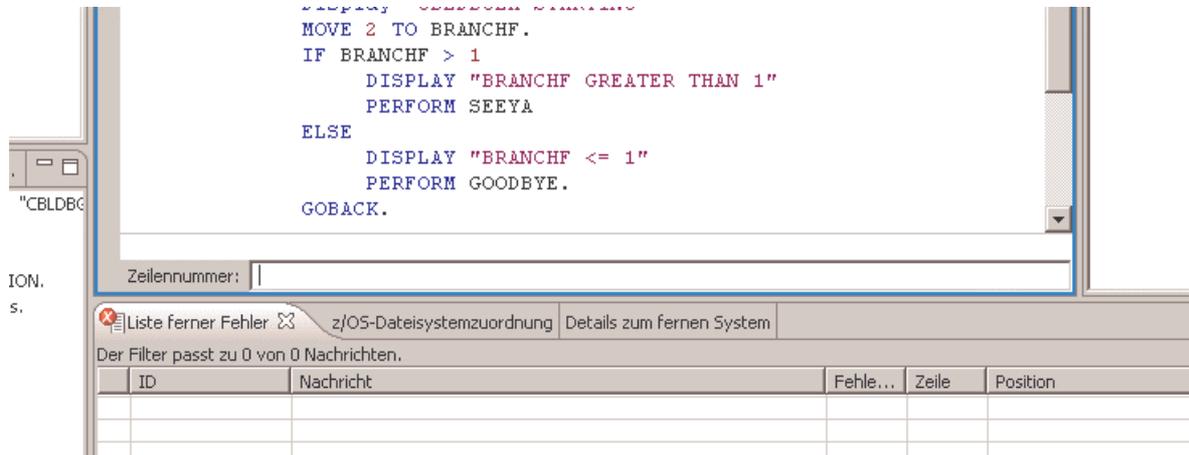


Abbildung 30

In diese Zeile zum Beispiel "25", gefolgt von der Eingabetaste, eingeben; so wird vom Programm-Code die Zeile 25 – wenn vorhanden – angezeigt.

## 5.5 Suche nach bestimmten Strings im Cobol-Programm-Code

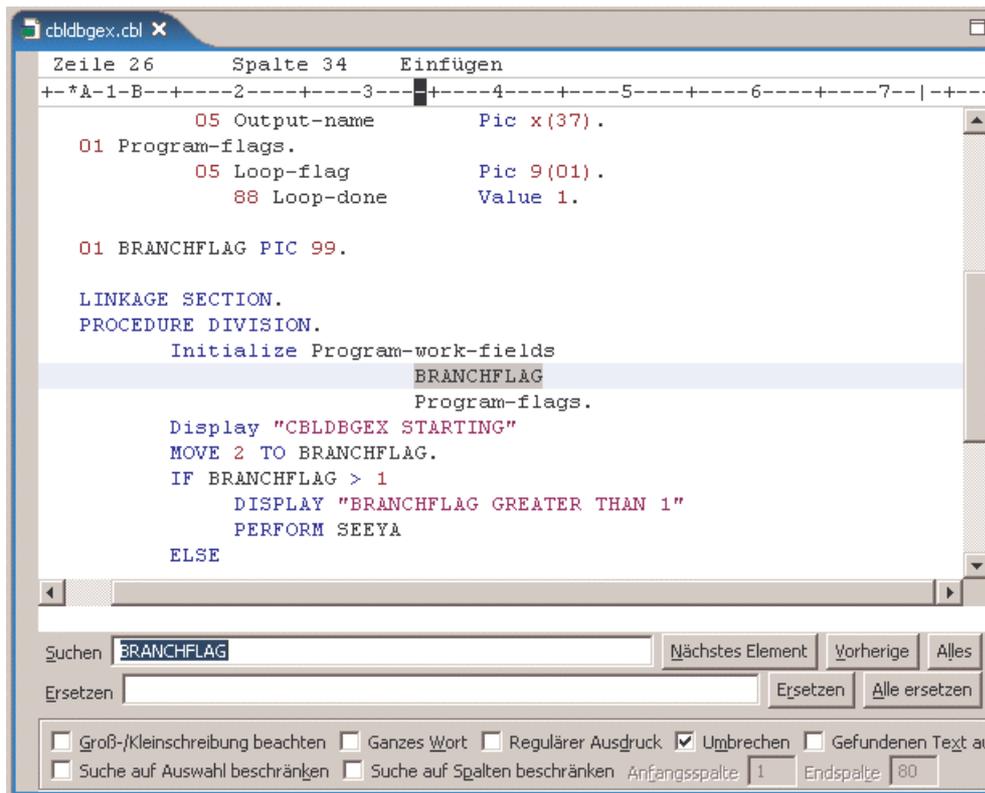
Die zentral gelegene View mit dem Cobol-Programm-Code muss die aktive View sein. Wenn nicht, dann Klick in diese View.

Die Tastenkombination Strg+F ermöglicht nun das suchen (und ersetzen) von Strings im Cobol-Programm-Code.

Soll zum Beispiel nach "Branchflag" gesucht werden:

In die Zeile "Suchen" den String "BRANCHFLAG" eingeben.

(Ein Abschließen mit der Eingabetaste ist nicht erforderlich.)



**Abbildung 31**

Es ist intuitiv sehr einfach, diese Suchen- und Ersetzen-Funktionalität zu verwenden.

Probieren Sie doch mal:

Nach dem nächsten String "BRANCHFLAG" suchen,  
 nach dem vorigen String "BRANCHFLAG" suchen,  
 Alle vorhandenen "BRANCHFLAG" z.B. durch "BRANCHF" ersetzen, etc.

## 5.6 Hilfefunktion zu den Cobol-Programm-Statements

Zu den Cobol-Programm-Statements existiert eine handliche Hilfe-Funktion. Man ruft diese auf, indem man in das jeweilige Schlüsselwort klickt und anschließend die Taste F1 betätigt.

Möchte man beispielsweise zu "DISPLAY" die Hilfe aufrufen, dann stellt man den Cursor auf "DISPLAY" und betätigt anschließend F1. Nach einigen Sekunden öffnet sich das nachfolgend dargestellte Hilfe-Fenster

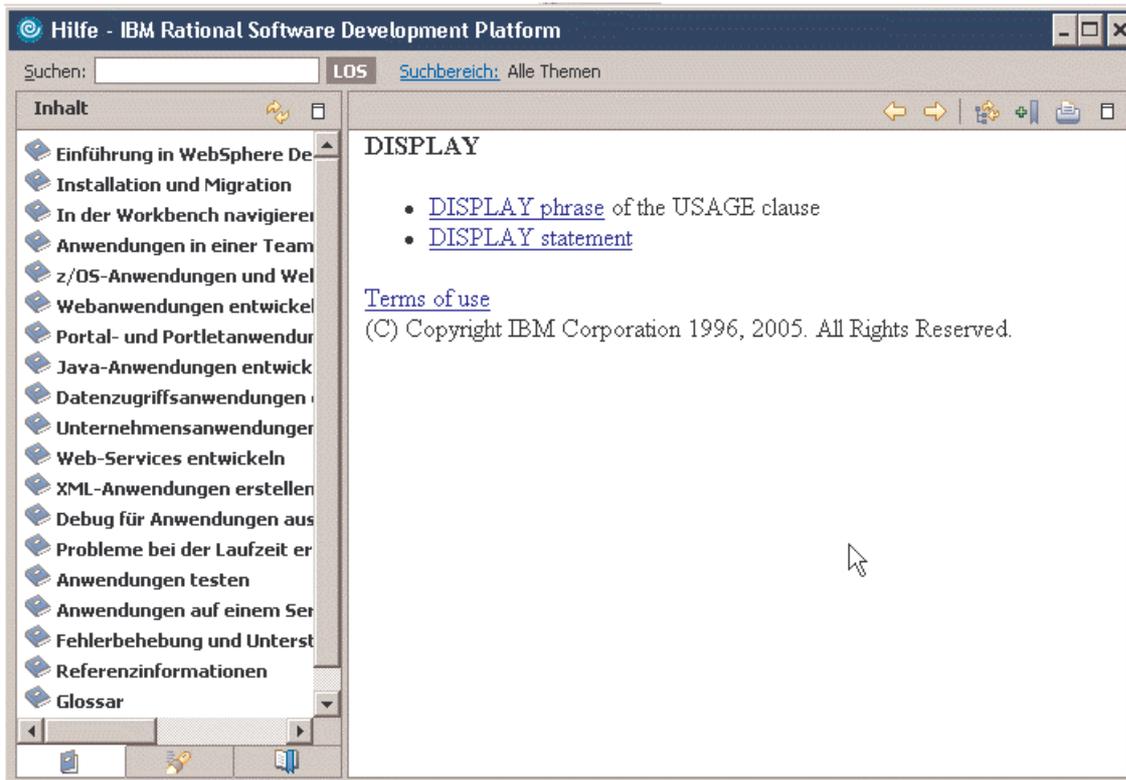


Abbildung 32

Anschließend Klick auf "DISPLAY statement".

## 5.7 Automatische Backups des Cobol-Programms wiederherstellen

Sobald eine Änderung am Cobol-Programm-Code erfolgte, wird das an einem Stern neben dem Dateinamen sichtbar (siehe nachstehende Abbildung links oben).

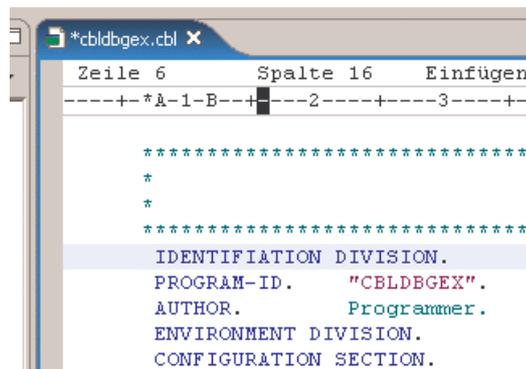
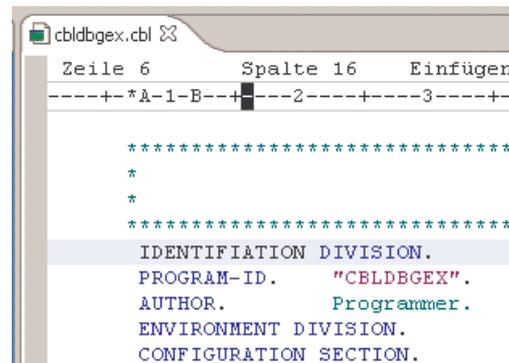


Abbildung 33

Nachdem die Änderung per Ctrl-S auf Platte gespeichert wurde, ist der Stern wieder verschwunden:



```
cbldbgex.cbl
Zeile 6      Spalte 16  Einfügen
-----*A-1-B-----2-----3-----+
*****:
*
*
*****:
IDENTIFIATION DIVISION.
PROGRAM-ID.      "CBLDBGEX".
AUTHOR.          Programmer.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
```

**Abbildung 34**

Es wurde aber eine Sicherheitskopie der alten Version des Dokuments gemacht, welche bei Bedarf wieder reaktiviert werden kann.

