EJB3 Zugriff auf DB2
Installation und Konfiguration

Unsere EJB Tutorials bestehen aus drei Teilen:

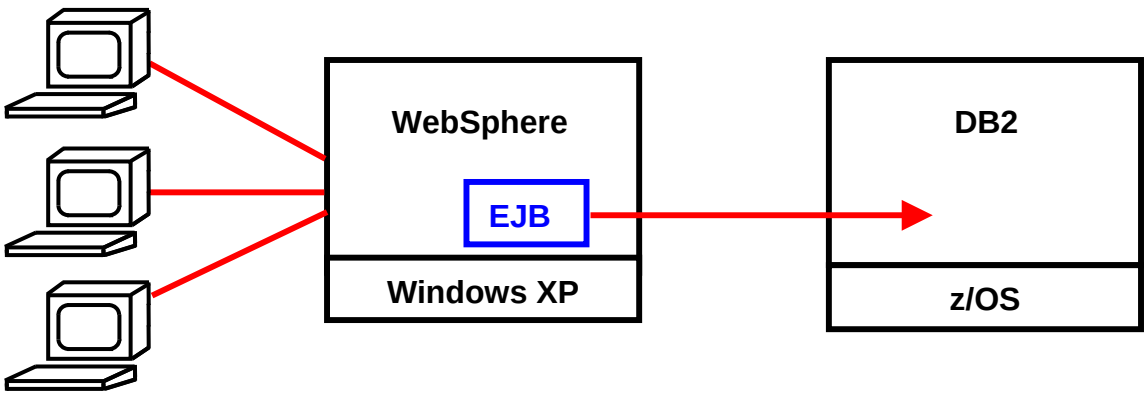
1. **Erstellen einer Test- und Entwicklungsumgebung mit WebSphere**
2. **EJB Zugriff auf z/OS DB2**
3. **Java Message driven Bean**

Dies ist Teil 1 mit folgendem Inhalt:

1. **Übersicht**
2. **Windows XP virtuelle Maschine**
3. **Konfiguration der Data Source**

1. Übersicht

Ziel von Teil 2. dieses Tutorials ist es, mittels einer Enterprise Java Bean (EJB) auf eine DB2-Datenbank unseres Rechners leia.informatik.uni-leipzig.de zuzugreifen.



Wir benutzen eine typische 3-Tier Konfiguration, mit Server-zentrischer Präsentationslogik.

Der WebSphere Application Server ist in unterschiedlichen Versionen verfügbar. Wir benutzen die einfachste Version, die auf der IBM Home Page kostenlos zur Verfügung gestellt wird, und auf einer Geronimo Engine der Apache Foundation aufbaut.

Enterprise JavaBeans (EJB) sind standardisierte Komponenten innerhalb eines Java Enterprise Edition (JEE)-Servers. Sie vereinfachen die Entwicklung komplexer mehrschichtiger verteilter Softwaresysteme mittels Java. Mit Enterprise Java Beans können wichtige Konzepte für Unternehmensanwendungen, z.B. Transaktions-, Namens- oder Sicherheitsdienste, umgesetzt werden, die für die Geschäftslogik einer Anwendung nötig sind.

Die EJB Spezifikation wurde ursprünglich in 1997 von IBM entwickelt und 1999 von Sun Microsystems als EJB 1.0 und 1.1 verbessert. Der Java Community Process brachte 2001 EJB 2.0 heraus.

Die Komplexität und die fehlende Objektorientiertheit der EJB-Technologie waren in der Vergangenheit Kritikpunkte. Entsprechend dem Feedback von einem großen Teil der J2EE Community betrachten viele Experten die Entwicklung mit früheren Versionen von JEE als unnötig komplex, obwohl alles gut funktioniert. Bei der Entwicklung mussten die Entwickler mehr Zeit verbringen, um APIs für den EJB-Container zu schreiben, als für die Implementierung der Geschäftslogik. Zum Beispiel:

- **Die EJB 2.x-Spezifikation erfordert, dass eine Session oder Entity-Bean die Remote und/oder die Home-Schnittstelle implementieren muss, welche die Schnittstelle aus dem EJB Framework-Paket erweitert (extends). Dies bewirkt eine enge Kopplung zwischen dem Entwickler-geschriebenen Code und den Interface-Klassen aus dem EJB Framework-Paket. Die Folge ist eine langweilige und wiederholte Arbeit, um mehrere unnötigen Callback-Methoden (ejbCreate (), ejbPassivate (), ejbActivate ()) zu implementiern, die keinen direkten Bezug zu der Business-Logik haben. Darüber hinaus müssen Exceptions, die durch die unnötigen Methoden verursacht werden, gehandhabt werden.**
- **Die XML-Deployment-Deskriptoren sind übermäßig verbose und komplex. Ein Großteil der Informationen in dem Deployment-Deskriptor könnten mit Standardwerten (Defaults) definiert werden.**
- **Auf Ressourcen muss durch JNDI zugegriffen werden.**
- **Das Container-Managed Persistence Modell ist kompliziert zu entwickeln und zu verwalten.**
- **Das Persistenz Mapping-Modell war nie gut in EJB2.x definiert. Dis bewirkt, dass Entity Beans nicht für alle J2EE-Container ohne Veränderungen passen.**

Diese negativen Aspekte haben die folgenden Gründe:

- **Die J2EE-Entwicklung ist zu komplex.**
- **Zu komplexe XML-basierte Konfiguration.**
- **Das Persistenz-Modell gilt heute als überholt.**

Aus diesem Grunde wurde 2006 die neue EJB 3.0-Spezifikation entwickelt, die eine deutliche Vereinfachung bringen soll. EJB 3.0 stellt einen erheblichen Bruch mit der Vergangenheit dar. Hervorzuheben ist insbesondere die Einführung eines neuen Plain Old Java Object (POJO)-basierten Programmiermodells, das die Entwicklung von J2EE-Anwendungen merklich vereinfacht. Das heißt, der Code muss nicht durch EJB-Implementierungsdetails „verschmutzt“ werden. Ein wesentliches Merkmal in EJB3 ist die breite Verwendung von Annotationen. Diese ermöglichen innovative Techniken wie Metadata Annotations, Lifecycle Interceptors, Dependency Injection, EJB Injection und Java Persistence API (JPA).

In der EJB3 Entwicklung Modell, müssen wir nur 2 einfache Schnittstellen definieren, deren Beziehung aus der Klasse hervorgeht, ohne (oder mit einer stark reduzierten) Konfiguration in einem Deployment Descriptor.

Die wichtigsten Neuerungen bei EJB3.0:

- **Annotations, die Angaben in den Deployment Deskriptoren ergänzen oder überschreiben**
- **Convention über Declarationen**
- **Verzicht auf das Homeinterface bei Session Beans**

Annotations werden als Statements in den normalen Java code eingefügt. Ein Annotation Statement beginnt mit dem Symbol @. Hier ist ein Beispiel, wie eine EJB mit Annotations aussieht:

```
public interface HelloService {
  String sayHello();
}

@Stateless
@Local(HelloService.class)
@Remote(HelloServiceRemote.class)
public class HelloServiceBean implements HelloService {
  public String sayHello() { return “Hello world”; }
}

public interface HelloServiceRemote {
  String sayHello();
}
```

Die EJB 3.1 (2009) Spezifikation und die EJB 3.2 (2012) Spezifikation brachten weitere Verbesserungen.

