

# RDz Tutorial 03

## Remote COBOL

© Abteilung Technische Informatik, Institut für Informatik, Universität Leipzig  
© Abteilung Technische Informatik, Wilhelm Schickard Institut für Informatik,  
Universität Tübingen  
Version 01, November 2012

**Dieser Text wurde ohne den RDZ Server mit einer Konfiguration erstellt, bei der RDz unmittelbar auf der Workstation installiert war. Als Folge können bei der Benutzung des RDz Servers einige Referenzen auf die Workstation anders aussehen. Siehe z.N. Abb. 2.1.3, 2.2.3, 3.3.2, 4.3.10, usw. Es ist deshalb erforderlich, den RDz Tutorial Text 03 nochmals unter Benutzung des RDz Servers durchzuspielen.**

## Übersicht

In diesem Tutorial werden Sie unter Benutzung von RDz eine Cobol Anwendung auf einem System z Mainframe entwickeln. Sie definieren eine remote z/OS-Verbindung, setzen ein MVS-Projekt auf und führen ein remote Bearbeiten, Kompilieren und Debuggen durch.

Das Tutorial besteht aus den folgenden Teilaufgaben:

1. Verbindung zu einem Remote-System herstellen
  - 1.1 Remote-Verbindung definieren
  - 1.2 Verbindung herstellen
  - 1.3 Aktivierung einer Verbindungs-Ressource
  - 1.4 Auflistung der Datasets
  - 1.5 Auflistung der JES-Jobs
  - 1.6 Auflistung der Unix System Services-Dateien
  - 1.7 Deaktivierung der Verbindung
2. Parameter einstellen
  - 2.1 z/OS System-Einstellungen
  - 2.2 Benötigte z/OS Datasets erstellen
  - 2.3 Host Code Page einstellen
  - 2.4. Emulation des 3270 Green Screens
- 3.0 Arbeiten mit MVS files
  - 3.1 Allocating a Data Set
  - 3.2 Mapping Data Sets
  - 3.3 Kopieren lokaler Files in einen allocated z/OS Data Set
4. Arbeiten mit entfernten Data Sets
  - 4.1 Wiederholung
  - 4.2 Arbeiten mit Projekten und Subprojekten
  - 4.3 Ein MVS Subproject erstellen
  - 4.4 Ressourcen dem Projekt hinzufügen
  - 4.5 Compile/Link/Execute des entfernten Cobol Programms.

# 1. Vorbereitungen für die RDz Remote-Entwicklung

Bevor wir anfangen zu programmieren, müssen einige Vorbereitungen durchgeführt werden.

Zunächst werden Sie eine Verbindung zu einem Remote-System herstellen, welches für das Arbeiten mit RDz eingerichtet wurde. Sie werden lernen, wie Sie Ihre Daten mithilfe von Filtern organisieren, und wie Sie Data Sets mit RDz zuordnen. Sie werden ein JCL Script ausführen, um benötigte Data Sets zu erstellen.

Details zum Einloggen in den Remote RDz Server und zum Starten von RDz sind im RDz-Tutorial 01 Abschnitt 3 beschrieben.

## 1.1 Verbindung zu einem Remote-System definieren

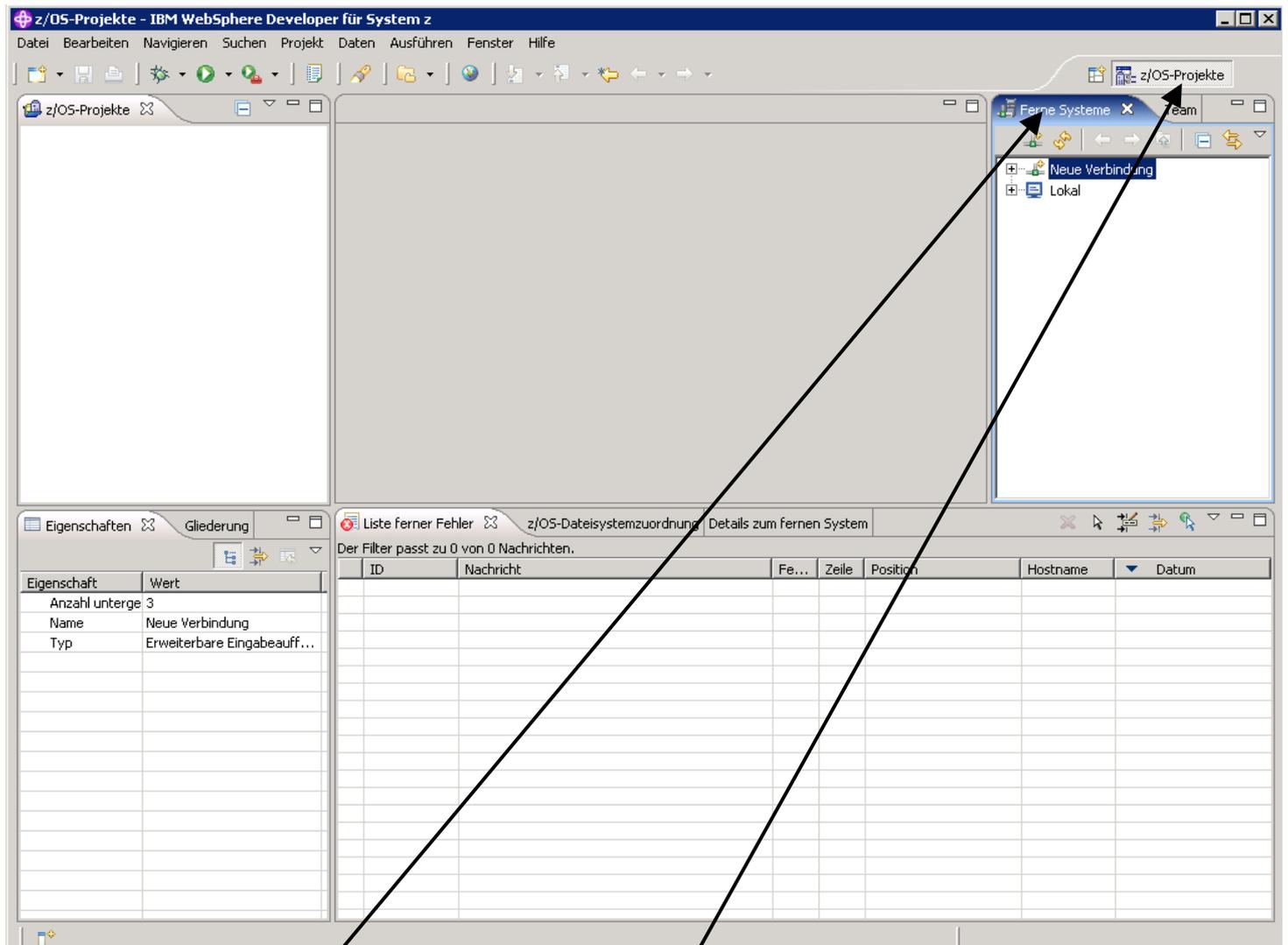


Abbildung 1.1

Stellen Sie sicher, dass die z/OS Projekte Perspektive eingeschaltet ist.

Der Remote Systems View zeigt alle bestehenden Verbindungen zu entfernten-Systemen an.

Verbindungen sind persistente System Connection Objekte. Sie enthalten die erforderlichen Informationen, um auf einen bestimmten Remote-Host zuzugreifen. Der Remote Systems View ermöglicht Pop-up-Menü Aktionen für das Umbenennen, Kopieren, Löschen und neu Anordnen von bestehenden Verbindungen.

Verbindungen enthalten Attribute oder Daten, die zwischen Workbench Sitzungen gespeichert werden. Diese Attribute sind der Verbindungsname des Remote-Systems, eine optionale Beschreibung, und eine Benutzer-ID, die beim Verbindungsaufbau standardmäßig von jedem untergeordneten z/OS Subsystem als Default verwendet wird. Darunter sind alle Verbindungen als Dateien in einer Eclipse-Projekt namens RemoteSystemsConnections untergebracht. Der Benutzer kann sie für eine Team-Unterstützung aktivieren. Hiermit können Verbindungen von einem Team gemeinsam genutzt (shared) werden.

In diesem Beispiel benutzt der Mainframe Host entweder

- den Namen frodo, und die IP Adresse 134.2.205.54, Port 423, oder
- den Namen leia und die IP adresse 139.18.4.30, port 23.

Die hier verwendete Benutzer-ID ist prak032. Bitte ersetzen Sie alle Verweise mit dem von Ihnen benutzten Mainframe-Host und Ihrer Benutzer-ID.

Im Tutorial verwendete Parameter

	Wert	Kommentar
User-ID	prak<xxx>	Im Tutorial benutzte User ID
Passwort	?????????	Erhalten sie vom Betreuer
IP Adresse	134.2.205.54, port 423 139.18.4.30, port 23	Telnet 3270-Emulator Telnet 3270-Emulator
Ports	139.18.4.30:4035 139.18.4.30:4035 139.18.4.30: 6715	MVS Port Entfernter Dämon JES Job-Monitor
Verknüpfungsbibliotheken	CEE.SCEELKED	Für den Verknüpfungsschritt erforderliche Bibliothek

Jeder Übungsteilnehmer erhält vom Betreuer eine User-ID mit einer ihn eindeutig zugeordneten Nummer. Im Tutorial ist prak<xxx> bzw. prak032 überall durch diese Nummer zu ersetzen.

Wenn Sie nicht unseren RDz Server benutzen, sondern eine Kopie von RDz 7.0 auf Ihrer Workstation (z.B. unter VMWare) installiert haben, stellen Sie sicher, dass auf Ihrer Workstation der Firewall sowohl auf Ihrem VMware Host und auf Ihrem Gastcomputer deaktiviert ist. Zumindest müssen alle Verbindungen von/zu 134.2.205.54 oder 139.18.4.30 offen sein.

## 1.2 Remote-Verbindung zu einem Mainframe herstellen

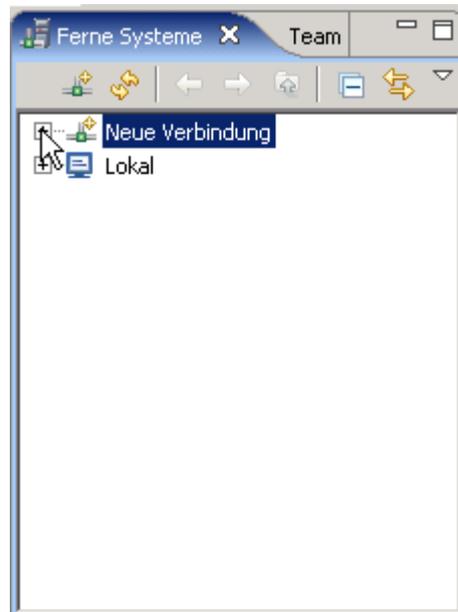


Abbildung 1.2

In der View "Ferne Systeme" (remote Systems) "Neue Verbindung" erweitern.

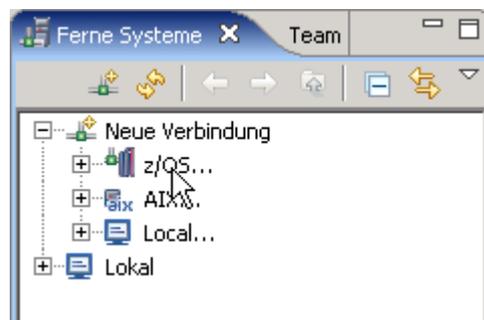


Abbildung 1.3

Rechtsklick auf "z/OS..." und

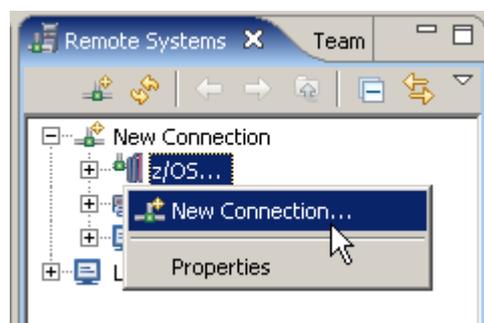
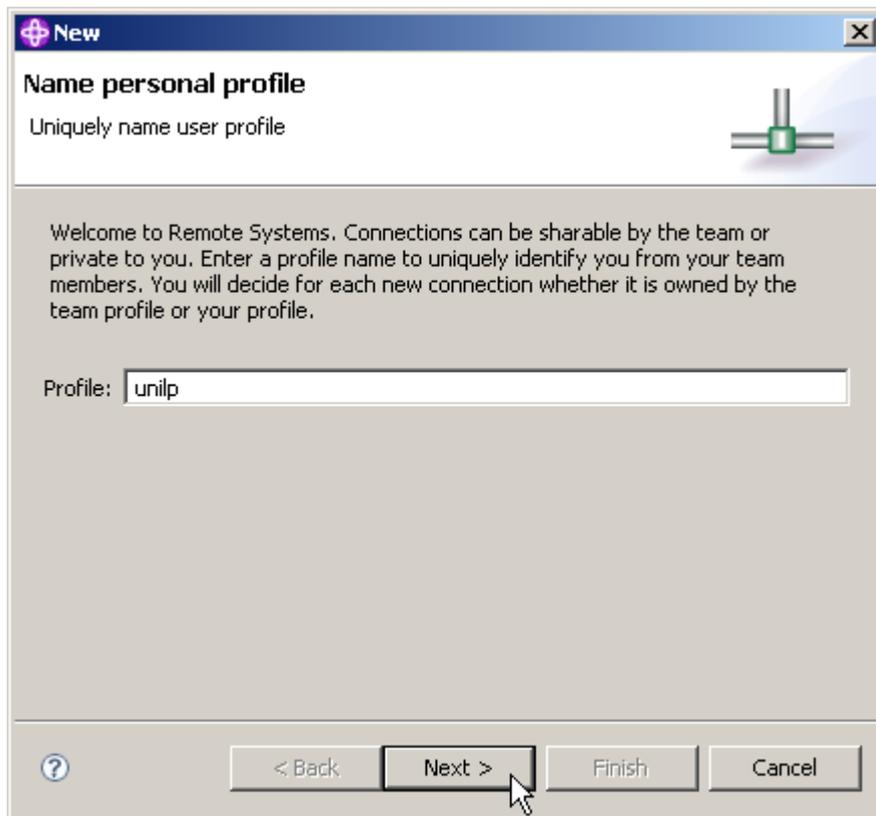


Abbildung 1.4

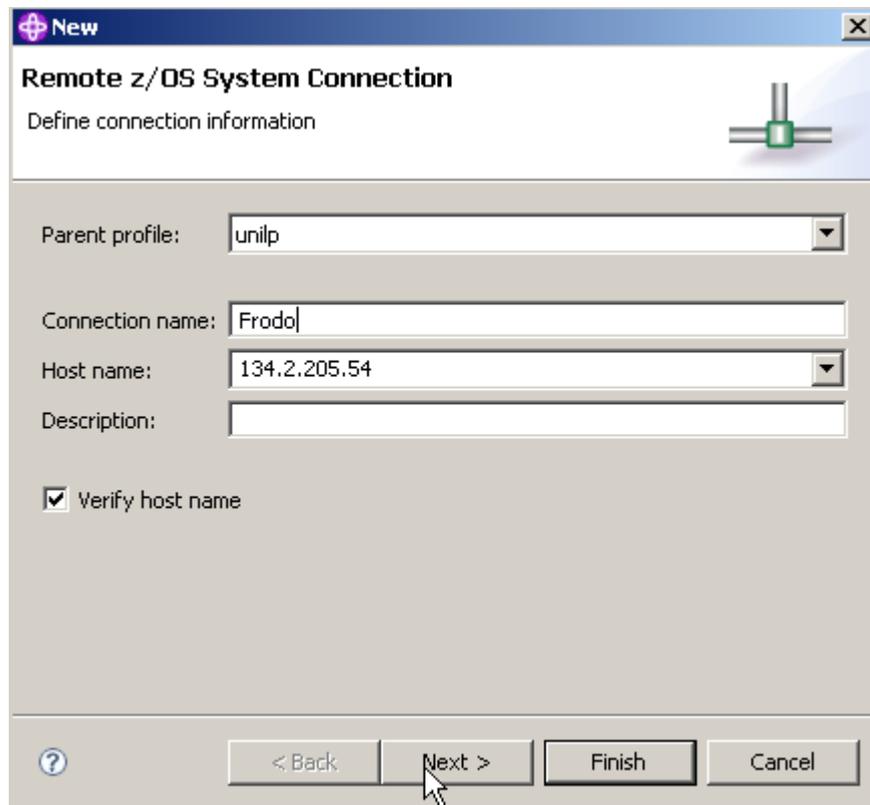
"Neue Verbindung" auswählen



**Abbildung 1.5**

**Wenn dies der erste Versuch ist eine Verbindung zu erstellen, werden Sie aufgefordert, ein Profil zu erstellen. Nennen Sie Ihr Profil „unilp“ und klicken Sie auf Weiter.**

**Hinweis: Wenn der Bildschirm nicht angezeigt wird gibt es bereits ein vorhandenes Profil, und Sie können diesen Schritt ignorieren.**



**Abbildung 1.6**

**Die anzulegende Verbindung benötigt einen eindeutigen Namen, z.B. Frodo.**

**Als Hostnamen die IP-Adresse "134.2.205.54" eintragen.**

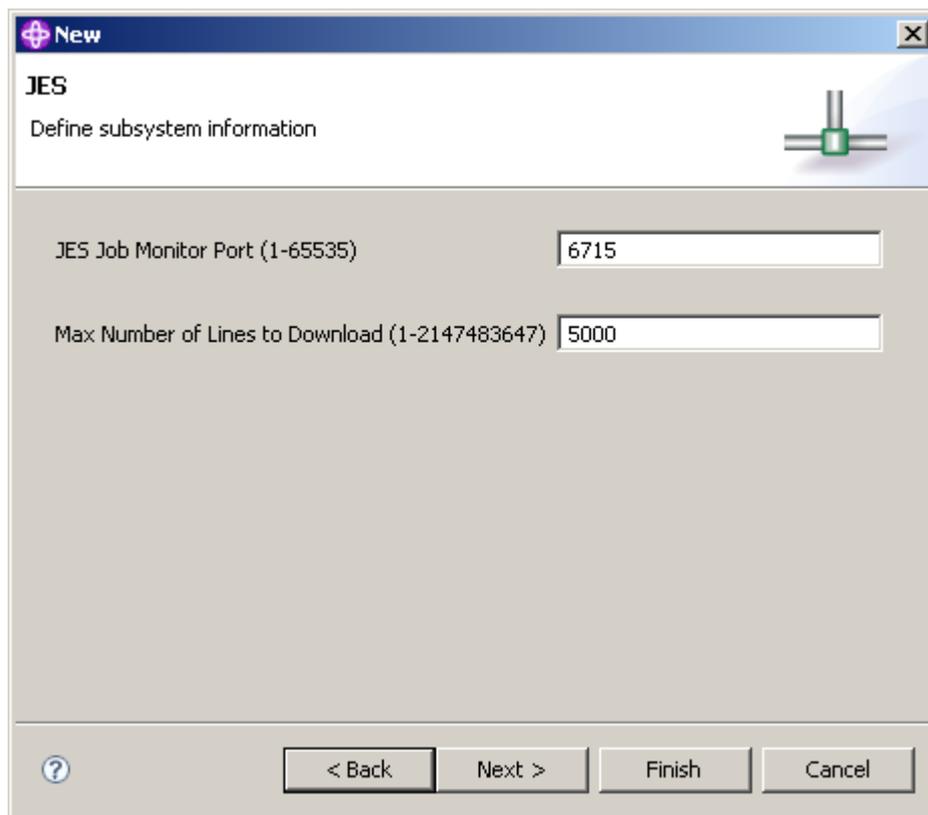
**Als "Beschreibung" kann ein beliebiger Text eingetragen werden, z.B. "Verbindung zu frodo.informatik.uni-tuebingen.de"**

**Wird ein Häkchen vor "Hostnamen prüfen" gesetzt, wird die Korrektheit des Hostnamens geprüft.**

**Frodo braucht Port 423 statt Port 23**

**Alternativ können Sie auch mit dem System leia.informatik.uni-leipzig.de oder 139.18.4.30 unter Port 23 arbeiten.**

**Klick auf "Weiter"**



**Abbildung 1.7**

**Im JES properties Panel, Port = 6715 eingeben. Klick auf "Weiter"**

**Wenn Sie einen anderen z/OS Server als frodo oder leia verwenden, werden diese Portnummern unterschiedlich sein.**

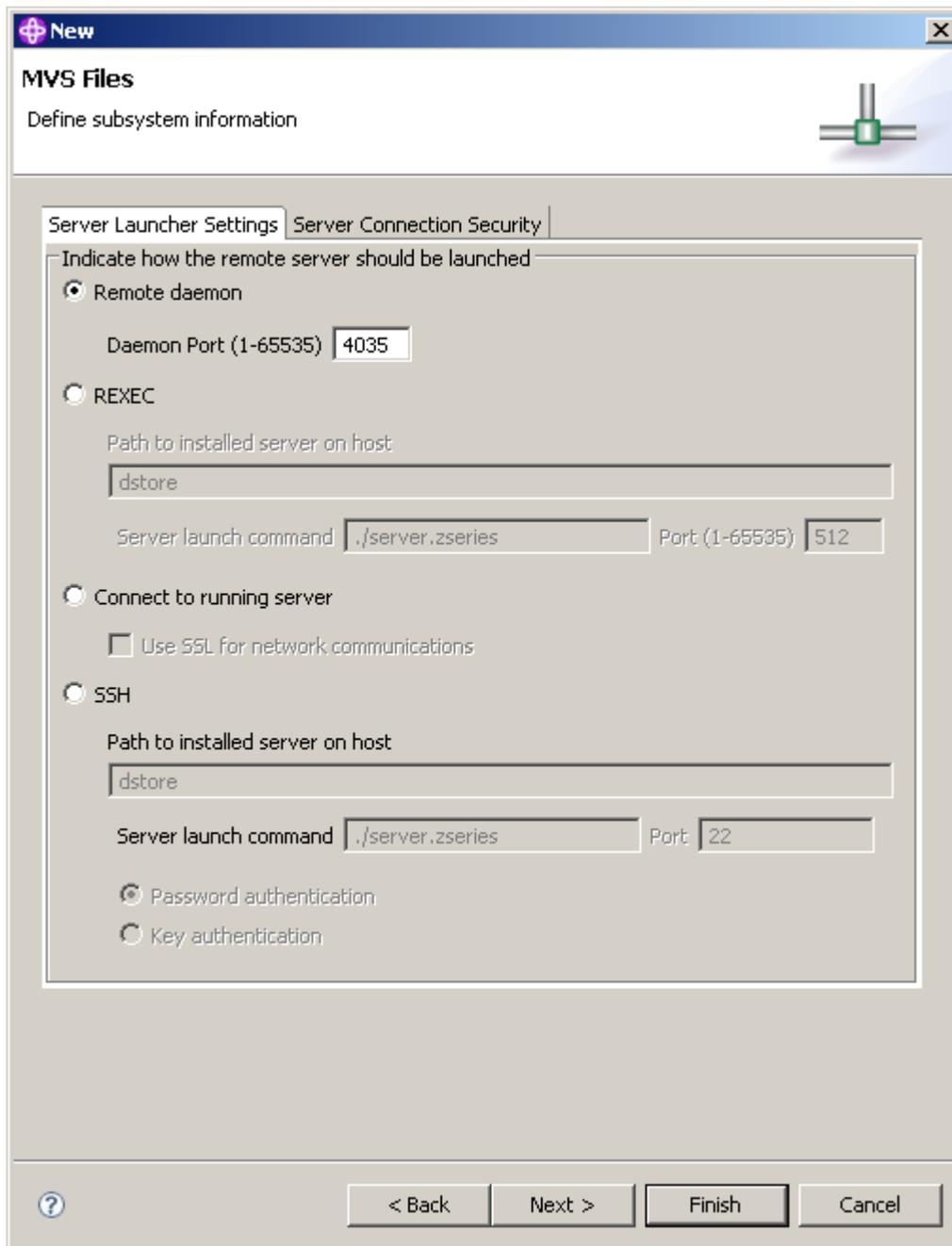


Abbildung 1.8

Im MVS Files Panel, den Defaults Port 4035 unverändert lassen. Klick auf "Weiter"

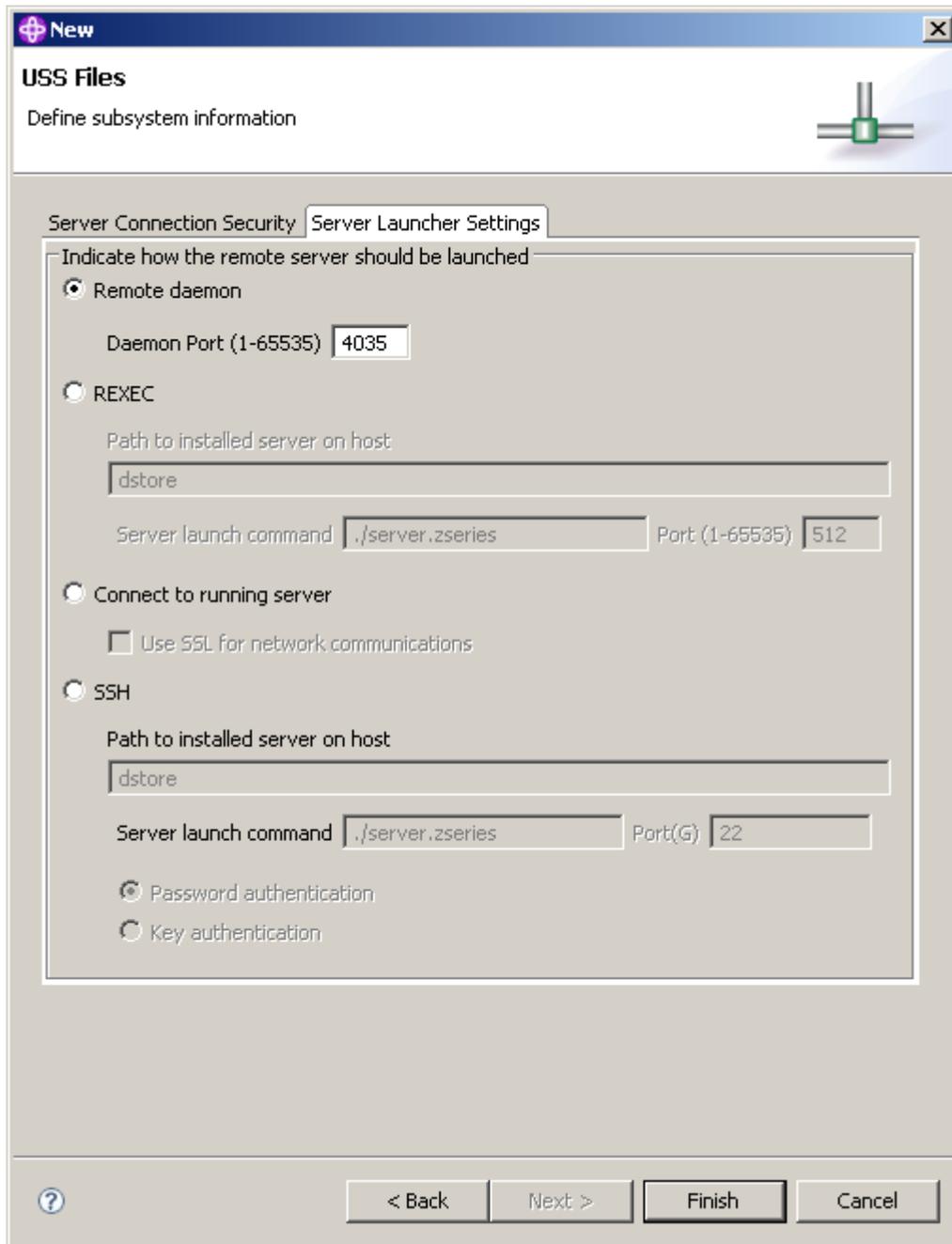


Abbildung 1.9

Im USS Files Panel, 1k auf **Server Launch Settings**. Verifizieren Sie **Daemon Port 4035**, lassen Sie die **Default Werte** unverändert, 1k auf **Finish**. Dies erstellt die neue **z/OS-Verbindung** und fügt sie in der **Remote Systems Perspektive** ein.

### Selbst-Test

- Warum müssen wir eine Verbindung erstellen, ehe wir uns einloggen ?
- Es muss zunächst eine Verbindung zu dem von Ihnen benutzten z/OS System aufgebaut werden. Diese benutzt standardmäßig port 23 (Port423 im Falle von Frodo).? Wozu braucht man dann die extra Ports für JES, MVS Files und USS ?

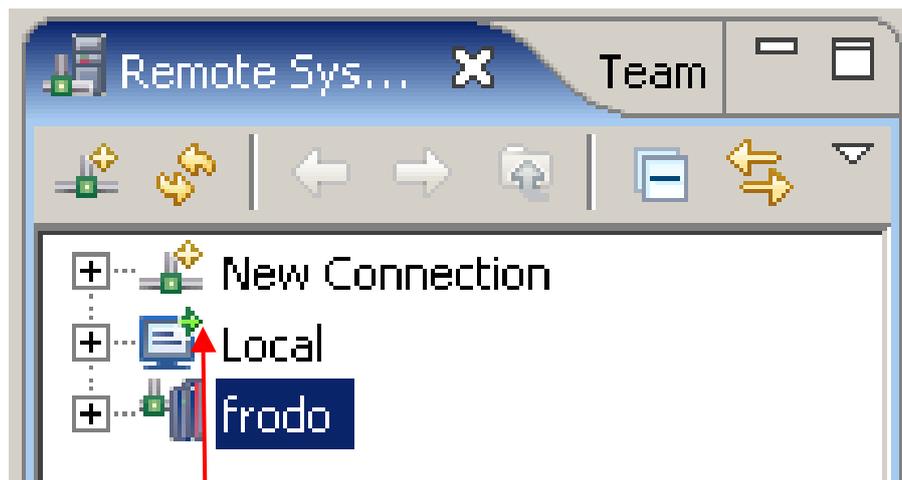


Abbildung 1.10

Nun ist in dem "Remote System"-View die Ressource "frodo" hinzugekommen.

