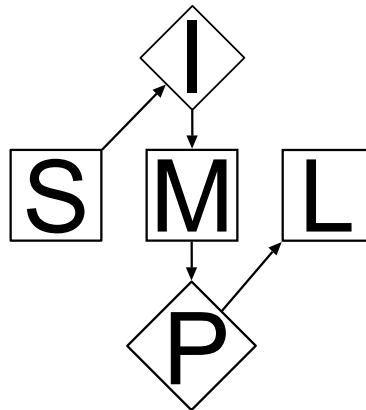


Installationsanleitung

Version 1.0

31. August 2010



Inhaltsverzeichnis

1	Einleitung	4
2	Systemanforderungen	5
3	Vorbereitung	6
4	All-In-One-Installation	7
5	Paketbasierte Installation	8
5.1	SIMPL Core, BPEL-DM Extension Activities und SIMPL Auditing	8
5.2	SIMPL Reference Resolution System (RRS) und RRS Transformation-Service	11
5.3	jUDDI Registry und SIMPL Web-Interface	12
6	Probleme und Lösungen	13
	Literaturverzeichnis	14
	Abbildungsverzeichnis	15
	Verzeichnis der Listings	16

Änderungsgeschichte

Version	Datum	Autor	Änderungen
0.1	13.07.2010	René Rehn	Erstellung der Dokumentstruktur und Skizzierung des Inhalts.
1.0	31.08.2010	Michael Hahn	Ausformulierung und Erweiterung der Kapitel. Fertigstellung des Dokuments.

1 Einleitung

Dieses Dokument soll eine Hilfestellung für die Installation und Einrichtung des SIMPL Rahmenwerks geben. Dafür werden die Systemanforderungen und durchzuführende Vorbereitungen für die Installation beschrieben, sowie die eigentliche Installation selbst. Es gibt dabei zwei verschiedene Installationsszenarien. Zum Einen kann eine All-In-One Installation durchgeführt werden, bei der eventuell bestehende Ressourcen überschrieben werden, die aber ein voll funktionsfähiges System ohne weitere Konfiguration liefert und zum Anderen kann eine paketbasierte Installation durchgeführt werden, bei der jede Komponente des SIMPL Rahmenwerks einzeln in eine bereits bestehende Infrastruktur installiert werden kann. Im Rahmen der paketbasierten Installation müssen teilweise bestehende Ressourcen nachträglich durch den Benutzer angepasst werden. Die nachfolgenden Kapitel beschreiben die aufgezeigten Aspekte genauer.

2 Systemanforderungen

Die Systemvoraussetzungen des Rahmenwerkes entsprechen den Voraussetzungen der verwendeten Java Runtime Environment und der Workflow-Engine. Empfohlen sind jedoch mindestens 1 GB freier Arbeitsspeicher sowie mindestens 500 MB freier Festplattenspeicher. Aufgrund der Skalierbarkeit richten sich die Systemvoraussetzungen nach der Anzahl der laufenden Prozesse in der Workflow-Engine. Je nach Anwendungskonstellation kann sich jedoch ein erheblich größerer Speicherbedarf ergeben. Die Anzahl der Prozesse wird demnach nur durch die Höhe des Speichers, welcher durch das Betriebssystem zur Verfügung gestellt wird, beschränkt. Durch die Integration des SIMPL Eclipse Plug-Ins in die Eclipse IDE kann das Rahmenwerk prinzipiell auf allen Systemen benutzt werden, die Eclipse unterstützen. Der SIMPL Core kann in allen Java fähigen Web Containern eingesetzt werden. Es muss beachtet werden, dass das Rahmenwerk momentan nur mit Apache ODE 2.0 verwendet werden kann, da nur diese Version die Anbindung von Extension Activities unterstützt. Die letzte stabile Version von Apache ODE 1.3.4 unterstützt noch keine Extension Activities und kann daher nicht mit SIMPL verwendet werden. Sollten die Extension Activities in einer Version 1.X unterstützt werden, so ist eine Verwendung prinzipiell möglich, jedoch wird keine Gewährleistung für den einwandfreien Betrieb übernommen, da SIMPL für Apache ODE 2.0 entwickelt wurde.