Eine Enterprise Application muss auf einem JEE-Anwendungsserver eingesetzt werden. Dieses Tutorial verwendet WebSphere Application Server 6.1. WebSphere Application Server 6.1 ist JEE-kompatibel und unterstützt Java Standard Edition 1.5. Mit der Installation des EJB3 Feature Packs bietet WebSphere Application Server 6.1 eine EJB 3.0 Unterstützung.

Das vorliegende Tutorium entstand aus einer Masterarbeit von Herrn Li Zhang. Sie können die Arbeit unter http://www.cedix.de/DiplArb/LiZheng.pdf herunterladen.

Java Enterprise Edition ist trotz mehr als 12-jähriger Entwicklung immer noch nicht sehr stabil. Einzelheiten unter: Doug Lyon: The Java Tree Withers. IEEE Computer, Jan. 2012, p.83-85, http://www.cedix.de/VorlesMirror/Band3/Lyon.pdf.

Als Folge implementieren die Hersteller neue JEE Spezifikationen nur mit einer nicht unerheblichen Zeitverzögerung in ihre Software Produkte. Zum Zeitpunkt der Erstellung der Masterarbeit war EJB 3.0 unter WebSphere soeben erst verfügbar geworden; EJB 3.1 kam erst deutlich später. Die zusätzlichen EJB 3.1 Erweiterungen wären allerdings für das vorliegende Tutorium auch uninteressant.

Hinweise

WinXP Login: UNILP
Passwort: unilp

WebSphere Application Server 6.1 (WAS 6.1),
Login: admin
Passwort: admin

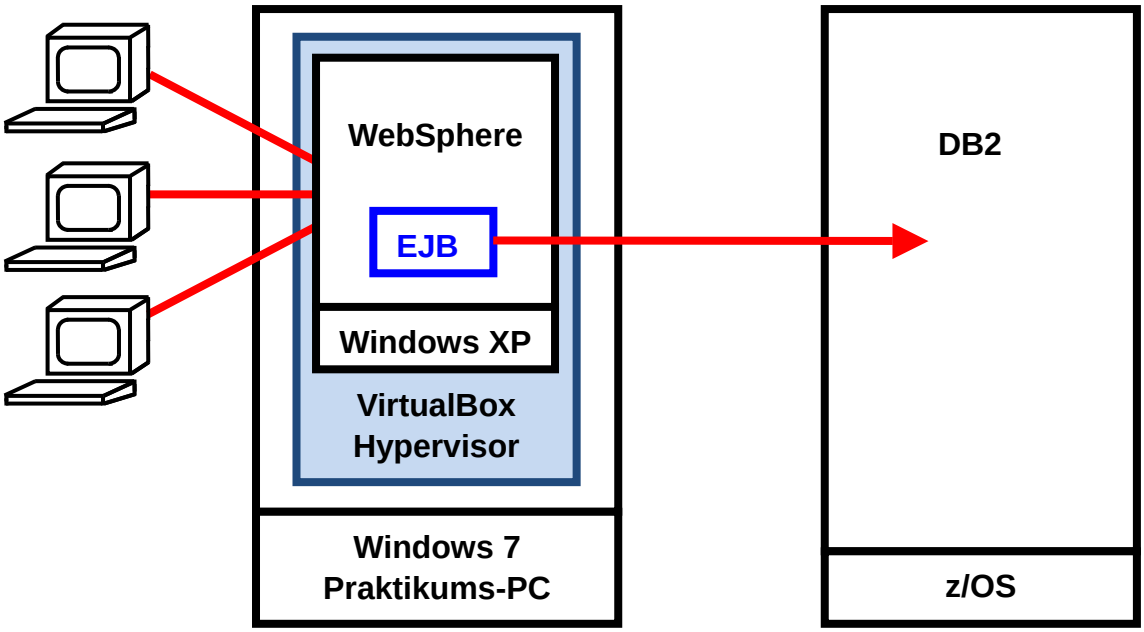
Starten/Stoppen des WAS 6.1 Server:
Programme -> IBM WebSphere -> Application Server V6.1 -> Profiles -> AppSrv01 -> Start/Stop the server

In dem hier vorliegenden Teil 1 der drei EJB Tutorials installieren wir eine Test- und Entwicklungsumgebung.

2. Windows XP virtuelle Maschine

Installation, Konfiguration, Anpassung und Einrichtung der Test- und Entwicklungsumgebung ist ein komplexer Vorgang. Wir möchten Sie nicht davon abhalten, diese Aktivität selbst vorzunehmen. Zu diesem Zweck müssen Sie die kostenlose WebSphere Application Server Community Edition (CE) aus dem Netz herunterladen (z.B. http://www-03.ibm.com/software/products/us/en/appserv-wasce) und falls erforderlich, das ebenfalls kostenlose EJB 3.0 Feature Pack. Wir empfehlen, das Ganze in einer virtuellen Maschine auf Ihrem PC zu installieren.

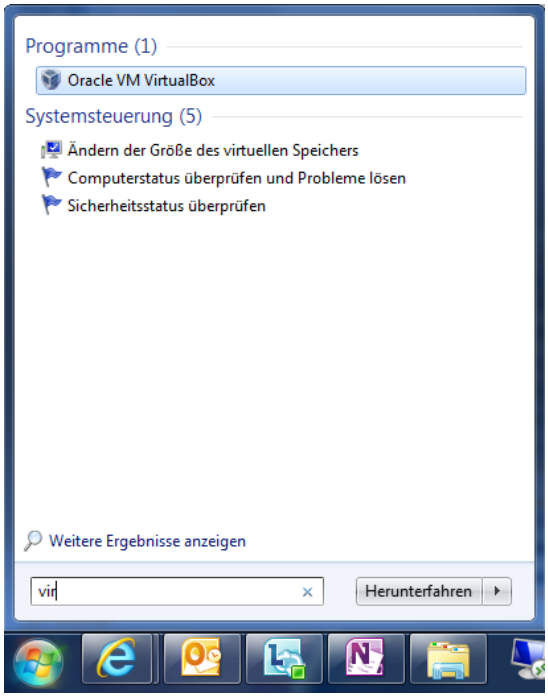
Zur Vereinfachung haben wir dies jedoch bereits für Sie vorgefertigt. Hierzu existiert eine virtuelle Maschine mit VirtualBox, in der ein Windows XP System mit der Entwicklungsumgebung einschließlich WebSphere bereits vorinstalliert ist. Diese virtuelle Maschine ist auf dem Praktikums-PC der Abteilung Technische Informatik eingerichtet und kann dort gestartet werden.