Beachten Sie: "Local" bezieht sich auf die RDz Funktionen, die unter Windows verfügbar sind. „frodo“ bezieht sich auf die RDz Funktionen, die unter dem z/OS System 134.2.205.54 potentiell verfügbar sind. Auf die „Local“ Funktionen kann unmittelbar zugegriffen werden. Dies wird durch einen kleinen grünen Pfeil gekennzeichnet. Bei „frodo“ fehlt der kleine grüne Pfeil, weil zunächst noch ein Logon zu frodo vorgenommen werden muss.

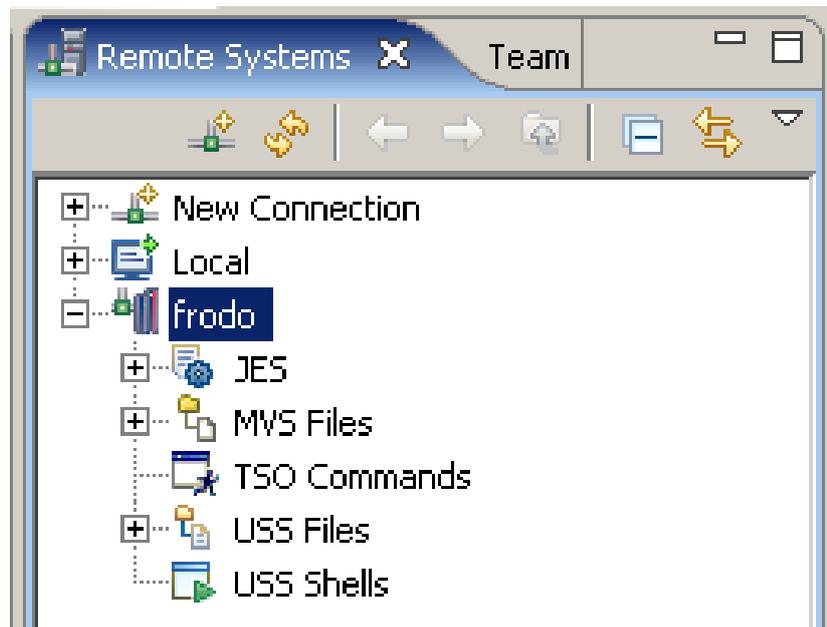


Abbildung 1.11

Expandieren Sie frodo. Es werden die die Ressourcen JES, MVS Dateien, TSO Commands, USS Dateien und USS Shells angezeigt. Diese angelegten Ressourcen sind zwar verfügbar, aber noch sind die Verbindungen zu diesen Ressourcen inaktiv.

Diese Verbindung stellen wir als Nächstes her, indem wir uns über RDz auf Frodo mit unserer User ID und unerem Passwort einloggen.

### 1.3 Aktivierung einer Verbindungs-Ressource

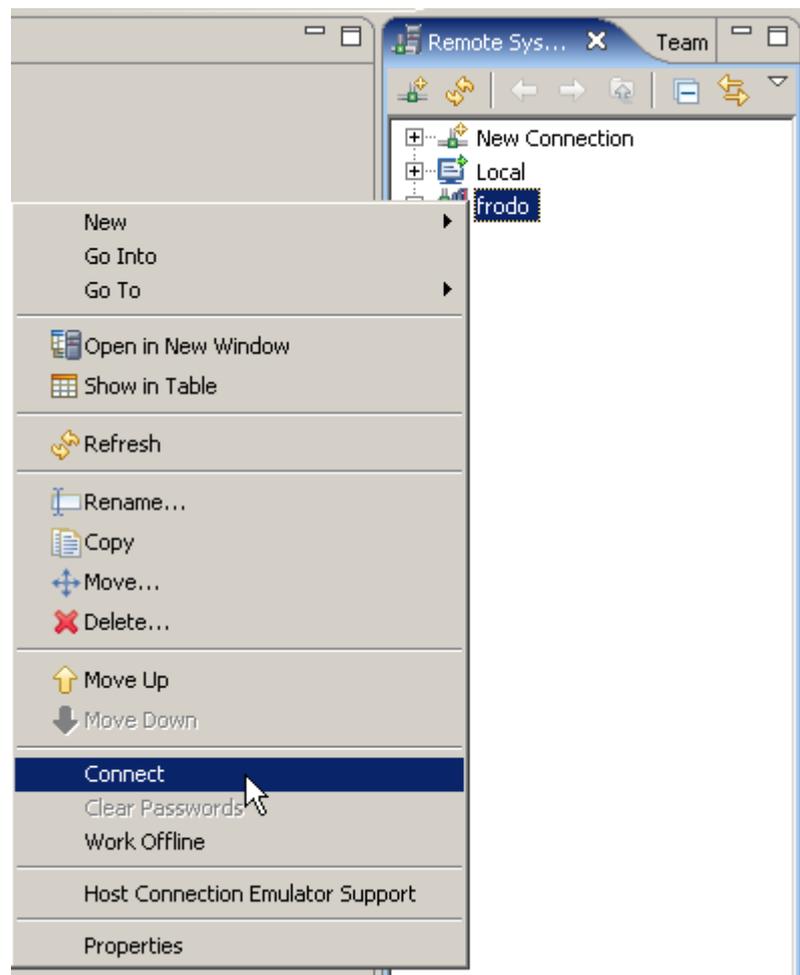


Abbildung 1.12

Rechte Maustaste auf "frodo". Im geöffneten Popup-Fenster Klick auf "Verbindung herstellen".

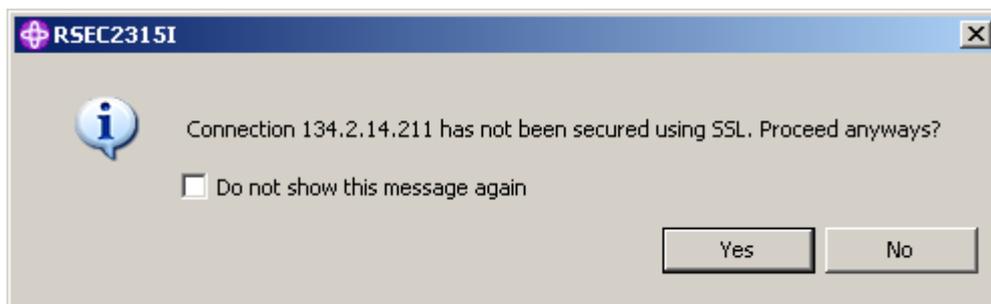


**Abbildung 1.13**

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

**Das Passwort darf kein Sonderzeichen enthalten.**

**Click OK um die Verbindung zu frodo herzustellen.**



**Abbildung 1.14**

**Natürlich.**

**Nach etwa 15 Sekunden sind alle Verbindungen aufgebaut.**



Abbildung 1.15

Wenn die Verbindung zum Remote-System erfolgreich hergestellt wurde, wird neben dem Symbol für Frodo ein kleiner grüner Pfeil sichtbar, ähnlich wie der kleine grüne Pfeil neben dem Symbol für Local.

**Wichtig:** Sie sind nun mit Ihrer Benutzer-ID auf dem z/OS Rechner verbunden. Wenn Sie die RDz Session beenden wollen, müssen Sie vorher ein z/OS Logout durchführen! Das Verfahren ist das Gleiche wie bei der Verbindung mit dem System. Der kleine grüne Pfeil wird dann wieder verschwinden.

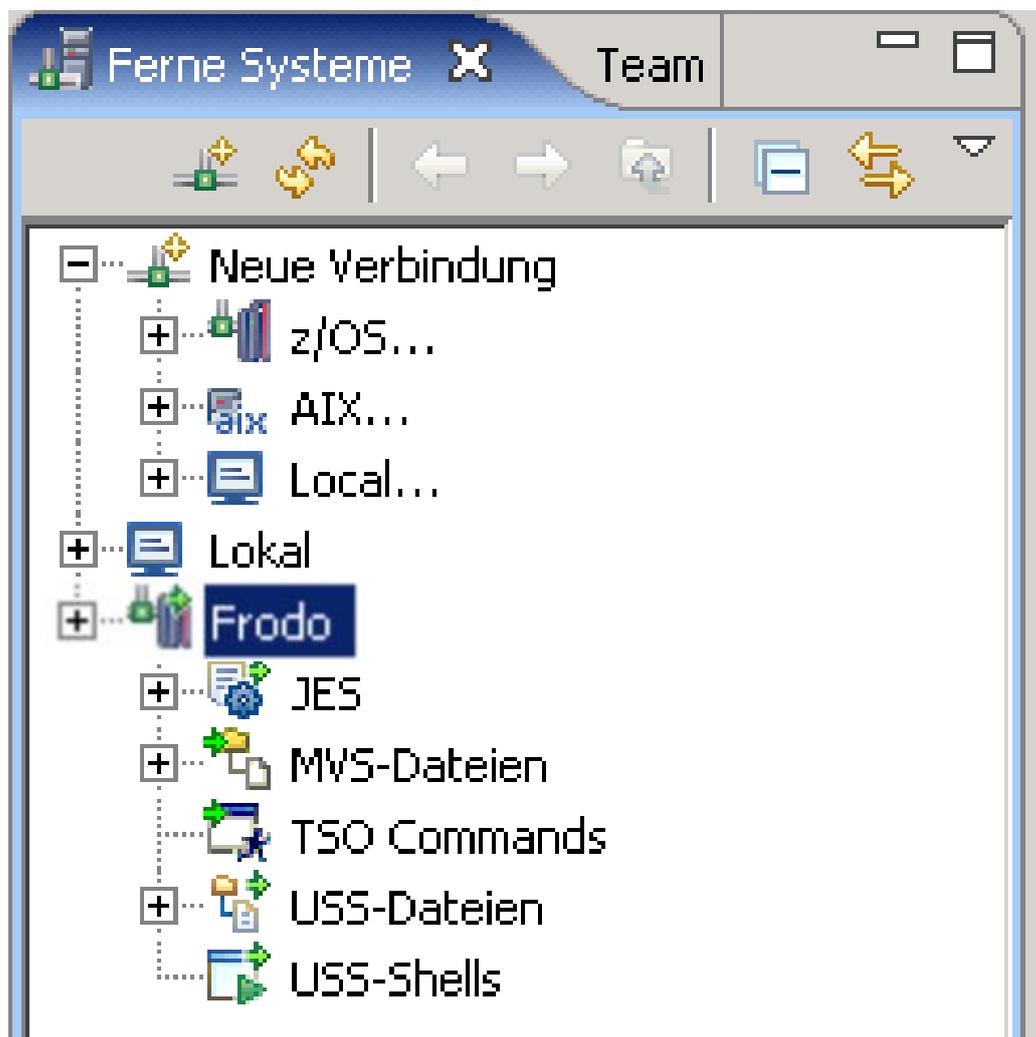


Abbildung 1.16

Wir hoffen, Sie haben gute Augen. Im Unterschied zu Abbildung 1.11 sind die kleinen grünen Pfeile hinzugekommen. Sie bedeuten, dass sowohl der Rechner frodo, als auch seine Ressourcen (spezifisch JES, MVS Dateien, TSO Commands, USS Dateien und USS Shells) mit ihrem Workspace verbunden sind. Sowohl frodo, als auch alle Ressourcen unter frodo, müssen den grünen Pfeil aufweisen, damit man damit arbeiten kann.

Wenn ein grüner Pfeil fehlt (z.B. bei JES) haben Sie ein Problem. Im Falle von JES kann die Ursache sein, dass Sie in Abb. 1.7 die falsche Port Nr. eingegeben haben. Wenn Sie das Problem nicht lösen können, wenden Sie sich an Ihren Betreuer.

Sie können unseren RDz Server benutzen, um mit unterschiedlichen Mainframe Rechnern zu arbeiten. Dies kann z.B. 139.18.4.30, eine zusätzliche unterschiedliche LPAR (virtuelle Maschine) von [jedi.informatik.uni-leipzig.de](http://jedi.informatik.uni-leipzig.de) sein, die wie Leia benannt haben.

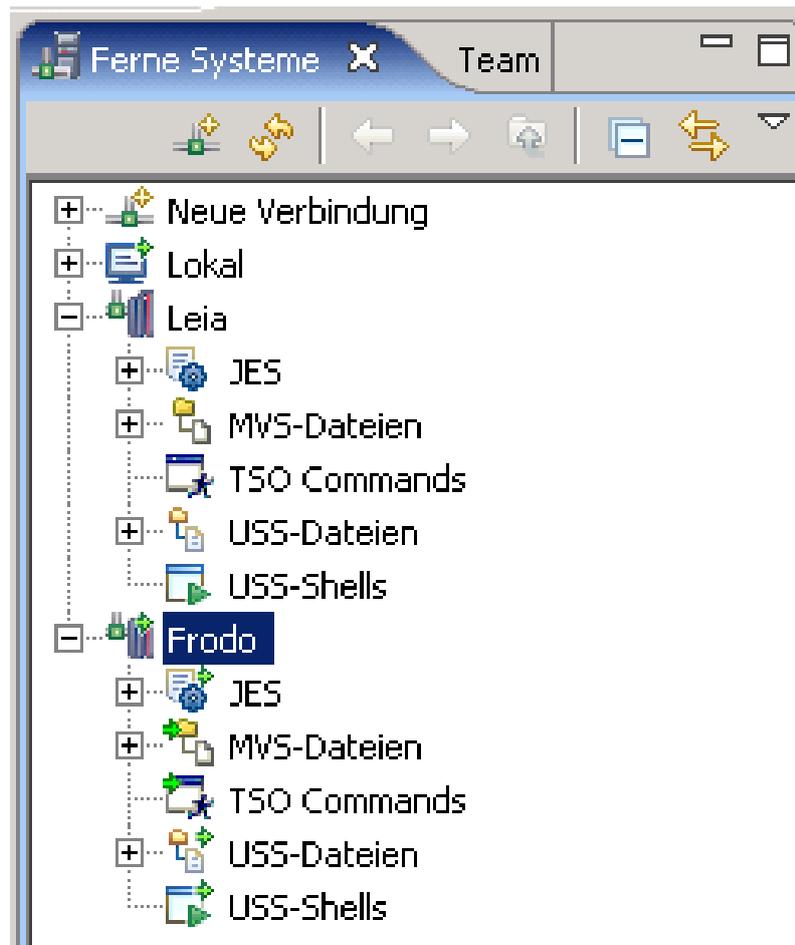


Abbildung 1.17

Sie sehen in Abb. 1.17 die Verfügbarkeit der z/OS Ressourcen sowohl unter Frodo, als auch unter Leia. Sie können aber in jedem Augenblick mit Ihrer Benutzerkennung nur mit einem der beiden Rechner verbunden sein, in diesem Beispiel mit Frodo.

## 1.4 Auflistung der Datasets

Nun können wir uns mit einer PRAK User ID in Leia einloggen. Die kleinen grünen Pfeile sind nun unter den Leia Ressourcen sichtbar.

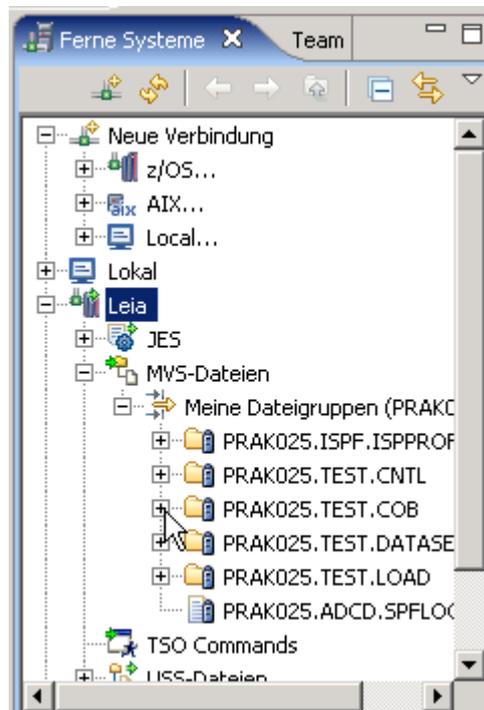


Abbildung 1.19

Die Erweiterung von "MVS-Dateien" und "Meine Dateigruppen..." macht – falls vorhanden – die dort vorhandenen Datasets sichtbar. Abb. 1.19 – 1.21 zeigt Data Sets, die ein Benutzer mit der Kennung PRAK025 auf Leia in der Vergangenheit angelegt hat.

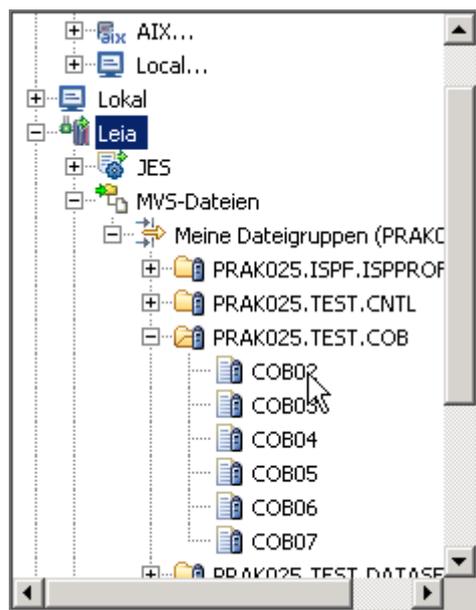


Abbildung 1.20

Der Partitioned Dataset PRAK025.TEST.COB lässt sich erweitern, es werden seine Member sichtbar.

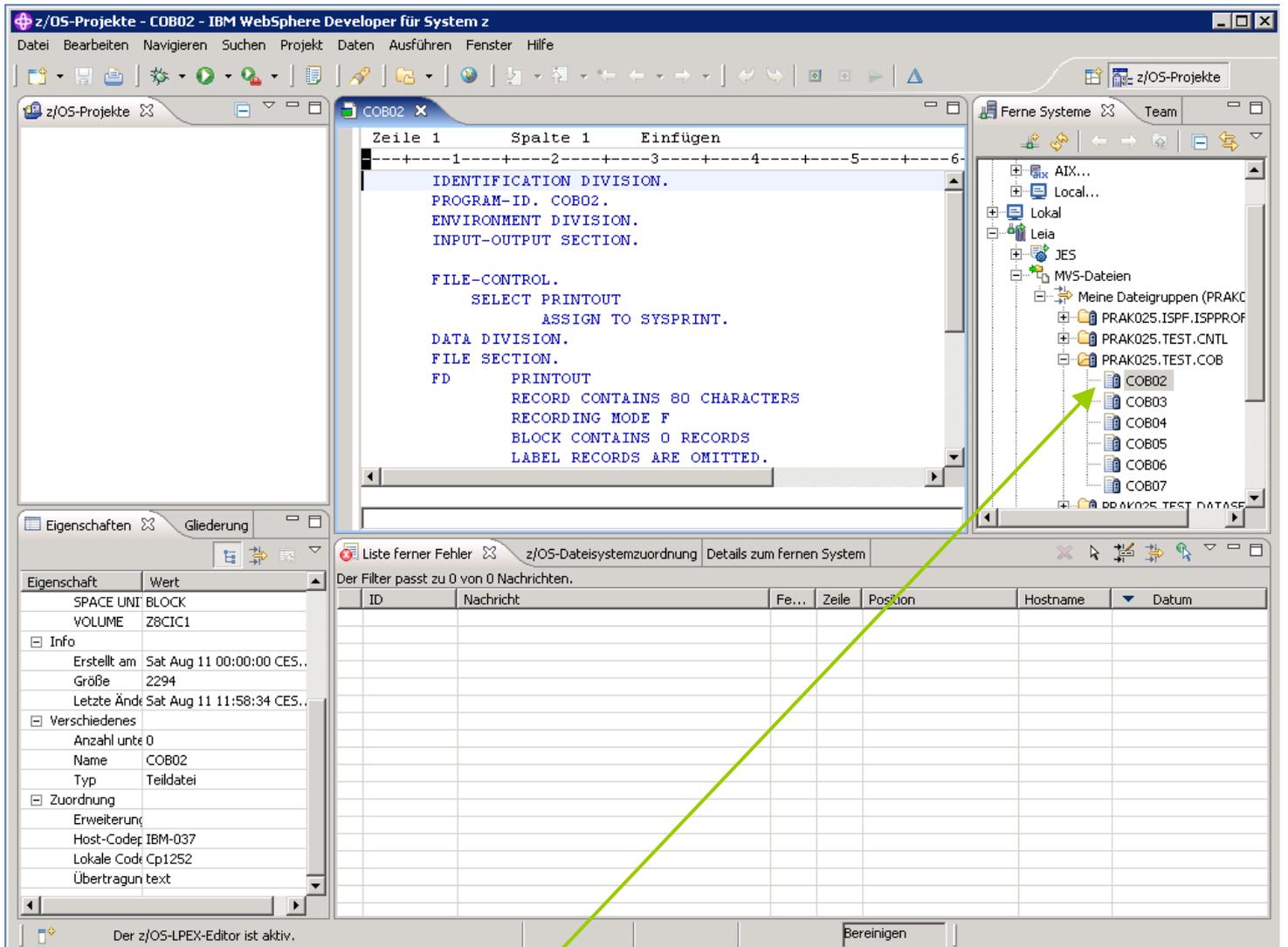


Abbildung 1.21

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

## 1.5 Auflistung der JES-Jobs

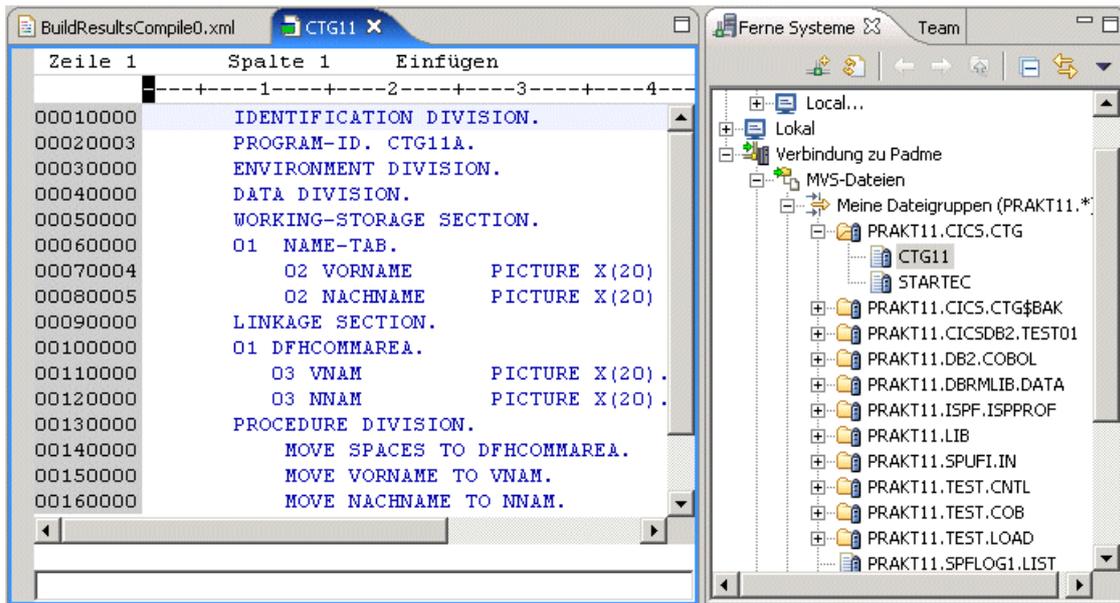


Abbildung 1.22

Abbildung 1.22 zeigt, dass im Mainframe-Account "PRAKT11" unter anderem der PDS "PRAKT11.CICS.CTG" mit seinen beiden Mitgliedern "CTG11" und "STARTEC" angelegt ist. In der Editor-View wird der Inhalt des Members "CTG11" (Cobol-Programm-Code) angezeigt.

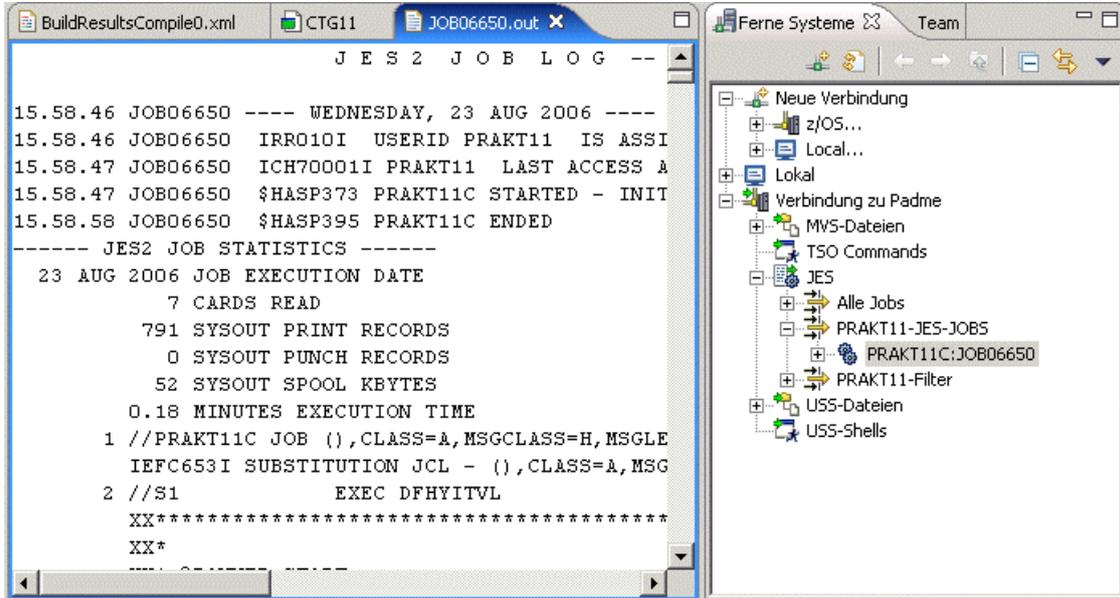


Abbildung 1.23

Die Expansion von "JES" und „PRAKT11-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.

Das Beispiel in Abbildung 1.23 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.

### Selbst-Test

- Was ist der Unterschied zu den ISPF Dlist und Edit Funktionen ?
- Welche zusätzlichen Funktionen bietet RDz, die in ISPF nicht vorhanden sind ?

## 1.6 Auflistung der Unix System Services-Dateien

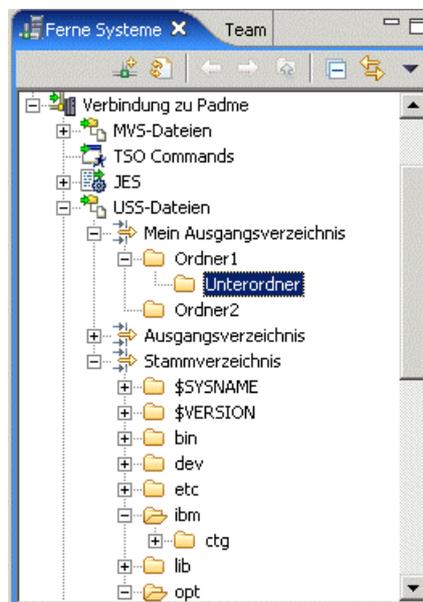


Abbildung 1.24

Wenn Sie "USS-Dateien" und "Mein Ausgangsverzeichnis" expandieren, werden – falls vorhanden – die eigenen USS-Ordner und -Dateien anzuzeigen.

Wenn Sie "Stammverzeichnis" expandieren, wird die Verzeichnisstruktur unter der Wurzel sichtbar gemacht.

## 1.7 Deaktivierung der Verbindung zum Mainframe

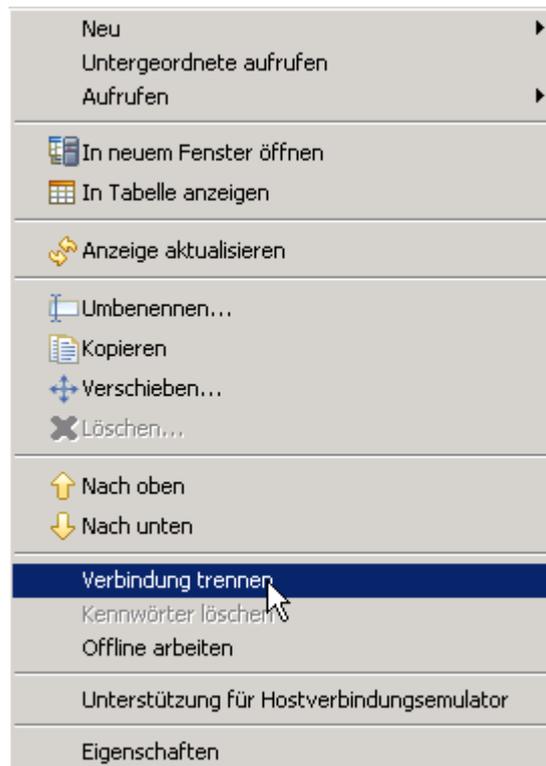


Abbildung 1.25

Rechtsklick auf den Namen der Verbindungsressource. Im sich öffnenden Fenster Klick auf "Verbindung trennen".

Sämtliche kleinen hellgrünen Pfeile verschwinden. Damit ist die Verbindung zum Mainframe geschlossen. An dieser Stelle können Sie jetzt - wenn sie wollen – die Sitzung mit RDz beenden.

Auf jeden Fall sollten Sie die Verbindung mit einem entfernten z/OS System trennen, ehe Sie eine RDz Sitzung beenden.