Rechter Mausklick auf den Cobol-Programm-Code, der durch eine ältere Version ersetzt werden soll, in diesem Beispiel rechter Mausklick auf "cbldbgex.cbl" (in View "z/OS-Projekte").

→ Ersetzen durch → Lokales Protokoll...

Im sich öffnenden Fenster kann man die aktuelle Version mit verschiedenen älteren Versionen vergleichen:

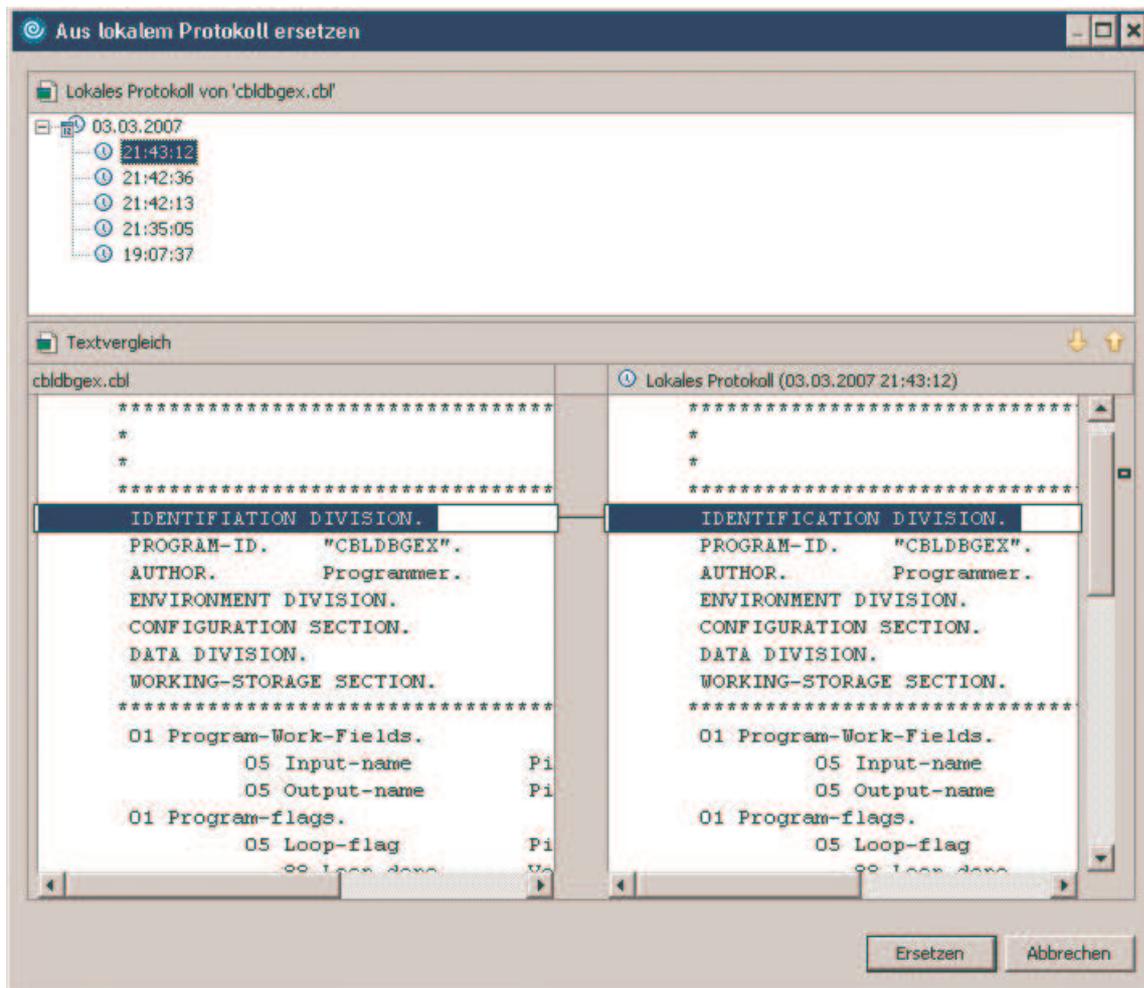


Abbildung 35

Man kann sich eine (von meistens mehreren) Sicherheitskopien zum Vergleich auswählen. Im Beispiel ist die Sicherheitskopie vom 03.03.2007 um 21:43:12 Uhr ausgewählt.

Möchte man diese ausgewählte Sicherheitskopie als neues aktuelles Dokument haben, dann ist das möglich durch einen Klick auf "Ersetzen".

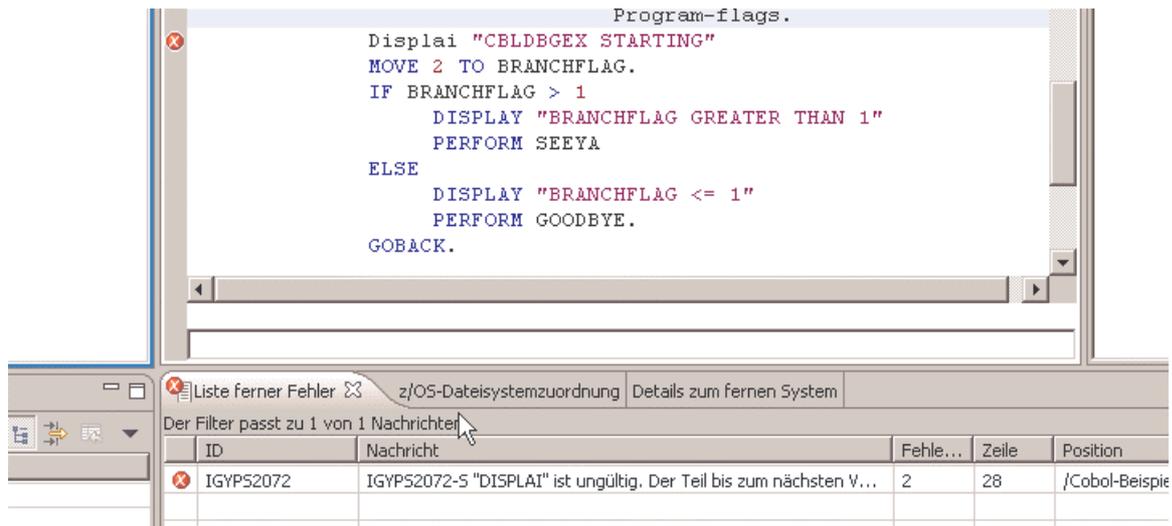
## 5.8 Lokaler Syntax-Check

Ein z.B. gerade geschriebenes Cobol-Programm kann mit einem lokalen Syntax-Check überprüft werden.

Klick mit rechter Maustaste auf den Namen des zu überprüfenden Programm-Codes, hier "cbldbgex.cbl"

→ Im sich öffnenden Popup-Fenster Klick auf "lokale Syntaxprüfung"

Das Ergebnis zeigt die folgende Abbildung. Beachten Sie insbesondere, dass in der "Liste ferner Fehler" jetzt eine Fehlermeldung steht.



**Abbildung 36**

Ersetzen von "Displai" nach dem korrekten Cobol-Schlüsselwort "Display".  
Abspeichern des nun korrekten Cobol-Codes per Cntl-S.

Wird nun die lokale Syntaxprüfung wiederholt, taucht in "Liste ferner Fehler" keine Fehlermeldung mehr auf.

## 5.9 Syntax-Check während der Code-Eingabe

Im folgenden wird der Editor als ISPF-Editor verwendet. Falls die ISPF-Editor-Emulation noch nicht aktiviert ist:

Fenster → Benutzervorgaben → LPEX-Editor → ISPF → OK

Typisch für die ISPF-Editor-Emulation sind die 6-stelligen Zeilennummern neben dem Cobol-Programm.

Einige Syntax-Fehler werden schon während des Schreibens von Cobol-Programm-Code erkannt und angezeigt.

Ein Beispiel dafür:

Auf der Zeile 25 mit dem Zeileninhalt

'DISPLAY "Enter name or O to quit" UPON CONSOLE'

einfach die Ausführungszeichen nach "quit" weglassen und anschließend den Cursor in eine andere Zeile bewegen. Die nachstehende Abbildung zeigt die auftretende Fehlermeldung. Der fehlerhafte Text ist rot unterlegt und die Fehlermeldung auffallend in violetter Farbe geschrieben.

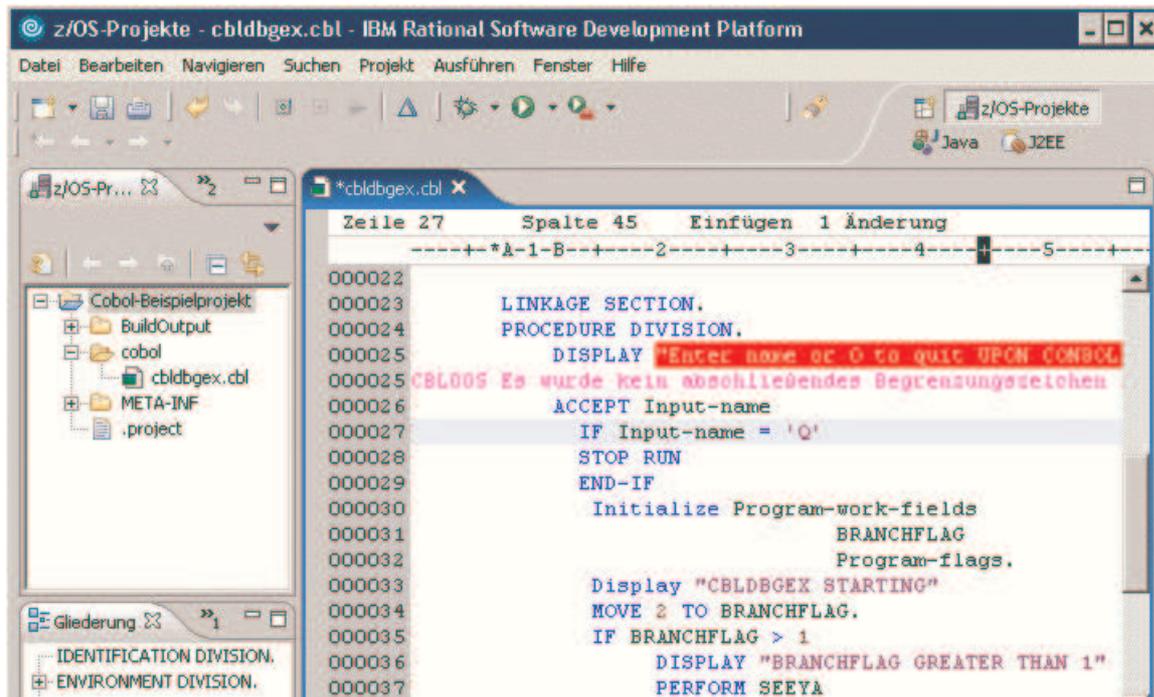


Abbildung 37

## 5.10 Die Hilfe beim Erstellen von Cobol-Programm-Code: Der Code-Assistent

- Der Code-Assistent (code assist) ist ein nützliches Feature, falls man bei der Benutzung von Cobol-Schlüsselwörtern unsicher ist.
- Am nützlichsten ist der Code-Assistent aber beim Gebrauch von Variablen. Insbesondere bei großen Cobol-Programmen gibt es häufig sehr viele Variablen und man verliert die Übersicht, welche Variable man mal wie deklariert hatte.