3 Vorbereitung

In diesem Kapitel werden alle nötigen Vorbereitungsmaßnahmen für die Installation des SIMPL Rahmenwerks aufgezeigt und beschrieben. Dabei spielt es keine Rolle, ob nach der Vorbereitung die All-In-One-Installation oder eine paketbasierte Installation einzelner Komponenten durchgeführt wird.

Für beide Installationsarten muss zunächst, falls nicht bereits vorhanden, Apache Tomcat wahlweise in Version 5.5 oder 6.0 heruntergeladen und installiert werden. Eine detaillierte Installationsanleitung für verschiedene Betriebssysteme findet sich auf der Homepage von Apache Tomcat (siehe [TOMCAT]). Weiterhin muss Eclipse vorhanden sein. Die beste und einfachste Möglichkeit ist die Verwendung einer Eclipse JEE Version, da diese bereits alle für den Eclipse BPEL Designer und SIMPL benötigten Pakete enthält und die Installation dadurch sehr einfach ausfällt. Zum Zeitpunkt des Projekts war Eclipse Galileo (Eclipse Version 3.5) die aktuellste Eclipse Version. Neuere Eclipse Versionen sollten theoretisch fehlerfrei mit den SIMPL Eclipse Plug-Ins und dem Eclipse BPEL Designer funktionieren, dies kann allerdings nicht garantiert werden. Wir empfehlen daher die Installation und Verwendung von Eclipse Galileo JEE. Das Installationspaket und entsprechende Anleitungen finden sich auf der Eclipse Homepage (siehe [ECLIPSE]).

Weiterhin können noch die SIMPL Eclipse Plug-Ins und der SIMPL BPEL Designer direkt installiert werden, da diese für beide Installationsarten gleich installiert werden. Dazu muss das ***simpl-eclipse.zip*** Installationspaket von der SIMPL Homepage ([SIMPL]) heruntergeladen werden. Dieses Paket enthält einen Ordner *plugins*, der wiederum alle durch SIMPL bereitgestellten Eclipse Plug-Ins enthält. Diese müssen in den *plugins*-Ordner des zuvor installierten Eclipse JEE entpackt werden. Damit ist die Installation der Eclipse Plug-Ins bereits abgeschlossen. Die korrekte Funktion kann durch starten von Eclipse und dem Anlegen eines BPEL-Projekts sowie eines BPEL-Prozesses überprüft werden. Eine detaillierte Anleitung dazu findet sich unter <http://www.se.uni-hannover.de/lehre/tutorials/BPEL-ODE-Eclipse-Getting-Started.php#h2-3> (zuletzt zugegriffen am 31.08.2010).

4 All-In-One-Installation

In diesem Kapitel wird die Installation des SIMPL Rahmenwerks mit Hilfe des All-In-One Installationspakets beschrieben.

Zuerst muss das entsprechende Installationspaket von der SIMPL Homepage ([SIMPL]) heruntergeladen werden. Für die All-In-One Installation ist dies das Paket ***simpl-tomcat.zip***. Anschließend muss das Paket nur noch in den Grundordner (normalerweise *C:/Programme/Tomcat X.X/*) der Apache Tomcat Installation entpackt werden. Danach muss der Tomcat-Server gestartet werden, entweder durch ausführen der *startup.bat* (Win) bzw. *startup.sh* (Unix) unter "*Tomcat-Installationspfad*"/*bin* oder aber in Windows durch den entsprechenden Service (z.B. über das Startmenü: Apache Tomcat X.X/Monitor Tomcat -> Start service), falls dieser installiert wurde (siehe Abbildung 1). Im Anschluss kann direkt geprüft werden, ob die SIMPL Web-Services erreichbar sind, dazu in einem Browser folgende URL <http://localhost:8080/ode/services/listServices> aufrufen und den AdministrationService oder DataSourceService anklicken. Dadurch sollte dann die autogenerierte WSDL des Web-Services erscheinen, falls dies nicht der Fall ist, liefert Punkt 1 in Kapitel 6 entsprechende Hilfe.