## **2. Parameter einstellen**

### **2.1 z/OS System-Einstellungen**

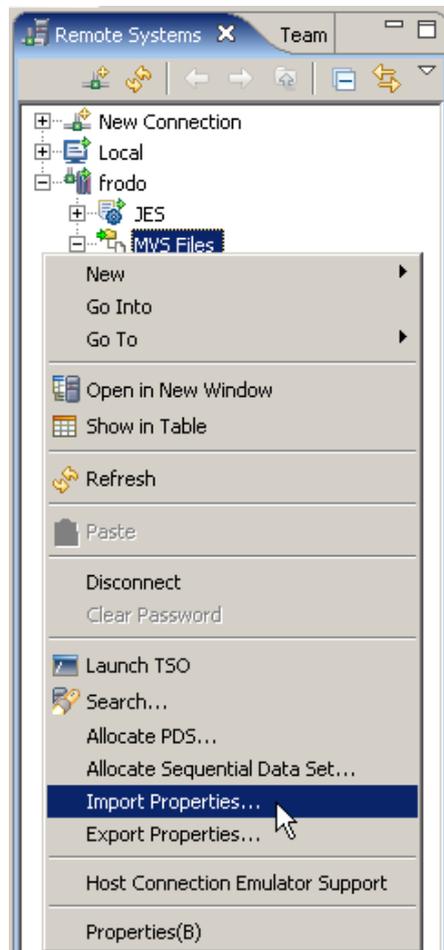
Wenn Sie zum ersten Male ein neues z/OS-System installieren, spezifizieren Sie eine Reihe von Installations-Parametern. Beispiele hierfür sind die Lokatition der System-Libraries, die Konfigurationen Ihrer CICS-oder DB2-Subsysteme, etc. Diese Daten werden als Systemeinstellungen (System Settings) bezeichnet.

Weiter benötigen wir die erforderlichen Einstellungen für z/OS Assets (z.B. z/OS Datasets) auf dem entfernten (remote) Host. Beispiele sind die richtigen COBOL Bibliotheken, CICS und DB2-Einstellungen usw. Normalerweise ist dies nur einmal für jeden neuen Workspace erforderlich.

RDz ordnet diesem Satz von System-Eigenschaften eine Reihe von Default Werten zu. Wenn Sie COBOL-Anwendungen in der Workbench entwickeln möchten, können Sie die Eigenschaften auf der Compileroptionen Registerkarte dieser Seite setzen. Sie überschreiben die JCL-Prozeduren, die sonst für compile, link, und die Ausführung verwendet werden würden.

Der String <HLQ> wird automatisch mit dem High Level Qualifier (Ihrer User ID) des Data Set Namens ersetzt. Später, je nach Verwendung von CICS, DB2 und IMS, müssen Sie möglicherweise zusätzliche Eigenschaften auf den übrigen Registerkarten (Tabs) einstellen.

Da diese Aufgabe nur einmal pro Installation getan werden muss, beschließen wir, die bereits definierten Eigenschaften für unsere Umgebung zu importieren.



**Abbildung 2.1.1**

**Im Remote Systems view, expandiere Frodo. 1kr auf MVS File, selektiere Import Properties.**

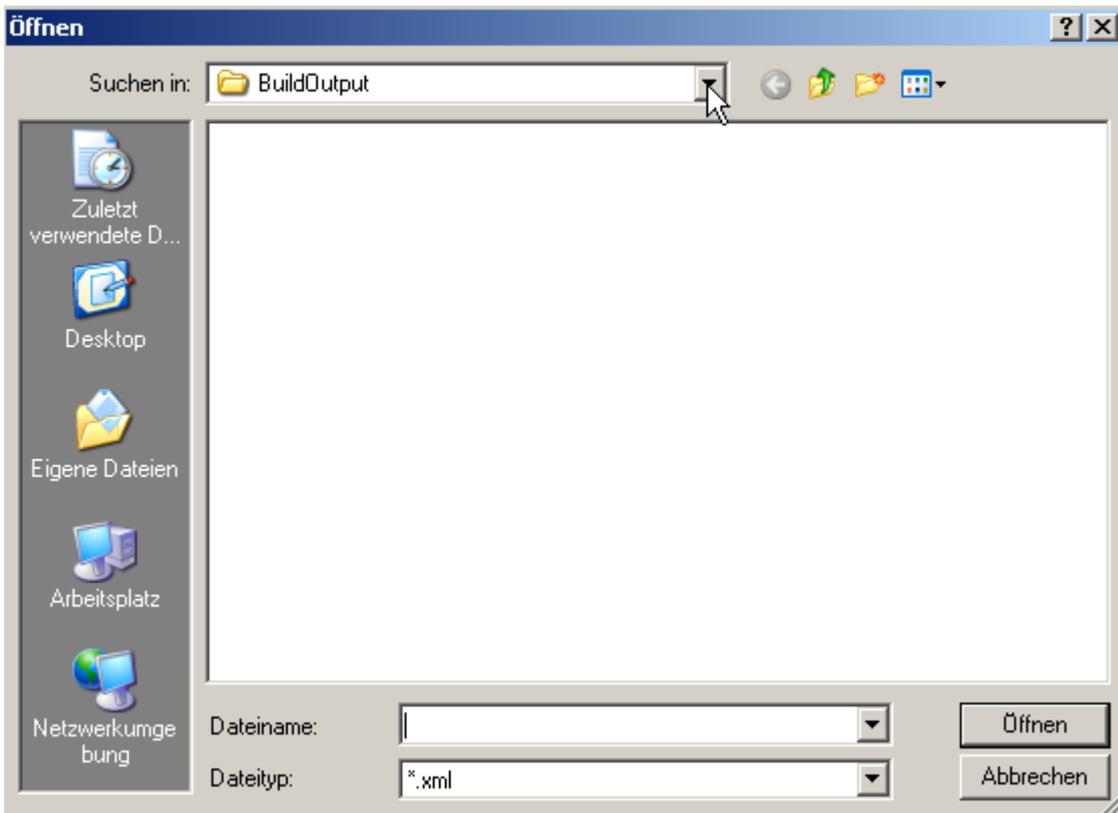


Abbildung 2.1.2

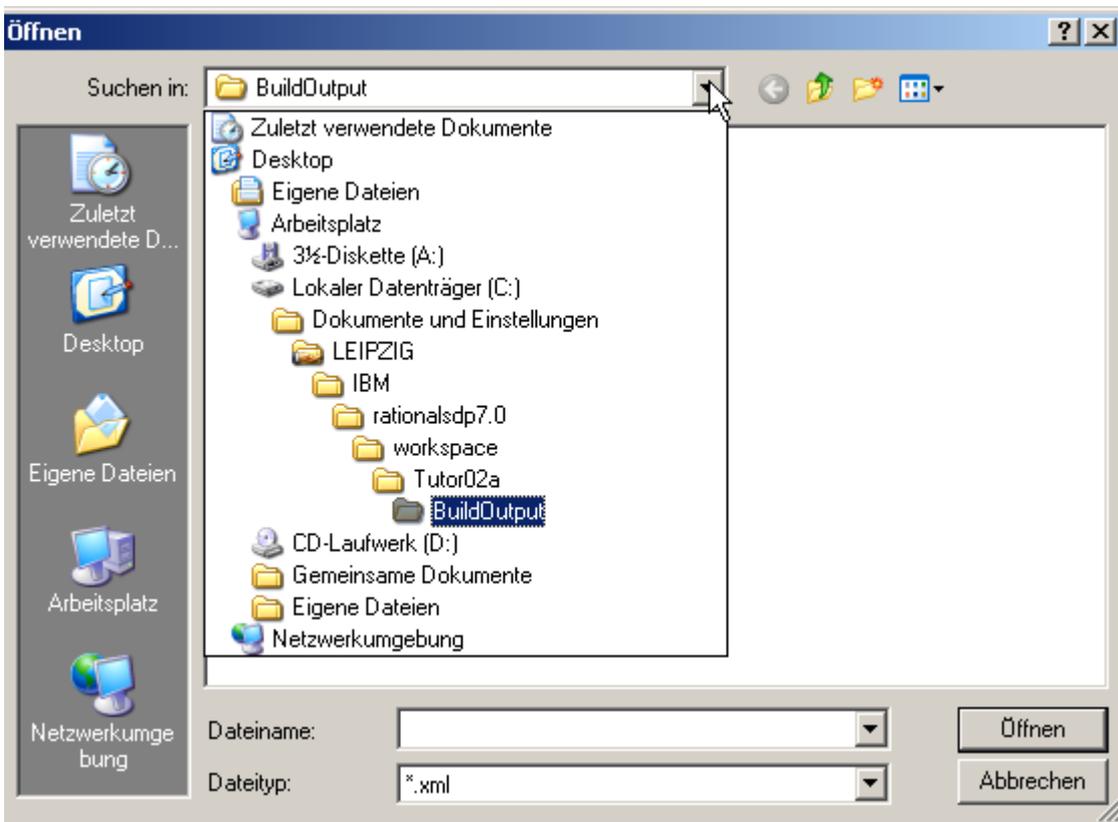


Abbildung 2.1.3

Auf dem RDz Server. Navigiere nach  
C:\Tutorials\Tutorial03\Resources\MVS\_Files\_Properties.xml , und dann ...

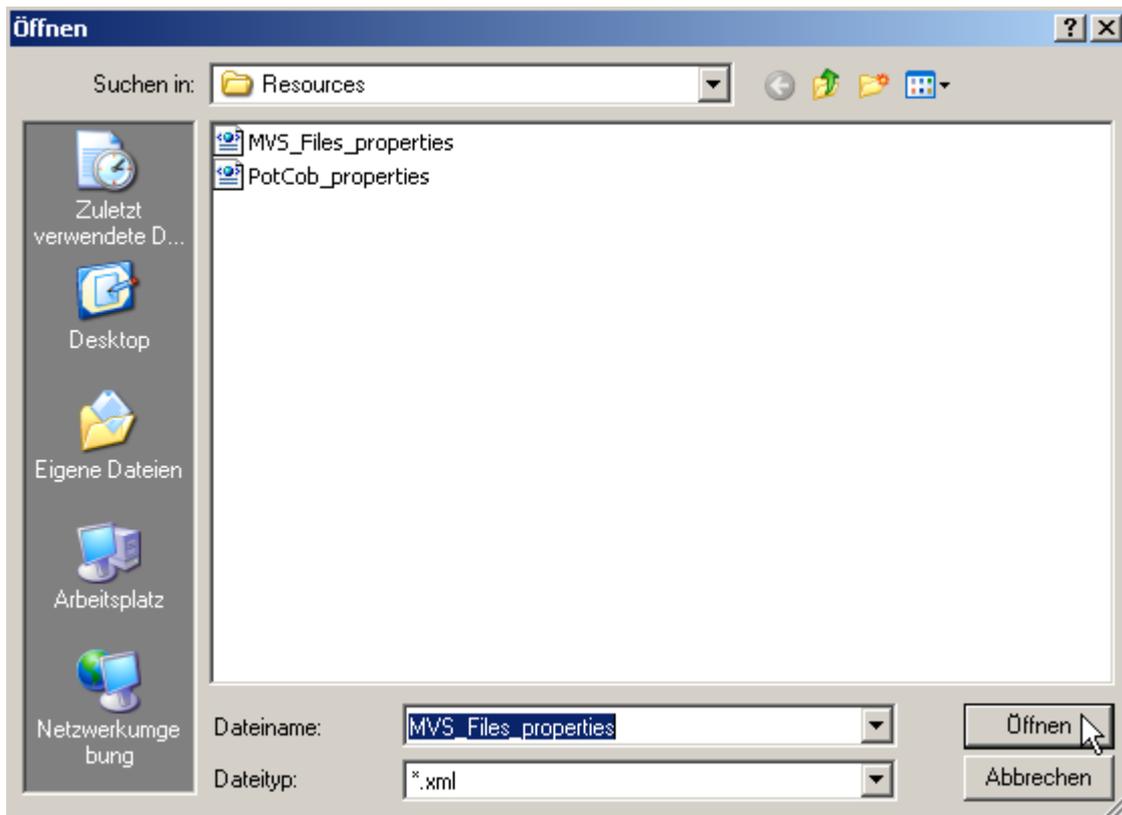
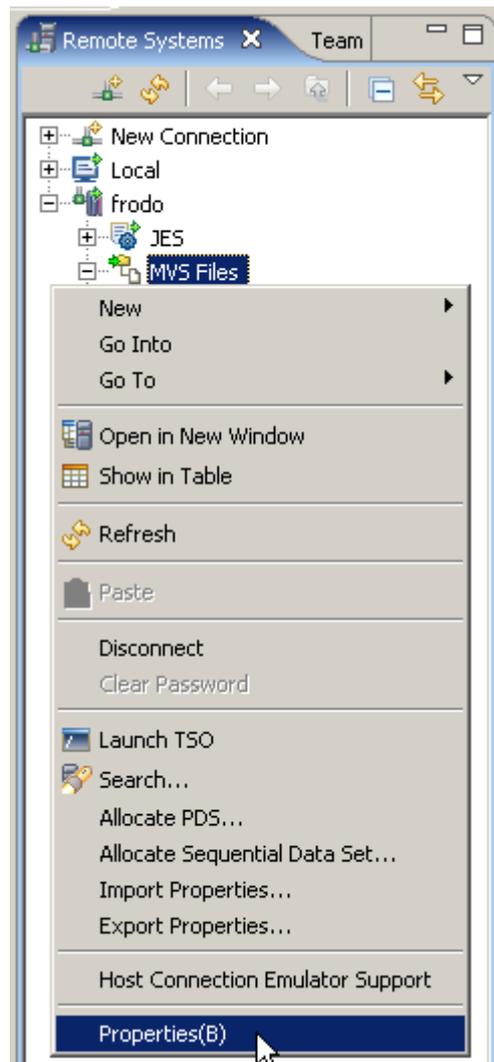


Abbildung 2.1.4

2k um es zu selektieren.  
(zum selektieren können sie es auch öffnen. Dazu 1k auf die „Öffnen“ Schaltfläche).

Öffnen.



**Abbildung 2.1.5**

**Nochmals 1kr auf MVS Files. Diesmal Properties selektieren.**

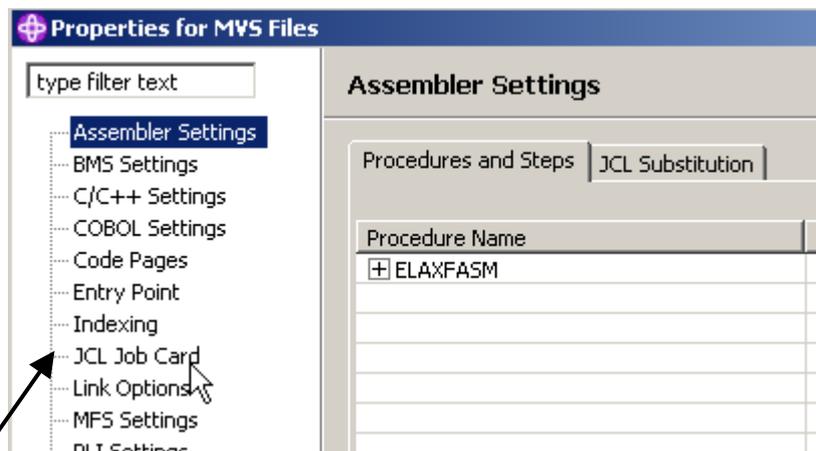


Abbildung 2.1.6

Die JCL Job Card muss geändert werden.

In dem "Properties for MVS Files" Fenster, 1k auf JCL Job Card

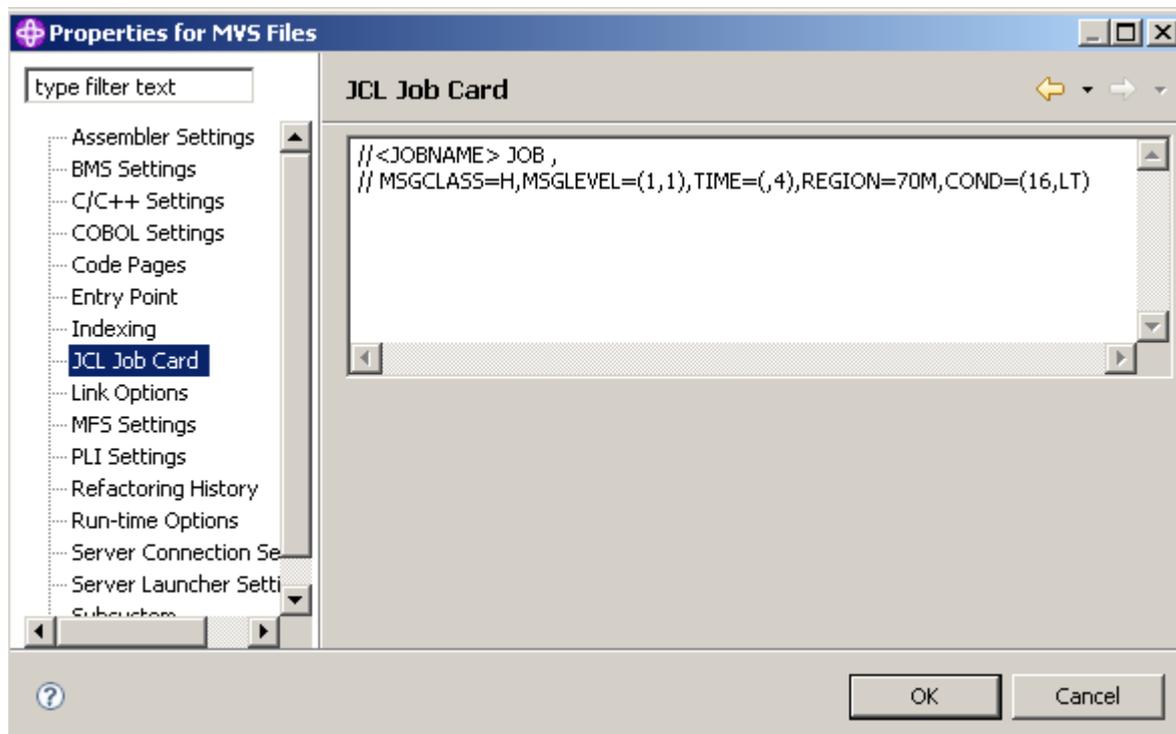


Abbildung 2.1.7

<JOBNAME> muss durch Ihre User ID ersetzt werden, z.B. prak032 in diesem Beispiel.

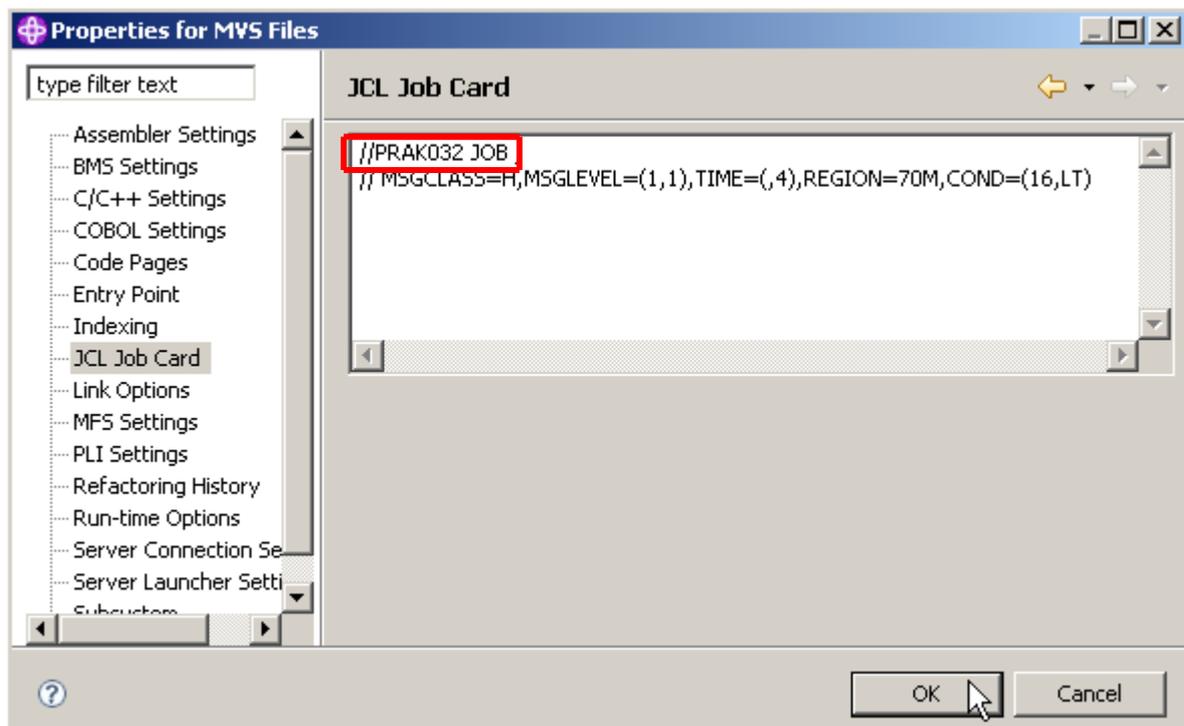


Abbildung 2.1.8

Der Rest ist ok.

Click OK to save. Das „Properties for MVS“ Fenster schließen.

### Selbst-Test

- Rekapitulieren Sie nochmals: Warum sind die Schritte in Abschnitt 2.1. erforderlich ?
- Lesen Sie ggf. die einführenden Sätze zu Beginn von Abschnitt 2.1 nochmals durch.

## 2.2 Benötigte z/OS Datasets erstellen

Bisher haben wir mit einem Remote-System verbunden. Wir haben die z/OS System-Einstellungen erfolgreich eingerichtet.

Jetzt erstellen wir (allocate) insgesamt 9 Datensätze, die für die Entwicklung von Cobol Programmen benötigt werden. Wir könnten dies manuell tun, so wie bisher. Stattdessen verwenden wir ein JCL Script, dass alle 9 Datensätze erstellt.

Wir modifizieren einen bereits existierenden JCL Script Prototypen und submit diesen zu z/OS um die 9 Datasets zu erstellen. Sie finden den Prototypen dieses JCL-Skripest auf Ihrer virtuellen IMachine unter C: .

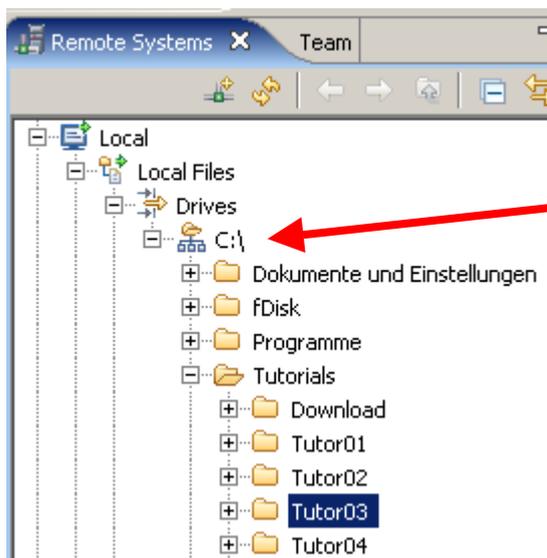


Abbildung 2.2.1

Denken Sie daran, RDz besteht aus einer Komponente auf Ihrer Workstation (Local) und einer Komponente auf Ihren Mainframe (Frodo in diesem Beispiel).

Die Dateien im Verzeichnis C:\ der virtuellen Maschine auf Ihrer Workstation sind in der Remote Systems Sicht unter Local angezeigt.

Unter Benutzung der Remote Systems Sicht, klicken auf das + Zeichen vor „Local“ um es zu erweitern (nicht das "Local ..." das Sie finden, wenn Sie New Connection expandieren !!).



Ihre virtuelle Maschine auf dem RDz Server enthält Partitionen C:\, Y:\, und Z:\

Abbildung 2.2.2

Navigieren sie nach C:\WDz\_Tutorials\Tutor03 auf Ihrer Virtuellen Maschine.

Navigieren sie zum Directory “Local Files\Drives\C\Tutorials\Tutorial03”.

Wenn das Directory nicht da ist, besorgen sie sich den Inhalt von Ihrem betreuer und kopieren sie es nach C:\ von Ihrer virtuellen Maschine.

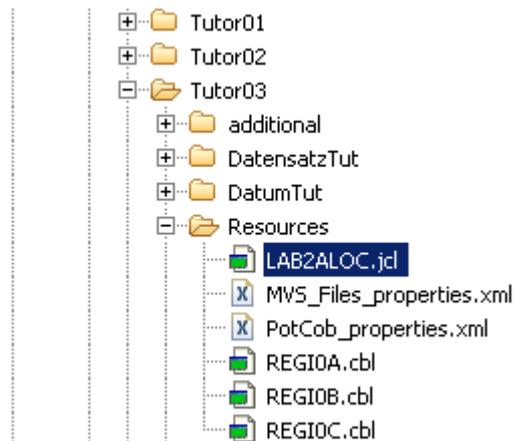


Abbildung 2.2.3

Jetzt Tutorl03, und die File LAB2ALOC.jcl in dem Directory Resources suchen.

2k auf LAB2ALOC.jcl um die file im editor Fenster zu öffnen.

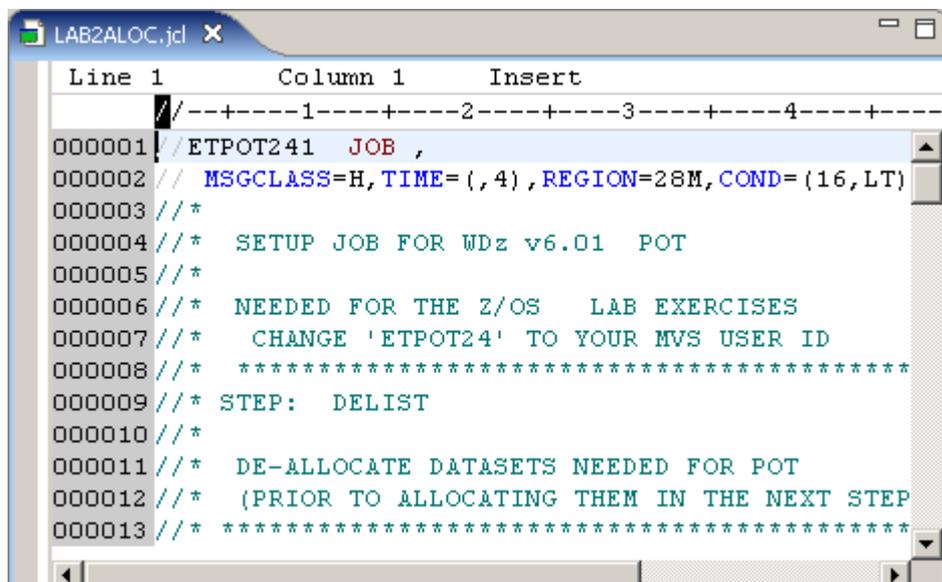
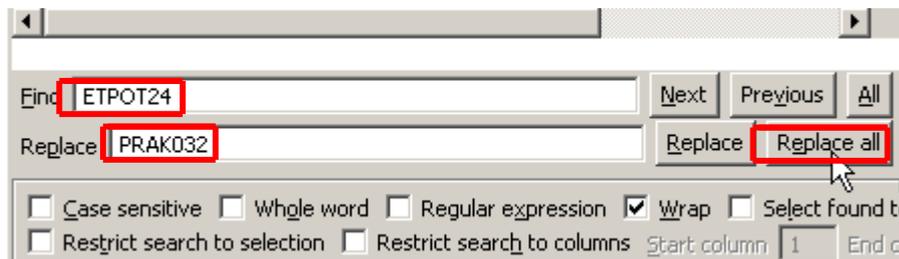


Abbildung 2.2.4

Das dargestellte Prototype JCL Script benutzt die UserID “ETPOT24”. ETPOT24 taucht in dem Script häufig auf. Alle Instanzen von ETPOT24 müssen durch Ihre eigene UserID ersetzt werden.

Die folgenden Schritte zeigen auf, wie ETPOT24 durch Ihre eigene User ID ersetzt wird.



**Abbildung 2.2.5**

**Die Search Facility CTRL+ F öffnen. In dem “Find” Feld ETPOT24 eingeben. In dem Replace Feld Ihre UserID eingeben(in diesem Beispiel ist die UserID = PRAK032). Es ist wichtig, Großbuchstaben zu benutzen. 1k auf „Replace all”.**

```
*LAB2ALOC.jcl
Line 13      Column 42      Insert  2 changes
//-----1-----2-----3-----4-----+
000001// PRAK032 JOB ,
000002//  MSGCLASS=H, TIME=(,4), REGION=28M, COND=(16,LT)
000003// *
000004// *  SETUP JOB FOR Wdz v6.01  POT
000005// *
000006// *  NEEDED FOR THE Z/OS  LAB EXERCISES
000007// *  CHANGE 'PRAK032' TO YOUR MVS USER ID
000008// *  *****
000009// *  STEP:  DELIST
000010// *
000011// *  DE-ALLOCATE DATASETS NEEDED FOR POT
000012// *  (PRIOR TO ALLOCATING THEM IN THE NEXT STEP
000013// *  *****
000014// DELIST EXEC PGM=IDCAMS
000015// SYSPRINT DD SYSOUT=*
000016 DELETE PRAK032.POT.COBOL
000017 IF LASTCC = 8 THEN SET MAXCC = 4
000018 DELETE PRAK032.POT.COPYLIB
000019 IF LASTCC = 8 THEN SET MAXCC = 4
000020 DELETE PRAK032.POT.OBJ
000021 IF LASTCC = 8 THEN SET MAXCC = 4
000022 DELETE PRAK032.POT.PLI
000023 IF LASTCC = 8 THEN SET MAXCC = 4
000024 DELETE PRAK032.POT.LISTING
000025 IF LASTCC = 8 THEN SET MAXCC = 4
000026 DELETE PRAK032.POT.PLI.LISTING
000027 IF LASTCC = 8 THEN SET MAXCC = 4
000028 DELETE PRAK032.POT.LOAD
000029 IF LASTCC = 8 THEN SET MAXCC = 4
000030 DELETE PRAK032.POT.JCL
000031 IF LASTCC = 8 THEN SET MAXCC = 4
000032 DELETE PRAK032.POT.SP2.COBOL
```

Abbildung 2.2.6

Das Ergebnis sollte ähnlich wie hier dargestellt aussehen.