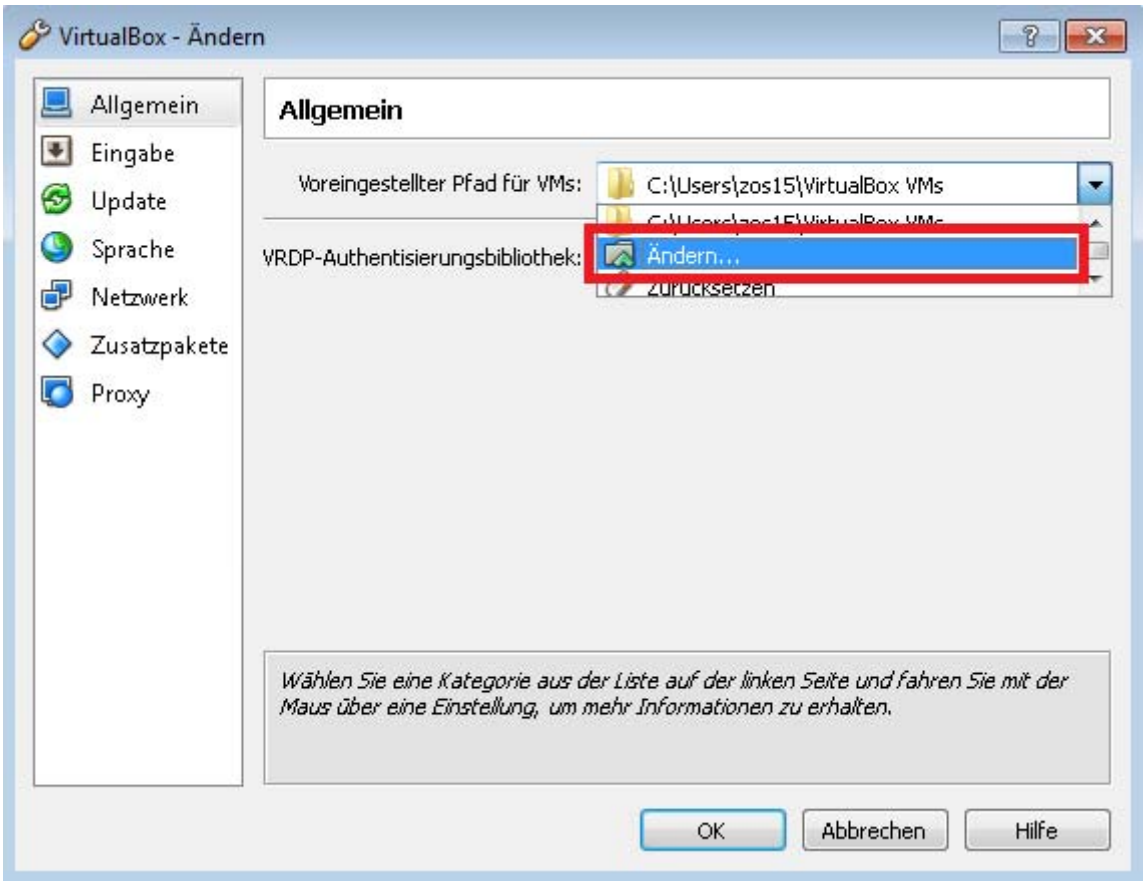
Das Praktikum kann auch auf dem eigenen Rechner durchgeführt werden, dazu ist das Virtual Disk Image (VDI) entsprechend zu kopieren.

Schritt 1 Oracle VM VirtualBox starten und konfigurieren

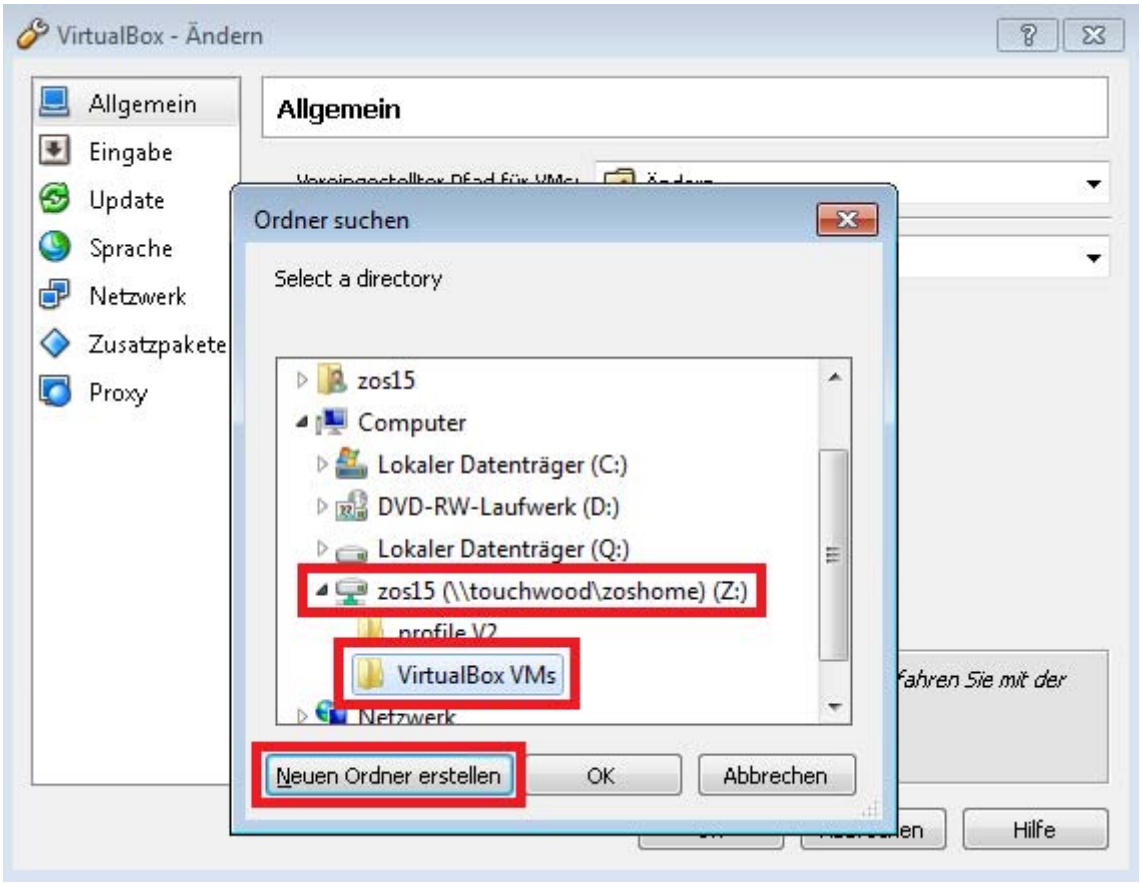
- **Starten Sie das Virtualisierungsprogramm Oracle VM VirtualBox.**



- **Datei**
- **Globale Einstellungen…**
- **Ändern…**


VirtualBox - Ändern
Allgemein
Eingabe
Update
Sprache
Netzwerk
Zusatzpakete
Proxy
Allgemein
Voreingestellter Pfad für VMs:
C:\Users\zos15\VirtualBox VMs
VRDP-Authentisierungsbibliothek:
Ändern...
Zurücksetzen
Wählen Sie eine Kategorie aus der Liste auf der linken Seite und fahren Sie mit der Maus über eine Einstellung, um mehr Informationen zu erhalten.
OK
Abbrechen
Hilfe