### Beispiel für a):

Es soll eine neue Zeile ins Cobol-Programm eingefügt werden. Dies ist möglich mit "i" gefolgt von der Eingabetaste:

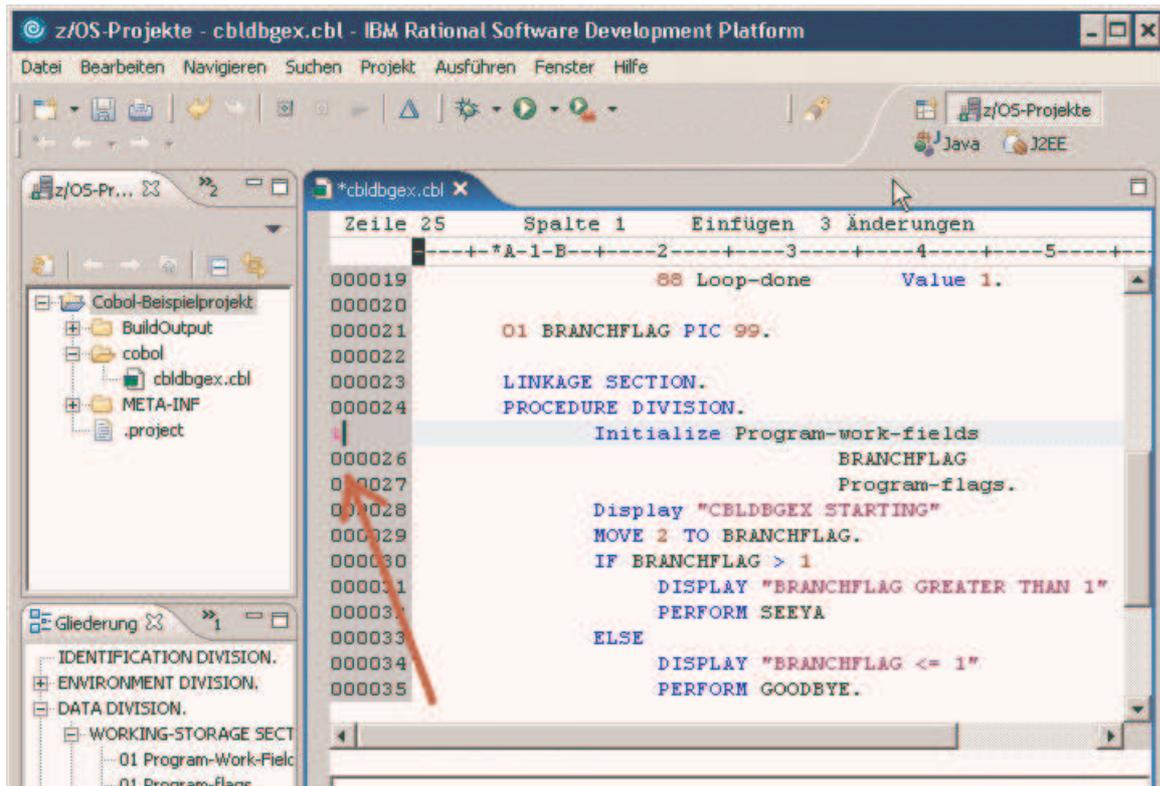


Abbildung 38

Positionieren Sie jetzt den Cursor an die Stelle, an der die neue Cobol-Programm-Zeile beginnen soll. Drücken Sie jetzt **Strg+Lehrtaste**. Es erscheint eine Liste mit allen **Variablen-Namen** oder **Cobol-Schlüsselworten**, die an der Cursor-Position jetzt eingefügt werden könnten (siehe nachfolgende Abbildung).

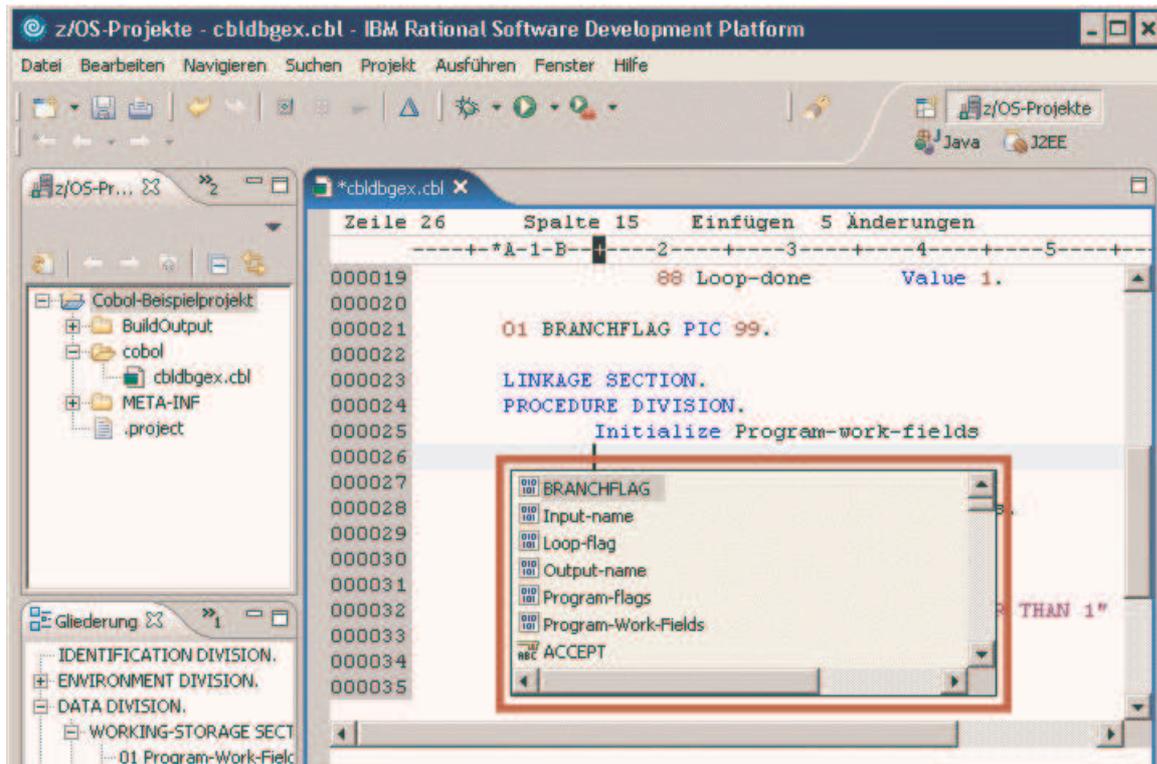


Abbildung 39

In der Liste der möglichen Cobol-Schlüsselworte oder Variablennamen kann man unter Nutzung der Pfeiltasten "nach oben" und "nach unten" blättern und durch Betätigung der Eingabetaste den gewünschten Bezeichner auswählen.

Alternativ dazu kann man den oder die ersten Buchstaben des gesuchten Bezeichners eingeben und danach **Strg+Leertaste** drücken. Als Beispiel tippen wir ein "D" als Anfangsbuchstaben des Schlüsselwortes "DISPLAY". Nach Betätigung von Strg+Leertaste erscheinen alle möglichen Bezeichner, die mit "D" beginnen.

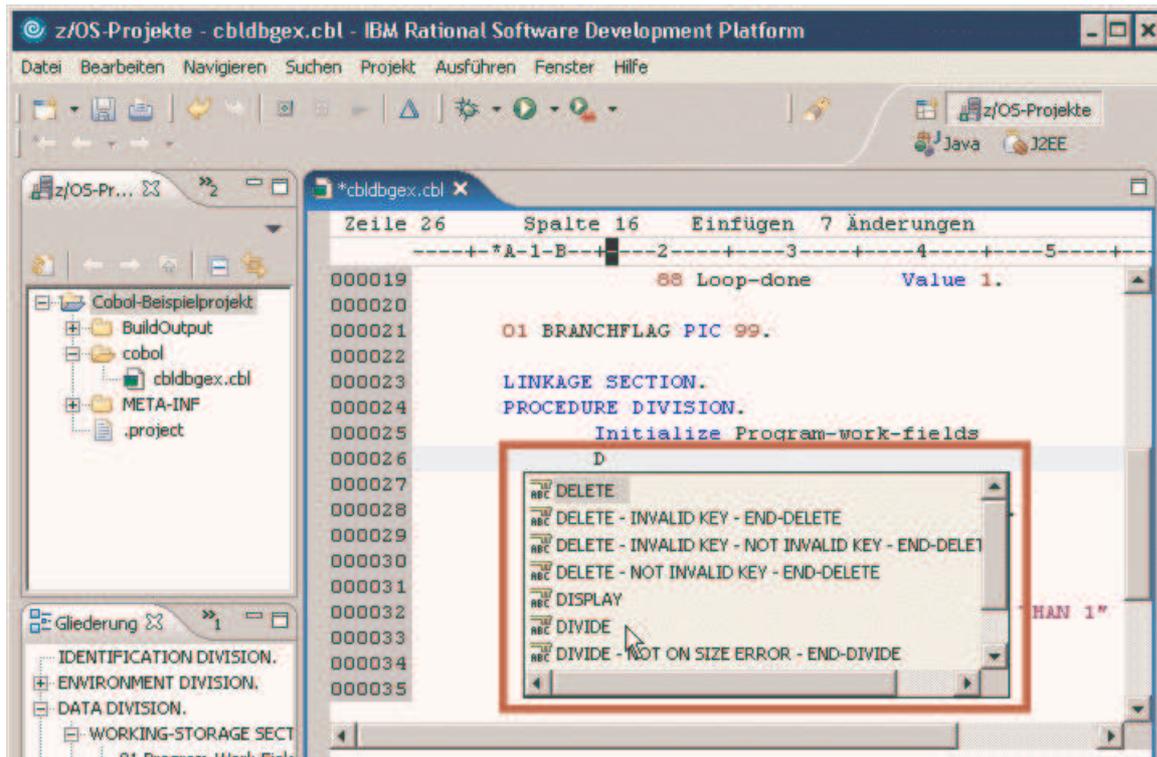


Abbildung 40

Wir wählen "DISPLAY" aus und betätigen die Eingabetaste.

Alternativ dazu hätten wir "DI" tippen können und anschließend hätte Strg+Leertaste eine Liste aller Schlüsselworte gebracht, die mit "DI" beginnen. Aus dieser Liste hätten wir "DISPLAY" oder "DIVIDE" auswählen können.

Vorschlag für weiteres Austesten des Code-Assistenten:

Schreiben Sie weiter an der gerade begonnenen DISPLAY-Zeile unter Einsatz des Code-Assistenten. Die Zeile soll so lauten:

DISPLAY "Enter name or Q to quit" UPON CONSOLE

Setzen Sie den Code-Assistenten ein, um die Worte "UPON" sowie "CONSOLE" zu vervollständigen.