Scroll down im Editor Fenster um die Statements anzusehen, welche das Allocate der Data Sets bewirken.

1k innerhalb des Editor Fensters, um die find/replace Function zu verbergen. Die erforderlichen Änderungen sind geschehen. Um die permanent zu machen, CTRL + S eingeben.

```
LAB2ALOC.jcl x
Line 39      Column 23      Insert
//-----1-----2-----3-----4-----
000010 /*
000011 /*  DE-ALLOCATE DATASETS NEEDED FOR POT
000012 /*  (PRIOR TO ALLOCATING THEM IN THE NEXT STEP
000013 /*  *****
000014 /* DELIST EXEC PGM=IDCAMS
000015 /* SYSPRINT DD SYSOUT=*
000016 DELETE PRAK032.POT.COBOL
000017 IF LASTCC = 8 THEN SET MAXCC = 4
000018 DELETE PRAK032.POT.COPYLIB
000019 IF LASTCC = 8 THEN SET MAXCC = 4
000020 DELETE PRAK032.POT.OBJ
000021 IF LASTCC = 8 THEN SET MAXCC = 4
000022 DELETE PRAK032.POT.PLI
000023 IF LASTCC = 8 THEN SET MAXCC = 4
000024 DELETE PRAK032.POT.LISTING
000025 IF LASTCC = 8 THEN SET MAXCC = 4
000026 DELETE PRAK032.POT.PLI.LISTING
000027 IF LASTCC = 8 THEN SET MAXCC = 4
000028 DELETE PRAK032.POT.LOAD
000029 IF LASTCC = 8 THEN SET MAXCC = 4
000030 DELETE PRAK032.POT.JCL
000031 IF LASTCC = 8 THEN SET MAXCC = 4
000032 DELETE PRAK032.POT.SP2.COBOL
000033 IF LASTCC = 8 THEN SET MAXCC = 4
000034 DELETE PRAK032.POT.DBRMLIB
000035 IF LASTCC = 8 THEN SET MAXCC = 4
000036 DELETE PRAK032.POT.LAB5
000037 IF LASTCC = 8 THEN SET MAXCC = 4
000038 DELETE PRAK032.ERRCOB
000039 IF LASTCC = 8 THEN SET MAXCC = 4
000040 DELETE PRAK032.POT.DEBUG
000041 IF LASTCC = 8 THEN SET MAXCC = 4

submit to Frodo
```

Abbildung 2.2.7

Command Line

Sie haben das lokale JCL-Skript auf Ihrer Workstation geändert.

Wir werden jetzt das Skript an Ihre Benutzerkennung auf dem fernen z/OS-Server senden. Eigentlich ist dies das gleiche wie mit TSO ein JCL-Skript zu generieren und das „Sub“-Kommando einzugeben.

Denken Sie daran, das JCL-Skript ist auf Ihrer Workstation gespeichert. Sie könnten das JCL-Skript auf Ihr remote z/OS-System kopieren und dort ausführen (submit) es dort. Stattdessen können Sie auf Ihrer lokalen Workstation das Command „Submit to Frodo“ eingeben, was zu dem gleichen Ergebnis führt.

Hierzu selektieren sie das LAB2ALOC.jcl Editor-Fenster, und drücken Sie die Esc-Taste. Der Cursor wird auf der Kommandozeile verschoben.

„Submit to Frodo“ auf der Kommandozeile eingeben, und drücken Sie Enter.

(Hinweis: Stellen Sie vorher sicher, dass Sie an Frodo angeschlossen sind) !

```
000039 IF LASTCC = 8 THEN SET MAXCC = 4
000040 DELETE PRAK032.POT.DEBUG
000041 IF LASTCC = 8 THEN SET MAXCC = 4
JOBID: JOB05969
```

Abbildung 2.2.8

Wurde der Job erfolgreich submitted, erhalten Sie eine JOBID (in diesem Beispiel ist die JOBID = JOB05969).

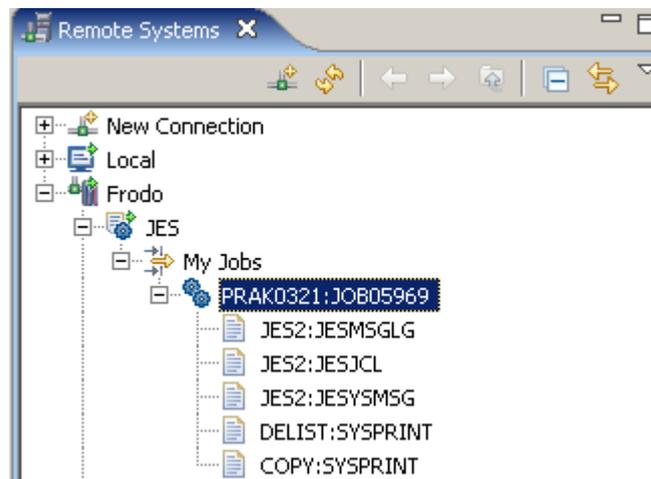


Abbildung 2.2.9

Sie haben den Job erfolgreich submitted. Dennoch sollten sie verifizieren, dass keine Fehler aufgetreten sind.

Hierfür, im Remote Systems View, expandiere JES → My Jobs. Dort finden sie die gerade generierte JOBID (in diesem Beispiel JOB05969). Um dies sichtbar zu machen, kann es sein, dass Sie für Frodo ein disconnect und reconnect durchführen müssen.

2k auf diesen Eintrag um den Editor zu öffnen.

```
IEB1098I 2 OF 2 MEMBERS COPIED FROM INPUT DATA SET REF
IEB144I THERE ARE 29 UNUSED TRACKS IN OUTPUT DATA SET
IEB149I THERE ARE 19 UNUSED DIRECTORY BLOCKS IN OUTPUT
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
```

Abbildung 2.2.10

Scroll down zum Ende. Wenn die Ausführung erfolgreich war, sollte die Botschaft wie oben dargestellt aussehen.



## Selbst-Test

- Sie hätten das gleiche Ergebnis auch erreichen können, wenn sie die benötigten 9 Data Sets wie bisher zu Fuß angelegt hätten. Schauen Sie sich den Abschnitt 2.2 nochmals an. Ist dies wirklich eine Arbeitserleichterung ?
- Sieht das anders aus, wenn sie häufig neue Cobol Programme schreiben ? Müssen die 9 Data Sets dann jedes Mal neu angelegt werden ? Wann ja und wann nein ?

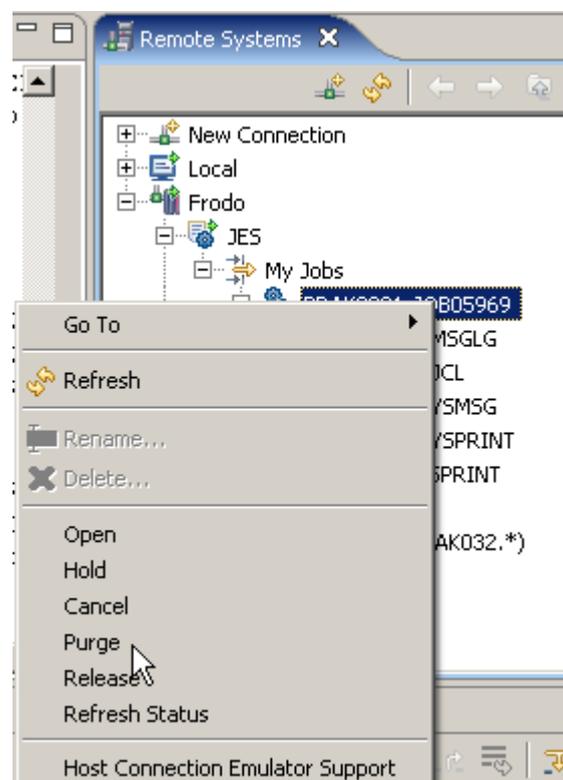


Abbildung 2.2.13

Wenn Sie einen Job Submitten, werden die Zwischenergebnisse in temporären Data Sets, den sog. Spool Files zwischengespeichert. Diese Spool Files für die Jobs laufen mit der Zeit voll. Deshalb ist es eine gute Idee, den Inhalt von Zeit zu Zeit zu leeren (purge). Um diese Aufgabe zu erfüllen, 1kr auf einen Job, und selektieren Sie Purge aus dem Kontext-Menü.

Bei einem längeren arbeiten mit RDz ist dies absolut notwendig. Bitte jedoch nicht gerade jetzt. Sie wollen die Jobs nicht löschen, die Sie gerade erstellt haben.

## 2.3 Host Code Page einstellen

Bei der Benutzung von Windows kennen Sie den Ärger, die richtige Code Page einzustellen. Bei der Verwendung der falschen Code Page werden z.B. die deutschen Umlaute oder das ß durch merkwürdige Buchstabenkombinationen dargestellt.

z/OS ist lange ohne Code Pages ausgekommen. Der Grund ist, der EBCDIC Code ist ein 8 Bit Code, und stellt die Sonderzeichen der wichtigsten europäischen Alphabete automatisch richtig dar. Der ASCII Code ist im Gegensatz dazu ein 7 Bit Code.

In der 2.Hälfte der 90er Jahre führte aber auch z/OS Code Pages ein. Die Standard z/OS Code Page is IBM-037, Local Codepage CP1252.

Wir empfehlen, bei der Benutzung von RDz eine andere Code Page zu benutzen. Wenn Sie die Code Page ändern wollen, gehen Sie in den Remote System View, und dann :

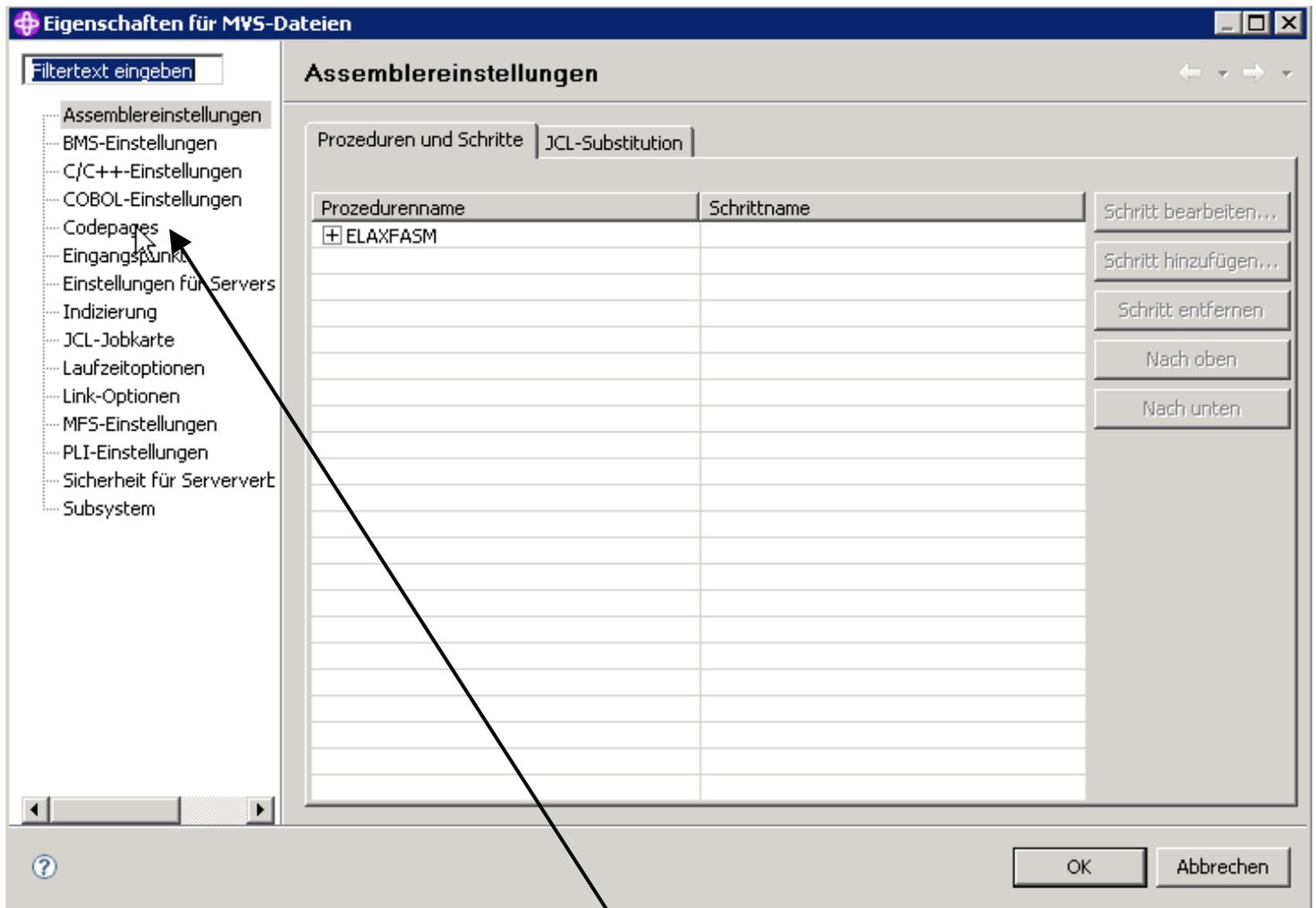


Abbildung 2.3.1

Frodo → MVS Files 1kr → Properties → 1k auf Code Pages

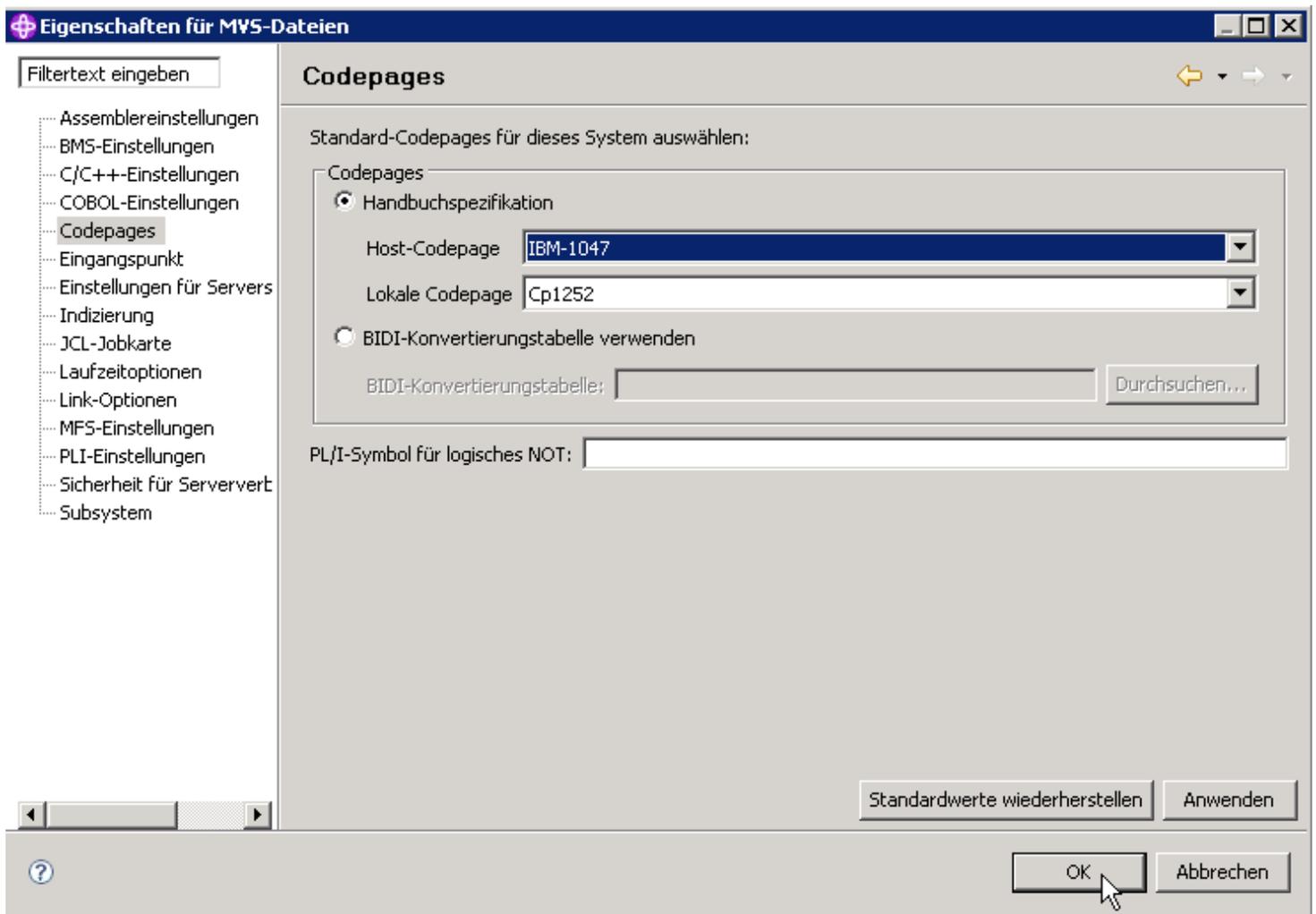


Abbildung 2.3.2

Jetzt selektiere IBM.1047, Local Codepage CP1252, 1k auf ok. Damit werden Sie hoffentlich einigen Schwierigkeiten aus dem Wege gehen.

Nichts ist vollkommen. Ein berüchtigtes Beispiel ist die Darstellung von rechteckigen Klammern beim Programmieren in C/C++. In Problemfällen hilft es, die Benutzung unterschiedlicher z/OS Code Pages auszuprobieren.

## 2.4. Emulation des 3270 Green Screens

RDz besitzt auch einen integrierten 3270-Emulator, den Sie statt Ihres normalen 3270 Emulators (z.B. Quick32 Freeware Edition). Nachfolgend wird dessen Aufruf demonstriert:

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

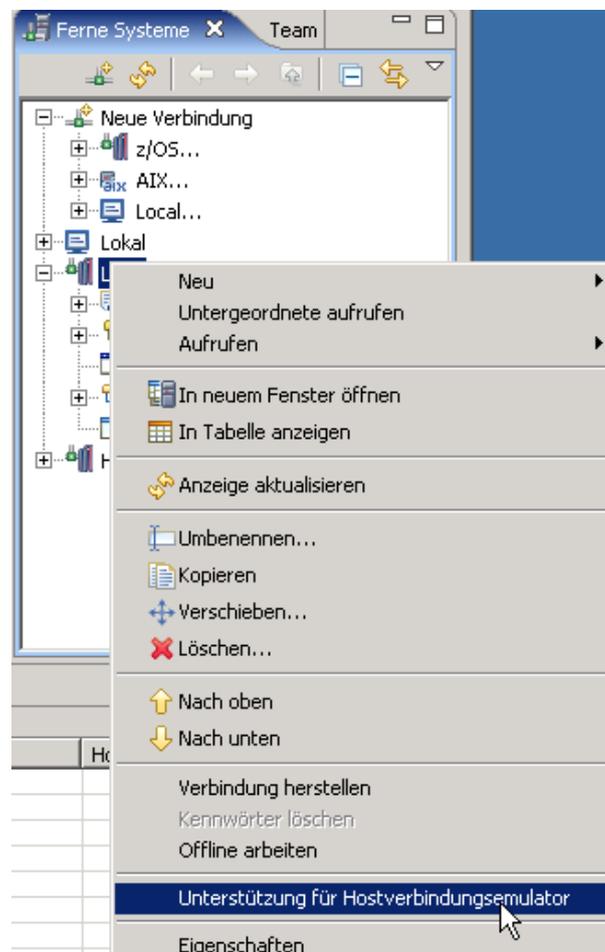


Abbildung 2.4.1

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

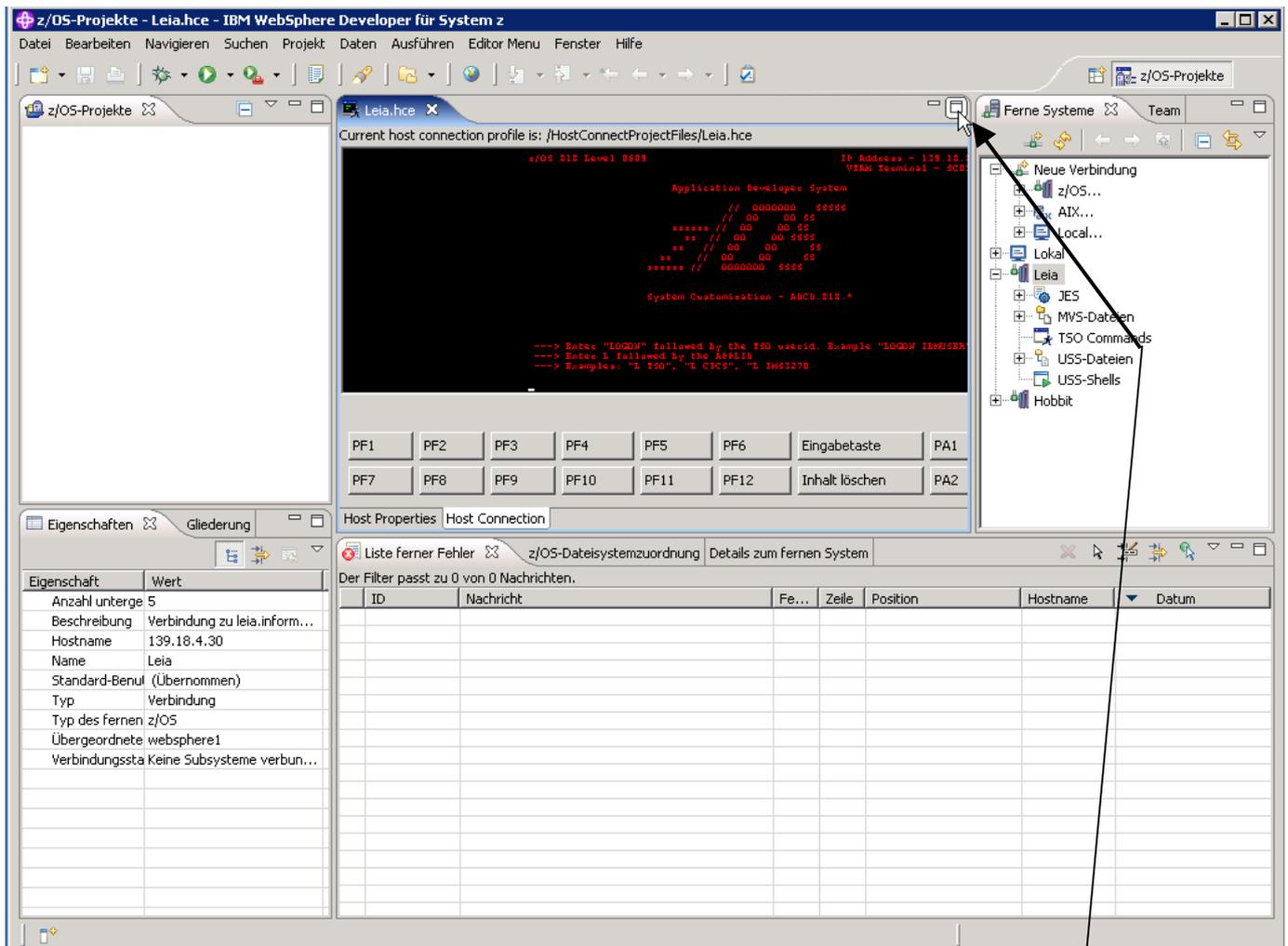


Abbildung 2.4.2

Um den 3270-Emulator auf die ganze RDz-Arbeitsfläche zu vergrößern, auf die Schaltfläche "Maximieren" klicken (siehe Abbildung 7.2).

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 Subsystem einloggen, oder mittels "L CICS", Mainframe-Login und Passwort ins CICS Subsystem.

## 3.0 Working with MVS files

Wir sind jetzt in der Lage, mit z/OS Assets zu arbeiten (besonders z/OS Datasets).

Wir haben 3 Cobol Quell-Programme REGIO0A.cbl, REGIO0B.cbl, und REGIO0C.cbl für Sie präpariert, welche Sie in dem lokalen Verzeichnis C:\WDz\_Tutorials\Tutorial03 finden können. Wir werden diese letztendlich kompilieren, linken und ausführen.

In Abb. 2.2.12 hatten wir neu erstellte Data Sets wiedergegeben, welche mit unseren Settings für die Cobol Programmierung notwendig sind. Zusätzlich werden wir jetzt einen weiteren neuen PDS (Partition Data Set) unter Ihrer z/OS User ID erstellen. Wir werden die vorgefertigten 3 Cobol Programme REGIO0A.cbl, REGIO0B.cbl, und REGIO0C.cbl in Member dieses Partitioned Data Sets kopieren, sowie Filter benutzen um Daten zu betrachten.

### 3.1 Allocating a Data Set

Erinnern Sie sich, in Ihrem allerersten Tutorial 01a, Dateiverwaltung unter TSO, war das Allocate eines Data Sets ihre erste Aufgabe. Für das Cobol RDz Tutorial machen wir das Gleiche, ausgenommen, dass wir hierfür RDz benutzen.

Im Remote Systems View, expandiere "MVS Files".

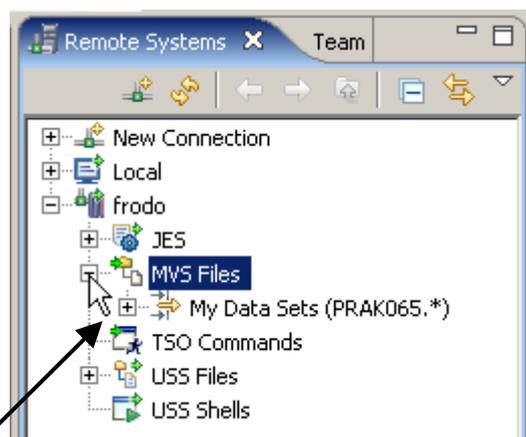


Abbildung 3.1.1

1k auf  My Data Sets um alle Data Sets zu sehen. (Keine Angst, wenn bereits viele Data Sets allocated sind; wir werden ein refresh des Systems in Kürze durchführen).

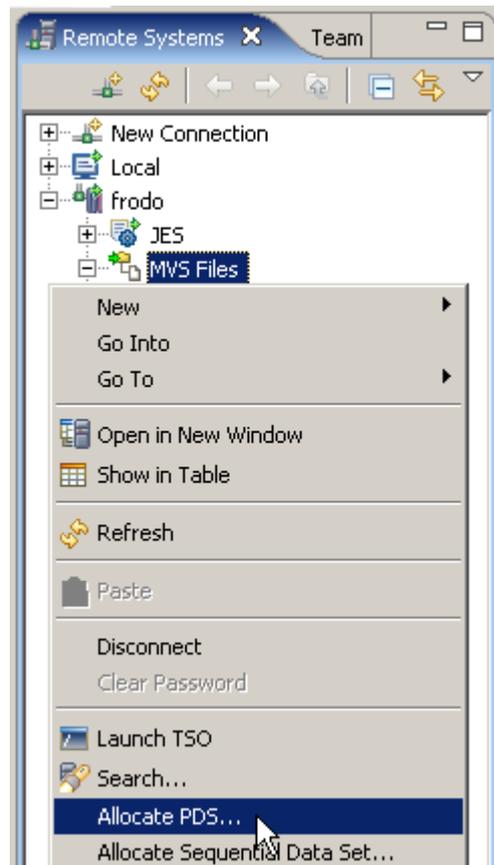


Abbildung 3.1.2

1kr auf MVS Files. Selektiere “Allocate PDS” aus dem Kontext Menu.

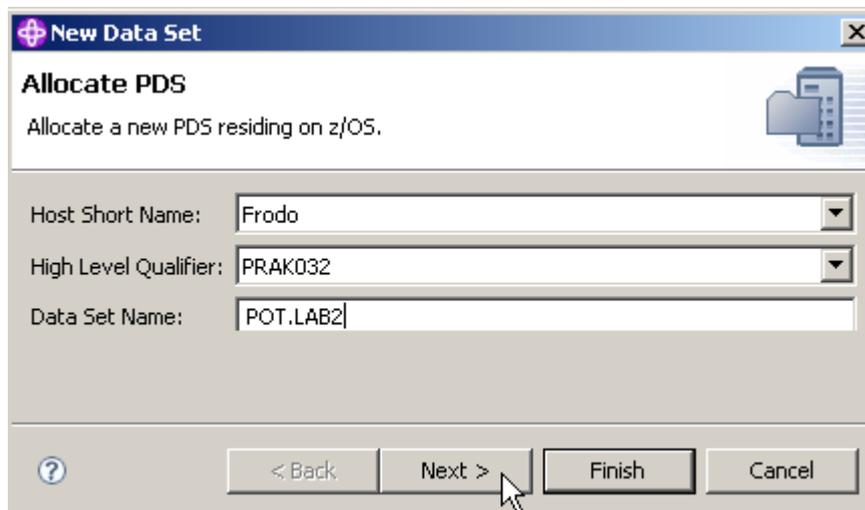


Abbildung 3.1.3

Im “Allocate PDS” Panel benutzen sie die Drop Down Liste um den Namen Frodo zu selektieren und Ihren High Level Qualifier (PRAK032 in diesem Beispiel). Geben Sie POT.LAB2 als den Data Set Name Namen ein. Hiermit soll ein Data Set allocated werden, der später Ihr Cobol Programm auf nimmt. 1k auf Next.

Die hier als Intermediate Level Qualifier gewählten Buchstaben “POT” sind eine Abkürzung für “Proof of Technology”.