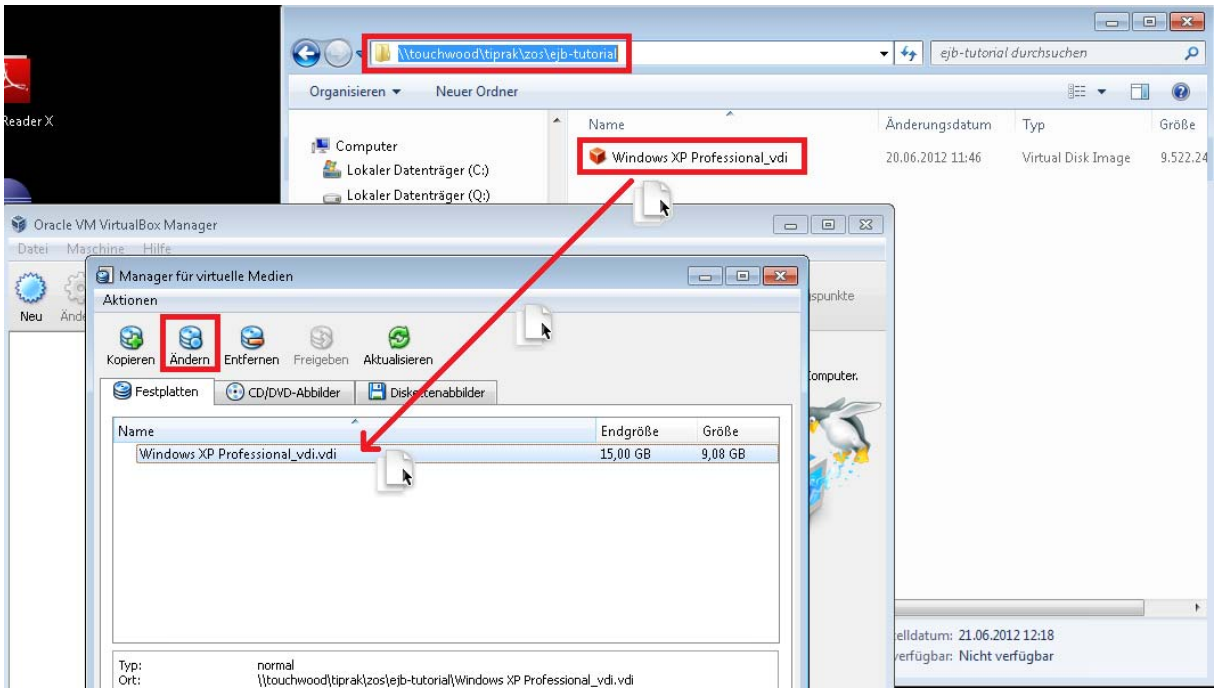
- **Laufwerk zosxx (\\touchwood\zoshome) (Z:) auswählen**
- **Neuen Ordner erstellen**
- **Namen des Ordners in VirtualBox VMs ändern**



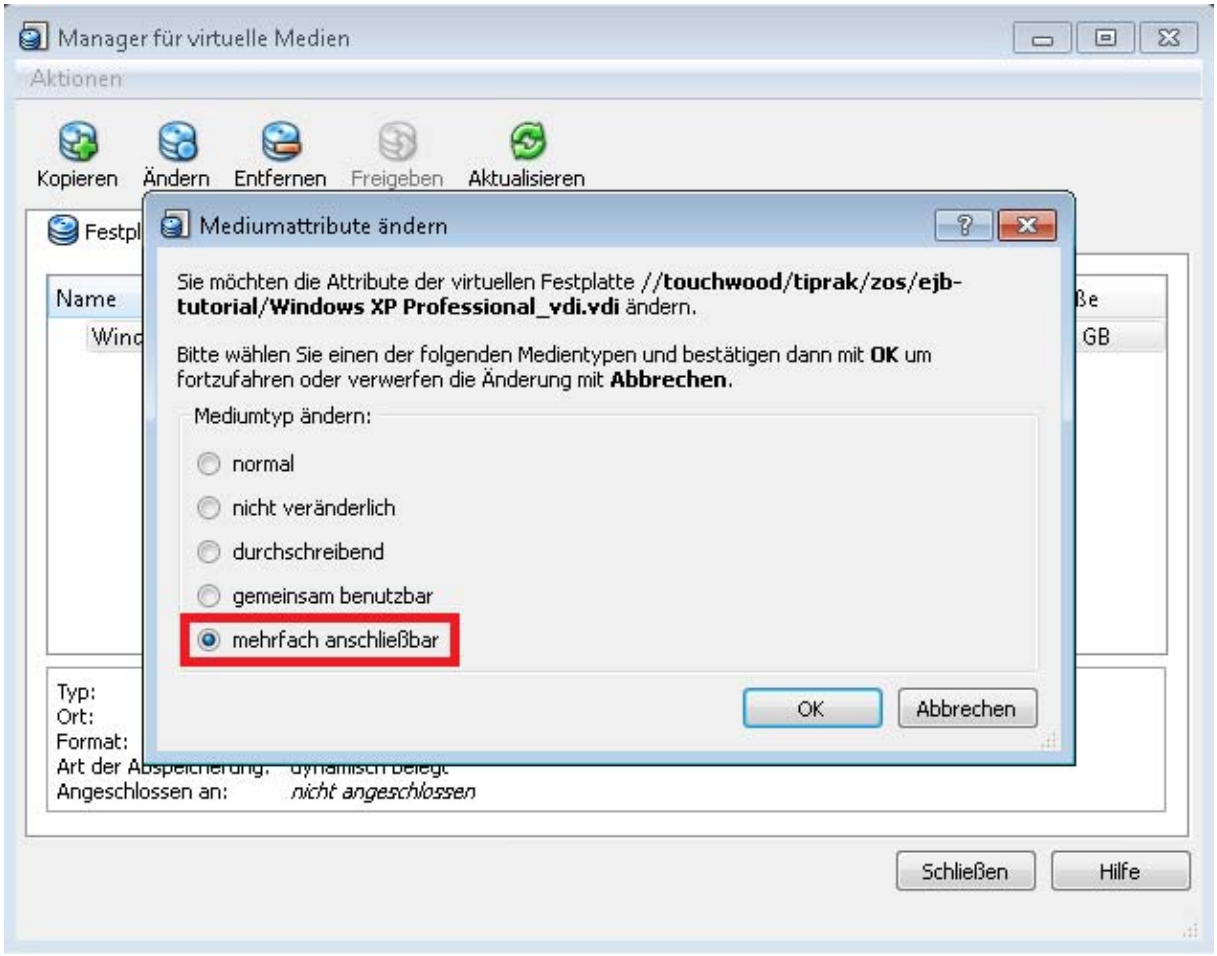
- **VirtualBox VMs (Ordner auswählen)**
- **OK**
  Alle Einstellungen und neuen Virtual Disk Image (VDI)-Dateien sowie Snapshots der VMs werden nun auf Laufwerk Z: und daher nicht im Profilordner (C:\Users\zosxx) abgelegt. Dadurch wird beim An- und Abmelden an einen Praktikums-PC der gesamte Inhalt des Ordners VirtualBox VMs nicht jedes Mal mit dem Server komplett herunter- und hochgeladen (sondern während der Sitzung automatisch synchronisiert). Die Größe des Ordners VirtualBox VMs kann während der Bearbeitung der Tutorien auf mehrere GB ansteigen!
- **OK**