### Beispiel für b)

An einem Beispiel wird gezeigt, wie der Code-Assistent verwendet werden kann, um einen früher deklarierten Variablennamen wiederzufinden.

Zuerst ist unter der gerade geschriebenen Cobol-Code-Zeile eine weitere Leerzeile mit Hilfe von "i" anzulegen.

In diese Leerzeile soll "ACCEPT Input-name" aufgenommen werden. Es wird unterstellt, dass vom Variablen-Namen sicher nur "input..." bekannt ist. Deshalb wird der Code-Assistent verwendet, um diesen Namen zu vervollständigen.

Tippen Sie eventuell unter Verwendung des Code-Assistenten zuerst "ACCEPT " ein. Tippen Sie dann "input" ein und befragen Sie mittels Strg+Leertaste wieder den Code-Assistenten. Dieser findet auch den gesuchten Variablennamen (siehe Abbildung 41).

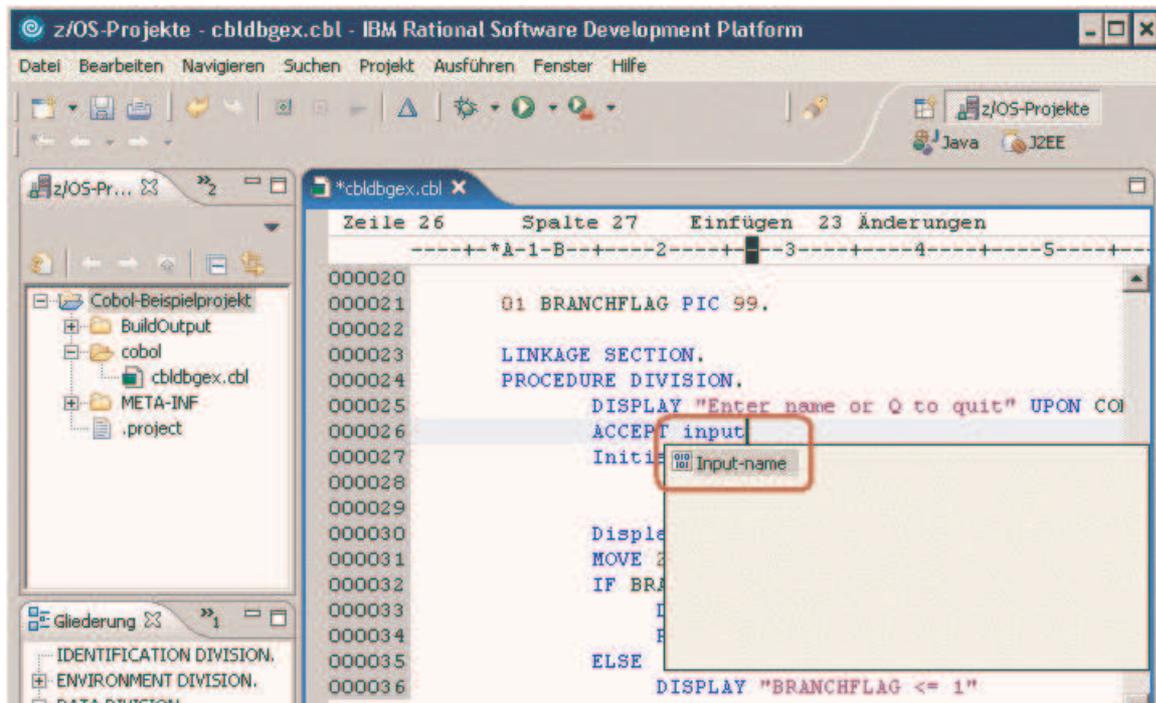


Abbildung 41

### Vorschlag für das Erstellen von weiteren 3 Zeilen Code unter Nutzung des Code-Assistenten

Fügen Sie unter der gerade geschriebenen Cobol-Zeile drei Leerzeilen mittels "i3" ein:

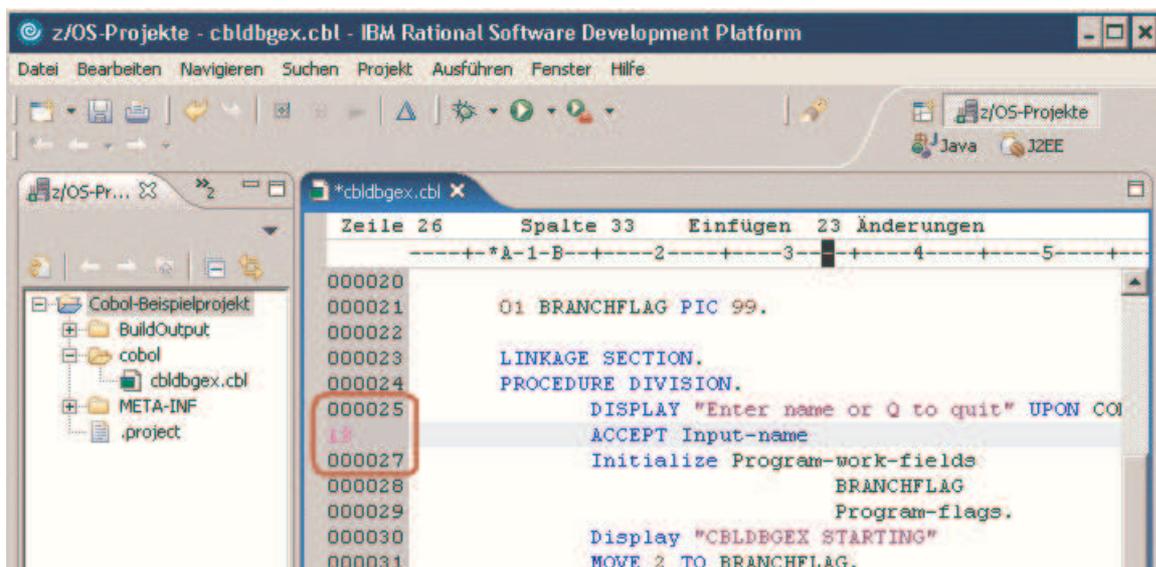


Abbildung 42

Erstellen Sie die folgenden drei Zeilen unter Einsatz des Code-Assistenten:

```
IF Input-name = 'Q'  
STOP RUN  
END-IF
```

## 6 Eine Remote-Verbindung zu einen Mainframe anlegen, auf- und abbauen

### 6.1 Eine Verbindungs-Ressource anlegen

In der View "Ferne Systeme" – wenn erforderlich – "Neue Verbindung" erweitern. Rechtsklick auf "z/OS..." und "Neue Verbindung..." auswählen (siehe nachstehende Abbildung)

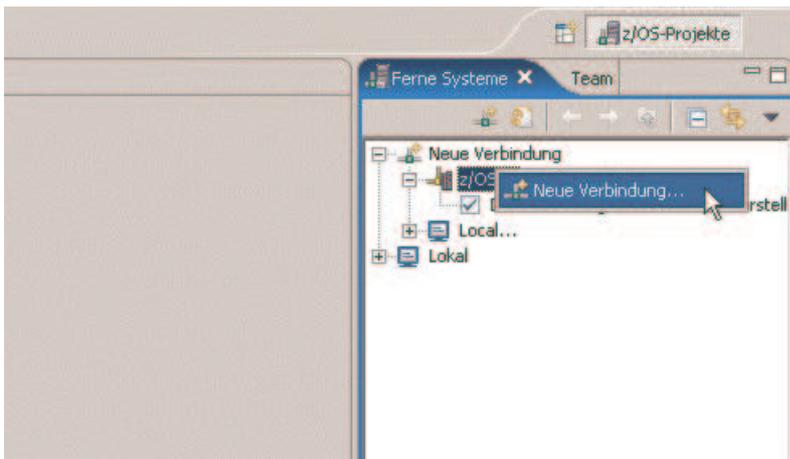


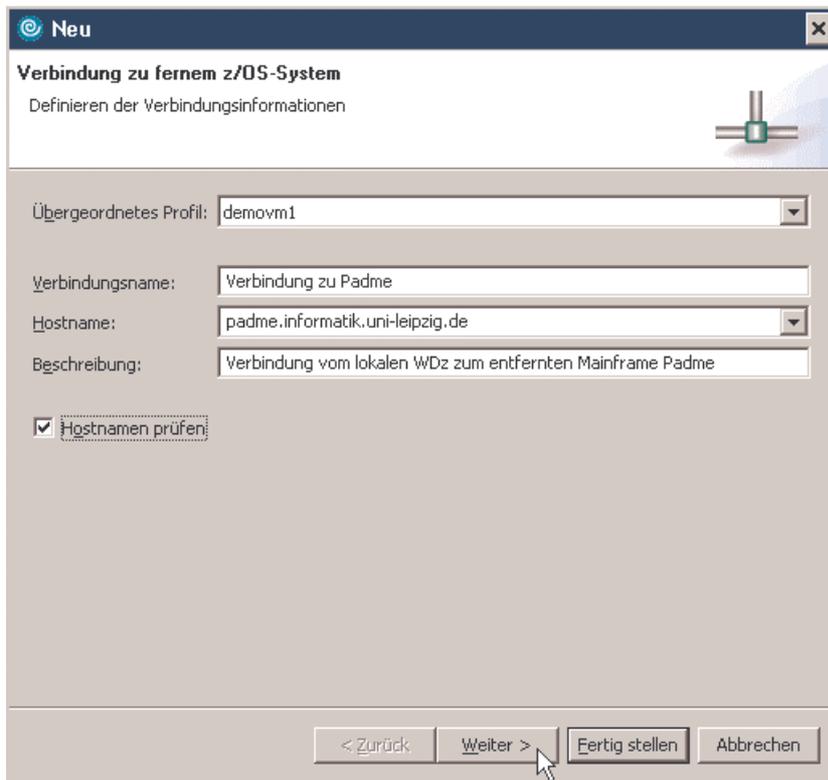
Abbildung 43

Die anzulegende Verbindung benötigt einen eindeutigen Namen, z.B. "Verbindung zu Padme".

Als Hostnamen lässt sich "padme.informatik.uni-leipzig.de" oder die IP-Adresse "139.18.4.35" eintragen.

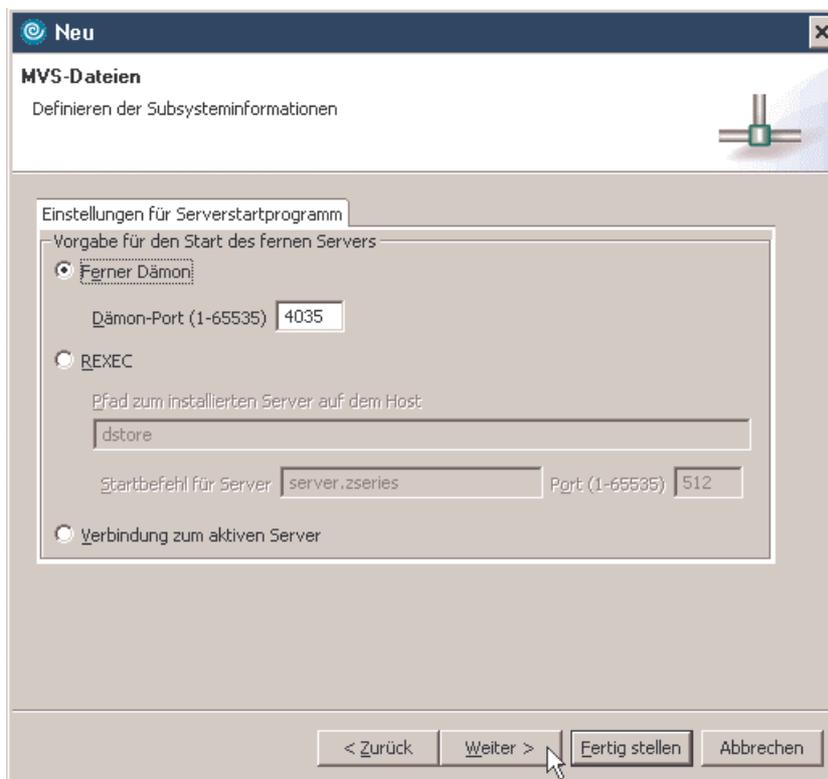
Als "Beschreibung" ist ein beliebiger Text einzutragen, z.B. "Verbindung vom lokalen WDz zum entfernten Mainframe Padme".

Wird ein Häkchen vor "Hostnamen prüfen" gesetzt, wird die Korrektheit des Hostnamens, hier "padme.informatik.uni-leipzig.de" geprüft.



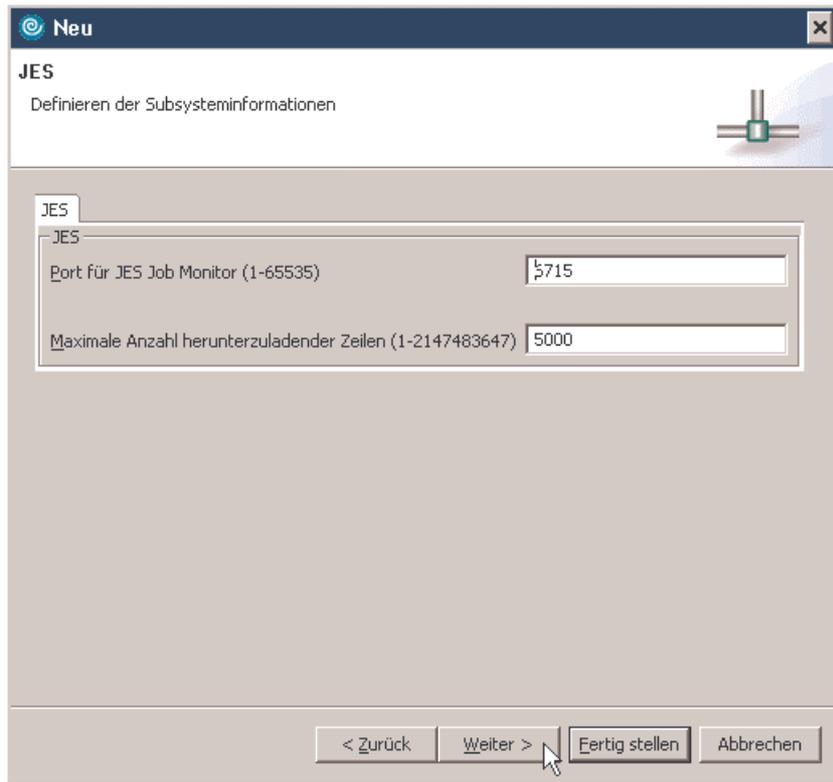
**Abbildung 44**

Klick auf "Weiter"



**Abbildung 45**

Klick auf "Weiter"



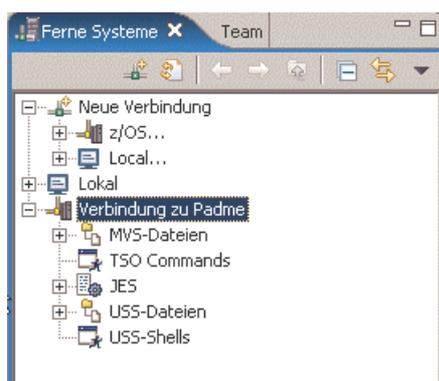
**Abbildung 46**

Klick auf "Weiter"

Es erscheint ein Ihnen schon bekanntes Fenster.

Klick auf "Fertig stellen".

Nun ist in der "Ferne Systeme"-View die Ressource "Verbindung zu Padme" dazugekommen:



**Abbildung 47**

Noch sind die Verbindungen dieser angelegten Ressource inaktiv.

## 6.2 Aktivierung der Verbindungen einer Verbindungs-Ressource

Rechte Maustaste auf "Verbindung zu Padme",  
im sich geöffneten Popup-Fenster Klick auf "Verbindung herstellen".

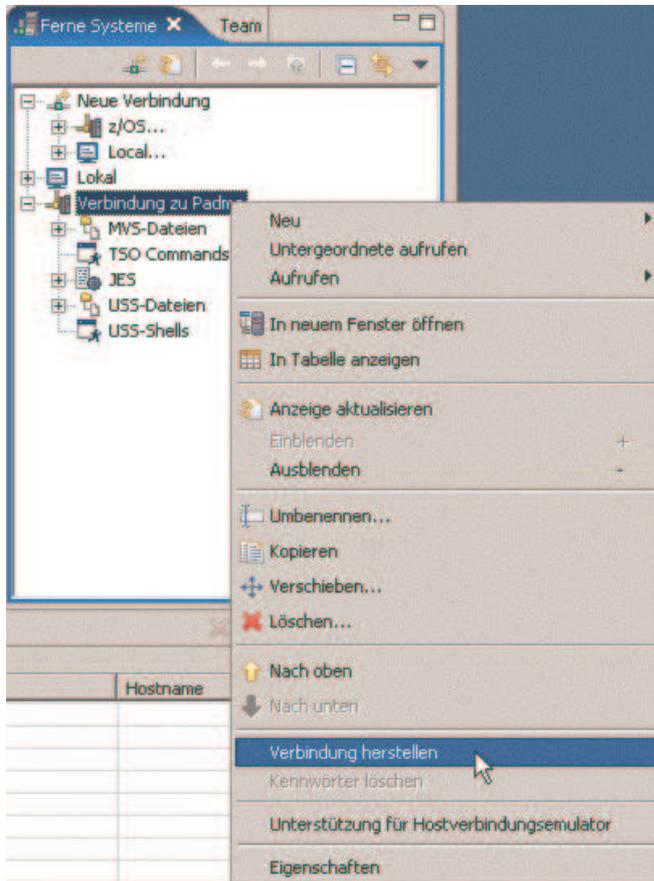


Abbildung 48

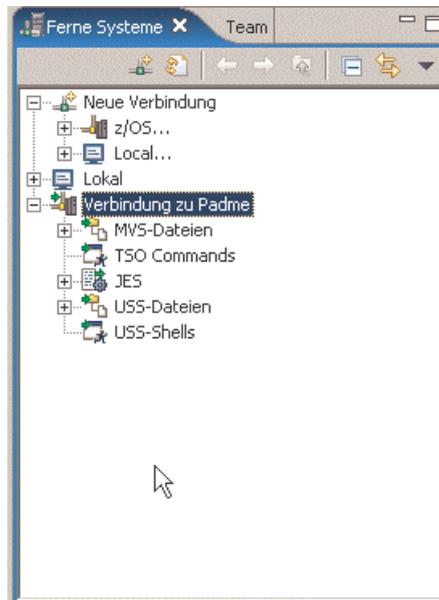
Nun sind Login-Name und Passwort des Mainframe-Logins, auf den man mittels der angelegten Verbindungsressource zugreifen möchte, einzugeben. Beide können auf Wunsch gespeichert werden.

Das Passwort darf kein Sonderzeichen enthalten.



Abbildung 49

Nach etwa 15 Sekunden sind alle Verbindungen aufgebaut. Dies erkennt man an den neu dazugekommenen kleinen hellgrünen Pfeilen:



**Abbildung 50**

### **6.3 Auflistung der Datasets**

Die Erweiterung von "MVS-Dateien" und "Meine Dateigruppen..." macht – falls vorhanden – die Datasets sichtbar.

Der Partitioned Dataset lässt sich auch erweitern, es werden seine Member sichtbar.

Ein Doppelklick auf einen Member öffnet diesen im LPEX-Editor, falls der Member lesbare Information enthält.

Nachstehende Abbildung zeigt, dass im Mainframe-Account "PRAKT11" unter anderem der PDS "PRAKT11.CICS.CTG" mit seinen beiden Membern "CTG11" und "STARTEC" angelegt sind. In der Editor-View wird der Inhalt des Members "CTG11" (Cobol-Programm-Code) angezeigt.

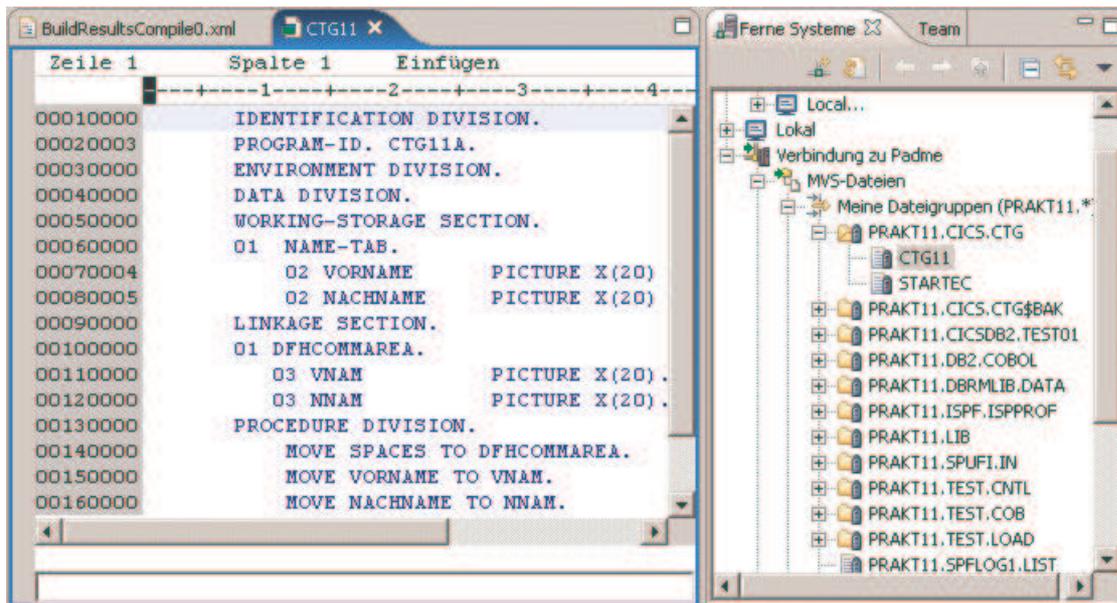


Abbildung 51

## 6.4 Auflistung der JES-Jobs

Die Erweiterung von "JES" und "<Login-Name>-JES-JOBS" macht – falls vorhanden – die von JES erzeugten Jobs sichtbar. Das sind genau die Jobs, die auch mit einem 3270-Emulator im SDSF aufgerufen werden können. Ein Doppelklick auf einen Job öffnet im LPEX-Editor seinen Inhalt, falls dieser lesbaren Text enthält.

Die nachfolgende Beispiel-Abbildung zeigt, dass es unter dem Login PRAKT11 nur einen einzigen JES-Job gibt, welcher den Jobnamen "PRAKT11C" und die Job-Nummer "06650" trägt. Der linke Teil der Abbildung enthält den lesbaren Output des Jobs.

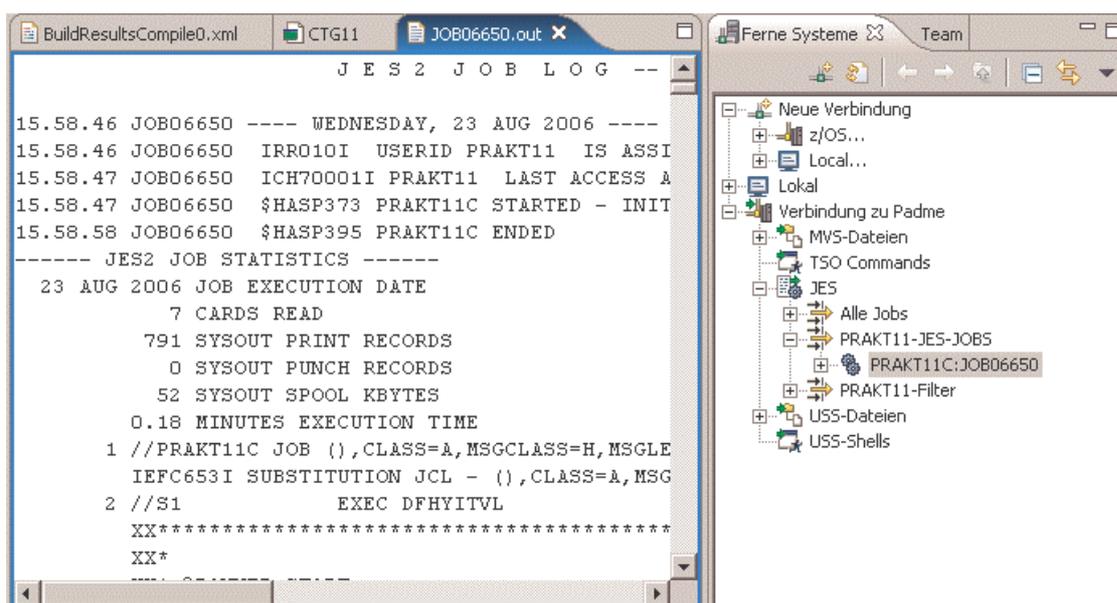


Abbildung 52

## 6.5 Auflistung der Unix System Services-Dateien

Man erweitert "USS-Dateien" und "Mein Ausgangsverzeichnis", um so – falls vorhanden – die eigenen USS-Ordner und -Dateien anzuzeigen.

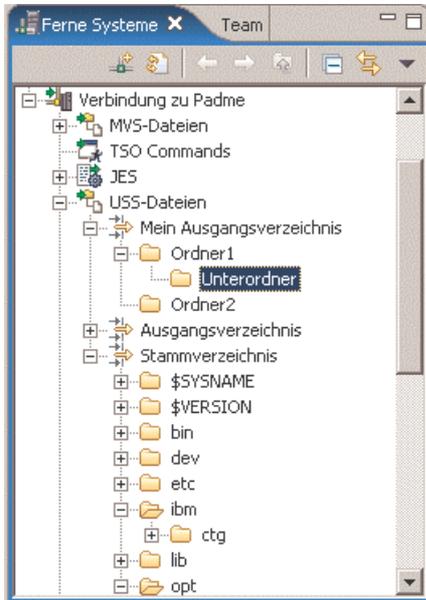
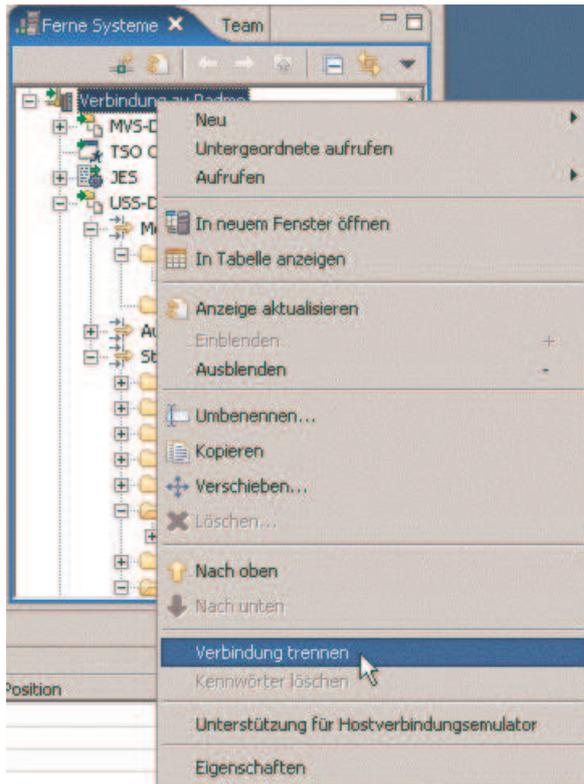


Abbildung 53

Man erweitert "Stammverzeichnis", um so die Verzeichnisstruktur unter der Wurzel sichtbar zu machen.

## 6.6 Deaktivierung der Verbindung zum Mainframe

Rechtsklick auf den Namen der Verbindungsressource (hier "Verbindung zu Padme"), im sich öffnenden Fenster Klick auf "Verbindung trennen".



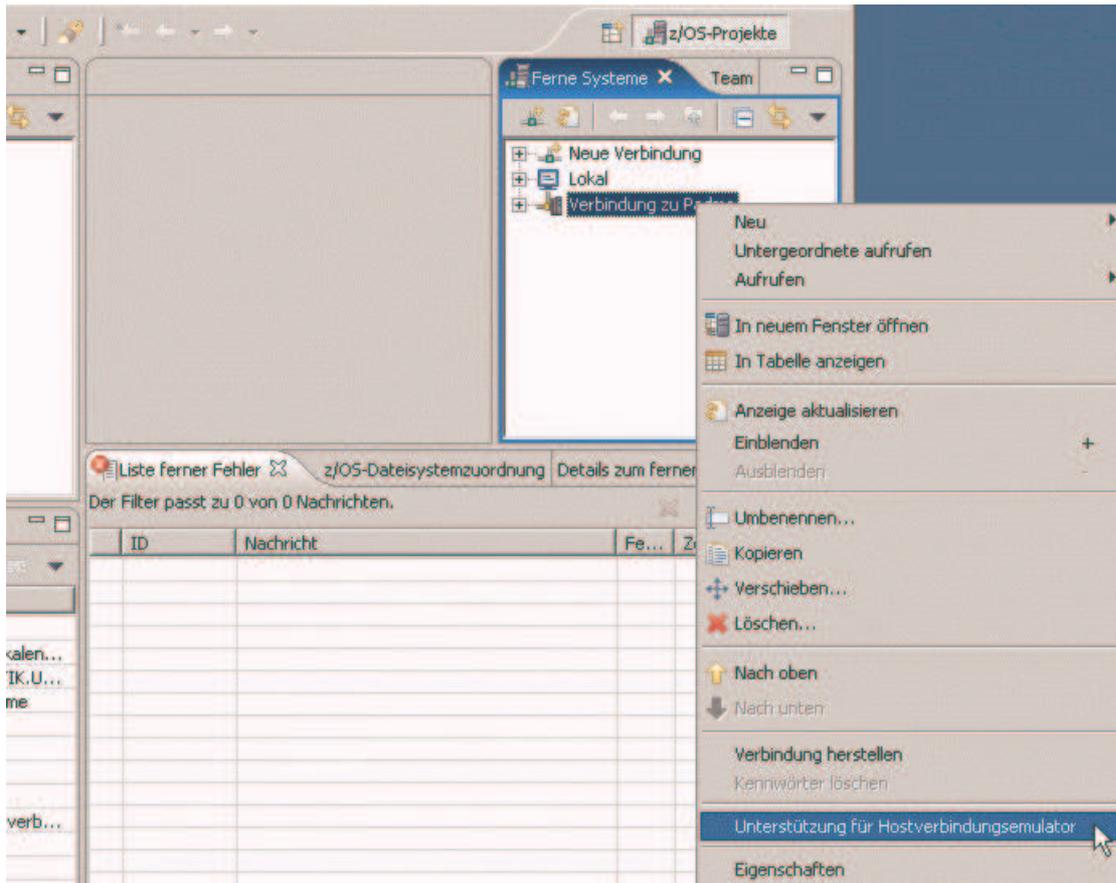
**Abbildung 54**

Sämtliche kleinen hellgrünen Pfeile verschwinden.

## 7 Emulation des 3270-Black-Screens

WDz besitzt auch einen integrierten 3270-Emulator. Nachfolgend wird dessen Aufruf demonstriert:

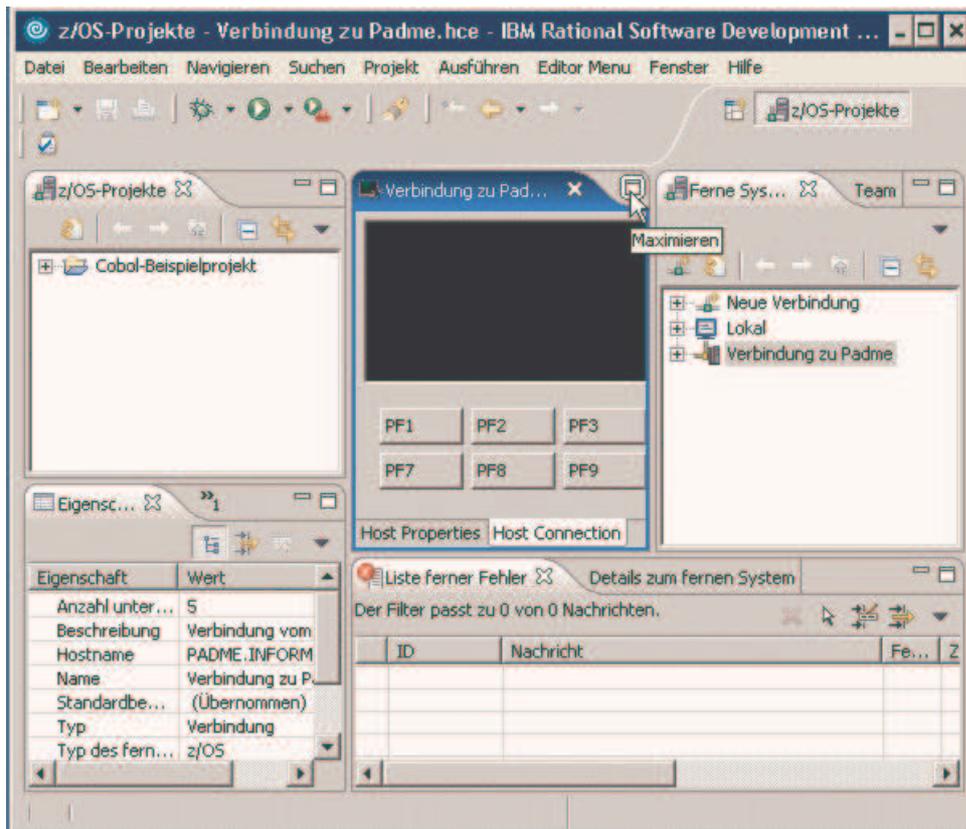
- Rechtsklick auf die Remote-Verbindung, die mit Hilfe eines 3270-Emulators aufgebaut werden soll, hier Rechtsklick auf "Verbindung zu Padme".
- "Unterstützung für Hostverbindungsemulator" auswählen



**Abbildung 55**

Es wird die Verbindung via 3270-Emulator aufgebaut. Der bekannte Black-Screen erscheint im Zentrum der WDz-Arbeitsfläche.