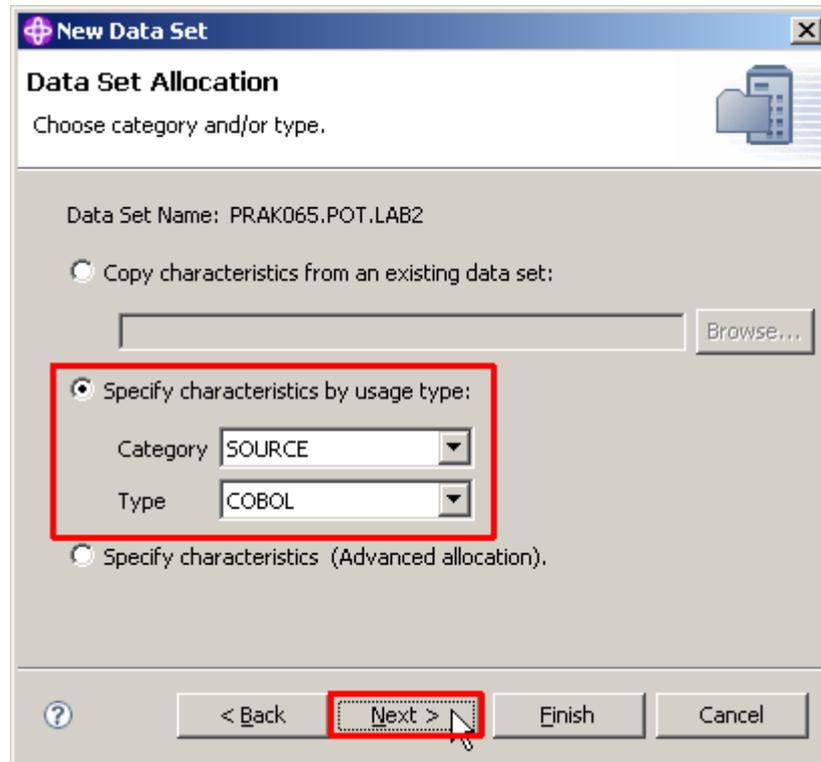


Abbildung 3.1.4

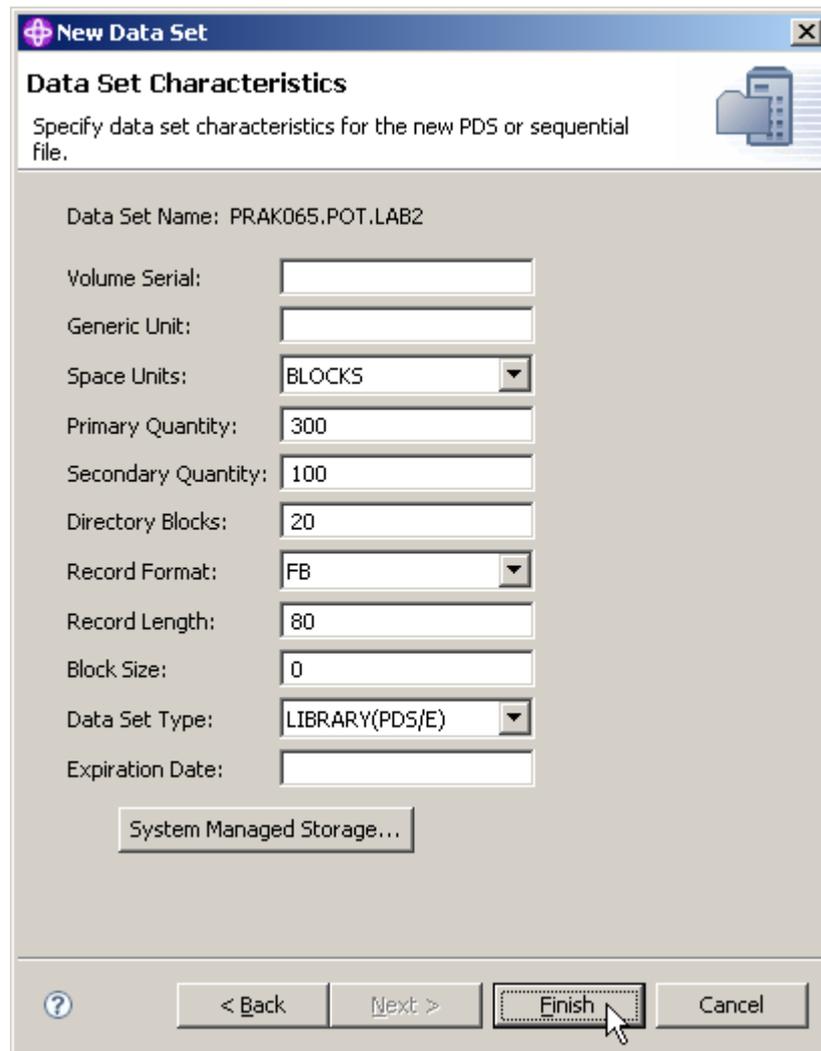
Im Data Set Allocation-Panel, selektieren Sie "Specify characteristics by usage type". Wählen Sie SOURCE für Kategorie und COBOL für Type. 1k auf Weiter.

Dies ist ein nettes Feature, da es einen Datensatz allocated, der die erforderlichen Eigenschaften für das Speichern von COBOL Quellcode aufweist.

Die Alternativen für SOURCE (Quellcode) wäre z.B. LISTING, oder andere. Die Alternativen für COBOL wäre ASM (Assembler), C/C++, JCL, PLI und REXX.

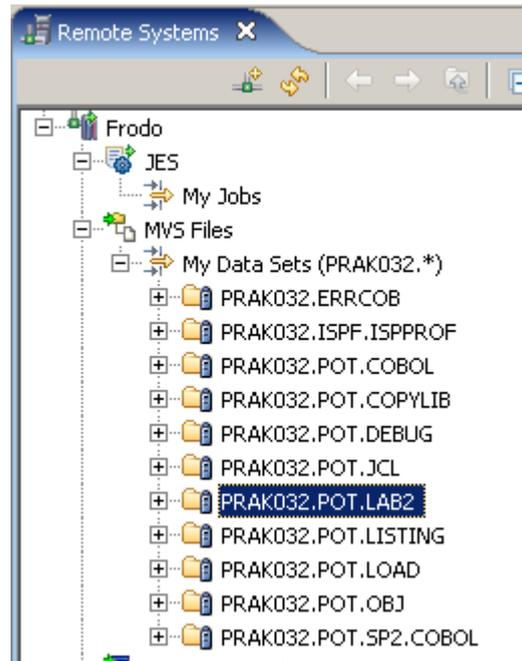
Anmerkung: Ganz im Gegensatz zu der x86 Architektur ist die Assembler Programmierung unter z/OS überraschend einfach. Die Bedeutung von z/OS Assembler ist in den letzten Jahren allerdings sehr stark zurückgegangen. Dennoch gibt es immer wieder Fälle, in denen z/OS Assembler eine Rolle spielt. Assemblerspielt unter z/OS eine bedeutendere Rolle als unter Linux, Unix und Windows.

Klick next



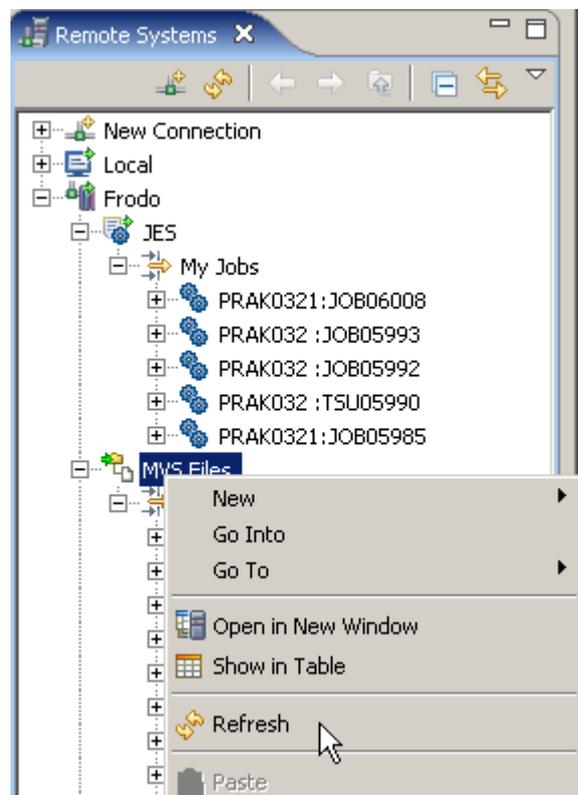
**Abbildung 3.1.5**

**Dies sieht ähnlich wie der vertraute TSO Allocate Screen aus dem allerersten Tutorial 01a (Dateiverwaltung unter TSO) aus, und hat die gleiche Funktion. Übernehmen sie die Default Werte und click Finish.**



**Abbildung 3.1.6**

Unter der Annahme, dass Plattenspeicherplatz unter z/OS verfügbar ist, wird der Partitioned Data Set PRAK032.POT.LAB2 angelegt. Zu diesem Zeitpunkt hat er noch keine Member.



**Abbildung 3.1.7**

Wenn Sie ihn jetzt nicht sehen, 1kr auf MVS Files, und selektiere Refresh (disconnecting und reconnecting Frodo würde den gleichen Effekt haben).

## 3.2 Mapping Data Sets

RDz speichert Identische Kopien einer File sowohl unter z/OS als auch auf der Workstation, die von RDz synchron manipuliert werden. Erfahrungsgemäß ist dies nicht ganz leicht zu verstehen, deshalb Abschnitt 3.2 sorgfältig lesen.

Es existieren unterschiedliche Konventionen für die Namensgebung von Dateien unter Windows und unter z/OS. Wir möchten eine Kopie der gleichen Datei unter beiden Betriebssystemen speichern und bearbeiten.

Das Problem ist, eine Datei Bezeichnung wie PRAK032.POT.COBOLO oder PRAK032.POT.LAB2 ist kein gültiger Name für eine Windows FAT32 oder NTFS Datei.

Erinnern wir uns, bei einem Data Set Namen wie PRAK032.POT.COBOLO bezeichnen wir „PRAK032“ als den High Level Qualifier (HLQ) und „COBOLO“ als den Lowest Level Qualifier (LLQ).

Ein „Data Set Mapping“ assoziiert den Lowest Level Qualifier in jedem MVS Data Set mit einer File-Name Extension, welche das Betriebssystem Ihrer Workstation(auf der RDz läuft), versteht. Beispielsweise auf dem Mainframe haben wir Data Sets wie PRAK032.POT.COBOLO oder PRAK032.POT.JCL. Wir mappen \*\*COBOLO auf cbl, und \*\*JCL auf jcl, usw. Hierbei repräsentieren die beiden Asterics „\*\*“ die beiden Qualifier PRAK032 und POT in dem Data Set Namen.

Dies ist erforderlich, damit die Workstation weiss, mit welchen Arten von Daten sie es zu tun hat. **Das Gleiche ist auch für die Member eines PDS Data Sets möglich.**

Wir führen jetzt ein Mapping für die Member des gerade neu angelegten z/OS PDS Data Sets PRAK032.POT.LAB2 durch. Spezifisch generieren wir drei Member, die Cobol Quellcode enthalten, mit den Namen

- PRAK032.POT.LAB2(REGI0A)
- PRAK032.POT.LAB2(REGI0B)
- PRAK032.POT.LAB2(REGI0C)

Wenn wir fertig sind, können Sie diese drei Member sich im Remote Systems View ansehen.

Gleichzeitig werden wir auf der Windows Workstation ein Directory PRAK032.POT.LAB2 einrichten, und in diesem Directory drei Files mit den Namen

- REGI0A.cbl
- REGI0B.cbl
- REGI0C.cbl

unterbringen, welche den identischen Cobol Quellcode wie die drei Member des PDS Data Sets enthalten.

Die drei Member des PDS Data Sets erhalten das Mapping cbl. Damit wird erreicht, dass RDz auf eine für den Benutzer transparente Art wahlweise mit den Membern des Datasets auf dem z/OS System oder den Files unter Windows arbeiten kann.

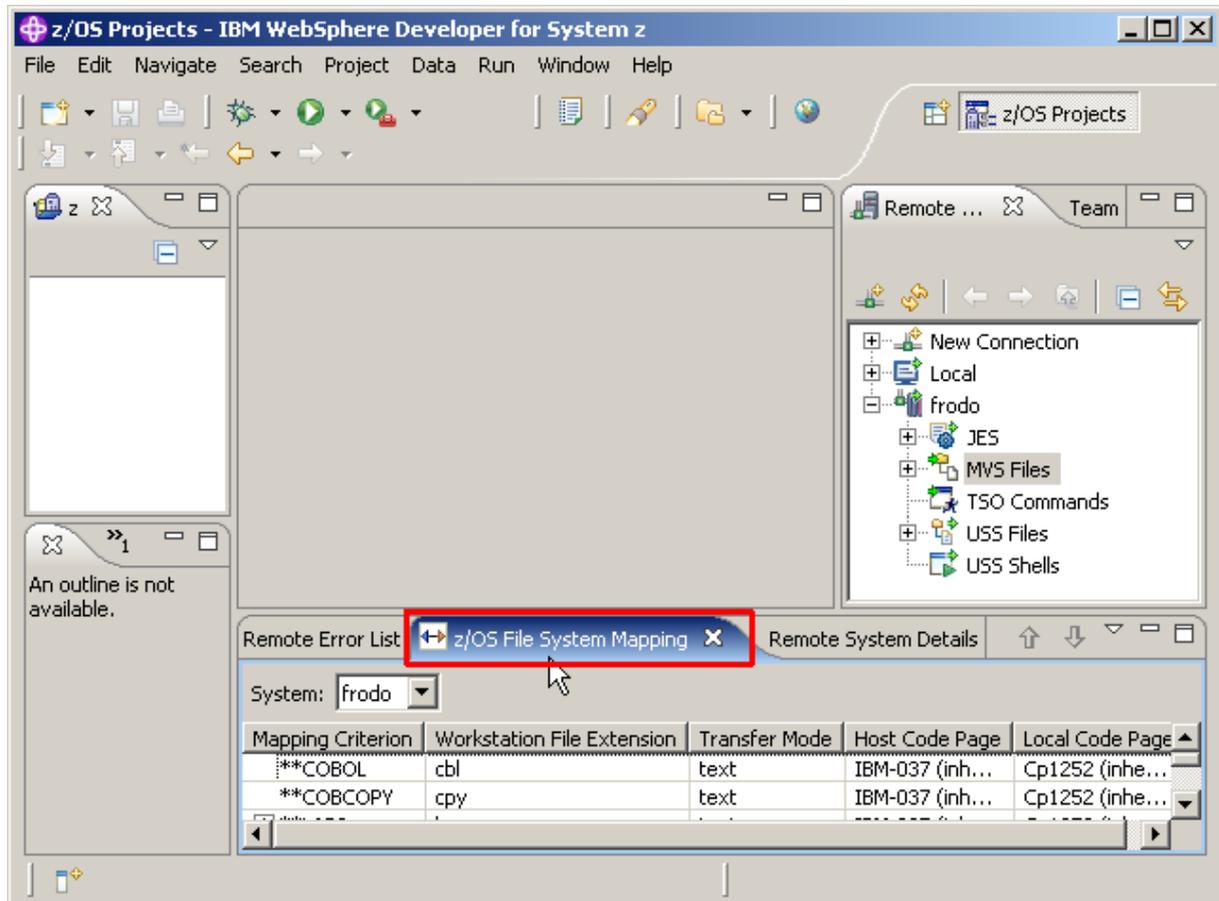


Abbildung 3.2.1

Wechseln Sie zu dem z/OS File System Mapping View mit 1k auf den Tab.

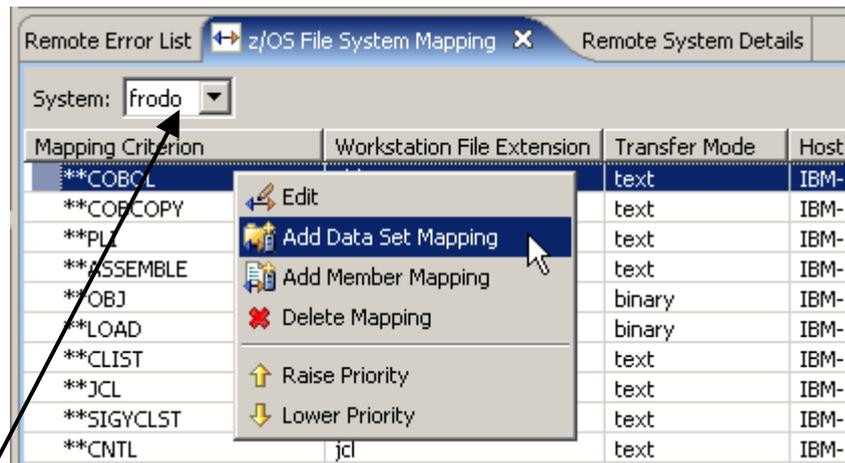


Abbildung 3.2.2

In Abb. 3.2.2 wird eine Liste von Standard mäßig vorgeschlagenen Mappings dargestellt, z.B. cbl für \*\*Cobol Data Sets, oder cpy für \*\*COBCOPY Data Sets (Cobol Copy Books).

1kr auf \*\*COBOL, selektiere "Add Data Set Mapping".

Wir wollen jetzt zu der Liste den Data Set PRAK032.POT.LAB2 hinzufügen, den wir gerade erstellt haben. "LAB2 ist dessen LLQ. Verifizieren Sie hierzu den Namen Ihrer Host Connection (Frodo in diesem Beispiel).

Wir machen dies in 2 Schritten. Im ersten Schritt werden alle Member des Data Sets PRAK032.POT.LAB2 als „bms“ abgebildet (mapped). In einem 2. Schritt wird eine Untermenge der Member von PRAK032.POT.LAB2 als "cbl" abgebildet. Dies geschieht für solche Member, deren Name mit den 4 Buchstaben REGI anfängt.

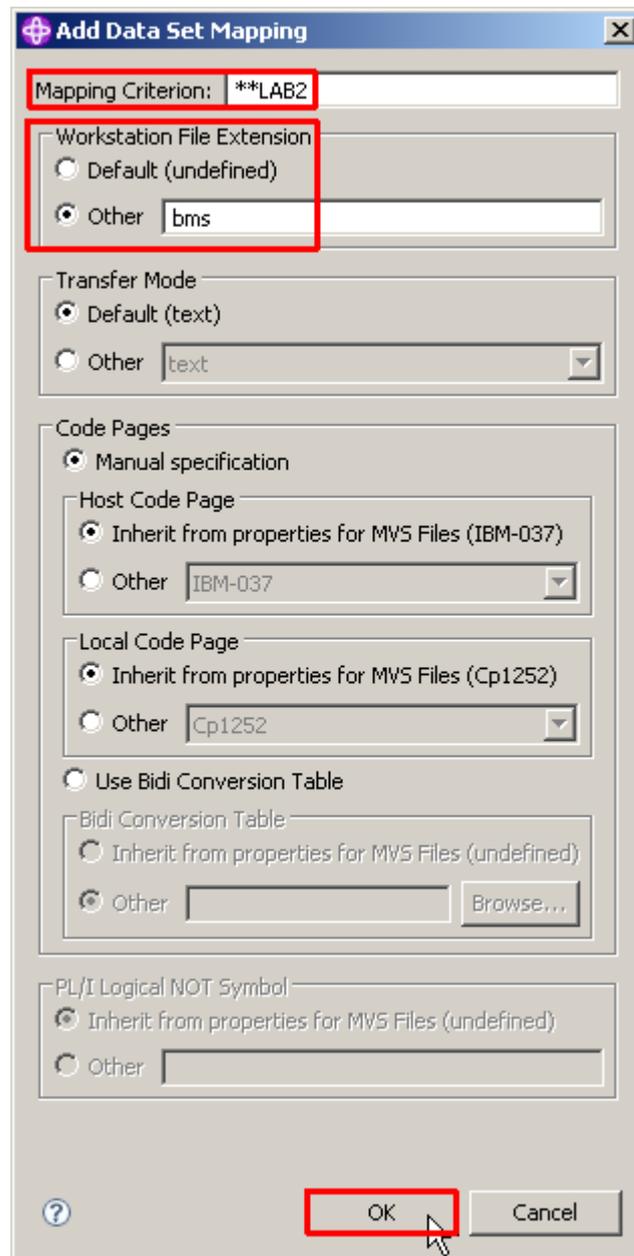


Abbildung 3.2.3

Als Mapping Criterion den Wert **\*\*LAB2** eingeben (für PRAK032.POT.LAB2), als „Workstation File Extension“ selektiere „Other“, und dann „bms“ eingeben. Alle Member dieses Data Sets werden als bms abgebildet. Die restlichen Default Werte unverändert übernehmen. 1k auf OK.

Mapping Criterion	Workstation File Extension	Transfer Mode	Host
**COBOL	cbl	text	IBM-
<b>**LAB2</b>	<b>bms</b>	<b>text</b>	<b>IBM-</b>
**COBCOPY	cpy	text	IBM-
**PLI	pli	text	IBM-
**ASSEMBLE	asm	text	IBM-

Abbildung 3.2.4

Als Ergebnis wurde **\*\*LAB2** erfolgreich zu der Liste hinzugefügt. Alle Member von **\*\*LAB2** erhalten das Mapping bms. Und damit zu Schritt 2.

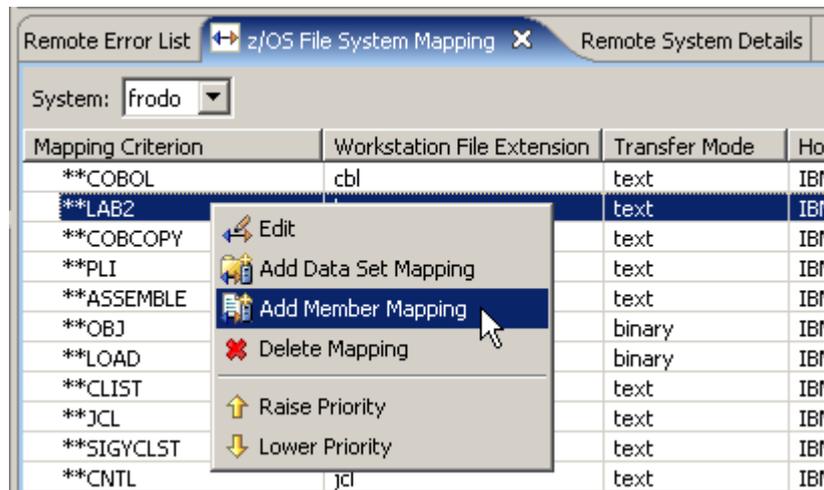


Abbildung 3.2.5

1kr auf **\*\*LAB2**, selektiere **Add Member Mapping** (nicht **Data Set Mapping** wie vorher).

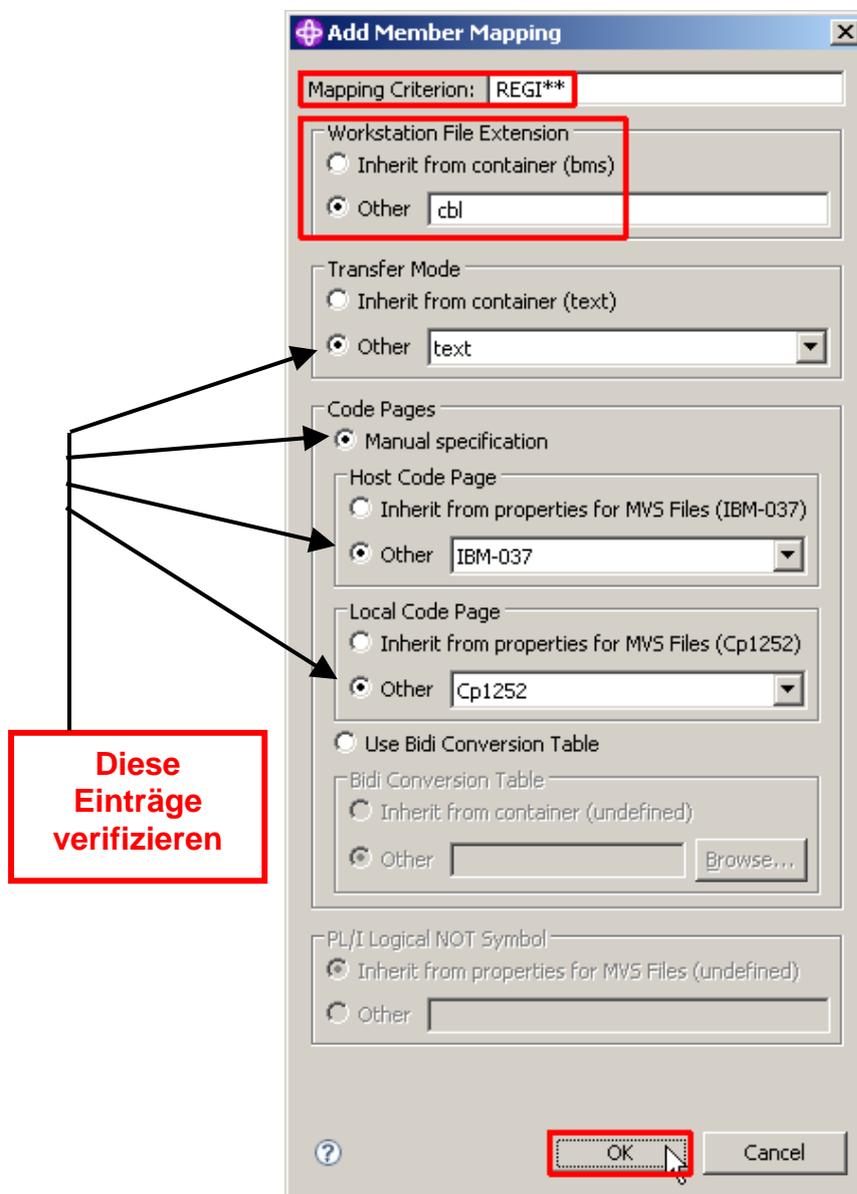


Abbildung 3.2.6

Als Mapping Criterion **REGI\*\*** eingeben, als Workstation File Extension selektiere **Other**, und **cbl** eingeben. Für den Rest die gezeigten Default Werte übernehmen. Click **OK**.

Erklärung: Alle Member des Data Sets mit der Endung **\*\*LAB2** werden als **bms** abgebildet (Schritt 1, Data Set Mapping), mit Ausnahme der Member, deren Name mit **REGI\*\*** anfängt (in Schritt 2, Add-Member Mapping). Letztere werden auf **cbl** abgebildet. Die **ASTERICS \*\*** stellen die Qualifier des Data Set Namens dar.

Mapping Criterion	Workstation File Extension	Transfer Mode	Host C
**COBOL	cbl	text	IBM-03
<b>**LAB2</b>	<b>bms</b>	<b>text</b>	<b>IBM-03</b>
REGI**	cbl	text	IBM-03
**COBCOPY	cpy	text	IBM-03
**PLI	pli	text	IBM-03

Abbildung 3.2.7

Expandiere das **+** von **\*\*LAB2**. Sie sehen ein anderes Mapping Kriterium unter **\*\*LAB2**. Damit wird jeder Member, der mit den 4 Buchstaben **REGI\*\*** im Partitioned Data Sets (PDS) **\*\*LAB2** anfängt, als **cbl** (COBOL) abgebildet. Die restlichen Member werden als **bms** abgebildet.

Hiermit haben wir erreicht, dass alle Member des Data Sets **PRAK032.POT.LAB2**, die mit den 4 Buchstaben **REGI** anfangen, unter Windows als Files mit der Extension **cbl** bearbeitet werden können

### Selbst-Test

- Befinden sich jetzt in dem Data Set **PRAK032.POT.LAB2** bereits die drei Member **Regio0A**, **Regio0B** und **Regio0C** ?
- Wenn nein, was haben wir dann erreicht ?

## 3.3 Kopieren lokaler Files in den allocated z/OS Data Set **PRAK032.POT.LAB2**

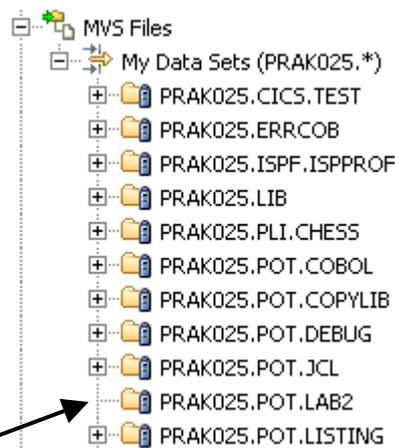


Abbildung 3.3.1

Der allocated Partitioned Data Set **PRAK032.POT.LAB2** ist leer (hat keine Member).

Wir kopieren jetzt drei COBOL Programme in drei Member des Partitioned Data Set PRAK032.POT.LAB2. Diese 3 Cobol Programme befinden sich bereits in dem Windows XP Directory Ihrer virtuellen Gast Maschine unter den Namen REGIO0A.cbl, REGIO0B.cbl, und REGIO0C.cbl. Nach dem Kopieren sollen sie unter z/OS als die drei Member

- PRAK032.POT.LAB2(REGI0A),
- PRAK032.POT.LAB2(REGI0B), und
- PRAK032.POT.LAB2(REGI0C)

verfügbar sein.

1k auf den Remote Systems View, 1k auf das + Zeichen der Node Local Files. Navigieren Sie zu dem Verzeichnis C:\WDz\_Tutorials\Tutorial03\Resources.