Schritt 2 VM mit vorhandener VDI erzeugen

- **Datei**
- **Manager für virtuelle Medien…**
- **Windows Explorer starten**
- **In Adresszeile \\touchwood\tiprak\zos\ejb-tutorial eingeben**
- **VDI-Datei Windows XP Professional_vdi in das Fenster von Manager für virtuelle Medien hineinziehen (drag&drop)**



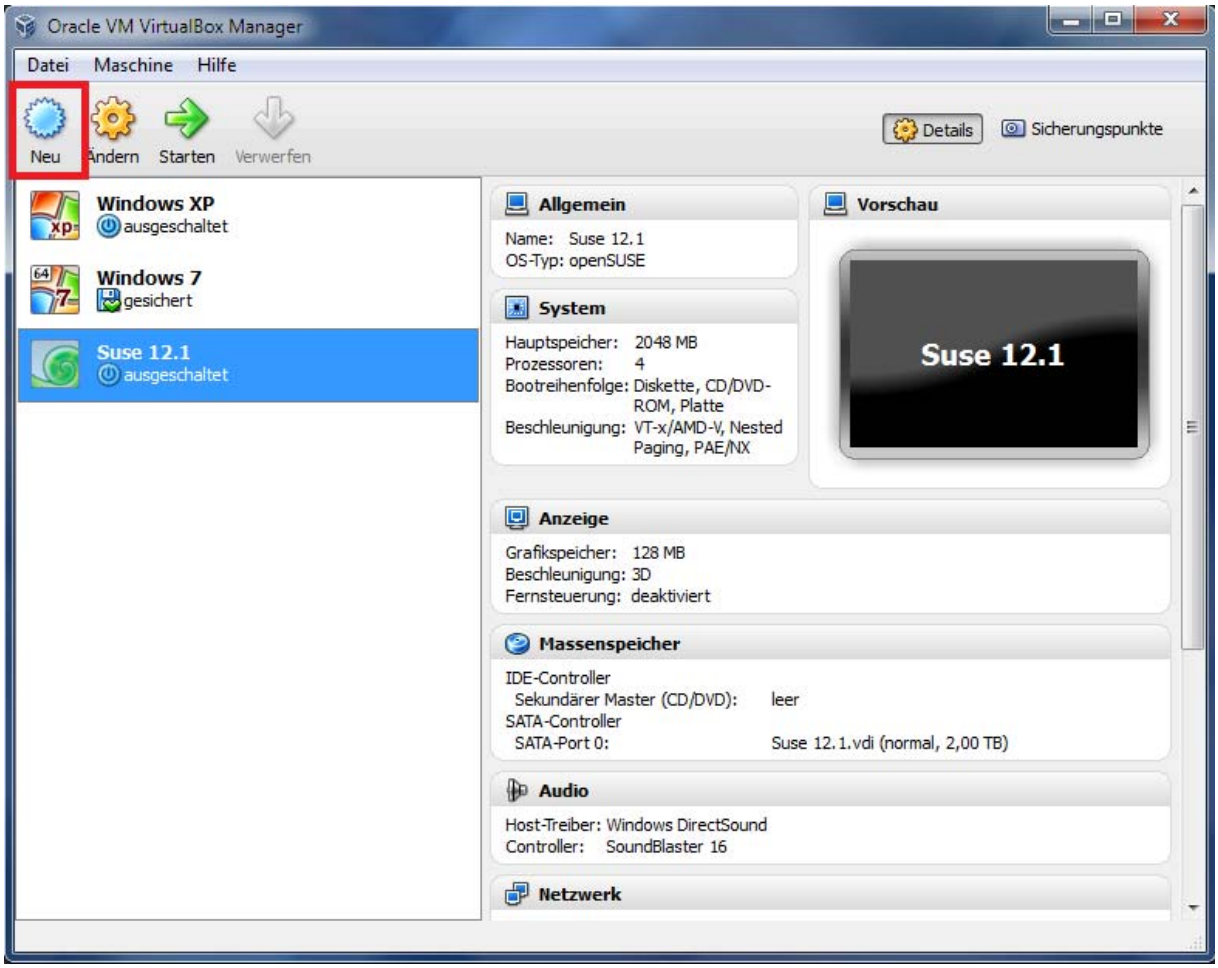
- **Ändern**
- **Medientyp mehrfach anschließbar auswählen**