Um den 3270-Emulator auf die ganze WDz-Arbeitsfläche zu vergrößern, ist auf die Schaltfläche "Maximieren" zu klicken (siehe nachstehende Abbildung).



**Abbildung 56**

Mit diesem integrierten 3270-Emulator kann man die gewohnten Aktionen ausführen, z.B.

- Einloggen und Arbeiten im TSO
- Einloggen und Arbeiten im CICS

Beispielsweise kann man sich mittels "L TSO", Mainframe-Login und Passwort ins TSO einloggen sowie mittels "L CICS", Mainframe-Login und Passwort ins CICS.

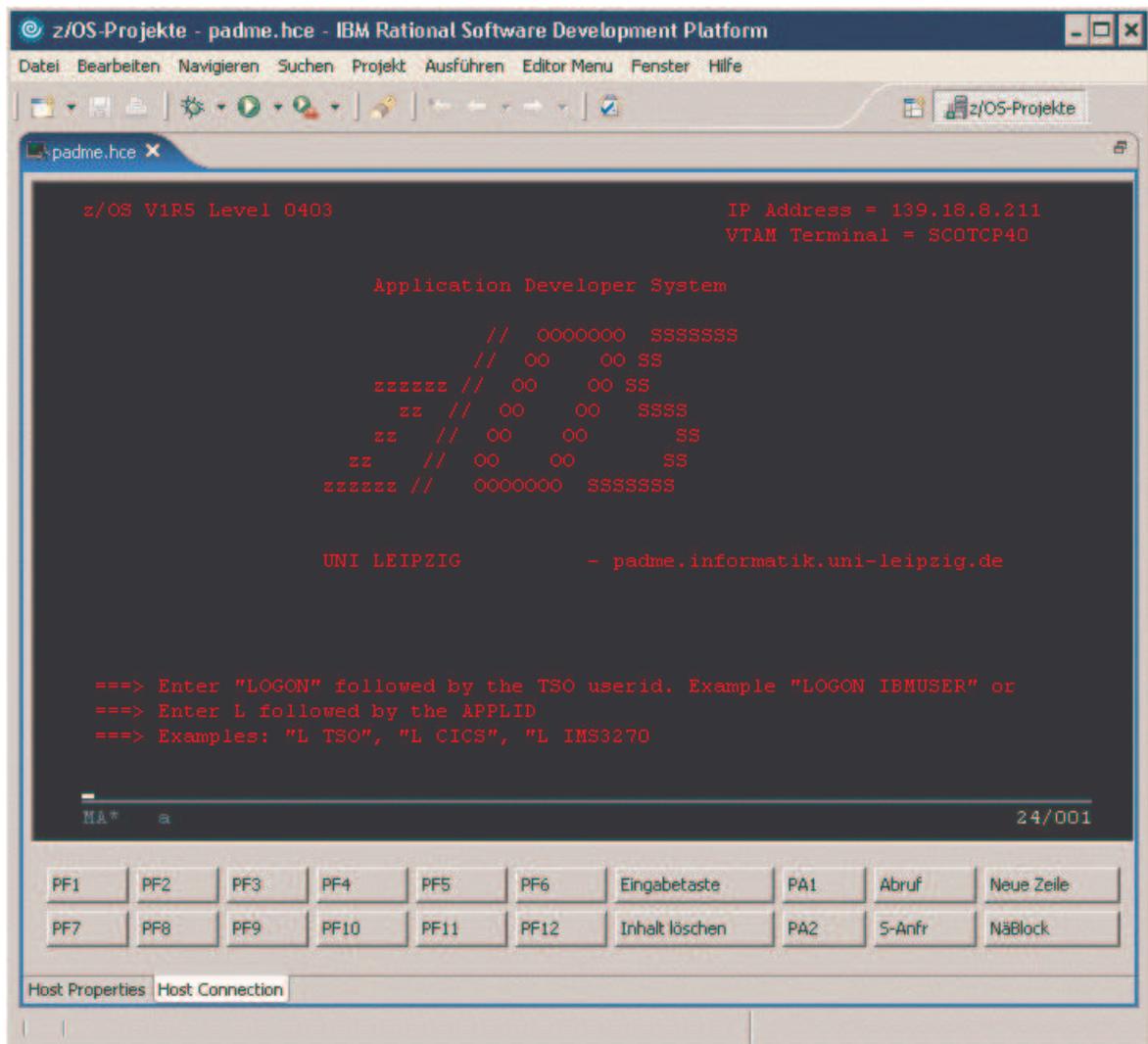


Abbildung 57

## 8 WDz beenden sowie Ausloggen aus dem entfernten XP-Server

WDz wird wie folgt beendet: Datei → Beenden

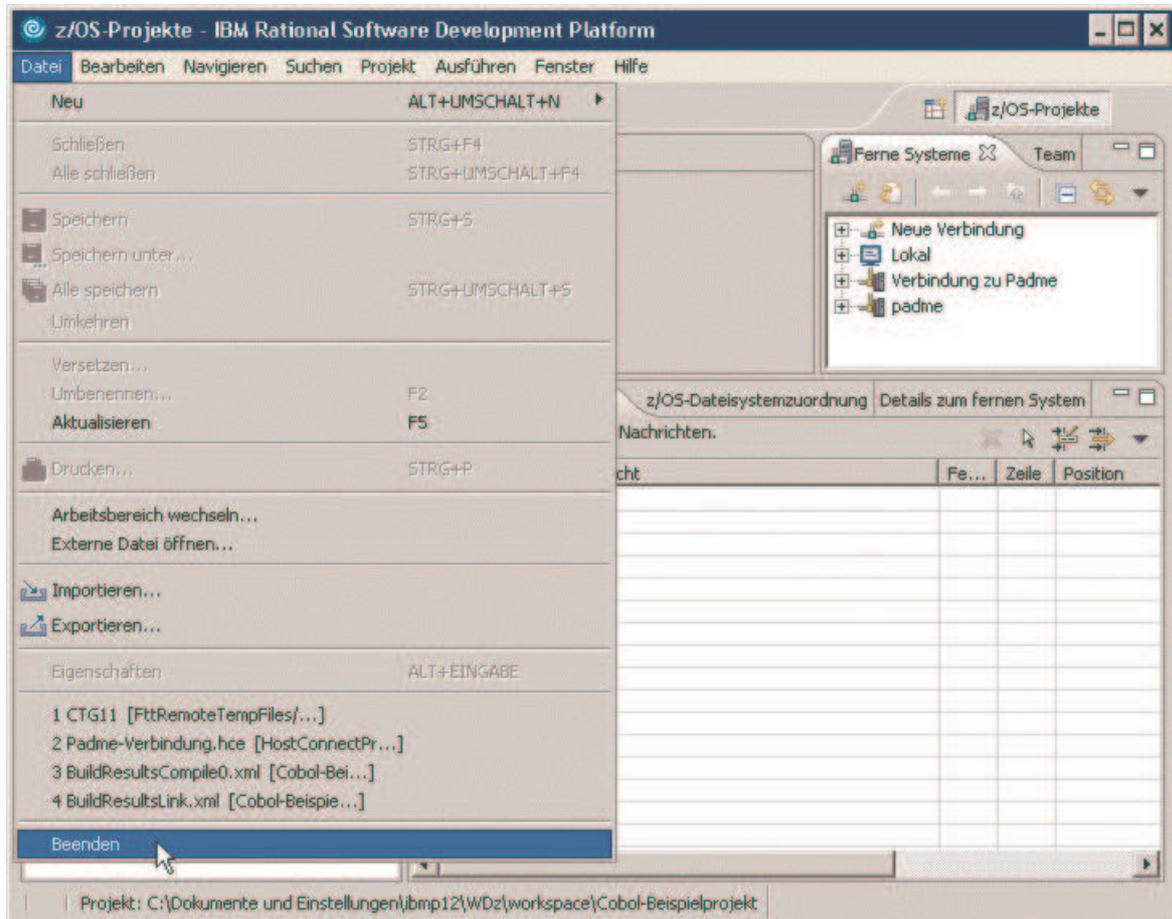
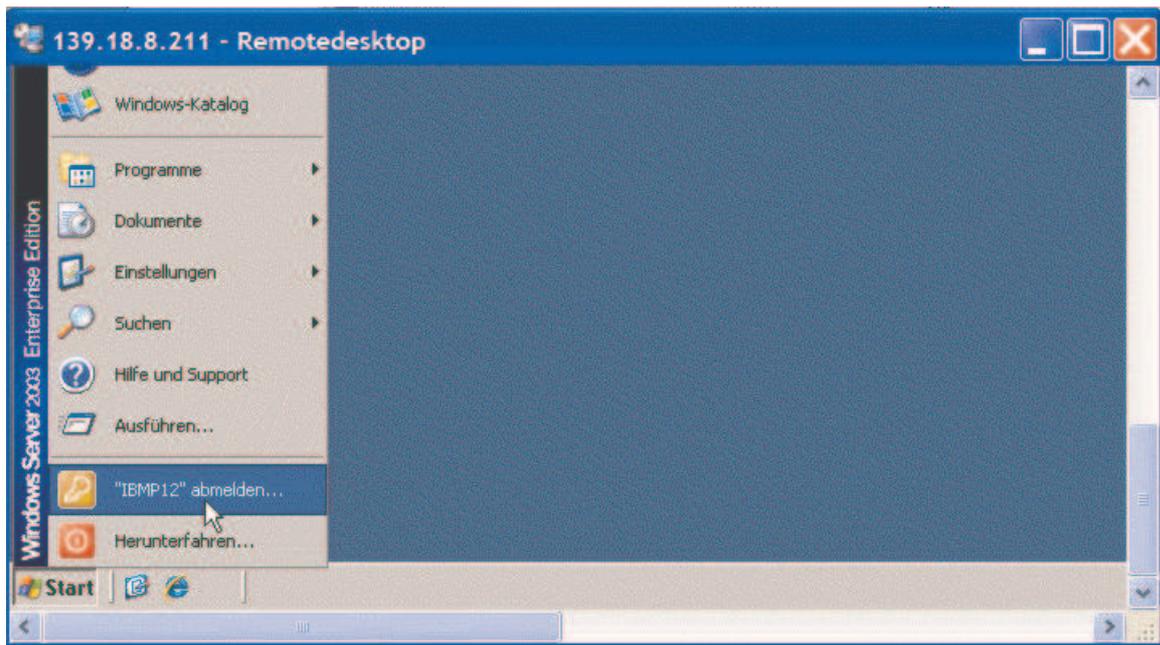


Abbildung 58

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

START → "IBMP<xx> abmelden..."