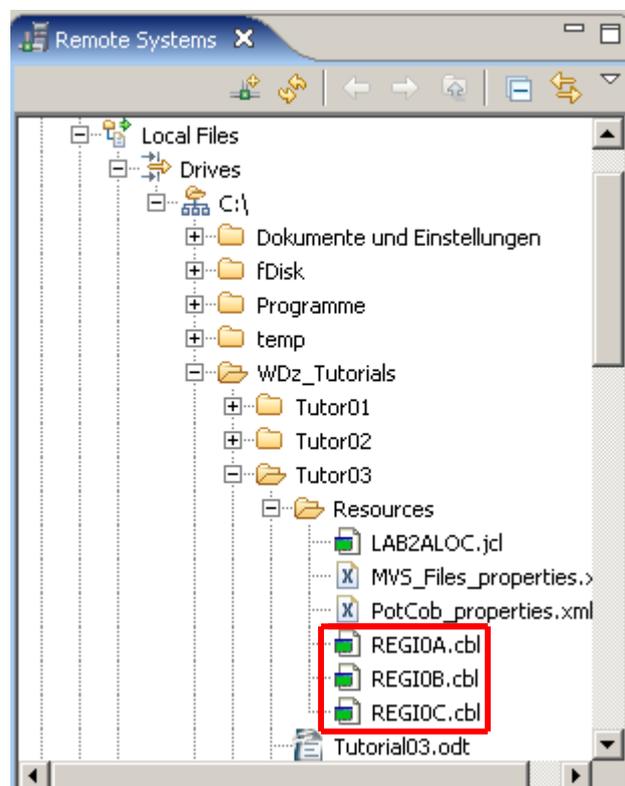


Abbildung 3.3.2

1k auf REGIO0A.cbl, Shift Taste halten, und 1k auf REGIO0C.cbl um alle drei cbl-Files zu markieren. Alternativ 1k auf REGIO0A.cbl, den Ctr Key halten, 1k auf REGIO0B.cbl und dann auf REGIO0C.cbl um alle 3 cbl-Files zu markieren.



Abbildung 3.3.3

1kr auf die markierte gruppe, select Copy. Die 3 Dateien sind nun im Clipboard.

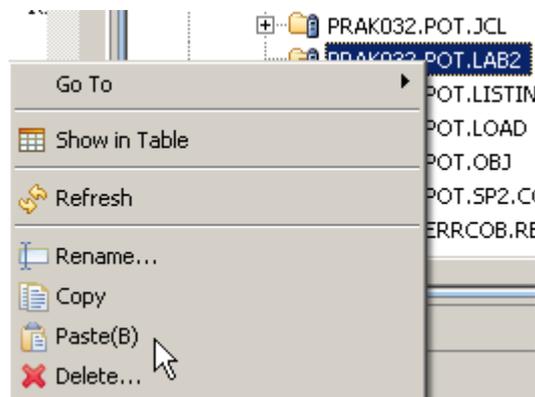


Abbildung 3.3.4

1kr auf PRAK032.POT.LAB2, selektiere Paste, und PRAK032 durch Ihre UserID ersetzen.

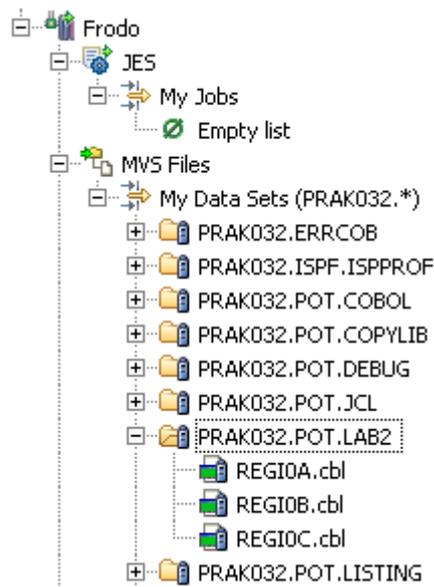


Abbildung 3.3.5

Ihr neuer PDS Data Set hat nun drei neue Members, REGIOA.cbl, REGIOB.cbl und REGIOC.cbl.

Ohne das gerade durchgeführte Member Mapping hätten die drei Files nicht die Zuordnung .cbl, die gebraucht wird, um mit den Kopien auf der workstation synchron zu arbeiten !!!

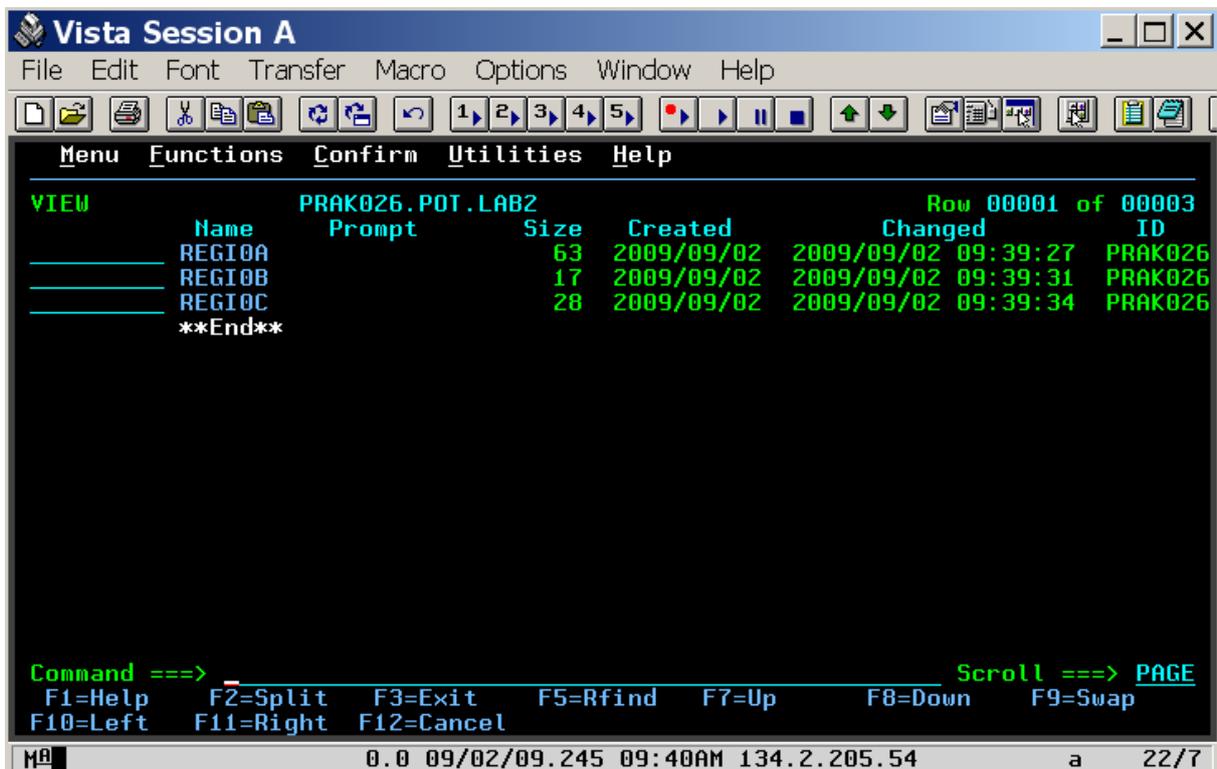


Abbildung 3.3.6

Wenn Sie unter ISPF nachsehen, PRAK032.POT.LAB2 hat nun 3 Member REGIO0A, REGIO0B, and REGIO0C . Die Zuordnung .cbl ist unter RDz, nicht aber unter ISPF sichtbar.

## Selbst-Test

- Befinden sich jetzt in dem Data Set PRAK032.POT.LAB2 bereits die drei Member Regio0A, 0B und 0C ?
- Wie ist das geschehen ?

## 4. Arbeiten mit entfernten Data Sets

### 4.1 Wiederholung

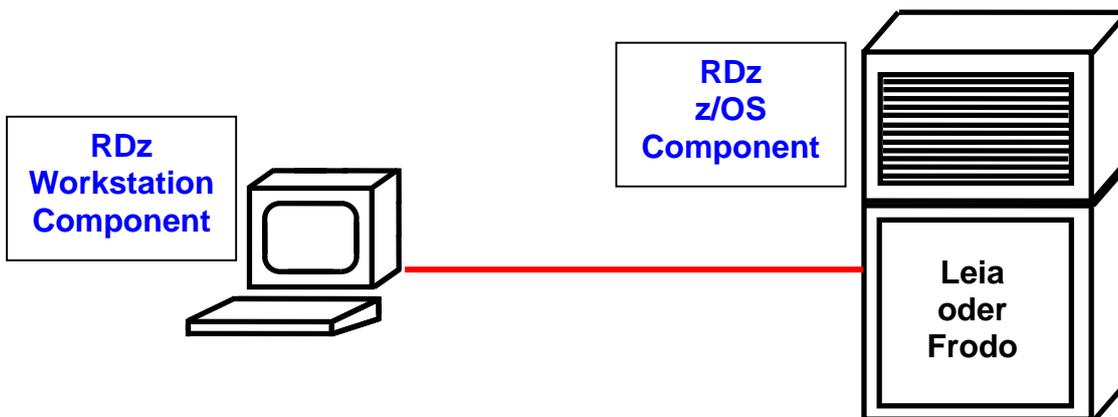


Abbildung 4.1.1

Eine übliche RDz Konfiguration besteht aus einer Workstation Komponente (installiert auf Ihrem PC), sowie einer z/OS Komponente, die auf einem Mainframe installiert ist. Die Workstation speichert Ihren Workspace.

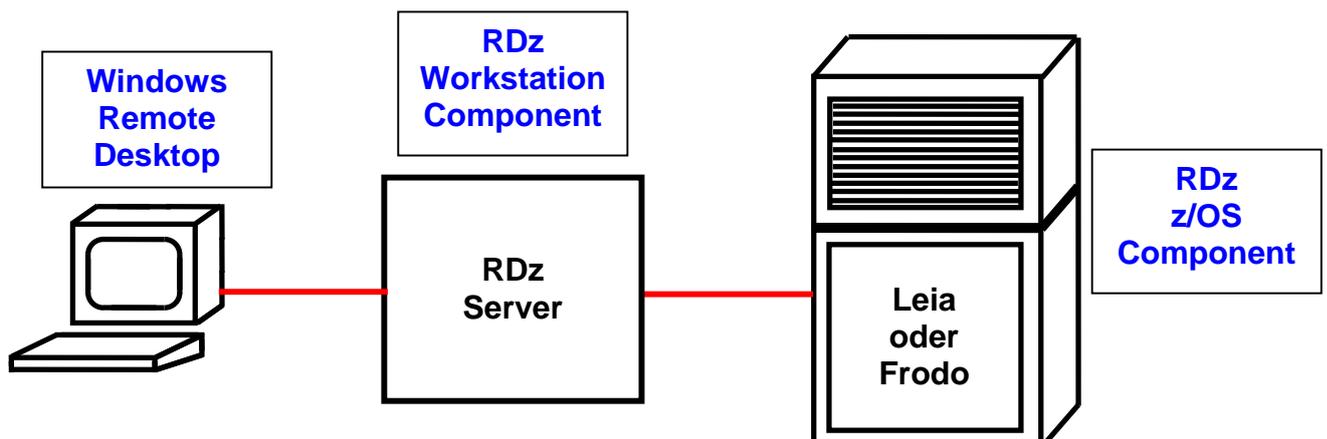


Abbildung 4.1.2

In unserem Fall benutzt Ihr PC nur den Standard Windows Remote Desktop um auf den RDz Server an der Universität Leipzig zuzugreifen. Die RDz Workstation Komponente ist auf dem Leipziger RDz Server in einer virtuellen Windows Maschine installiert. Letztere speichert auch Ihren RDz Workspace.

In beiden Fällen befinden sich die Data Sets Ihres Cobol Projektes unter z/OS unter Ihrer User ID (PRAK032 in diesem Beispiel). Zusätzlich ist eine 1:1 Kopie der Data Sets in Ihrem Workspace gespeichert. Wir haben die Data Set mapping Funktion in Abschnitt 2.2 dieses Tutorials durchgeführt, um Inkompatibilitäten in der Naming Syntax für Windows Files und z/OS Data Sets aufzulösen..

In Abschnitt 2.3 kopierten wir die 3 Files REGI0A.cbl, REGI0B.cbl und REGI0C.cbl in Member des z/OS Data Sets PRAK032.POT.LAB2 von irgendwoher. In unserem Fall befand sich das "irgendwoher" ebenfalls auf dem RDz Server, aber außerhalb der RDz Workstation Komponente, und damit außerhalb Ihres Workspace. Wir werden später die drei Files ebenfalls in Ihren workspace kopieren.

## 4.2 Arbeiten mit Projekten und Subprojekten

Das Arbeiten mit RDz erfolgt traditionell mit Hilfe von Projekten. Ein Projekt enthält eine Sammlung von Dateien. Sie können mehrere Projekte auf Ihrer Workstation speichern, aber Sie werden zu jedem Zeitpunkt nur mit einem einzigen Projekt arbeiten.

Sie definieren ein **lokales Projekt**, wenn es Ihre Absicht ist, COBOL (oder PL/I) Code auf der Workstation zu entwickeln und dort auszuführen. Sie arbeiten zum Beispiel mit einem lokalen Projekt, wenn Ihre neue COBOL-Anwendung auf einer Windows-Workstation ausgeführt werden soll. Rational Developer für System z bietet eine Reihe von Tools an, mit deren Hilfe Sie lokale Projekte erstellen, welche Host-basierten Ressourcen benutzen. Beispiele für solche Host-basierte Ressourcen sind Enterprise Service Tools und der Database Application Generator. Lokale Projekte benutzen lokale Compiler und Übersetzer für die Syntaxprüfung. Sie zeigen Abhängigkeiten auf und bewirken Projekt-Builds. Rational Developer für System z bietet auch Werkzeuge für die Remote-Synchronisierung an. Dies erlaubt es, ein lokales Projekt mit einem Remote-System zu verbinden. Mit der Remote-Synchronisierung können Sie COBOL (und PL/I) -Anwendungen lokal entwickeln und das Ergebnis dann auf ein Remote-System übertragen

Sie definieren ein **z/OS Projekt**, wenn es Ihre Absicht ist, COBOL (oder PL/I) Code zu entwickeln, der unter z/OS gespeichert und dort ausgeführt wird. Für z/OS bedeutet dies, dass Sie Ihre Assets (vor allem z/OS-Datensätze) in einer Projektstruktur verknüpfen, nämlich in „z/OS-Projekte“. z/OS-Projekte dienen in erster Linie als Werkzeug für das Organisieren und Arbeiten mit mehreren Load Modulen.

z/OS-Projekte sind Container, die ein oder mehrere Subprojekte enthalten. Subprojekte können entweder MVS Subprojekte oder Unix System Services (USS) Subprojekte sein, je nachdem, ob Sie mit PDS Members in MVS-Data Sets, oder Dateien in einem USS hierarchischen Filesystem arbeiten. In beiden Fällen bleiben ihre Assets (z.B. z/OS Datensätze) auf dem Host, und sind nur logisch mit den Teilprojekten verknüpft.

Ein Teilprojekt enthält die Dateien und Programme für ein einziges Load-Modul. (Mit mehreren Teilprojekten, können Sie mehrere Lademodule mit einem einzigen Build-Befehl bauen). Teilprojekte werden in einem bestimmten Remote-System erstellt; die Eigenschaften dieses Systems wirken als Default Werte für Subprojekt Eigenschaften.

Dieses Kapitel führt Sie durch den Prozess der Erstellung eines Remote-Projekts, und das Festlegen seiner Properties (Eigenschaften) für verschiedene Sprachen und Runtimes (Laufzeiten).



Abbildung 4.2.1

Selektiere File → New → Project... von der Menu Bar.

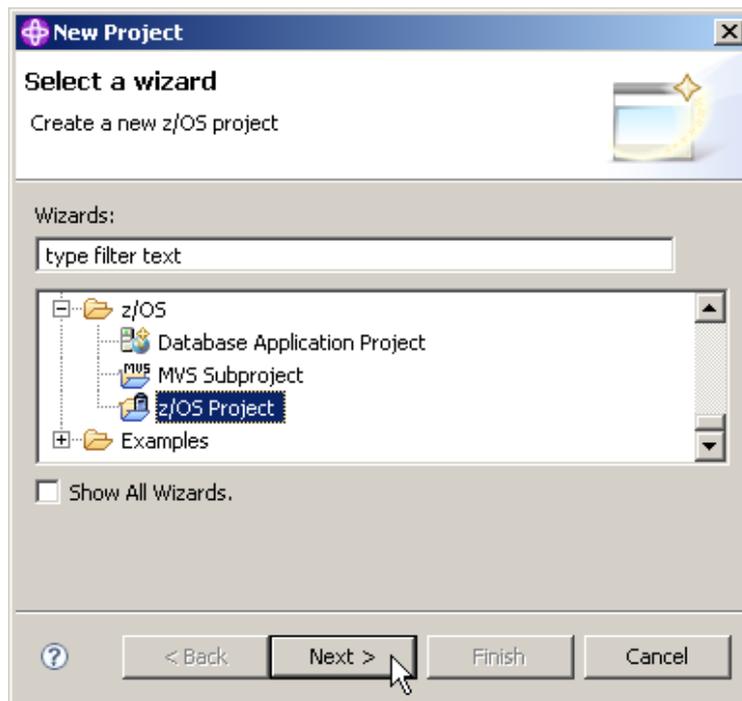


Abbildung 4.2.2

Scroll down, expandiere z/OS, selektiere z/OS Projekt, 1k auf Next.

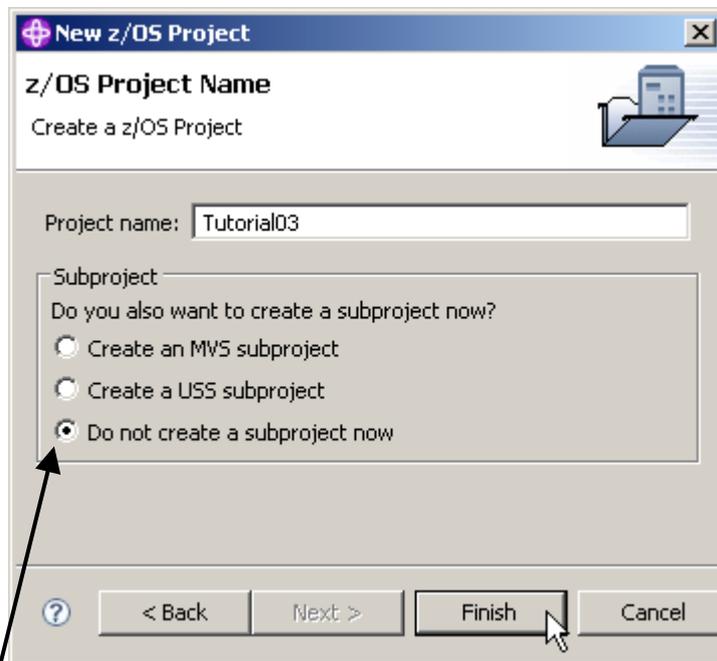


Abbildung 4.2.3

Im z/OS Project Name Panel, benennen Sie das Projekt "Tutorial03". Selektieren Sie den "Do not create a subproject now" Radio Button.

(Anmerkung: We werden ein Subproject später erstellen. Es könnte auch hier sofort geschehen.)

1k auf Finish. Dies erzeugt ein z/OS Parent Projekt.

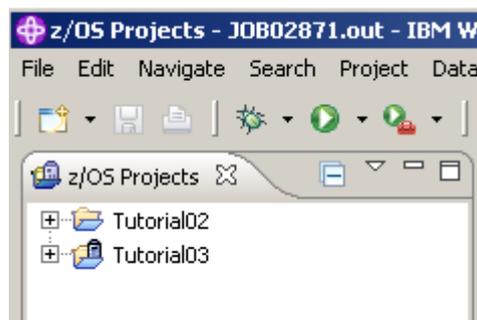


Abbildung 4.2.4

Erinnern Sie sich: Der z/OS Projects View lists lokale Projekte in Ihrem RDz Workspace. Project Tutorial02 war in dem vorhergehenden RDz Tutorial 02 "Local Cobol" erstellt worden.

### 4.3 Create an MVS Subproject

Zuallererst können Sie ein MVS-Subprojekt nur erstellen, wenn Sie mit dem System(Frodo in diesem Fall) verbunden sind. Ein MVS-Projekt befindet sich jeweils in einem von zwei Zuständen:

Im Online-Zustand ist das Projekt mit dem System verbunden, auf das sich das Projekt bezieht. Sie können die Dateien direkt ändern, die in diesem System gespeichert sind.

Im Offline-Zustand kann das Projekt nur auf Dateien auf dem Arbeitsplatzrechner zugreifen. Dies können neue Dateien oder auch Kopien von z/OS Dateien sein.

Wenn Sie sich von z/OS trennen (disconnect), können Sie die Data Sets und Member angeben, die zum Arbeitsplatzrechner übertragen werden sollen. Wenn Sie zurück zum Online-Zustand umschalten, werden die angegebenen Dateien zum Großrechner automatisch hoch geladen. Dies erfolgt mit einer Bestätigungsmeldung, welche Sie davon abhält, Ressourcen unbeabsichtigt zu überschreiben.

In dem z/OS Projects View, klicken Sie auf Tutorial03. Selektieren Sie New → MVS Subproject von dem Context Menü.

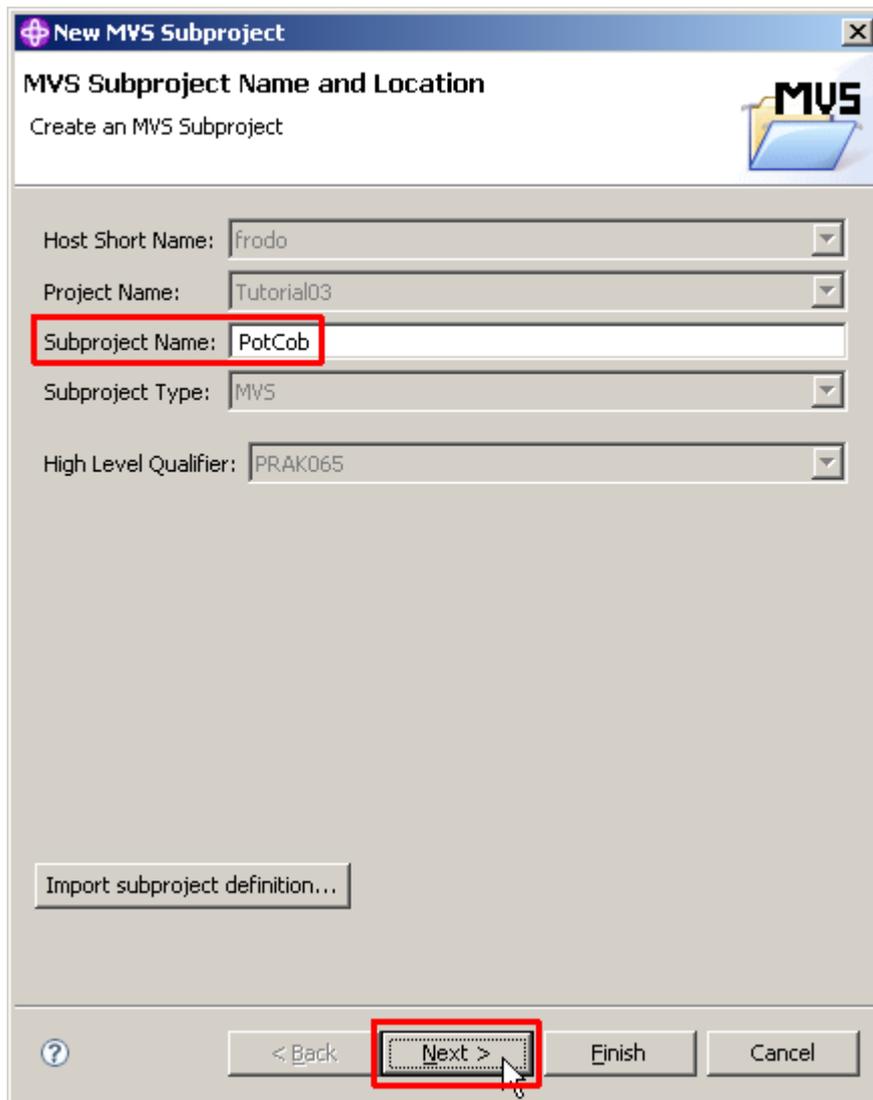


Abbildung 4.3.1

Benennen Sie Ihr MVS Subproject PotCob, und click Next..

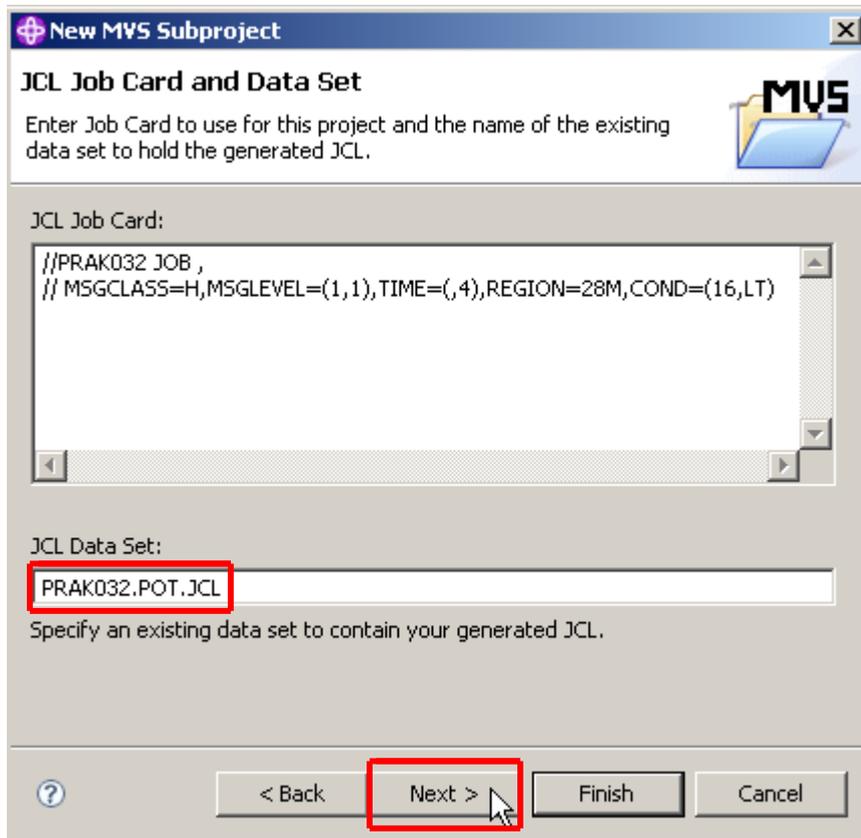


Abbildung 4.3.2

Auf dem "JCL Job Card and JCL Data Set panel", ändere den JCL Data Set to PRAK###.POT.JCL where ### Ihre UserID ist (032 in diesem Beispiel). Nicht den Default Wert! (PRAK###.GENERATED.JCL) benutzen !!!!!

Erinnern Sie sich: Die Job Card wurde von den z/OS System Settings geerbt (inherited), die wir in Abschnitt 2.1 „z/OS System-Einstellungen“ spezifiziert hatten )

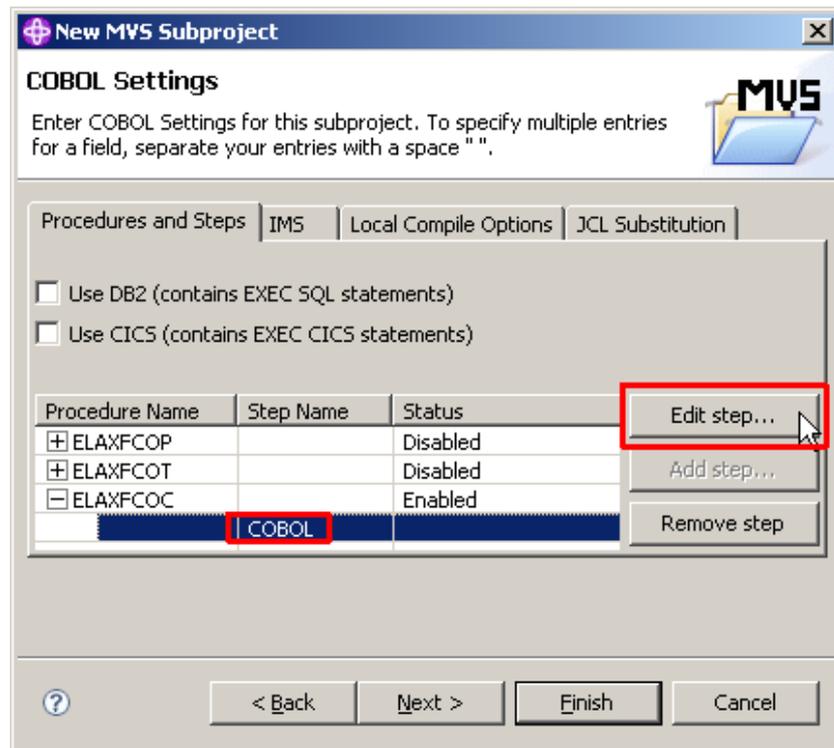


Abbildung 4.3.3

In dem COBOL Settings Panel, expandiere die ELAXFCOC Prozedur. Selektiere COBOL, 1k auf Edit Step. ELAXFCOC ist der Name einer z/OS Prozedur, welche das Kompilieren des Programms bewirkt.