Falls die WSDL des Web-Services erscheint, kann Eclipse gestartet und mit SIMPL gearbeitet werden, näheres dazu liefert das SIMPL Benutzerhandbuch.

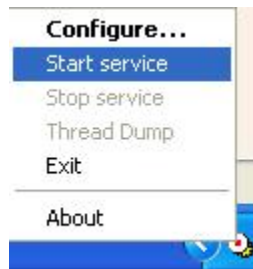


Abbildung 1: Start des Tomcat Services

5 Paketbasierte Installation

In diesem Kapitel wird die paketbasierte Installation des SIMPL Rahmenwerks beschrieben. Hierbei ist zu beachten, dass davon ausgegangen wird, dass benötigte Komponenten wie Apache ODE 2.0 ([ODE2]) oder auch Apache Axis2 ([AXIS2]) bereits vorhanden und korrekt installiert sind. Alle Abhängigkeiten werden aber nochmal explizit in den nachfolgenden Punkten aufgeführt. Alle nachfolgenden Verzeichnissangaben beziehen sich stets auf das Tomcat-Installationsverzeichnis, d.h. */webapps/ode* bezieht sich auf *“Tomcat-Installationspfad”/webapps/ode*.

5.1 SIMPL Core, BPEL-DM Extension Activities und SIMPL Auditing

Zunächst muss das entsprechende Installationspaket (*simpl-core.zip*) von der SIMPL Homepage ([SIMPL]) heruntergeladen werden. Dieses enthält alle Ressourcen des SIMPL Cores und der BPEL-DM Extension Activities. Für die Installation des SIMPL Cores muss Apache ODE 2.0 bereits vorhanden sein, ist dies nicht der Fall muss zuerst ODE installiert werden. Die benötigten Pakete und eine Installationsanleitung finden sich hier [ODE2].

Nachdem ODE installiert bzw. falls es bereits vorhanden ist, gibt es zwei mögliche Installationsszenarien:

1. Sollte man bisher keine Änderungen/Konfigurationen in den Dateien *ode-axis2.properties* (*webapps/ode/conf/ode-axis2.properties*) und *axis2.xml* (*webapps/ode/conf/axis2.xml*) vorgenommen haben, kann man das Installationspaket einfach in das Tomcat-Installationsverzeichnis entpacken und alle bereits vorhandenen Ressourcen überschreiben. Damit ist der SIMPL Core, die BPEL-DM Extension Activities und das SIMPL Auditing installiert.
2. Wurden an einer der in Punkt 1 genannten Dateien Änderungen vorgenommen, die erhalten werden sollen, so muss die Installation schrittweise durchgeführt werden. Zunächst können alle Dateien, bis auf die oben genannten zwei, in das Tomcat-Installationsverzeichnis entpackt werden. Im Anschluss müssen nun die Änderungen der zwei verbleibenden Dateien von Hand übernommen werden.
 - (a) *axis2.xml*: Die Zeile aus Listing 1 muss am Ende der Datei eingefügt werden (siehe Abbildung 2). Dadurch wird das dynamische Deployment der SIMPL Web-Services ermöglicht, die sich im Ordner *webapps/ode/WEB-INF/servicejars* in der Datei *simpl-core-webservices.jar* befinden.
 - (b) *ode-axis2.properties*: Es müssen die Zeilen aus Listing 2 hinzugefügt werden (siehe Abbildung 3). Dadurch werden die BPEL-DM Extension Activities und der SIMPL EventListener in ODE registriert. Die dahinter liegende Implementierung dafür liefert die Datei *simpl-ode-extension.jar*, die in den Ordner *webapps/ode/WEB-INF/lib* entpackt wird.