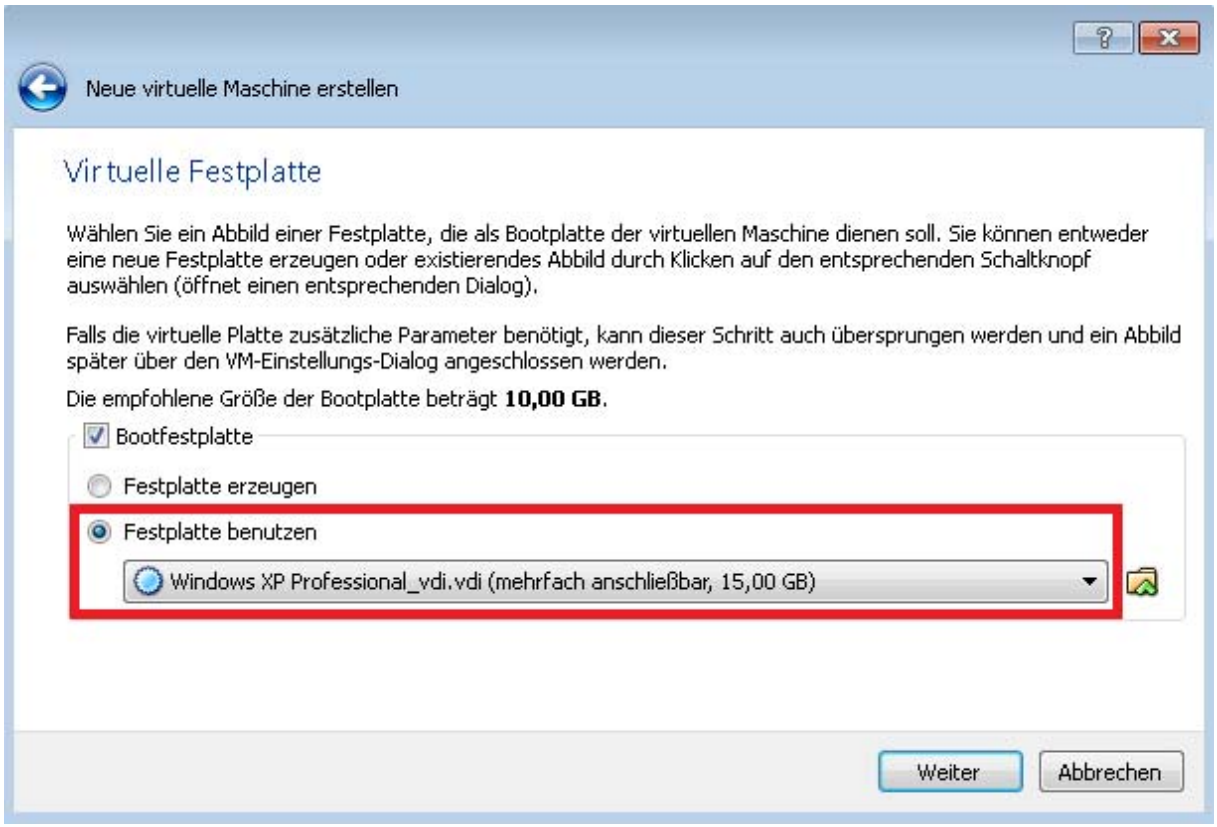
- **OK**

Das VDI ist jetzt konfiguriert. Mehrfach anschließbar bedeutet, dass alle Änderungen, die durch eine VM auf dem virtuellen Medium durchgeführt werden, nicht in dieser VDI-Datei landen, sondern in einem separaten Differenzimage abgelegt werden. In VirtualBox werden diese Differenzimages auch als Snapshots bezeichnet. Später durch angelegte Sicherungspunkte erzeugte Snapshots sind also auch jeweils ein Differenzimage. Alle Differenzimages einer VM werden im Ordner VirtualBox VMs\VM-Name abgelegt.

- **Neu**



- **Weiter**
- **Vergeben Sie der virtuellen Maschine (VM) einen Namen**
  Der Name erscheint anschließend in der obigen Liste. Da der Benutzername des Windows-Accounts in der VM unilp heißt, wird als VM-Namen unilp empfohlen.
- **Weiter**
- **Stellen Sie die Größe des Hauptspeichers auf 1024MB ein**
  Es können auch größere Werte eingestellt werden, für das Praktikum reichen 1GB RAM für die VM für ein flüssiges Arbeiten allerdings aus (die Praktikums-PCs besitzen 12GB RAM).
- **Weiter**
- **Wählen Sie „Festplatte benutzen“ aus**
- **Wählen Sie als virtuelles Plattenabbild die Datei Windows XP Professional_vdi.vdi aus**

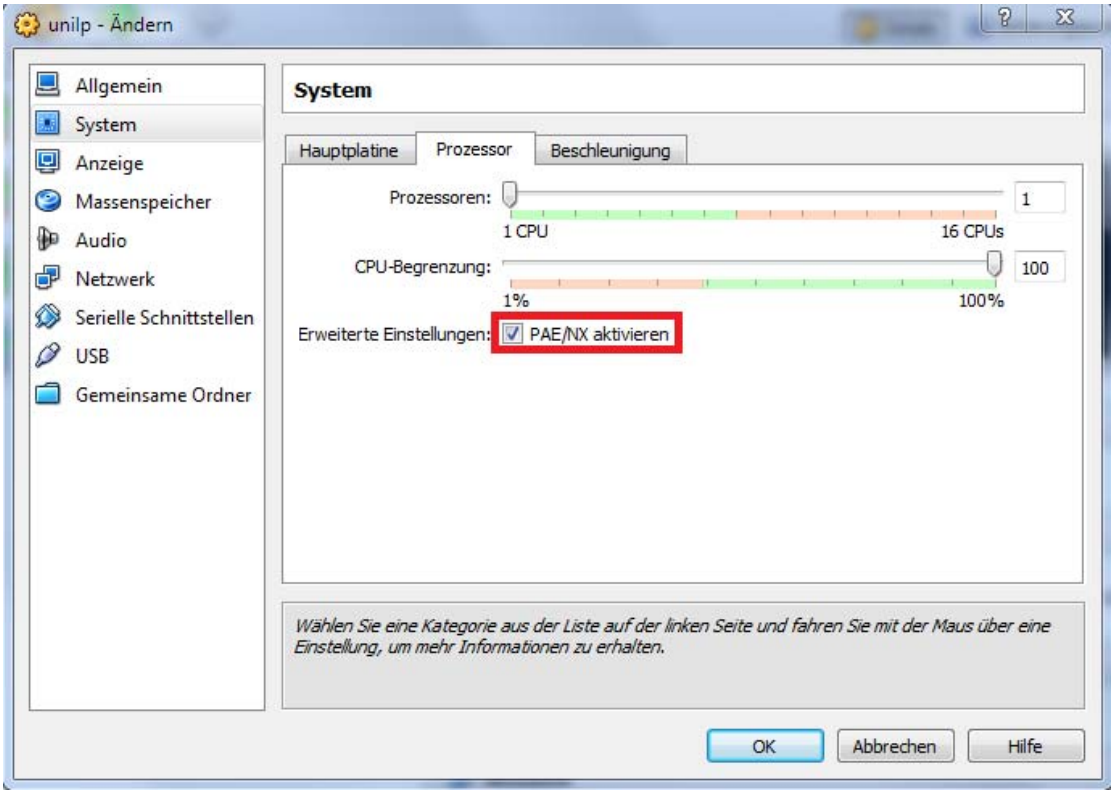


- **Weiter**
- **Erzeugen**

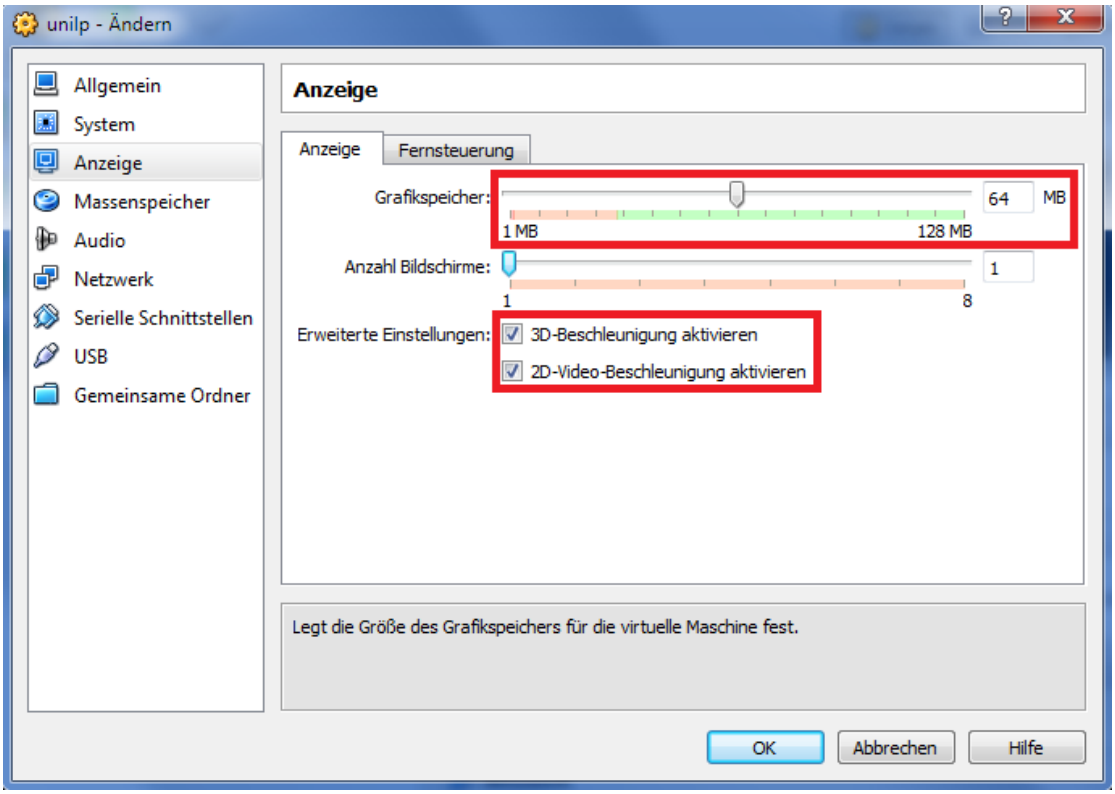
Schritt 3 Virtuelle Maschine (VM) konfigurieren