**Abbildung 59**

Es ist noch die Frage, ob man sich wirklich abmelden möchte, mit einem Klick auf den Button "Abmelden" zu beantworten.

Das Fenster mit der Remote Desktop Verbindung wird geschlossen.

Aufgabe: Legen Sie ein zweites Projekt an ("Workstation COBOL oder PL/I"-Projekt). Es soll den Namen "Uebung-1-<xxx>" erhalten, wobei "<xxx>" für Ihre dreistellige Mainframe-Login-Nummer steht.

Aufgabe: Nutzen Sie zur Bearbeitung der Aufgaben dieses Tutorials den LPEX-Editor als ISPF-Editor.

Aufgabe: Schreiben Sie im Projekt "Uebung-1-<xxx>" ein einfaches Cobol-Programm. Nutzen Sie zum Erstellen des Programm-Codes den Code-Assistenten. Bezeichnen Sie die Cobol-Programm-Code-Datei mit "Uebung-1-<xxx>.cbl".

Aufgabe: Führen Sie einen lokalen Syntax-Check durch und prüfen Sie so die syntaktische Korrektheit des Cobol-Programms. Sollte die Syntaxprüfung Fehler aufdecken, sind diese Fehler natürlich zu beseitigen.

Aufgabe: Erstellen Sie einen Screenshot (siehe Abbildung 60), welcher einen Code-Auszug Ihres Cobol-Programm zeigt sowie die fehlerfreie lokale Syntax-Prüfung demonstriert (Fenster "Liste ferner Fehler" ist leer). Auf dem Screenshot muss sich unter der View "z/OS-Projekte" der Projektname "Uebung-1-<xxx>" und der Dateiname "Uebung-1-<xxx>.cbl" befinden. Der Screenshot darf eine Größe von 290 Kilobytes nicht überschreiten.

Aufgabe: Legen Sie zu Übungszwecken eine zweite Verbindungsressource zu Ihren Mainframe-PRAKT-Account hin an. Diese soll ebenfalls den Namen "Uebung-1-<xxx>" erhalten.

Aufgabe: Aktivieren Sie die Verbindungsressource "Uebung-1-<xxx>".

Aufgabe: Legen Sie – unter Nutzung von WDz – einen weiteren Dataset an. Dieser soll den Namen "<Ihre User-ID>.WDZ.UEBUNG1" erhalten. Die Allocate-Parameter sind so zu wählen, dass der Dataset maximal 99 Kilobytes an Plattenplatz belegt.

Aufgabe: Erweitern Sie in der View "Ferne Systeme" unter Ihrer Verbindungsressource "Uebung-1-<xxx>" "MVS-Dateien", so dass Ihr gerade neu angelegter Dataset angezeigt wird. Markieren Sie diesen Dataset, so dass die View "Eigenschaften" die Allocate-Parameter des Datasets zeigt. Erzeugen Sie davon einen weiteren Screenshot (siehe Abbildung 61),, welcher die View "Ferne Systeme" mit Ihrem gerade angelegten Dataset "<Ihre User-ID>.WDZ.UEBUNG1" enthält. Ferner soll die View "Eigenschaften" mit den Allocate-Parametern auch mit auf dem Screenshot zu sehen sein. Dieser Screenshot darf ebenfalls nicht größer als 290 Kilobytes groß sein.

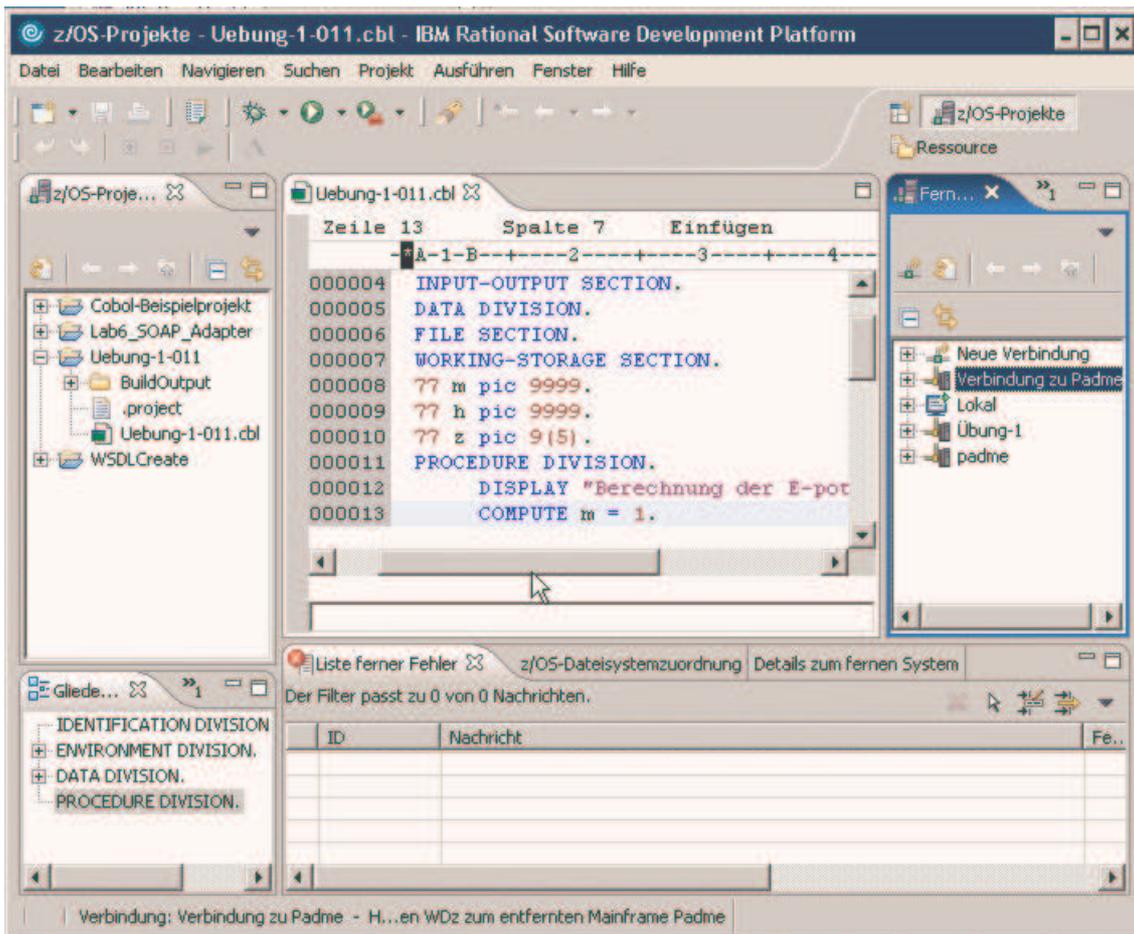


Abbildung 60

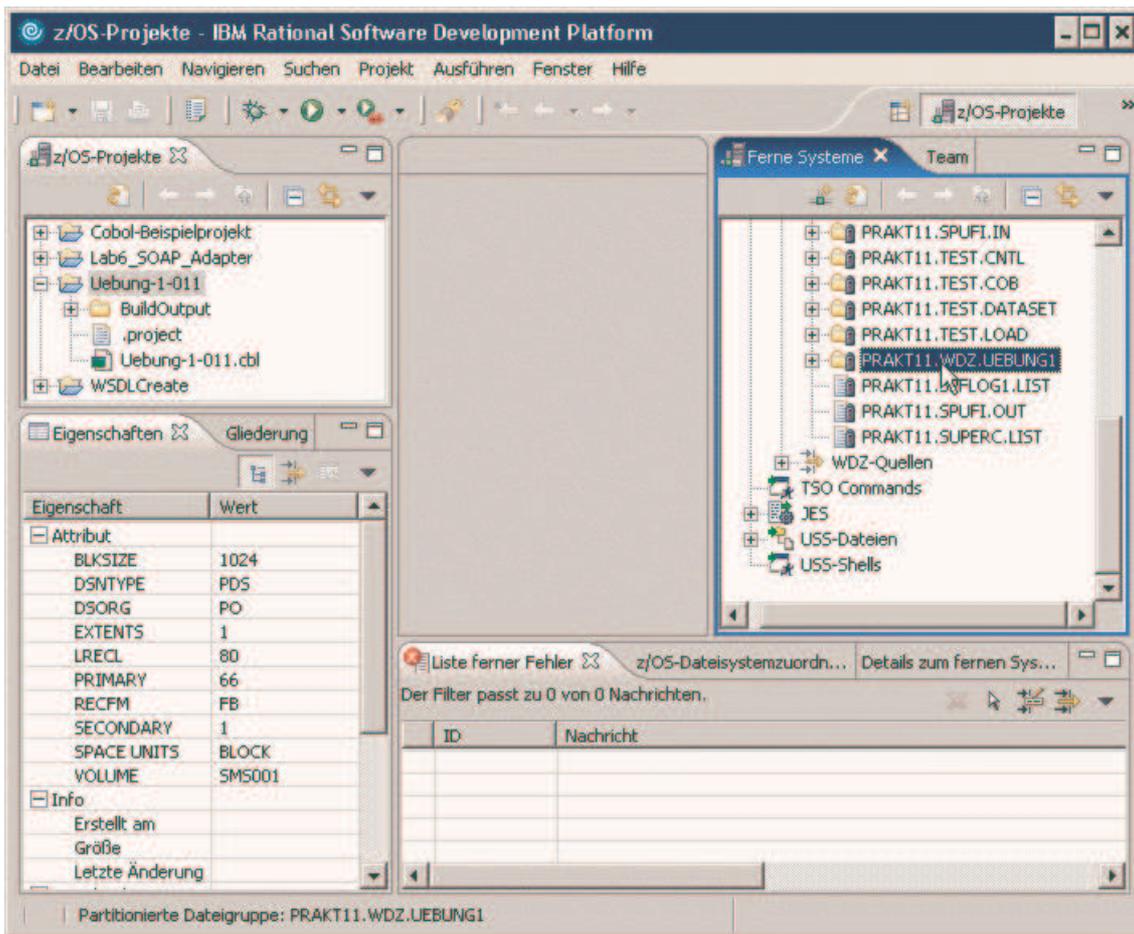


Abbildung 61