Danach kann der Tomcat-Server gestartet werden, entweder durch ausführen der *startup.bat* (Win) bzw. *startup.sh* (Unix) unter *“Tomcat-Installationspfad”/bin* oder aber in Windows durch den entsprechenden Service (z.B. über das Startmenü: Apache Tomcat X.X/Monitor Tomcat -> Start service), falls dieser installiert wurde (siehe Abbildung 1). Im Anschluss kann direkt geprüft werden, ob die SIMPL Web-Services erreichbar sind, dazu in einem Browser einfach *http://localhost:8080/ode/services/listServices* aufrufen und den *AdministrationService* oder *DataSourceService* aufrufen. Dadurch sollte dann die autogenerierte WSDL des Web-Services erscheinen, falls dies nicht der Fall ist, liefert Punkt 1 in Kapitel 6 entsprechende Hilfe.

Falls die WSDL des Web-Services erscheint, kann Eclipse gestartet und mit SIMPL gearbeitet werden, näheres dazu liefert das SIMPL Benutzerhandbuch.


```
<deployer extension=".jar"
    directory="servicejars"
    class="org.apache.axis2.jaxws.framework.JAXWSDeployer"/>
```

Listing 1: Aktivierung des dynamischen Deployments von Web-Services

```

    <handler name="AddressingBasedDispatcher"
        class="org.apache.axis2.dispatchers.AddressingBasedDispatcher"/>
</phase>
<phase name="Security"/>
<phase name="PreDispatch"/>
<phase name="Dispatch" class="org.apache.axis2.engine.DispatchPhase">
    <handler name="RequestURIBasedDispatcher"
        class="org.apache.axis2.dispatchers.RequestURIBasedDispatcher"/>
    <handler name="SOAPActionBasedDispatcher"
        class="org.apache.axis2.dispatchers.SOAPActionBasedDispatcher"/>
    <handler name="SOAPMessageBodyBasedDispatcher"
        class="org.apache.axis2.dispatchers.SOAPMessageBodyBasedDispatcher"/>
</phase>
<!-- user can add his own phases to this area -->
<phase name="OperationInFaultPhase"/>
<phase name="soapmonitorPhase"/>
</phaseOrder>
<phaseOrder type="OutFaultFlow">
    <!-- user can add his own phases to this area -->
    <phase name="soapmonitorPhase"/>
    <phase name="OperationOutFaultPhase"/>
    <phase name="PolicyDetermination"/>
    <phase name="MessageOut"/>
    <phase name="Security"/>
</phaseOrder>
<deployer extension=".jar" directory="servicejars" class="org.apache.axis2.jaxws.framework.JAXWSDeployer"/>
</axisconfig>
```

Abbildung 2: Änderung der Datei axis2.xml

```

## JCA connector port (default 2099), set to 0 to disable JCA connector
#ode-axis2.jca.port =

## JCA connector name (default 'ode')
#ode-axis2.jca.name =

## Working dir
#ode-axis2.working.dir =

## MEX Interceptors
#ode-axis2.mex.interceptors =

## Process dehydration
#ode-axis2.process.dehydration =

## Extension Bundles
## FQCNs, comma separated.
ode-axis2.extension.bundles.runtime = org.apache.ode.simpl.ea.SimplExtensionBundle
ode-axis2.extension.bundles.validation = org.apache.ode.simpl.ea.SimplExtensionBundle
## SIMPL EventListener registration
ode-axis2.event.listeners = org.apache.ode.simpl.events.listener.SimplSdoEventListener
```

Abbildung 3: Änderung der Datei ode-axis2.properties

```

## Extension Bundles
## FQCNs, comma separated.
ode-axis2.extension.bundles.runtime =
    org.apache.ode.simpl.ea.SimplExtensionBundle
ode-axis2.extension.bundles.validation =
    org.apache.ode.simpl.ea.SimplExtensionBundle

## SIMPL Eventlistener registration
ode-axis2.event.listeners =
    org.apache.ode.simpl.events.listener.SimplSdoEventListener

```

Listing 2: Registrierung der BPEL-DM Extension Activities und des SIMPL EventListeners

Nachfolgend werden noch alle Bestandteile des Installationspakets und deren Bedeutung beschrieben. Dies ist für die eigentliche Installation nicht von Bedeutung, aber für eventuelle Anpassungen oder Portierungen in andere Laufzeitumgebungen.