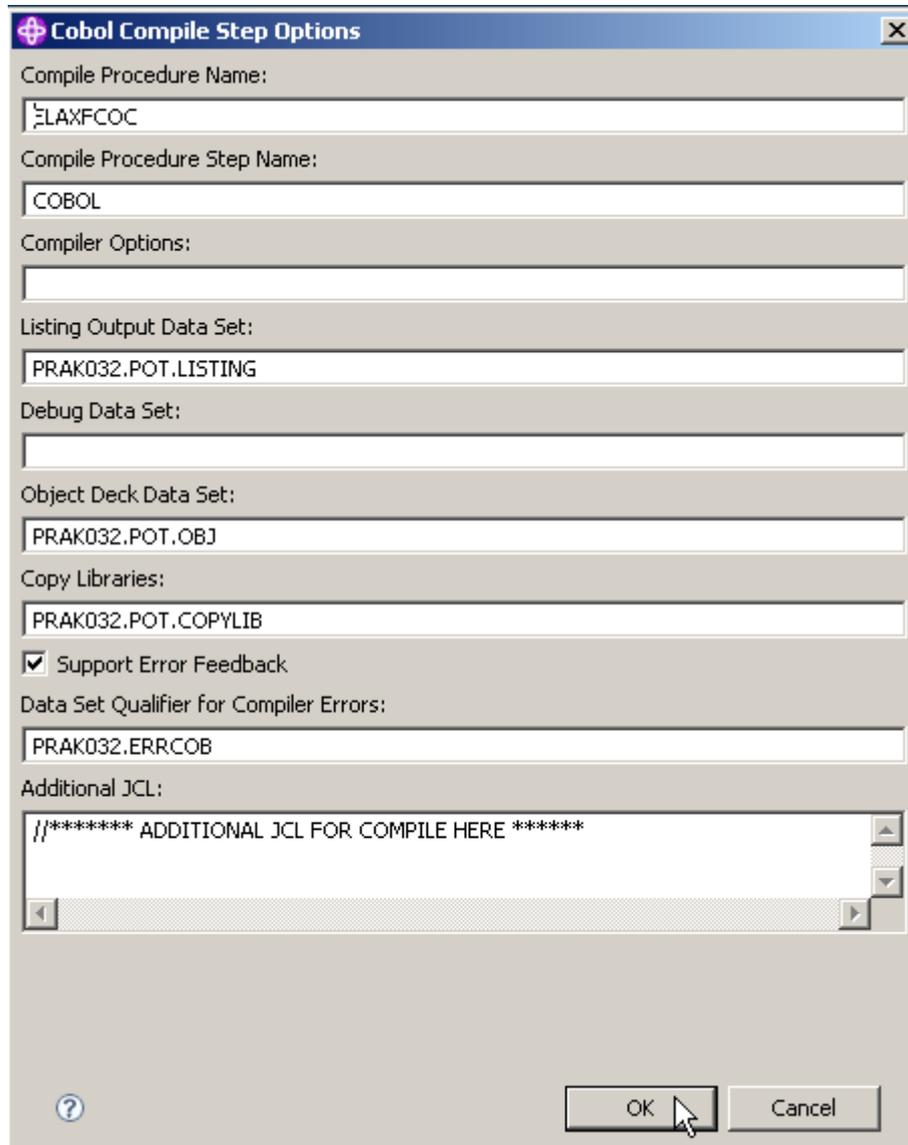


Abbildung 4.3.4

Schauen sie sich die Einträge an. Beachten Sie, die Werte wurden "inherited" von den frodo System Settings in Abschnitt 2.1.. Benutzen sie einen Text Editor, um sich die „MVS\_Files\_properties.xml“ File anzusehen. Sie finden dort ähnliche Werte.

Beachten Sie auch, der <HLQ> wurde automatisch auf Ihre User ID geändert (PRAK032 in diesem Beispiel). 1k auf OK.

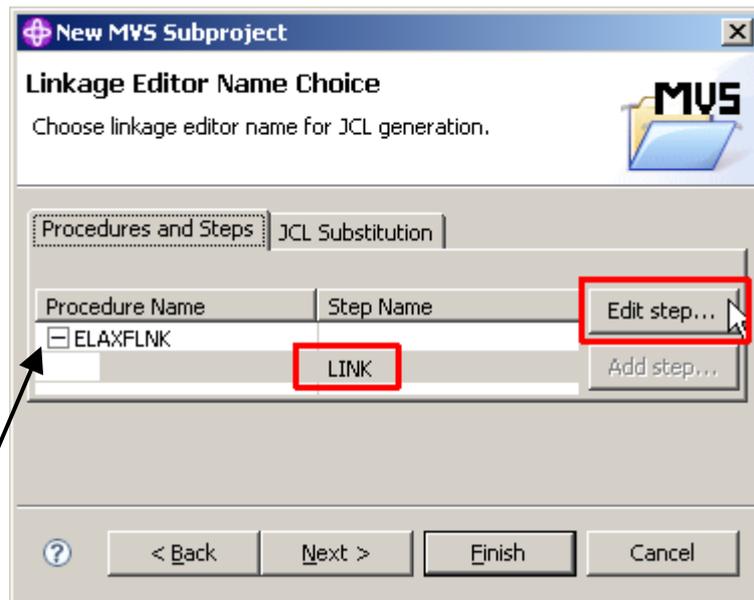


Abbildung 4.3.5

Mehrfach auf Next klicken, bis Sie zu dem „Linkage Editor Name Choice“ Panel kommen (etwa 6 mal).

Die ELAXFLNK Prozedur expandieren.

Selektiere LINK und click Edit Step.

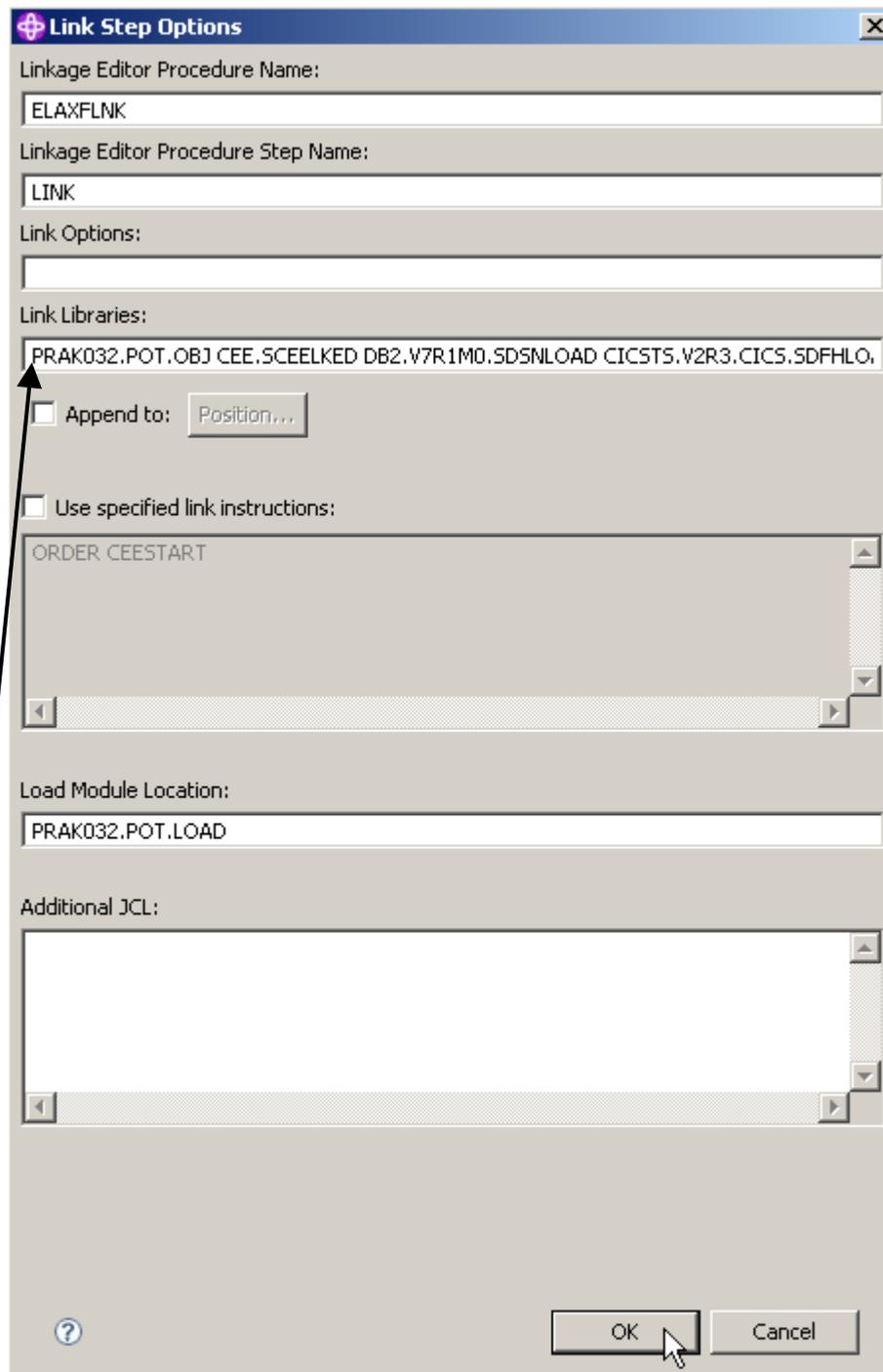


Abbildung 4.3.6

Das Link step Options Panel erscheint. In dem Link Libraries Field sehen sie diesen Eintrag:  
**PRAK032.POT.OBJ CEE.SCEELKED DB2.V7R1M0.SDSNLOAD CICSTS.V2R3.CICS.SDFHLOAD**

Ein potentielles Problem ist, dass nicht alle dargestellten User IDs gegen Ihre eigene User ID ausgetauscht werden. Bitte alle Änderungen sorgfältig überprüfen.

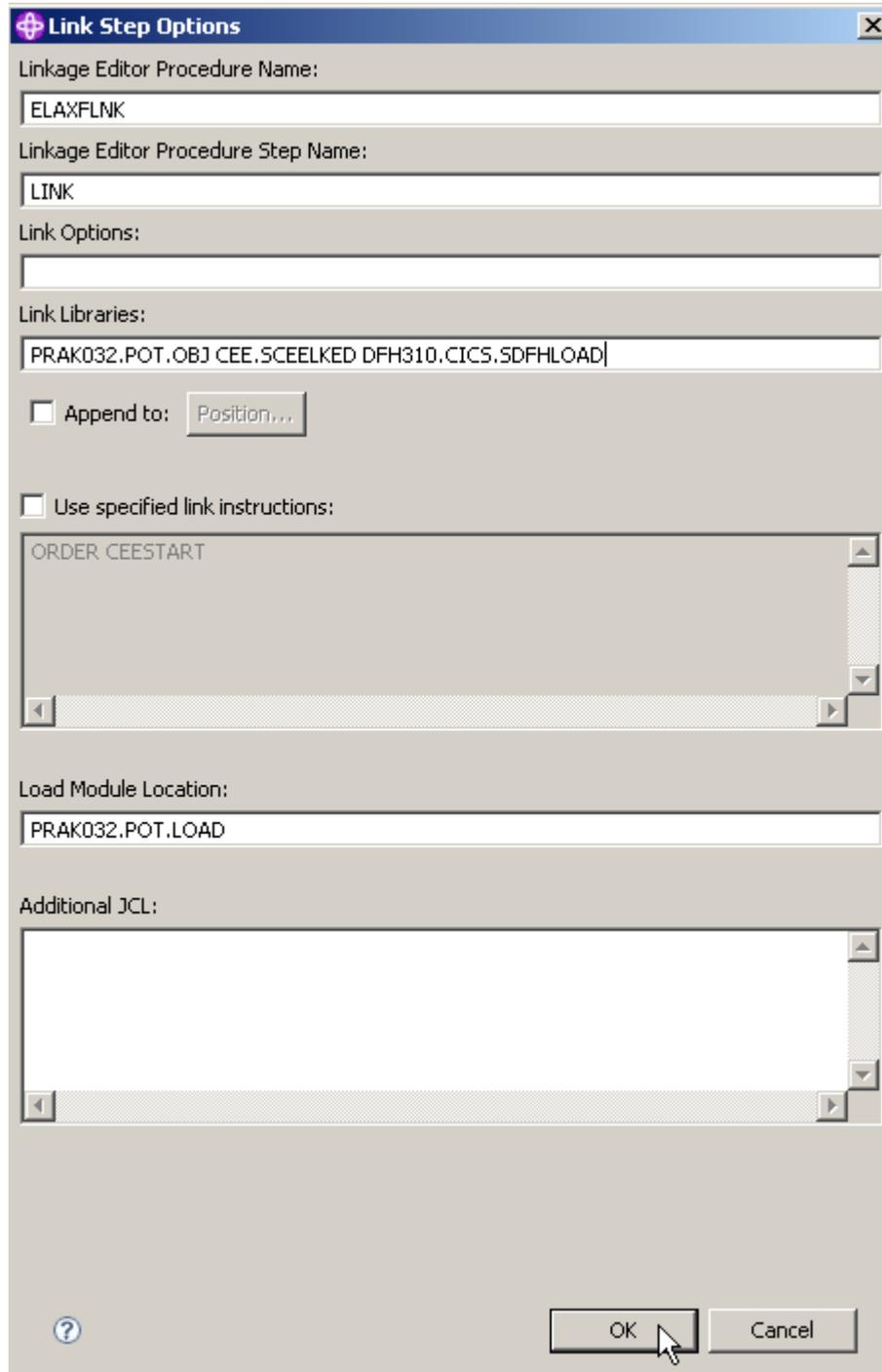


Abbildung 4.3.7

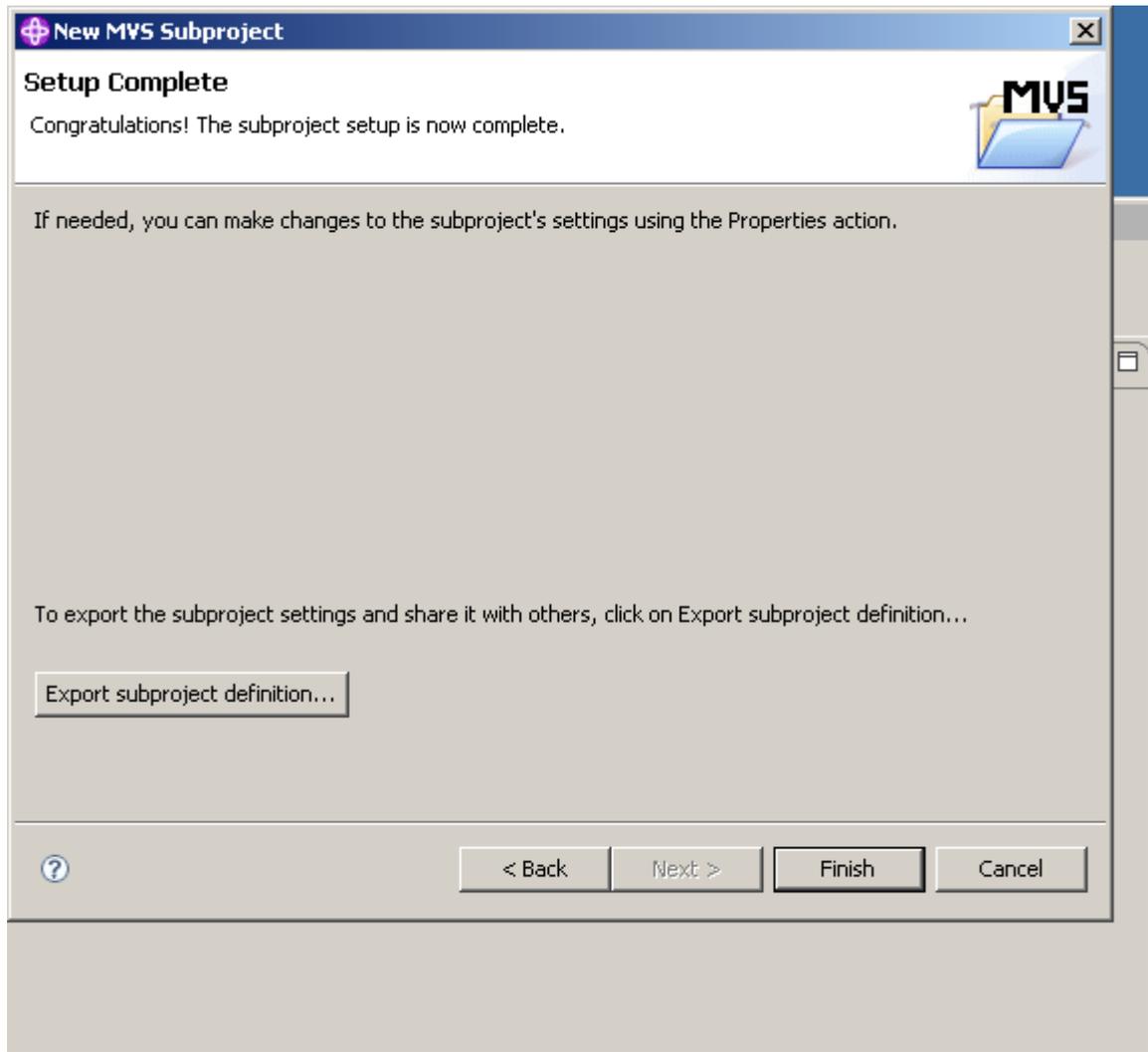
Den Link Libraries Eintrag ändern auf

PRAK032.POT.OBJ CEE.SCEELKED DFH310.CICS.SDFHLOAD

space

Die Link Libraries enthalten mehr als einen Data Set. To concatenate datasets, just add as many as you want with a space between. "CEE.SCEELKED" und „DFH310.CICS.SDFHLOAD“ sind die Namen von zwei z/OS System Libraries, die für das Linking benötigt werden.

Click OK.

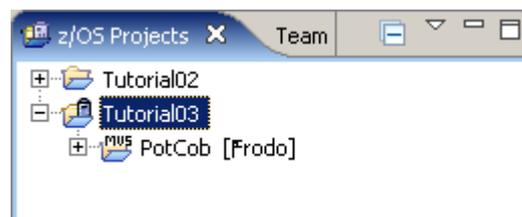


**Abbildung 4.3.8**

**Mehrfaches Klick auf Next (etwa 3 mal) bis Sie zu dem „Setup Complete“ Panel kommen.**

**Anmerkung: Für zukünftigen Gebrauch können sie die Subproject Definitionen an dieser Stelle in eine xml-file exportieren.**

**1k auf Finish um das Fenster zu schließen.**



**Abbildung 4.3.9**

**Sie sollten das neu erstellte MVS Project PotCob in Ihrem z/OS Projects View sehen. Wenn sie auf das + zeichen klicken um das Projekt zu expandieren, werden sie finden, dass PotCob leer ist.**

Ihr Workspace C:\Dokumente und Einstellungen\LEIPZIG\IBM\rationalSDP7.0\workspace

c:\Dokumente und Einstellungen\Leipzig\Ibm\rationalSDP7.0\workspace\*.*				
↑Name	Ext	Size	Date	Attr
↑[...]		<DIR>	09/02/2009 10:25---	
↑[.metadata]		<DIR>	09/02/2009 09:34---	
↑[project]		<DIR>	03/31/2008 10:55---	
↑[RemoteSystemsConnections]		<DIR>	03/31/2008 17:40---	
↑[RemoteSystemsTempFiles]		<DIR>	09/01/2009 17:06---	
↑[SCLMConfigProject]		<DIR>	03/31/2008 10:57---	
↑[Tutor02a]		<DIR>	09/01/2009 18:13---	
↑[Tutorial02]		<DIR>	09/01/2009 15:59---	
↑[wdz_proj_Tutorial03]		<DIR>	09/02/2009 10:03---	
↑[wdz_proj_Tutorial03_PotCob]		<DIR>	09/02/2009 10:25---	

Abbildung 4.3.10

hat jetzt zwei neue Einträge. Ihr Inhalt sieht so aus:

c:\Dokumente und Einstellungen\Leipzig\Ibm\rationalSDP7.0\workspace\wdz_proj_Tutorial03\*.*				
↑Name	Ext	Size	Date	Attr
↑[...]		<DIR>	09/02/2009 10:03---	
↑.project		296	09/02/2009 10:03-a--	
↑Tutorial03	xml	483	09/02/2009 10:25-a--	

Abbildung 4.3.11

und:

c:\Dokumente und Einstellungen\Leipzig\Ibm\rationalSDP7.0\workspace\wdz_proj_Tutorial03_PotCob\*.*				
↑Name	Ext	Size	Date	Attr
↑[...]		<DIR>	09/02/2009 10:25---	
↑.project		306	09/02/2009 10:25-a--	
↑PotCob	properties	14,725	09/02/2009 10:25-a--	
↑PotCob	xml	589	09/02/2009 10:25-a--	

Abbildung 4.3.12

Anmerkung: Wenn Sie in den folgenden Schritten Probleme haben sollten, könnten Sie die gerade erstellte Konfiguration von der xml-file PotCob\_properties.xml importieren. Sie finden diese auf der Workstation unter C:\Tutorials\Tutorial03\Resource.

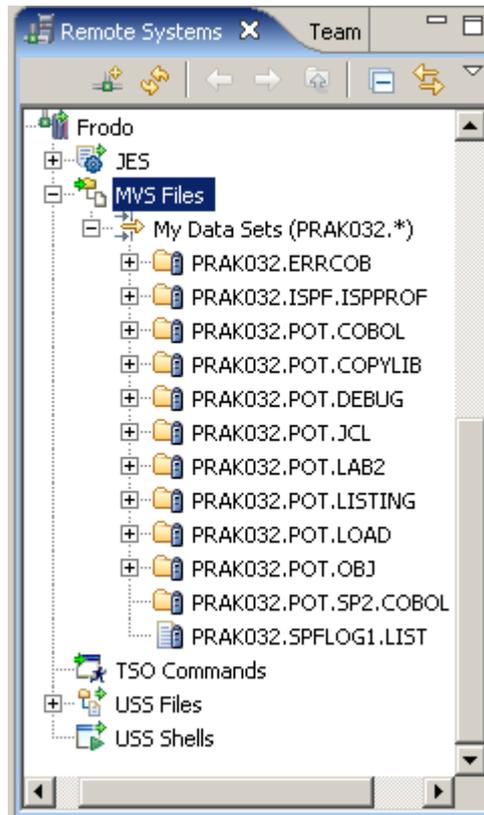


Abbildung 4.3.13

Remote Systems sollte jetzt so aussehen:

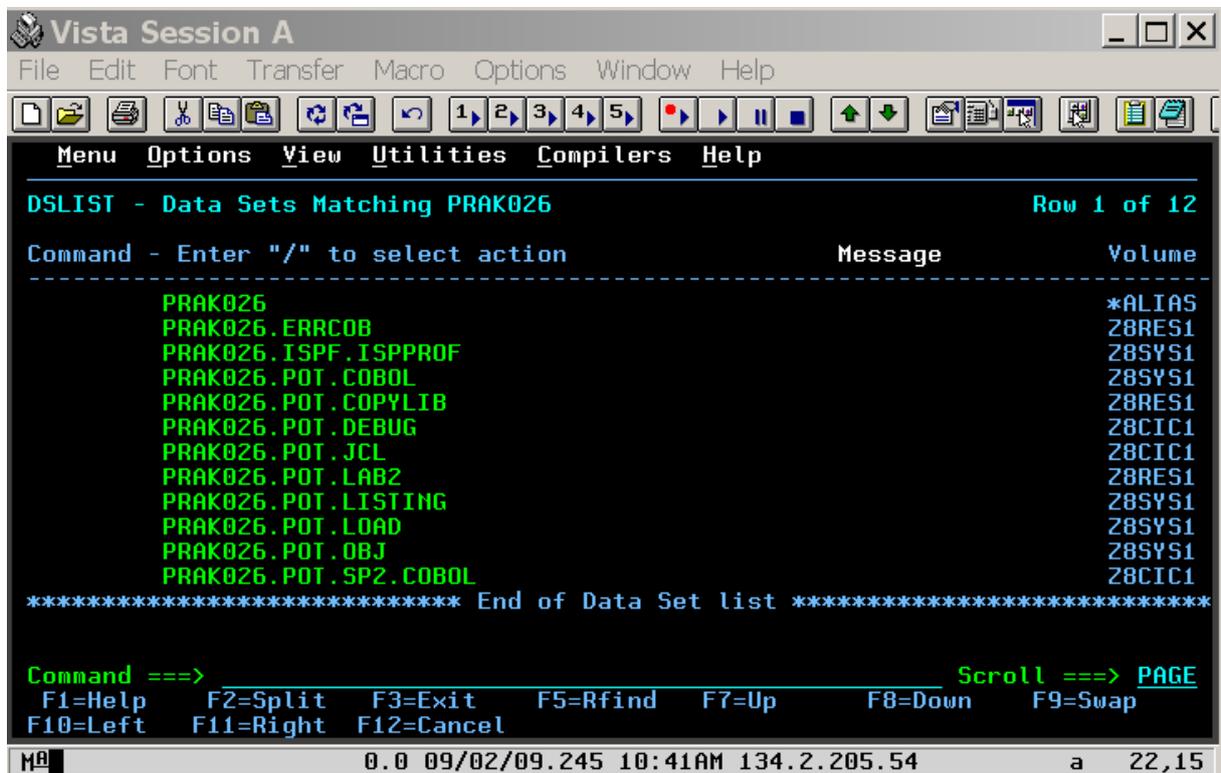


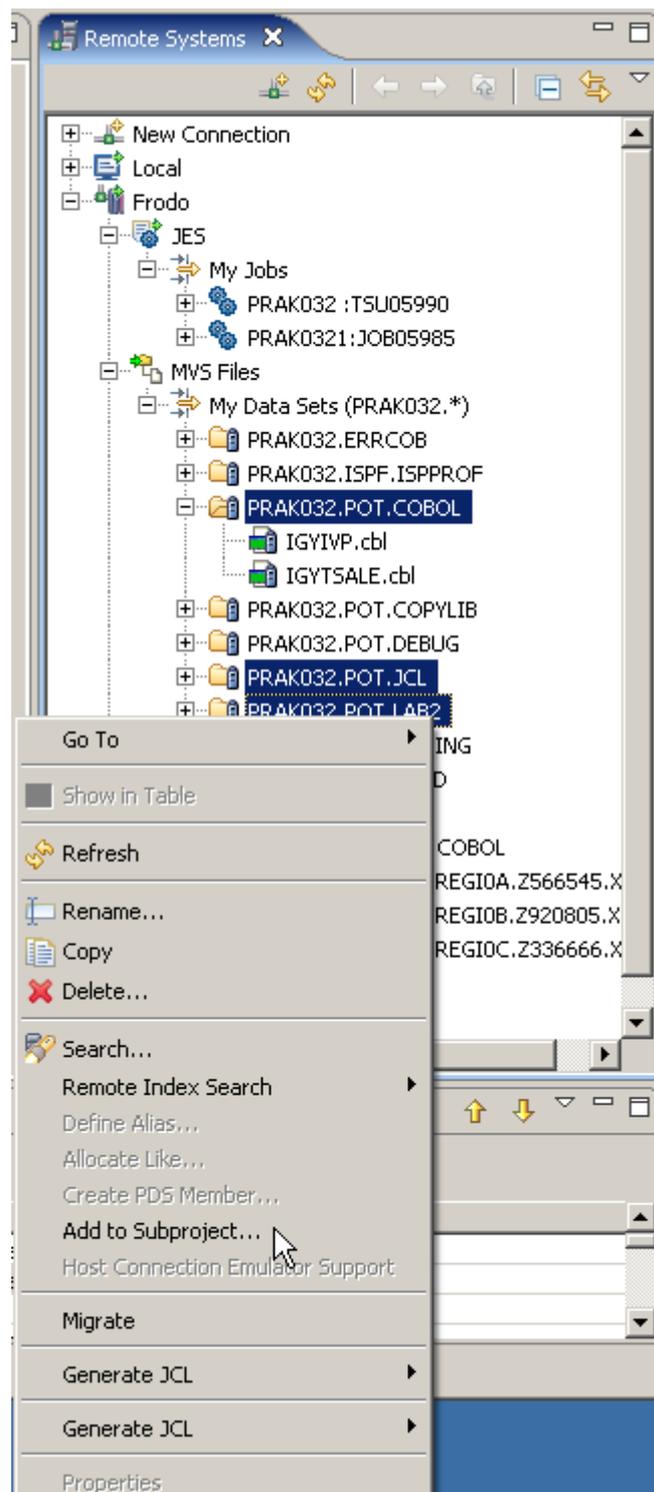
Abbildung 4.3.14

Wenn Sie einen 3270 Emulator benutzen, würde das ISPF DLIST Panel jetzt so aussehen.

## **4.4 Ressourcen dem Projekt hinzufügen**

**Wir installieren jetzt einige Data Sets unter Ihrer z/OS User ID. Hierzu fügen wir diese Ihrem Remote Projekt hinzu.**

**Wechseln sie zum Remote Systems View. Expandieren Sie die MVS Files unter Frodo bis Sie die Data sets sehen, welche Sie vorher allocated haben.**



**Abbildung 4.4.1**

Den CTRL key drücken (-> multiple select), markieren Sie die drei Data Sets PRAK032.POT.COBOL, PRAK032.POT.JCL und PRAK032.POT.LAB2. 1kr auf die Gruppe, selektiere „Add to Subproject“.

(Ersetzen Sie 032 mit Ihrer User ID)

Anmerkung: Sie könnten das Gleiche mit Drag and Drop erreichen..

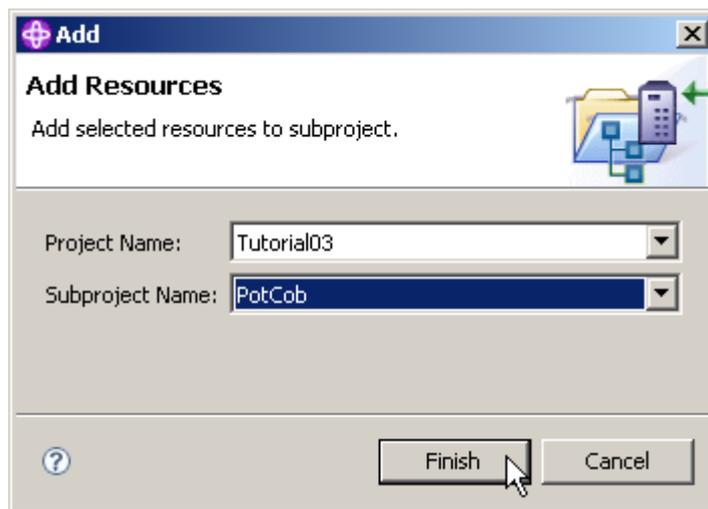


Abbildung 4.4.2

Akzeptieren Sie den Projekt Namen Tutorial03 und den Subprojekt Namen PotCob. 1k um die Data Sets hinzuzufügen.

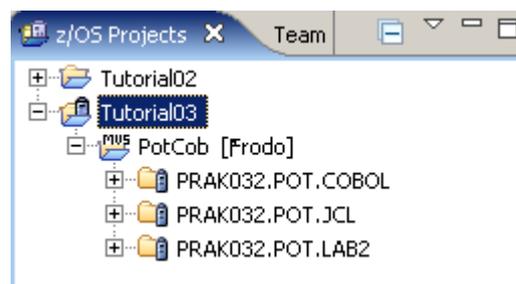


Abbildung 4.4.3

Wechseln Sie zum z/OS Projects View. Sie werden sehen, dass die drei Data Sets dem PotCob Projekt hinzugefügt wurden. Der z/OS Projects View sollte wie hier gezeigt aussehen.

Denken Sie daran: Das Arbeiten mit RDz erfolgt traditionell mit Projekten. Für z/S bedeutet dies, dass Ihre Assets (besonders z/OS-Datensätze) in eine Projektstruktur zu verknüpfen sind: in ein z/OS-Projekt.

Diese Projekte können entweder Unix System Services (USS) oder MVS Subprojekte sein, je nachdem, ob Sie mit Dateien eines USS hierarchischen Filesystems oder mit PDS-Membren in MVS arbeiten. In beiden Fällen bleiben Ihre Assets auf dem Mainframe-Host und werden nur logisch mit den Subprojekten verknüpft.

### Selbst-Test

- Befinden sich jetzt Kopien der in dem Data Set PRAK032.POT.LAB2 enthaltenen drei Member Regio0A, Regio0B und Regio0C in ihrem Workspace ?
- Von welchen z/OS Data Sets befinden sich jetzt ebenfalls Kopien in Ihrem workspace ?

## **4.5 Compile/Link/Execute des entfernten Cobol Programms.**

**Wir werden jetzt Ihre Programme unter z/OS kompilieren, verknüpfen (link) und ausführen. Sie verwenden hierfür eine Funktion „JCL Generation“. Wir benutzen die Properties des Remote z/OS-Projektes, um JCL nur für Compile, Compile und Link oder Compile, Link und Go zu generieren.**

**Wir verwenden diese Funktion nur zu Demonstrationszwecken. In der realen Welt würden Sie vermutlich Compile Verfahren einer bestehende Mainframe-Installation verwenden, oder ein Quellcode-Management-System wie SCLM (ISPF Software Configuration and Library Manager) oder den Endeavor Software Change Manager von Computer Associates benutzen (<http://www.ca.com/us/products/detail/ca-endavor-software-change-manager.aspx>).**

**In diesem Beispiel brauchen wir eine kompilierte Version von REGI0B.cbl, bevor wir mit der Kompilation von REGI0A.cbl fortfahren können. Wir fangen daher mit der Kompilation von REGI0B.cbl an.**

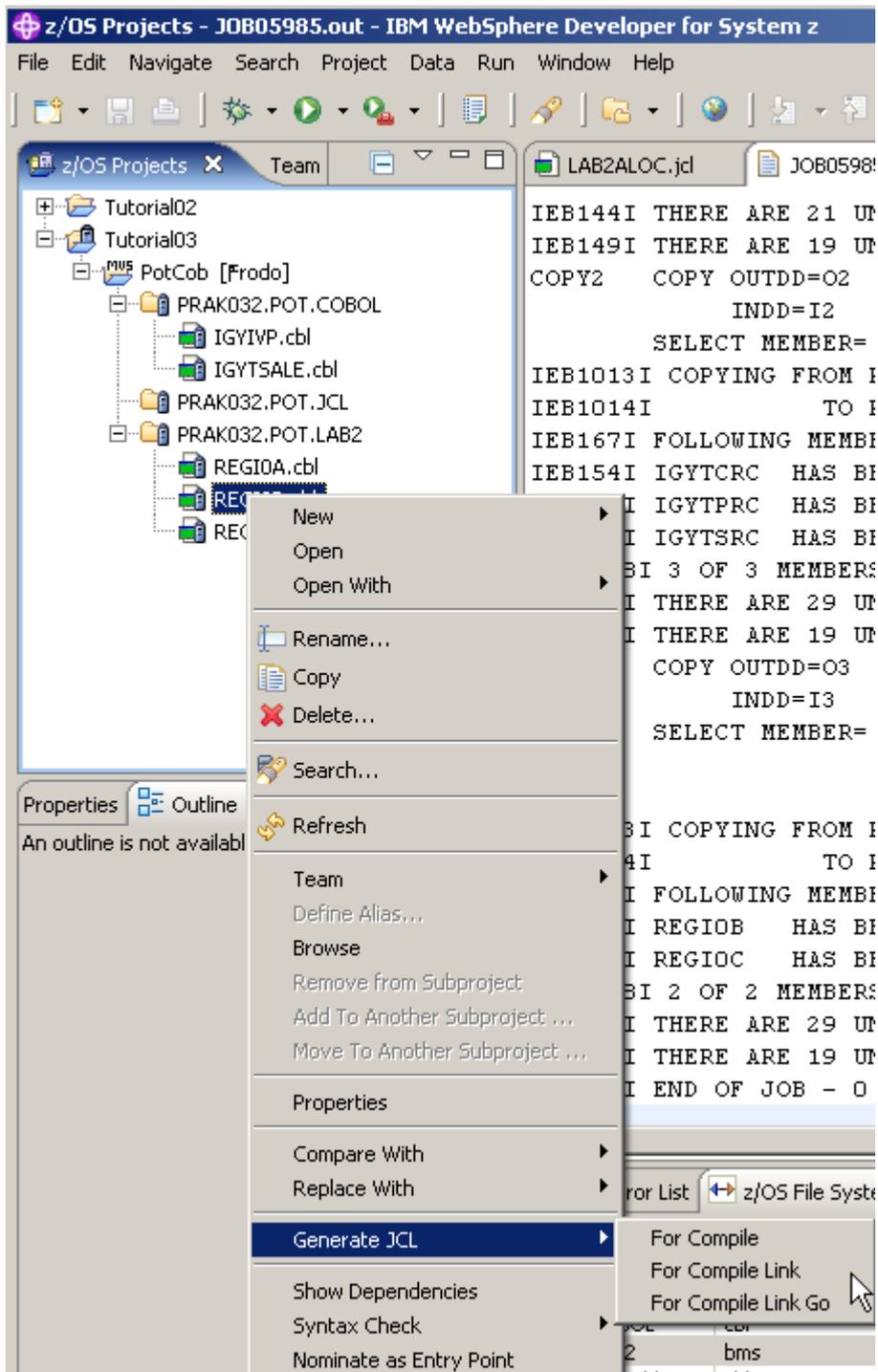


Abbildung 4.5.1

Im z/OS projects View, expandiere PotCob und PRAK032.POT:LAB. 1kr auf REGI0B.cbl. Selektiere Generate JCL → For Compile Link. 1k

Bitte beachten, sowohl in dem z/OS Projects View als auch in dem Remote Systems View ist PRAK032.POT.JCL leer ( enthält keine Member).



Abbildung 4.5.2

Im „JCL Data Set and Member Name“ Window hat der JCL Data Set Name den Wert, den Sie in Ihren Project Settings spezifizierten.

### muss durch Ihre User ID ersetzt werden. Click OK.

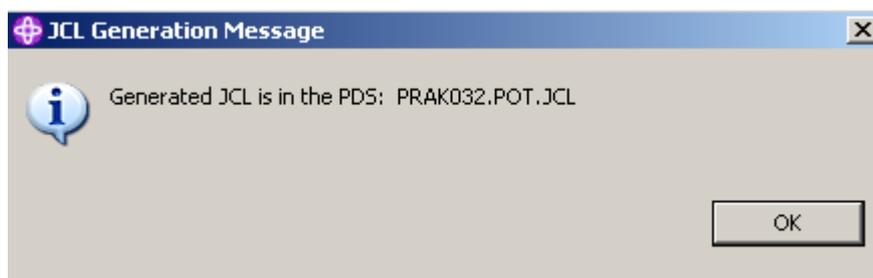


Abbildung 4.5.3

Sie sollten diese Nachricht sehen (Hier für die UserID PRAK032). Nochmals click OK.

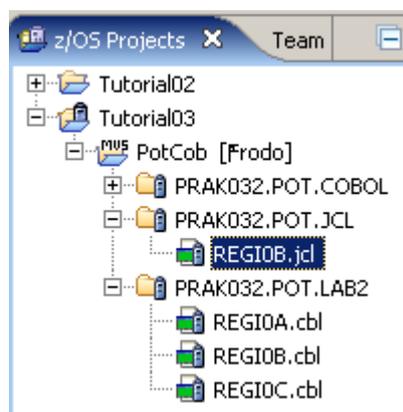


Abbildung 4.5.4

Gehen Sie zum z/OS Projects View. Sie werden sehen, dass REGIOB.jcl als Member von PRAK032.POT.JCL generiert wurde.

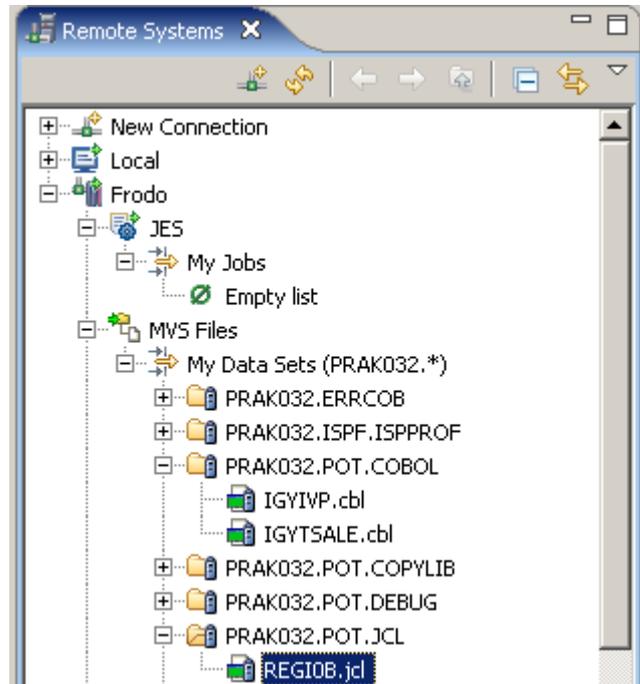


Abbildung 4.5.5

Beachten Sie, im Remote Systems View enthält Frodo einen Data Set mit dem gleichen Namen.

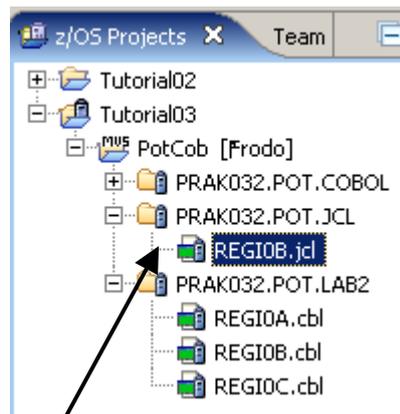


Abbildung 4.5.6

Im z/OS Projects view , 2k auf REGIO0B.jcl um die JCL File im Editor Fenster zu öffnen.

```

LAB2ALOC.jcl  JOB05969.out  REGIOB.jcl x
Line 1      Column 1      Insert
  /-+-----1-----+-----2-----+-----3-----+-----4-----+-----5-
000001 //PRAK032 JOB ,
000002 //  MSGCLASS=H,MSGLEVEL=(1,1),TIME=(,4),REGION=28M
000003 //*
000004 //STPO000 EXEC PROC=ELAXFCOC,
000005 //  CICS=,
000006 //  DB2=,
000007 //  COMP=
000008 //COBOL.SYSPRINT DD DSN=PRAK032.POT.LISTING(REGIO
000009 //                DISP=SHR
000010 //COBOL.SYSLIN DD DSN=PRAK032.POT.OBJ(REGIOB) ,
000011 //                DISP=SHR
000012 //COBOL.SYSLIB DD DSN=PRAK032.POT.COPYLIB,DISP=SH
000013 //COBOL.SYSXMLSD DD DUMMY
000014 //COBOL.SYSIN DD DSN=PRAK032.POT.LAB2 (REGIOB) ,DIS
sub

```

Abbildung 4.5.7

Stellen Sie sicher, der Job Name ist PRAK###, wobei ### Ihre UserID ist (PRAK032 in diesem Beispiel).

In der Command Zeile (unten im Editor Fenster), sub eingeben, und dann Enter, to submit das JCL Script. Mit der Escape Taste können Sie den Cursor auf der Command Zeile positionieren.

```

000012 //COBOL.SYSLIB DD DSN=PRAK032.POT.COPYLIB,DISP=SH
000013 //COBOL.SYSXMLSD DD DUMMY
000014 //COBOL.SYSIN DD DSN=PRAK032.POT.LAB2 (REGIOB) ,DIS
JOBID: JOB05971

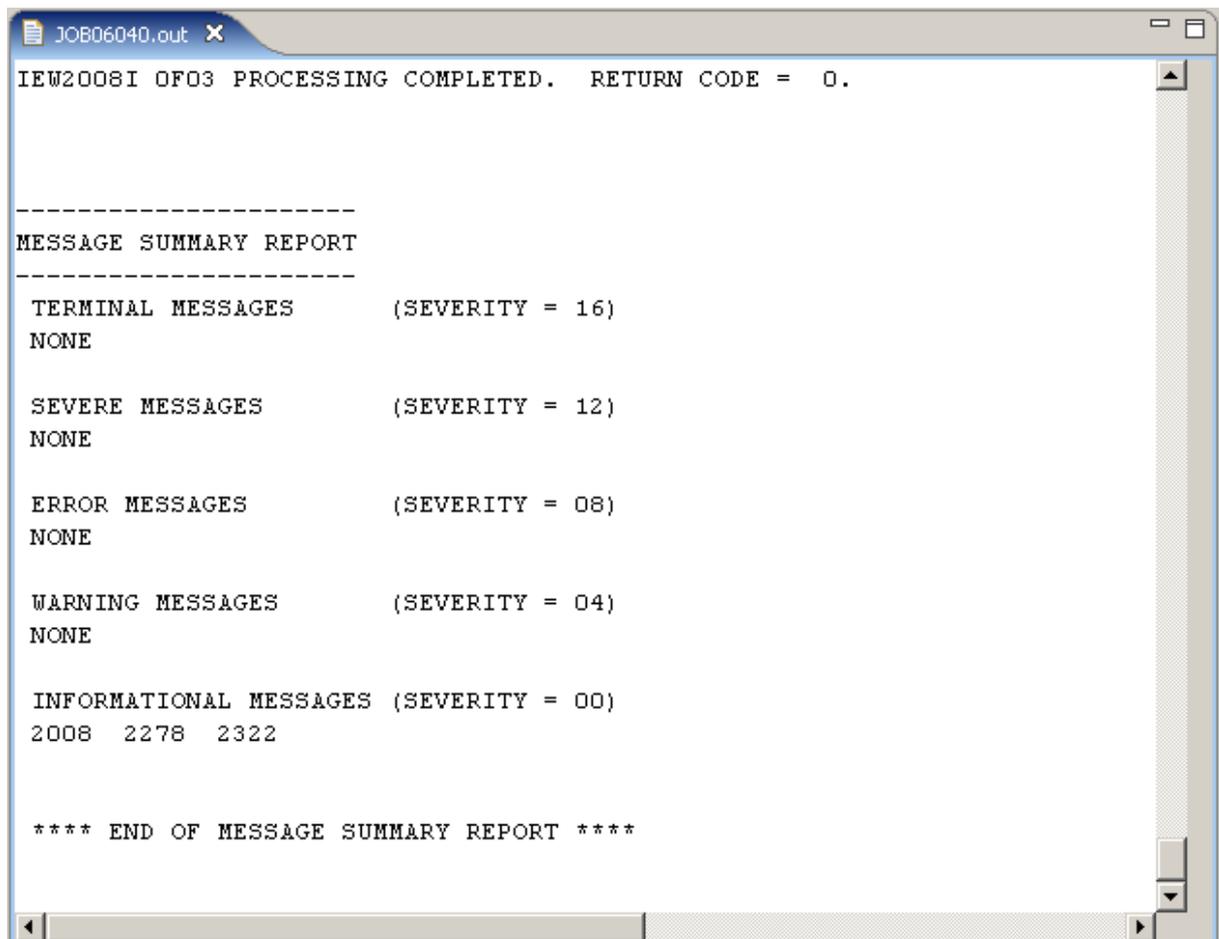
```

Abbildung 4.5.8

Angenommen, Sie sind mit dem Host verbunden. Sie erhalten eine Bestätigung mit der JOB ID.

Mit dieser JOB ID können nun Sie nach dem Job im JES Subsystem in dem Remote Systems Window schauen. Überprüfen Sie die erfolgreich Durchführung des Jobs. Die JOBID sollte unter " Frodo – JES – My Jobs" aufgeführt sein.

Sie müssen möglicherweise ein Disconnect und Reconnect für Frodo ausführen, um es zu sehen.



```
JOB06040.out X
IEW2008I OF03 PROCESSING COMPLETED. RETURN CODE = 0.

-----
MESSAGE SUMMARY REPORT
-----
TERMINAL MESSAGES      (SEVERITY = 16)
NONE

SEVERE MESSAGES        (SEVERITY = 12)
NONE

ERROR MESSAGES         (SEVERITY = 08)
NONE

WARNING MESSAGES       (SEVERITY = 04)
NONE

INFORMATIONAL MESSAGES (SEVERITY = 00)
2008 2278 2322

**** END OF MESSAGE SUMMARY REPORT ****
```

Abbildung 4.5.9

2k auf die JOB ID in dem Remote Systems View um es in dem Editor Window zu öffnen. Scroll down zum Ende. Sie sollten etwas ähnliches sehen wie in Abb. 4.5.9 dargestellt.

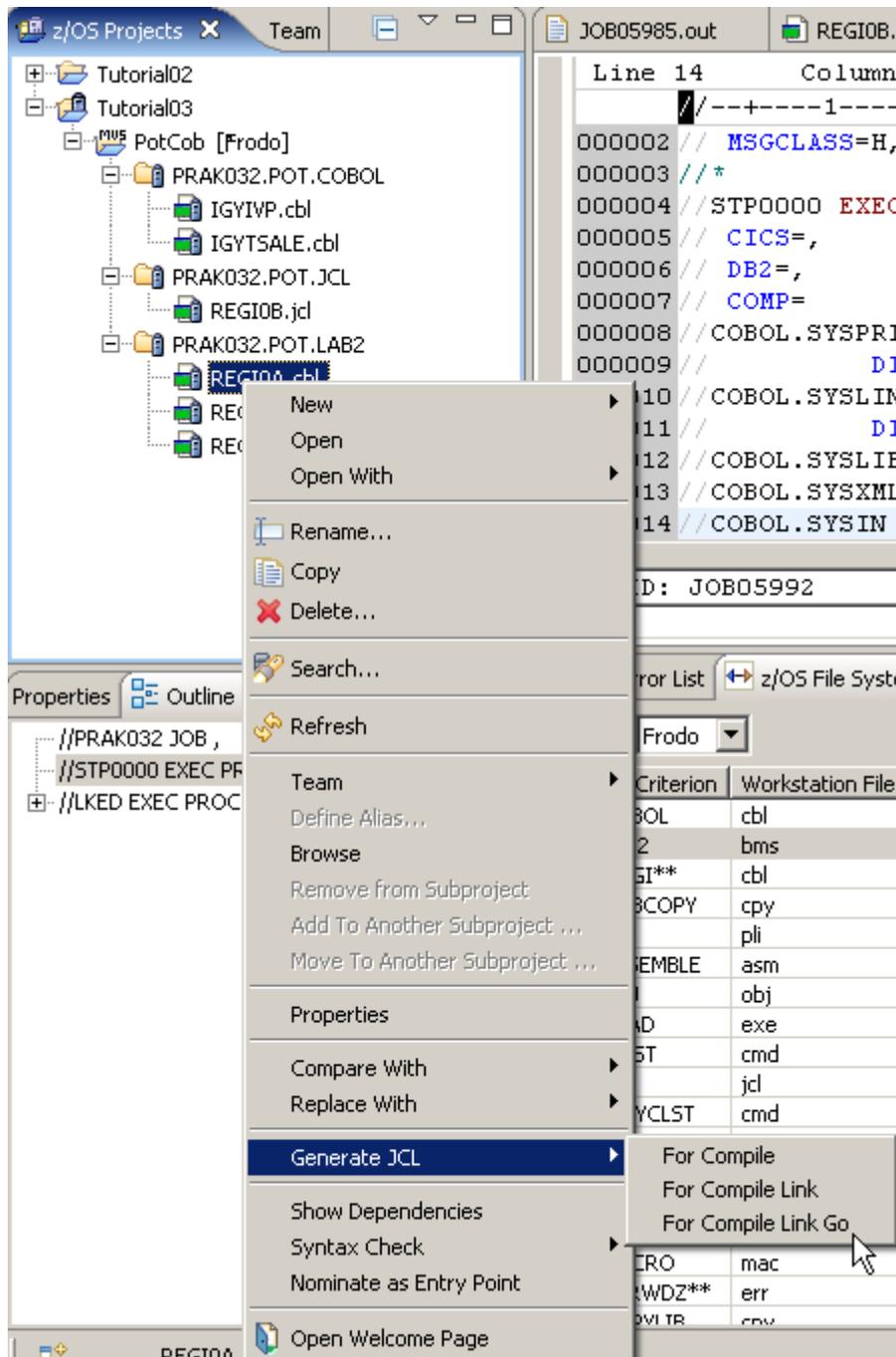
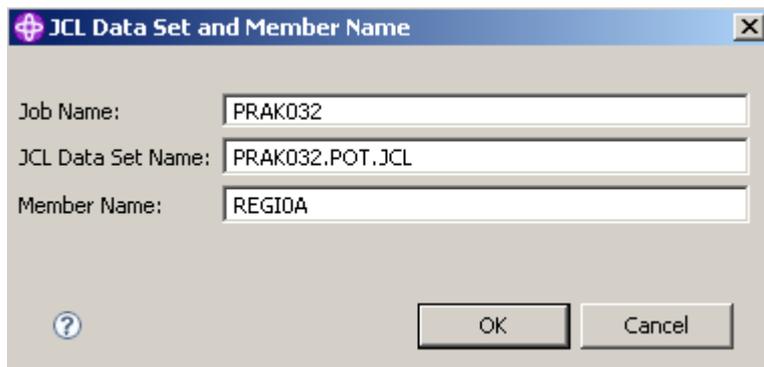


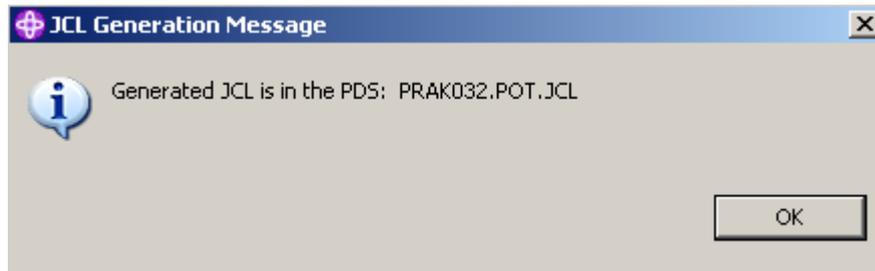
Abbildung 4.5.10

1kr auf REGI0A.cbl. Selektiere Generate JCL → For Compile Link Go.



**Abbildung 4.5.11**

**Verifizieren Sie, dass alles richtig ist. Ersetzen Sie PRAK032 durch Ihre User ID. Click OK.**



**Abbildung 4.5.12**

**Click OK.**

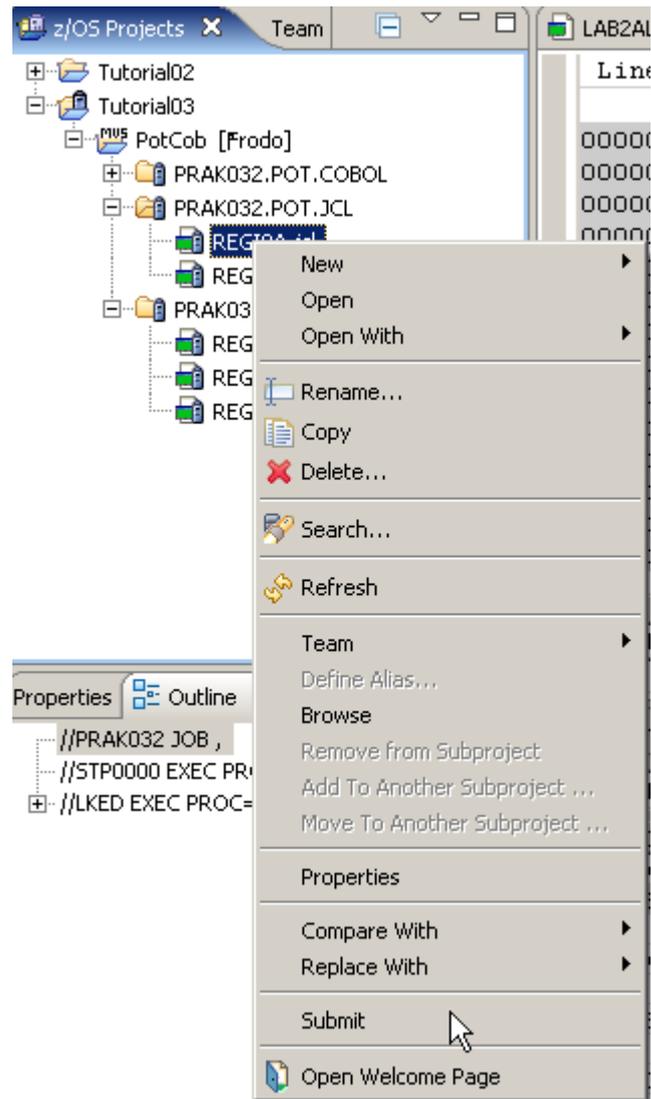


Abbildung 4.5.13

Die neu erstellte File REGI0A.jcl befindet sich in PRAK032.POT.JCL.  
1kr; selektiere Submit.

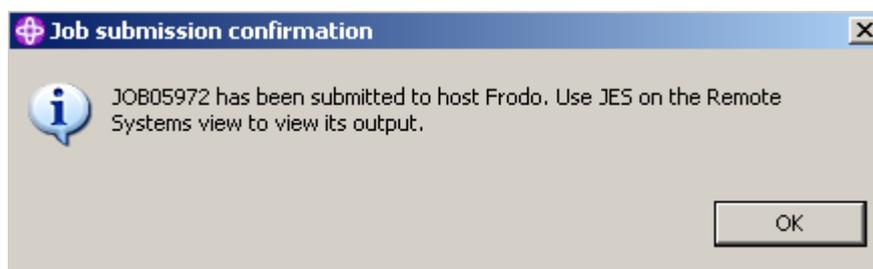
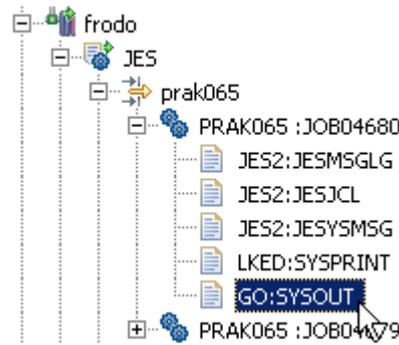


Abbildung 4.5.14

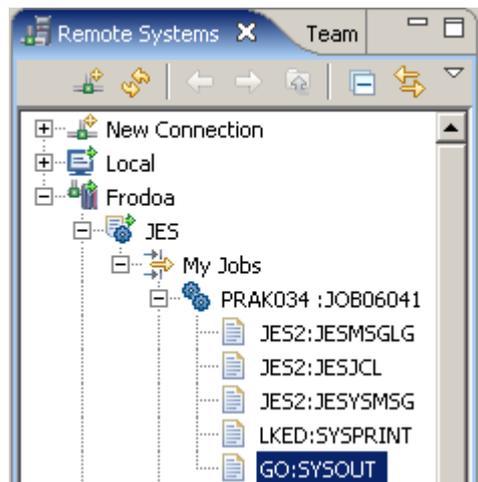
Click OK.



**Abbildung 4.5.15**

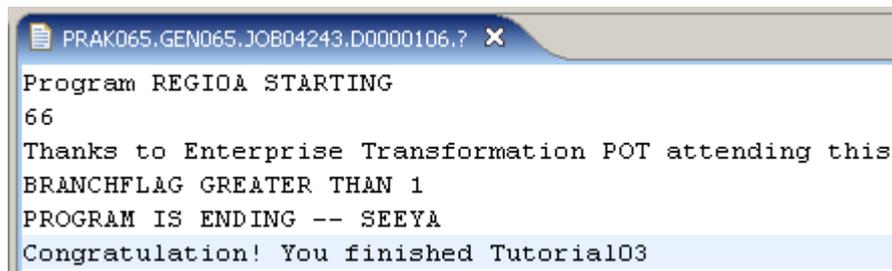
Um den Output Ihres Programms zu sehen, wechseln sie zum Remote Systems Window. Expandiere JES und My Jobs.

Schauen Sie nach der JOB ID Ihres submitted Jobs. Expandieren Sie, ....



**Abbildung 4.5.16**

und 2k auf GO:SYSOUT. Sie müssen möglicherweise ein Disconnect und Reconnect für Frodo ausführen, um es zu sehen.



**Abbildung 4.5.17**

Falls Sie alles richtig gemacht haben, sollten Sie diesen Output im Editor Fenster sehen.

### **Selbst-Test**

- Schauen Sie sich den Quelltext der drei Programme REHIO0A, REGIO0B und REGIO0C an. Warum wurde REGIO0B als erstes Kompiliert ?
- Was ist mit der Kompilierung von REGIO0C ?

**Herzlichen Glückwunsch,  
Sie haben Tutorial03 erfolgreich abgeschlossen !**