Dieser Schritt dient dazu, die VM für den Betrieb optimal zu konfigurieren. Mit diesen Einstellungen wurde die VM getestet, so dass sie auf den Praktikums-PCs schnell und einwandfrei läuft. Benutzen Sie zuhause z.B. eine Mehrbildschirmumgebung, so kann es sinnvoll sein, entsprechende Anpassungen an der Konfiguration vorzunehmen.

- **Ändern**
- **System**
- **Prozessor**
- **Häkchen bei „PAE/NX aktivieren“ setzen**
  Dadurch wird die „Physical Adress Extension“ (Physikalische Adresserweiterung) sowie das „No eXecute“-Bit (hardwareunterstützte Dateiausführungsverhinderung) aktiviert.



- **Kontrollieren, ob im Reiter Beschleunigung bereits die Häkchen bei „VT-x/AMD-V aktivieren“ sowie „Nested Paging aktivieren“ gesetzt sind**
  ➡VT-X/AMD-V ermöglicht VirtualBox, den Gast (hier Windows XP) para- anstatt vollvirtualisiert auszuführen, was zu einer deutlich höheren Performance des Gastes führt. Paravirtualisierung wird auch Hardwarevirtualisierung genannt und Vollvirtualisierung dementsprechend Softwarevirtualisierung. Mit VT-x/AMD-V können Prozesse des Gastes direkt auf der Host-Hardware arbeiten, weil der Prozessor zusätzliche Sicherheitsfunktionen (Abschottung) bereitstellt, welche verhindern, dass die Gastprozesse in Hostprozesse eingreifen können.
  ➡Nested Paging, auch bekannt als Rapid Virtualization Indexing, ist eine hardwarebasierte Virtualisierung für die Memory Management Unit (MMU) des Prozessor, um die Berechnungen in den Seitentabellen für die Adressierung des virtuellen Speichers des Gastes in Hardware durchzuführen (auch Shadow Page Tables genannt).
- **Anzeige**
- **Als Grafikspeicher 64 MB einstellen**
- **Häkchen bei „3D-Beschleunigung aktivieren“ sowie „2D-Video-Beschleunigung aktivieren“ setzen**



- **Für den Datenaustausch können Sie unter „gemeinsame Ordner“ einen Ordner des Hosts der virtuellen Maschine zur Verfügung stellen. Der verwendete Gast bietet allerdings bereits eine Dateifreigabe über das Netzwerk an.**
- **OK**

Die VM ist jetzt fertig eingerichtet. Sie können die VM jetzt starten. Das Passwort für den eingerichteten Benutzer unilp lautet unilp. Der Benutzername und Passwort für die Websphere-Administration-Backends sind identisch (unilp/unilp).

3. Konfiguration der Data Source

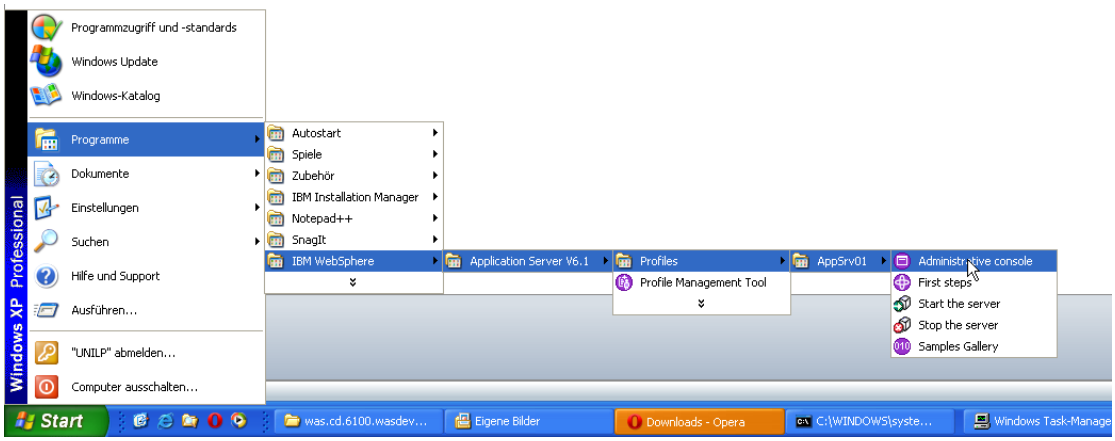
In diesem Abschnitt erläutern wir, wie Sie eine DB2 Data Source auf dem z/OS System leia.informatik.uni-leipzig.de einrichten können. Bitte besorgen Sie sich vorher einen DB2 Benutzernamen mit Passwort von Ihrem Betreuer. In diesem Text verwenden wir den Benutzernamen prak224; bitte ersetzen Sie ihn durch Ihren eigenen.

Anmerkung: Wir haben das Tutorium mit dem Internet Explorer ausgetestet. Wir können nicht garantieren, dass es mit einem anderen Browser funktioniert.