- *simplDB*: Embedded Derby Datenbank in der die Einstellungen des SIMPL Cores, die über die SIMPL Admin-Konsole in Eclipse vorgenommen werden, hinterlegt sind. Diese ist nicht dringend erforderlich, da sie dynamisch neu generiert wird, falls sie nicht vorhanden ist.
- *log4j.properties*: Enthält Logging-Einstellungen für den SIMPL Core.
- *webapps/ode/conf/*: Wie bereits weiter oben beschrieben, wird in der Datei *axis2.xml* das dynamische Deployment der SIMPL Web-Services aktiviert und in der Datei *ode-axis2.properties* die BPEL-DM Extension Activities sowie der SIMPL EventListener registriert. Weiterhin enthält dieser Ordner die Datei *simpl-core-config.xml* in der SIMPL Core Plug-Ins registriert werden können. Die Implementierung aller im Moment registrierten Plug-Ins befindet sich in der Library *webapps/ode/lib/simpl-core-plugins.jar*.
- *webapps/ode/lib*: Dieser Ordner enthält alle Libraries, die der SIMPL Core benötigt und den SIMPL Core selbst. Diese lassen sich noch in folgende Gruppen gliedern:
 - SIMPL Core
 - * *simpl-core.jar*
 - * *simpl-core-plugins.jar*
 - * *simpl-ode-extension.jar*
 - * *simpl-uddi-client.jar*
 - Datenbanktreiber
 - * *db2jcc.jar*
 - * *mysql-connector-java-5.0.8-bin.jar*
 - Service Data Objects (SDO)
 - * *tuscany-sdo-api-r2.1-1.0-incubating.jar*
 - * *tuscany-sdo-impl-1.0-incubating.jar*
 - * *tuscany-sdo-lib-1.0-incubating.jar*
 - Universal Description, Discovery and Integration (UDDI)
 - * *uddi-ws-3.0.1.jar*
 - * *juddi-core-3.0.1.jar*
 - * *juddi-client-3.0.1.jar*

– XML, XSD Verarbeitung

- * jdom.jar
- * stax-api-1.0.1.jar
- * xsd-2.2.3.jar
- * wstx-asl-3.2.0.jar

– Hilfsfunktionen

- * common-2.2.3.jar
- * ecore-2.2.3.jar
- * ecore-change-2.2.3.jar
- * ecore-xmi-2.2.3.jar
- * tools.jar

- *webapps/ode/servicejars/simpl-core-webservices.jar*: Dies sind die Web-Services des SIMPL Cores, die über Axis2 dynamisch generiert und deployt werden.

5.2 SIMPL Reference Resolution System (RRS) und RRS Transformation-Service

Für die Installation des SIMPL RRS muss Apache Axis2 Version 1.5 oder höher installiert sein. Ein Installationspaket und eine Installationsanleitung findet sich hier [AXIS2]. Danach muss wieder das entsprechende Installationspaket (*simpl-rrs.zip*) von der SIMPL Homepage ([SIMPL]) heruntergeladen werden. Im Anschluss muss einfach das Installationspaket in das Tomcat-Installationsverzeichnis entpackt werden. Die Pfade sind bereits so vordefiniert, dass die entsprechenden Dateien direkt in die entsprechenden Axis2-Unterordner entpackt werden. Sollten die Pfade eventuell nicht passen, können einfach direkt die Axis2-Unterordner (*services*, *lib*, *conf*) bzw. deren Inhalt aus dem Installationspaket in den lokalen Axis2-Ordner entpackt werden. Die Installation ist damit abgeschlossen und kann in Eclipse durch die RRS Management View getestet werden, näheres dazu findet sich im SIMPL Benutzerhandbuch.

Nachfolgend werden noch alle Bestandteile des Installationspakets und deren Bedeutung beschrieben. Dies ist für die eigentliche Installation nicht von Bedeutung, aber für eventuelle Anpassungen oder Portierungen in andere Laufzeitumgebungen.

- *webapps/axis2/WEB-INF/services*: Dieser Ordner enthält die SIMPL RRS Web-Services und die dahinter liegende Implementierung (*simpl-rrs.aar*) sowie den SIMPL RRS Transformation Web-Service und dessen Implementierung (*simpl-rrs-transformation.aar*).
- *webapps/axis2/WEB-INF/lib*: Hier sind befinden Datenbanktreiber, die für die Datenbankzugriffe beim Auflösen von Referenzen benötigt werden. Im Moment sind dies Treiber für die Datenbanken Apache Derby (*derby.jar*), IBM DB2 (*db2jcc.jar*) und MySQL (*mysql-connector.jar*).
- *webapps/axis2/WEB-INF/conf*: Dieser Ordner enthält die Datei *rrs-config.xml* in der RRS Datenquellen-Adapter registriert werden können. Im Moment sind dies, analog zu den vorher aufgezählten Datenbanktreiber, Adapter für Apache Derby, IBM DB2 und MySQL.

Natürlich muss das RRS nicht zwangsweise in einer eigenen Axis2 Installation deployt werden, ebenso könnte man das RRS in der in ODE integrierten Axis2 Installation deployen. Dabei ist allerdings zu beachten, dass sich dadurch die Web-Service Adressen ändern. Diese müssten dann in Eclipse angepasst werden, damit das RRS wieder erreichbar ist, da dort die Adressen der hier beschriebenen separaten Installation hinterlegt sind. Dazu müssen in Eclipse einfach im *SIMPL*-Menü die *Settings* geöffnet und unter *Reference Resolution System* die Adressen angepasst werden (siehe SIMPL Benutzerhandbuch).

```
<tomcat-users>
  <role rolename="secure"/>
  <user name="admin" password="" roles="admin,manager,secure"/>
</tomcat-users>
```

Listing 3: Mitgelieferte *tomcat-users.xml* Datei

5.3 jUDDI Registry und SIMPL Web-Interface

Zunächst muss das entsprechende Installationspaket (*simpl-jUDDI.zip*) von der SIMPL Homepage heruntergeladen werden. Dieses Paket enthält bereits eine vollständige Apache jUDDI (jUDDI) Installation, deshalb ist keine weitere Vorbereitung notwendig und das Paket kann direkt in das Tomcat-Installationsverzeichnis entpackt werden. Es ist lediglich zu beachten, dass für das SIMPL jUDDI Web-Interface eine neue Tomcat Benutzerrolle *secure* benötigt wird, deshalb wird im Installationspaket eine entsprechende *tomcat-users.xml* Datei mitgeliefert. Dadurch wird natürlich die bereits bestehende Datei überschrieben, falls dies nicht gewünscht ist, muss der Ordner *conf* einfach nicht mit entpackt werden und eine neue Rolle *secure* von Hand angelegt und einem bestehenden oder neu erstellten Benutzer zugewiesen werden (siehe Listing 3).

Damit ist die Installation schon abgeschlossen und kann direkt durch aufrufen folgender Adressen getestet werden: Test der jUDDI Installation <http://localhost:8080/juddiv3/> und Test der SIMPL jUDDI Web-Interface Installation <http://localhost:8080/juddiweb/> (standardmäßig: Benutzername *admin*, kein Passwort). Es empfiehlt sich direkt im Anschluss die in Kapitel 6 unter Punkt 2 beschriebene Speicheranpassung vorzunehmen.

6 Probleme und Lösungen

1. Die SIMPL Web-Services wurden deployt, die WSDLs können aber nicht generiert werden:
 - (a) Für die Generierung wird die *tools.jar* aus dem Java Development Toolkit (JDK, siehe [JAVA6]) benötigt. Diese wird aus dem JDK 1.6.0_14 in der ***simpl-tomcat.zip*** mitgeliefert. Je nachdem welche Version des Java Runtime Environment (JRE) auf dem Betriebssystem installiert ist, kann es unter Umständen nötig sein die *tools.jar* aus der entsprechend anderen Version des JDK zu verwenden. Diese muss unter *Tomcat x.x/webapps/ode/WEB-INF/lib/* die bereits vorhandene *tools.jar* ersetzen.
 - (b) Der Eclipse BPEL Designer kann nicht geöffnet werden, ein Fehler wird angezeigt
 - Apache Tomcat wurde nicht gestartet, der SIMPL Core ist nicht erreichbar
 - Die ***simpl-tomcat.zip*** wurde nicht deployt
 - Die ***simpl-eclipse.zip*** wurde nicht deployt
 - Es besteht Problem Nr. 1
2. “Heap Space”, “OutOfMemory” oder “PermGen Space” Fehlermeldungen von Tomcat
 - Falls Speicherprobleme auftreten, muss der Speicher in der von Tomcat verwendeten Java Virtual Machine erhöht werden. Dazu muss die Zeile “-XX:MaxPermSize=256m” im Windows Service in der Registerkarte “Java” (Apache Monitor -> Configure... -> Java -> Java Options) unter “Java Options” hinzugefügt werden. 256m steht dabei für 256Mb Arbeitsspeicher, dieser Wert ist nur eine Empfehlung und kann entsprechend dem verwendeten System nach oben oder unten korrigiert werden. Dieses Problem tritt nahezu immer bei der Verwendung von jUDDI auf, daher empfiehlt es sich diese Anpassung direkt nach der Installation vorzunehmen.
 - Sollte man Tomcat über die Kommandozeile mit startup.bat bzw. startup.sh starten, dann kann man den Speicher über eine entsprechende Umgebungsvariable setzen. Dazu muss einfach eine neue Umgebungsvariable “JAVA_OPTS” mit dem Wert “-XX:MaxPermSize=256m” erstellt werden.

Literatur

[AXIS2] Axis2, <http://ws.apache.org/axis2/>, zuletzt zugegriffen am 24.08.2010

[ODE2] Apache ODE, <http://ode.apache.org/>, zuletzt zugegriffen am 24.08.2010

[TOMCAT] Apache Tomcat, <http://tomcat.apache.org/>, zuletzt zugegriffen am 24.08.2010

[JAVA6] Java 6, <http://java.sun.com/>, zuletzt zugegriffen am 24.08.2010

[JAX-WS] JAX-WS, <https://jax-ws.dev.java.net/>, zuletzt zugegriffen am 24.08.2010

[ECLIPSE] Eclipse, <http://www.eclipse.org/>, zuletzt zugegriffen am 24.08.2010

[jUDDI] jUDDI, <http://ws.apache.org/juddi/>, zuletzt zugegriffen am 24.08.2010

[SIMPL] SIMPL, <http://code.google.com/p/simpl09/>, zuletzt zugegriffen am 24.08.2010

Abbildungsverzeichnis

1	Start des Tomcat Services	7
2	Änderung der Datei axis2.xml	9
3	Änderung der Datei ode-axis2.properties	9

Verzeichnis der Listings

1	Aktivierung des dynamischen Deployments von Web-Services	9
2	Registrierung der BPEL-DM Extension Activities und des SIMPL EventListeners	10
3	Mitgelieferte <i>tomcat-users.xml</i> Datei	12