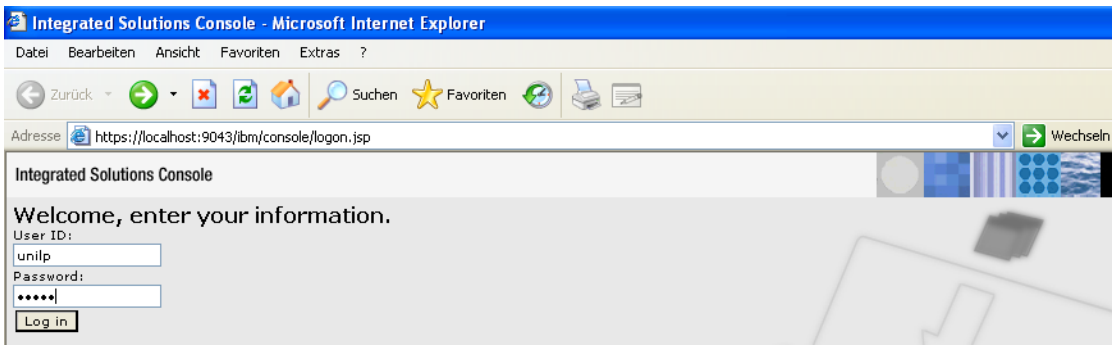
Wir verwenden die folgenden Abkürzungen

1k einmal mit der linken Maustaste klicken
2k zweimal mit der linken Maustaste klicken
1kr einmal mit der rechten Maustaste klicken

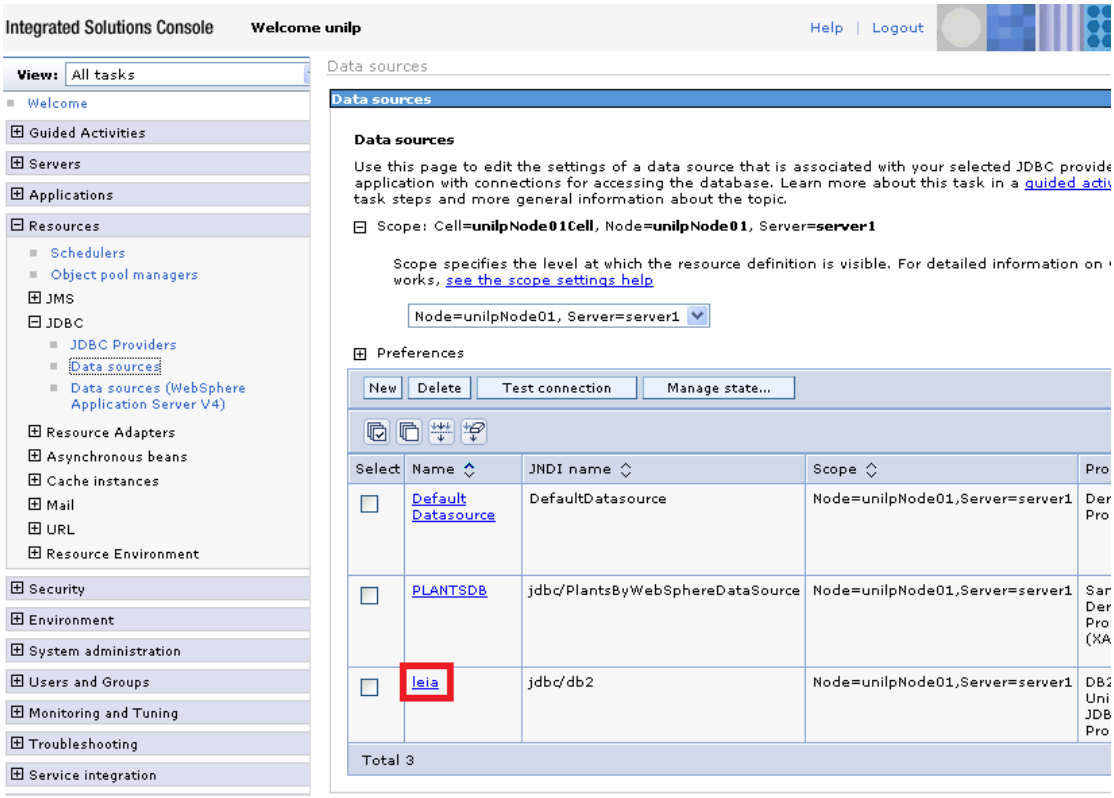
Schritt 1 1k Administive console um die Login Seite auf Ihrem Browser zu öffnen.



Schritt 2 Geben Sie Benutzername und Passwort ein, dann 1k auf denn Login Button.

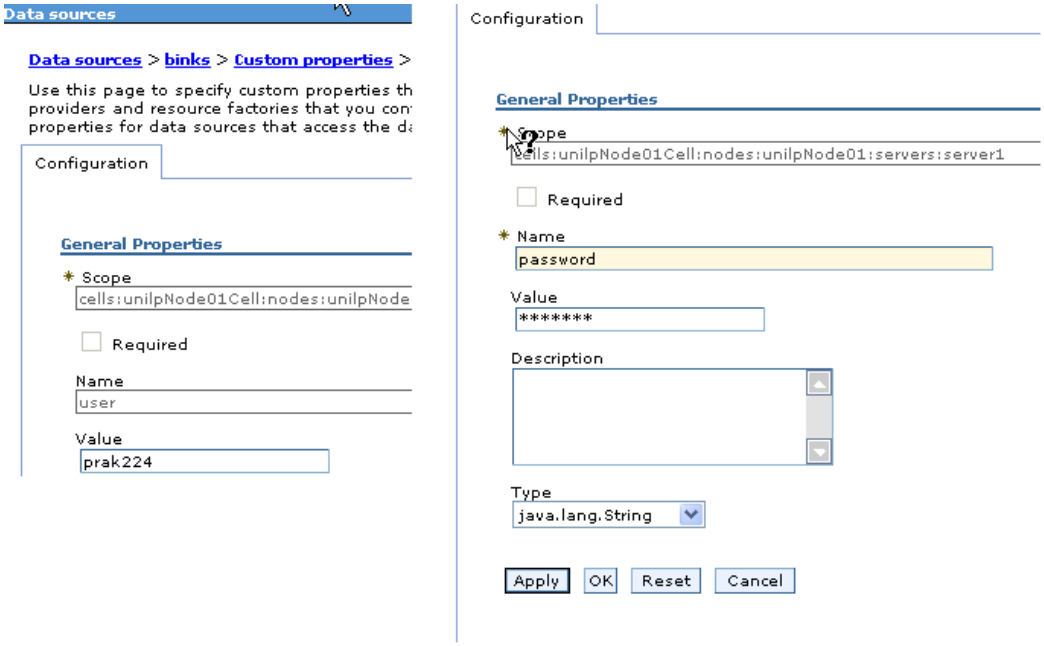


Schritt 3 In der Liste Resources → JDBC → Data Sources ist jetzt ein neuer Eintrag „leia“ zu sehen. Darauf 1k um die Configuration Page zu öffnen.



Schritt 4 Um die Verbindung zu der DB2 Data Source herzustellen, muss der Benutzername und das Passwort eingegeben werden. 1k auf Custom properties um die beiden Werte user und password eingeben.

Schritt 5 1k auf New um die Property user hinzuzufügen. In diesem Text verwenden wir den Wert „prak224“. Sie müssen Ihren eigenen Benutzernamen verwenden, den sie von dem DB2 Administrator enthalten haben. 1k auf Apply und Save. Nochmals 1k auf New um das Passwort zu setzen.



Schritt 6 Testen der Verbindung. 1k auf Data sources auf der linken Seite und dann 1k auf leia auf der rechten Seite. 1k auf Test connection. Wenn Sie alles richtig gemacht haben, bekommen Sie eine Nachricht wie in der folgenden Abbildung: