

# Web-Services aus der Datenbank

Ein Vergleich

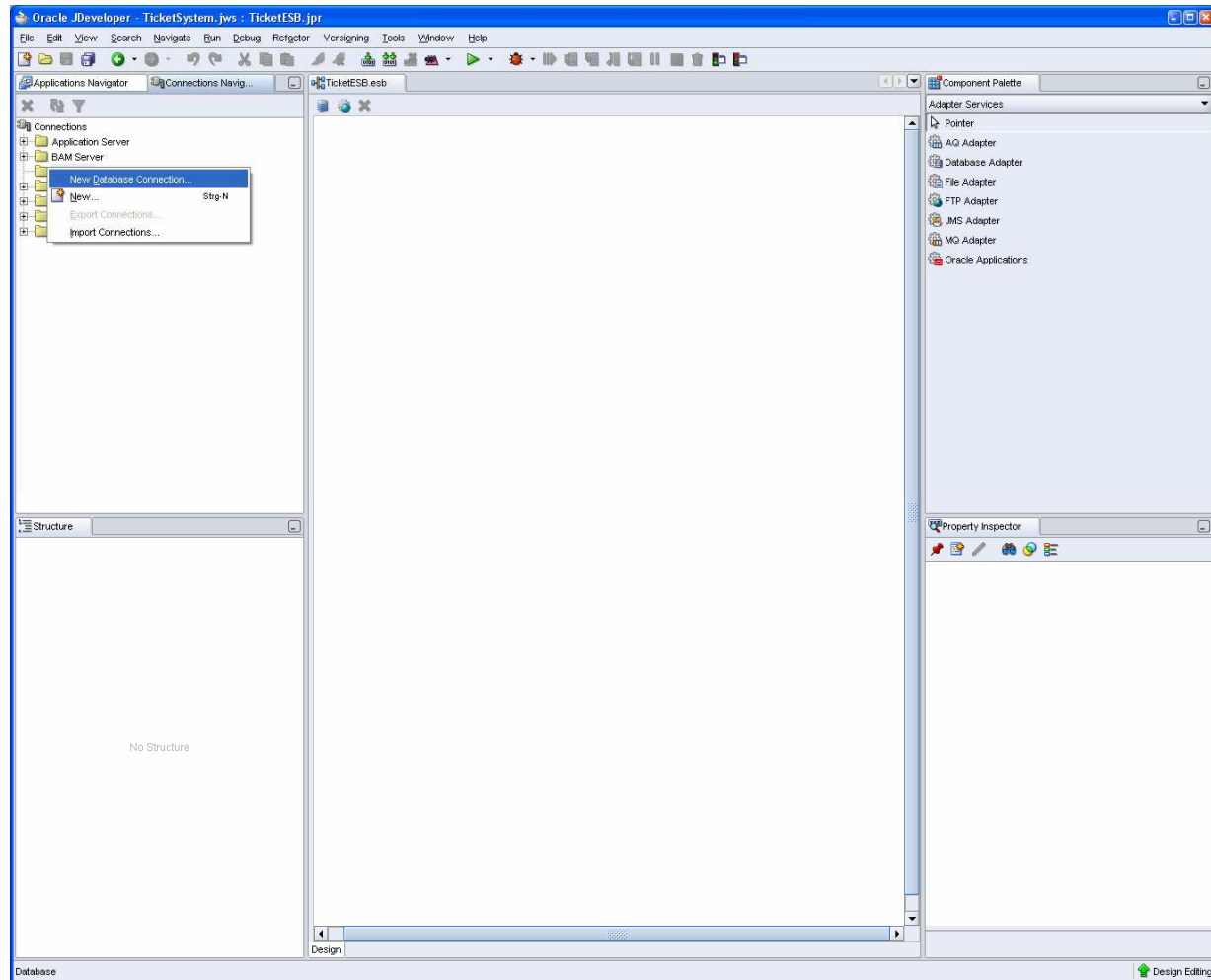
Alexei Ovetchkine, Dr. Götz Lichtwald  
merlin-zwo InfoDesign GmbH & Co. KG

Dieses HowTo beschreibt an Hand eines Beispiels, wie mit der SOA-Suite und dem JDeveloper eine Stored Procedure und ein SELECT aus der Datenbanktabelle als Webservices zu Verfügung gestellt werden können.

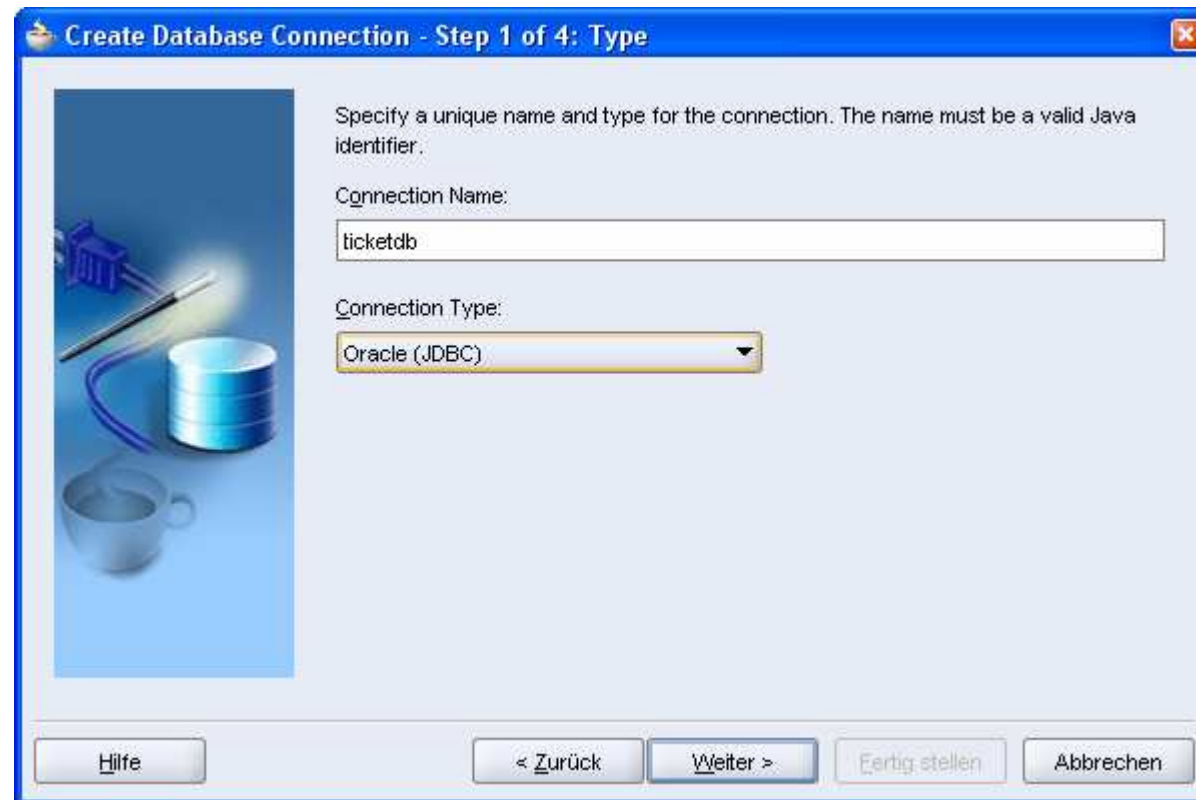
In dem Beispiel handelt es um ein TroubleTicket System. Die Stored Procedure erzeugt ein neues Ticket. Mittels der SELECT-Operation wird – anhand der angegebenen Ticketnummer – der Status des Tickets ermittelt.

Zunächst müssen im JDeveloper Connections zu der Datenbank, dem Application Server und Integration Server erstellt werden.

1. Datenbank Connection wird erstellt.



2. Der Connection Name ist „ticketdb“.



3. Username und Passwort werden eingegeben.



**Create Database Connection - Step 2 of 4: Authentication**

Specify a username and password to authenticate the connection. To bypass authentication at runtime, select Deploy Password.

Username:  
m2itracker

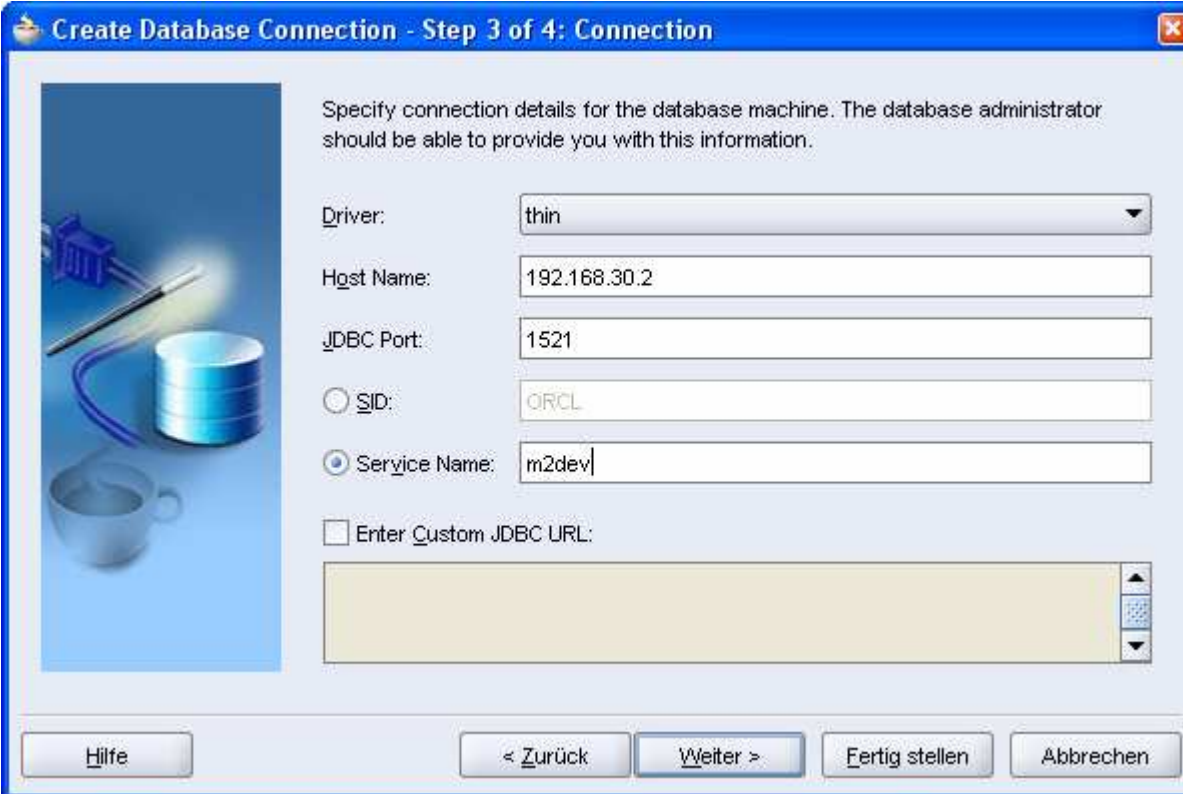
Password:  
\*\*\*\*\*

Role:

☒ Deploy Password

Hilfe   < Zurück   Weiter >   Fertig stellen   Abbrechen

4. Datenbank Host, Port und Service Name werden eingegeben.



**Create Database Connection - Step 3 of 4: Connection**

Specify connection details for the database machine. The database administrator should be able to provide you with this information.

Driver: thin

Host Name: 192.168.30.2

JDBC Port: 1521

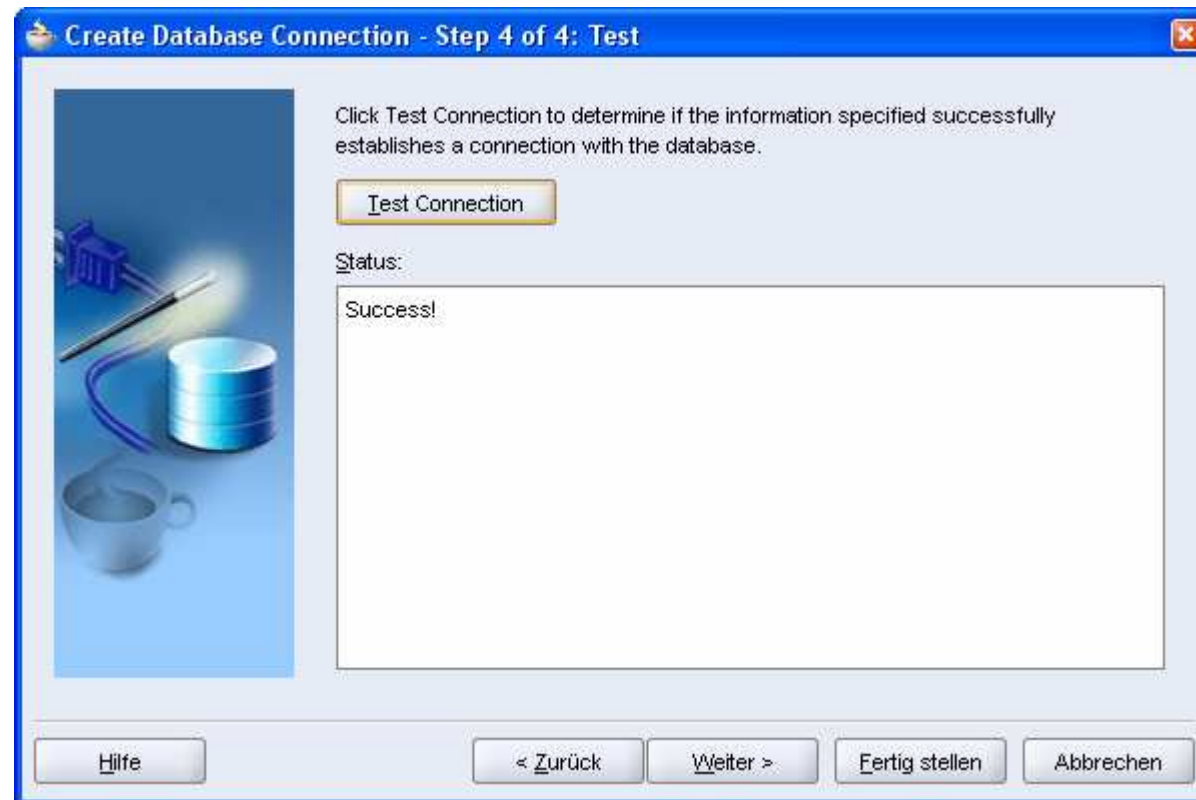
☐ SID: ORCL

☒ Service Name: m2dev

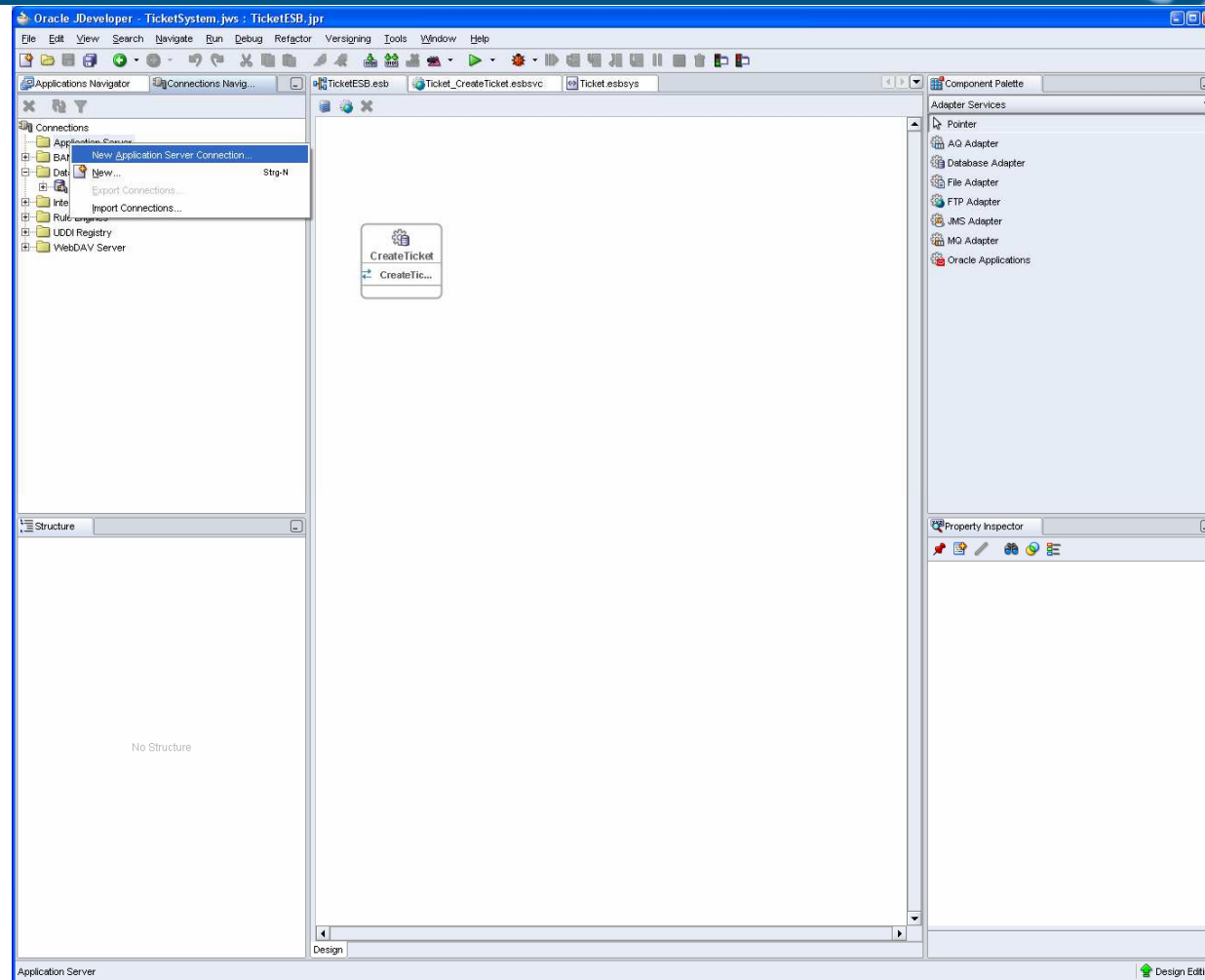
☐ Enter Custom JDBC URL:

Hilfe < Zurück Weiter > Fertig stellen Abbrechen

5. Der Connection Test muss jetzt erfolgreich durchlaufen.

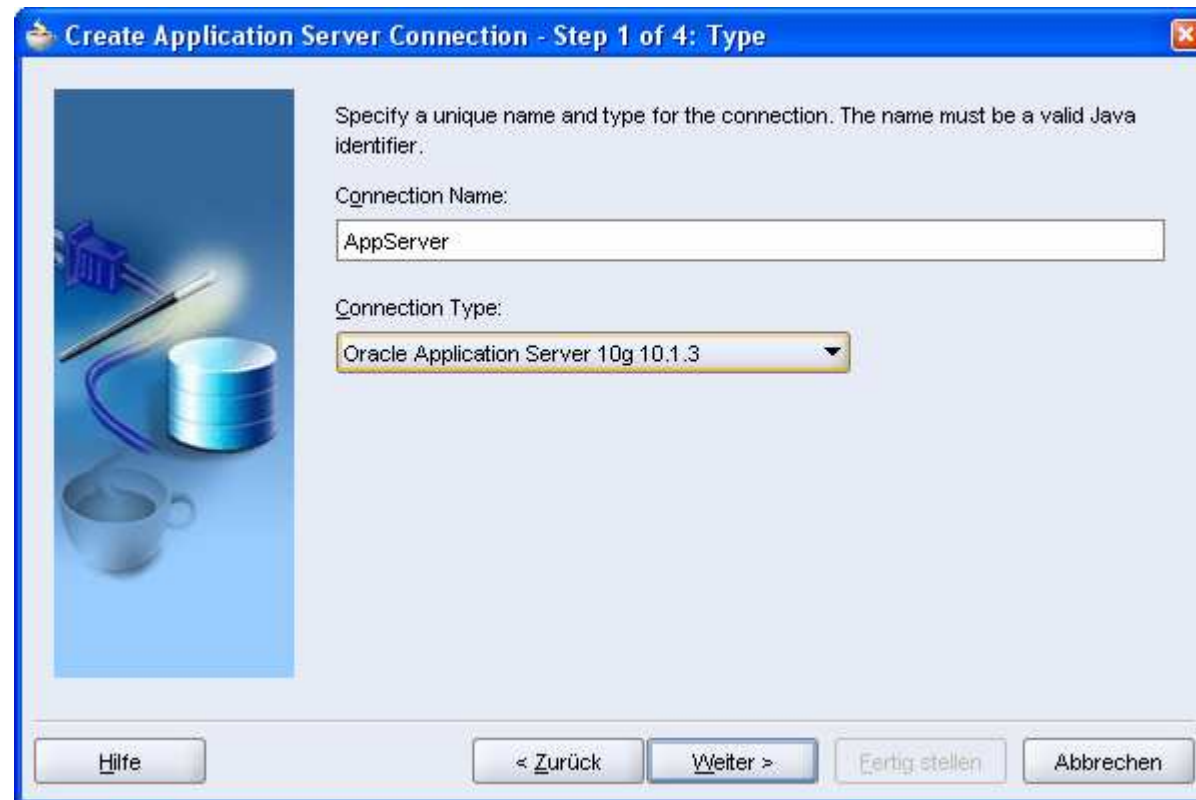


## 6. Die Application Server Connection wird erstellt.





7. Der Connection Name ist „AppServer“. Connection Type ist „Oracle Application Server 10g 10.1.3“.



8. Application server Username (default „oc4jadmin“) und Passwort werden eingegeben.



**Create Application Server Connection - Step 2 of 4: Authentication**

Specify a username and password to authenticate the connection. To bypass authentication at runtime, select Deploy Password.

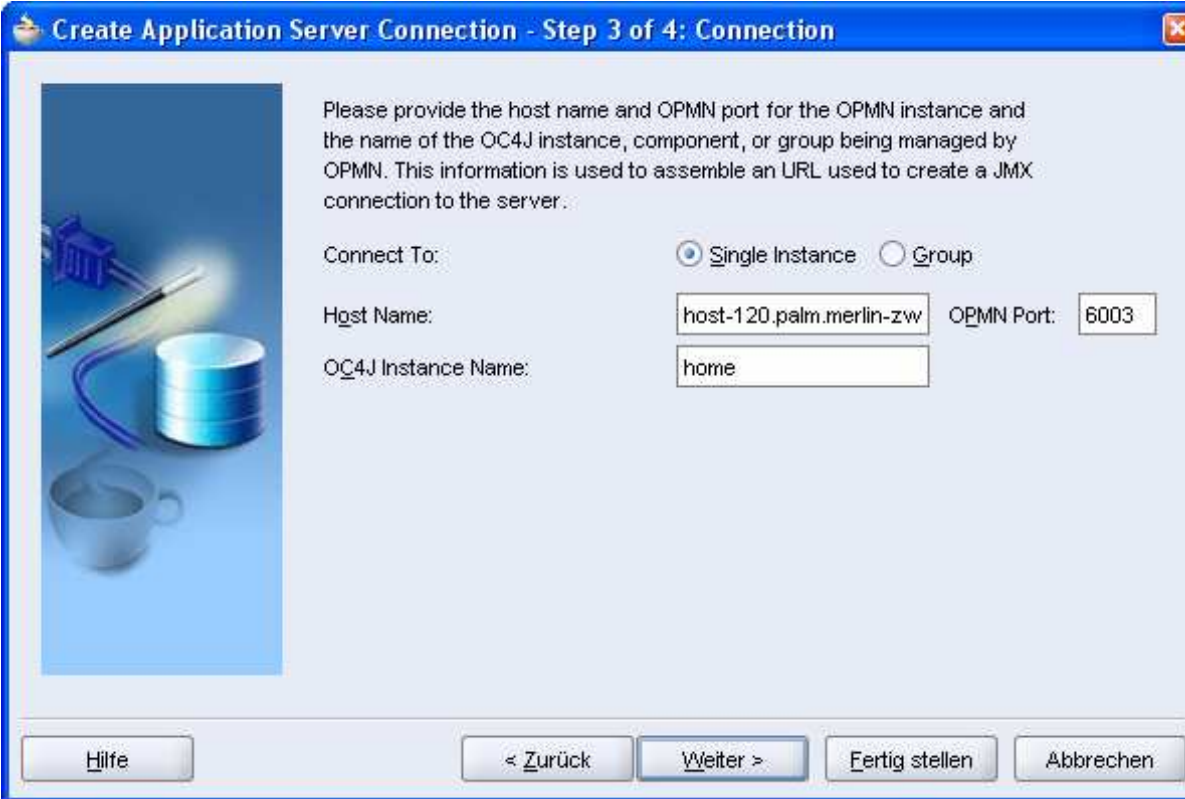
Username:  
oc4jadmin

Password:  
\*\*\*\*\*

☒ Deploy Password

Hilfe   < Zurück   Weiter >   Fertig stellen   Abbrechen

9. Application Server Host, Port (default 6003) und OC4J Instance Name (default „home“) werden eingegeben.



**Create Application Server Connection - Step 3 of 4: Connection**

Please provide the host name and OPMN port for the OPMN instance and the name of the OC4J instance, component, or group being managed by OPMN. This information is used to assemble an URL used to create a JMX connection to the server.

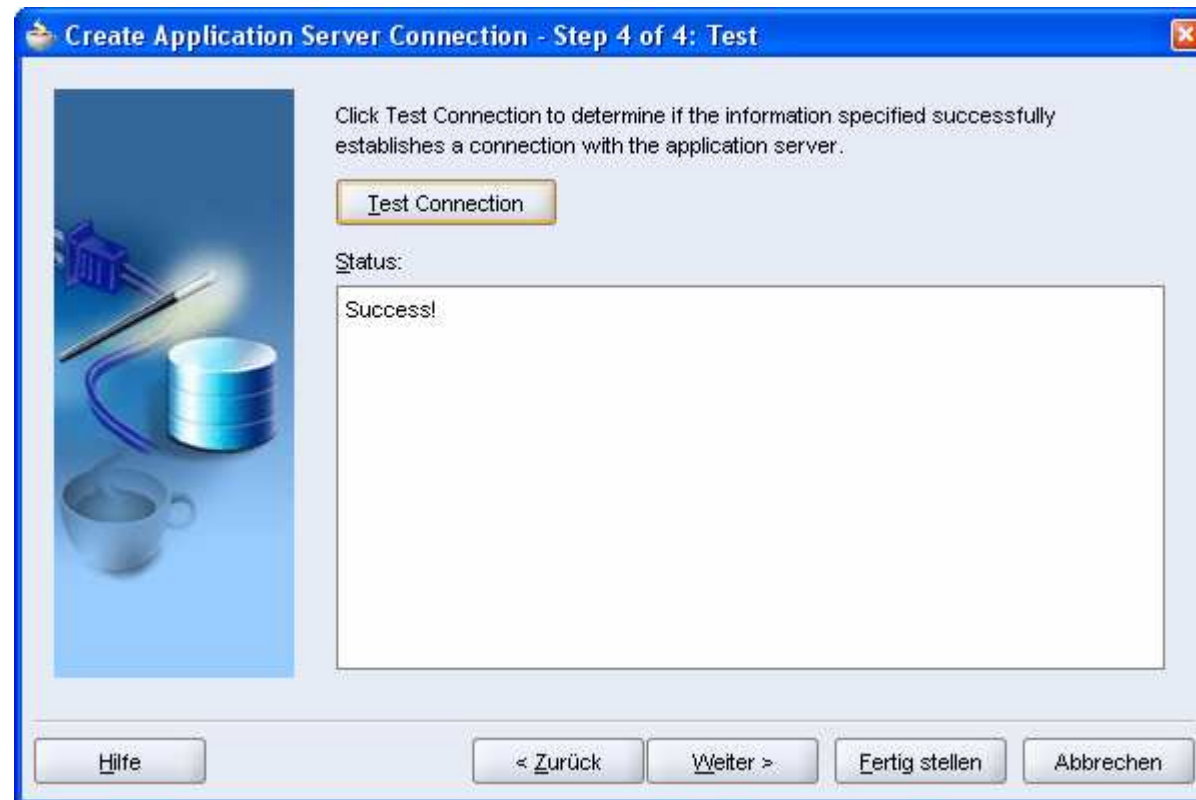
Connect To: ☒ Single Instance ☐ Group

Host Name:  OPMN Port:

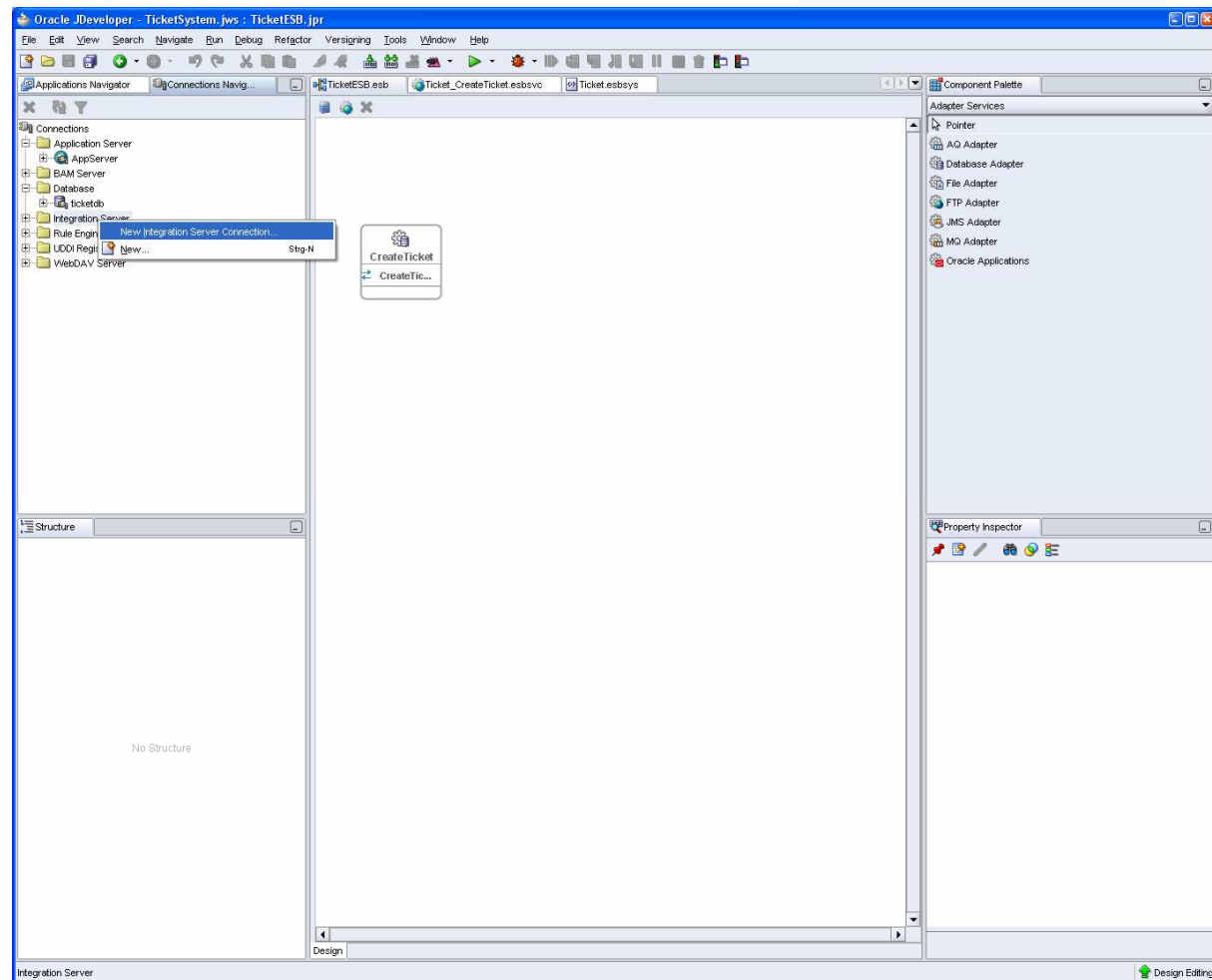
OC4J Instance Name:

Hilfe < Zurück Weiter > Fertig stellen Abbrechen

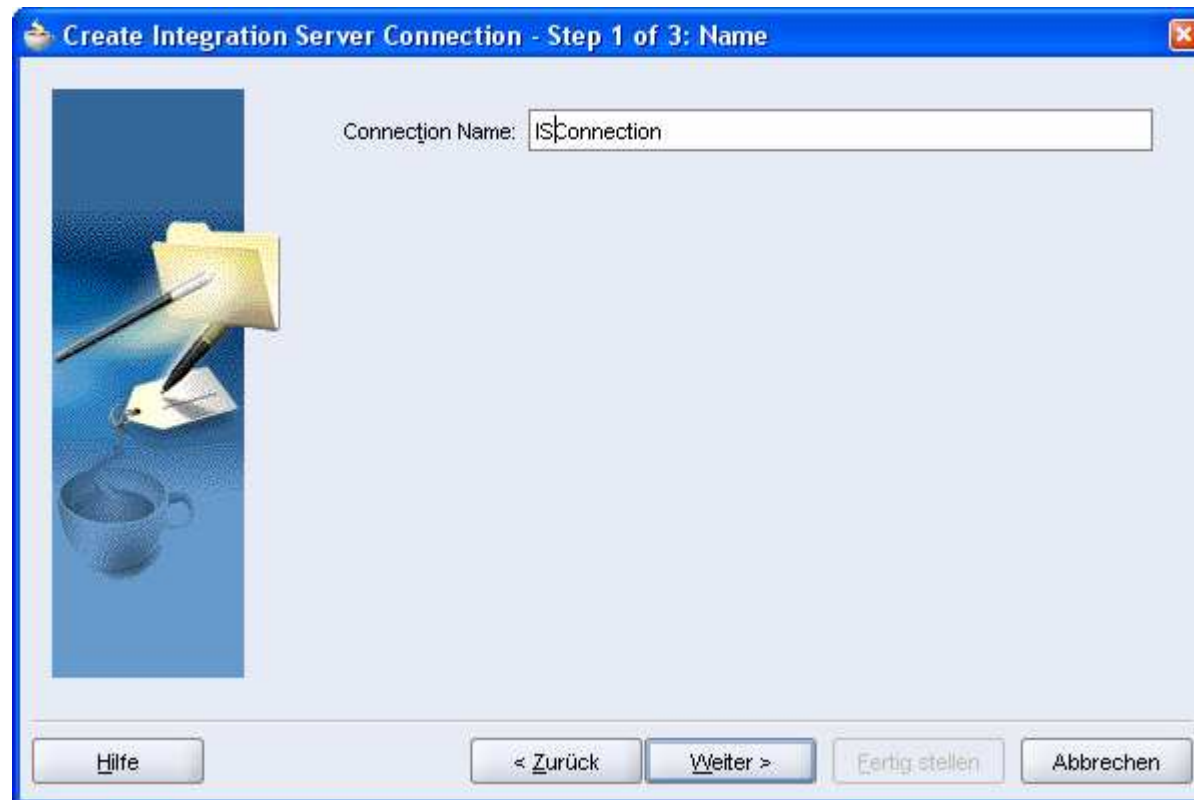
10. Connection Test muss jetzt erfolgreich durchlaufen.



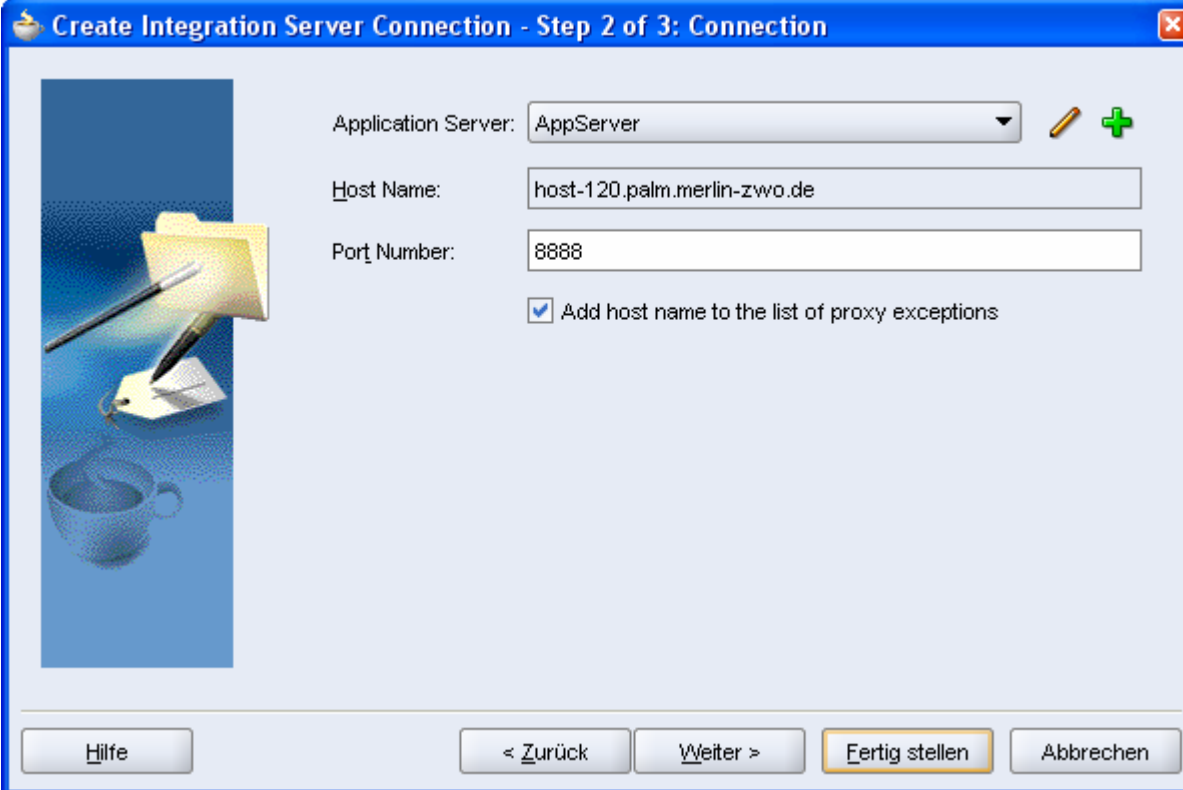
## 11. Integration Server Connection wird erstellt.



12. Connection Name ist „ISConnection“.



13. Integration Server Host und Port (default 8888) werden eingegeben.



Create Integration Server Connection - Step 2 of 3: Connection

Application Server: AppServer

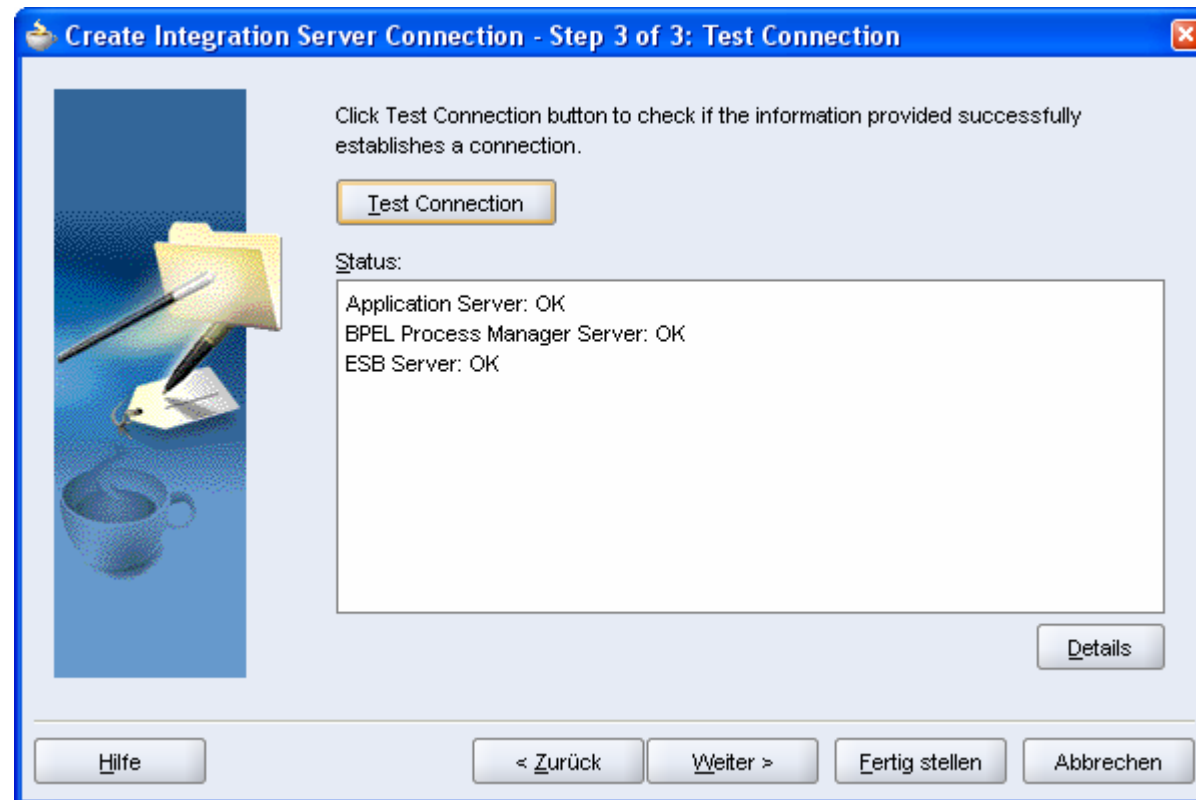
Host Name: host-120.palm.merlin-zwo.de

Port Number: 8888

☒ Add host name to the list of proxy exceptions

Hilfe < Zurück Weiter > Fertig stellen Abbrechen

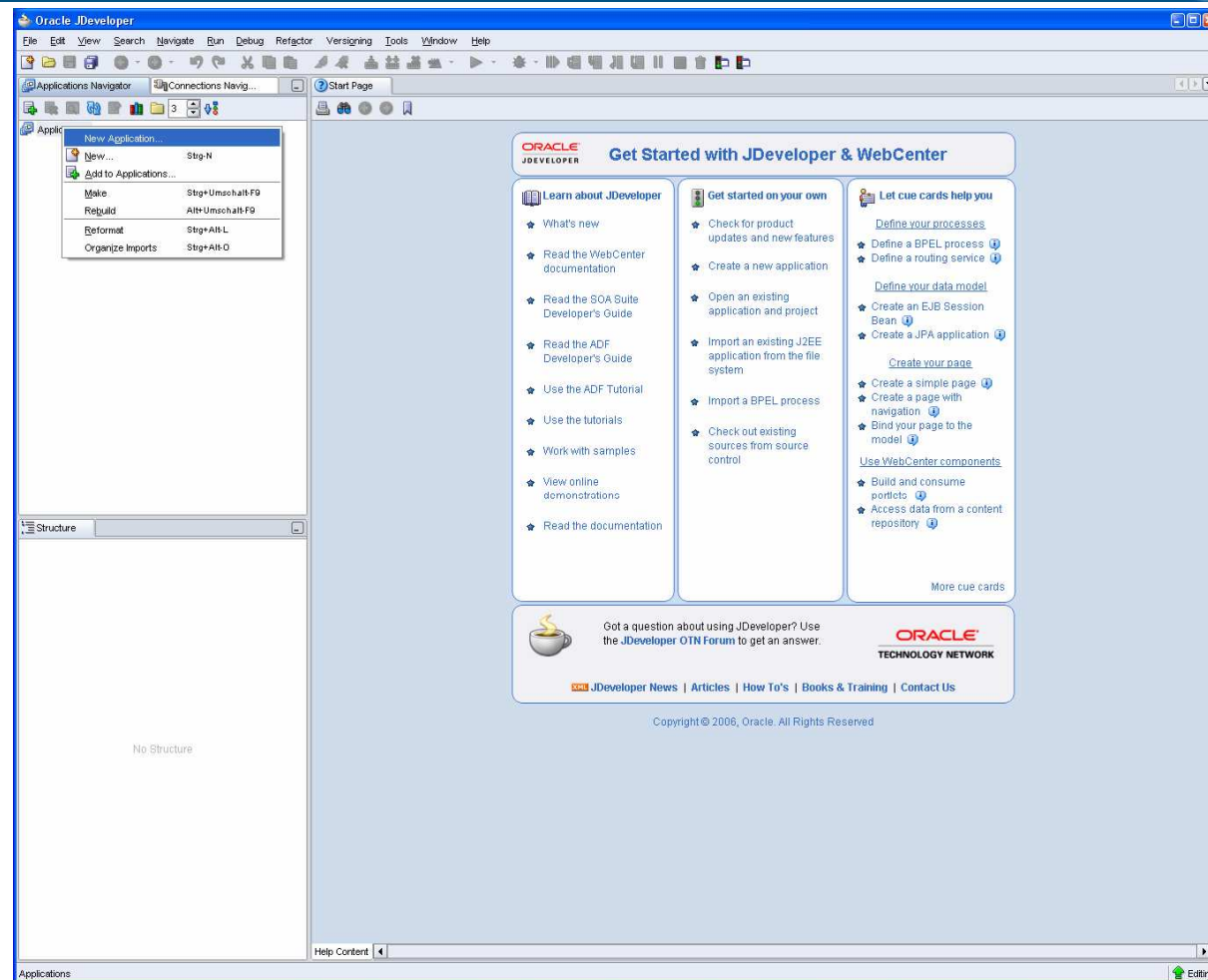
14. Connection Test muss jetzt erfolgreich durchlaufen.



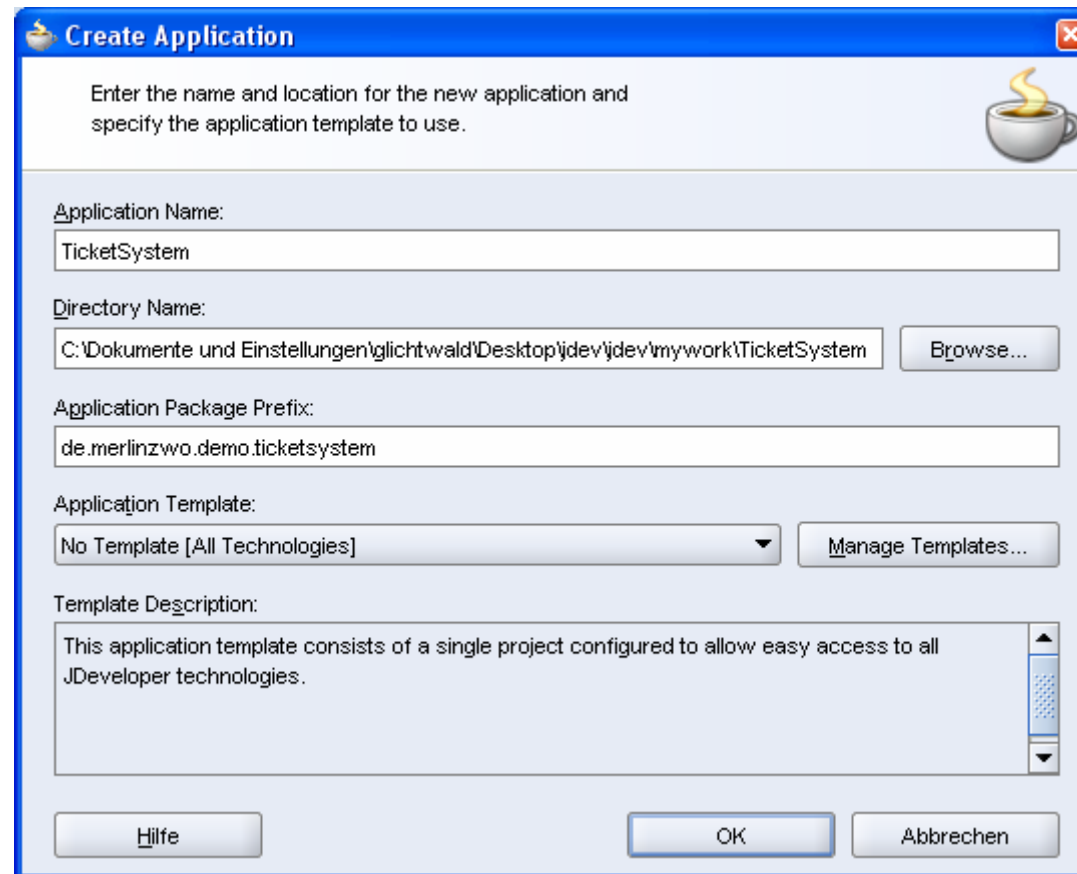


Weiter werden die Webservices in Rahmen eines ESB-Projektes erstellt.

15. In JDeveloper wird eine neue Application erstellt.



16. Der Name der Application ist „TicketSystem“. Application Package Prefix Ist „de.merlinzwo.demo.ticketsystem“.



**Create Application**

Enter the name and location for the new application and specify the application template to use.

Application Name:  
TicketSystem

Directory Name:  
C:\Dokumente und Einstellungen\glichtwald\Desktop\jdev\jdev\mywork\TicketSystem [Browse...](#)

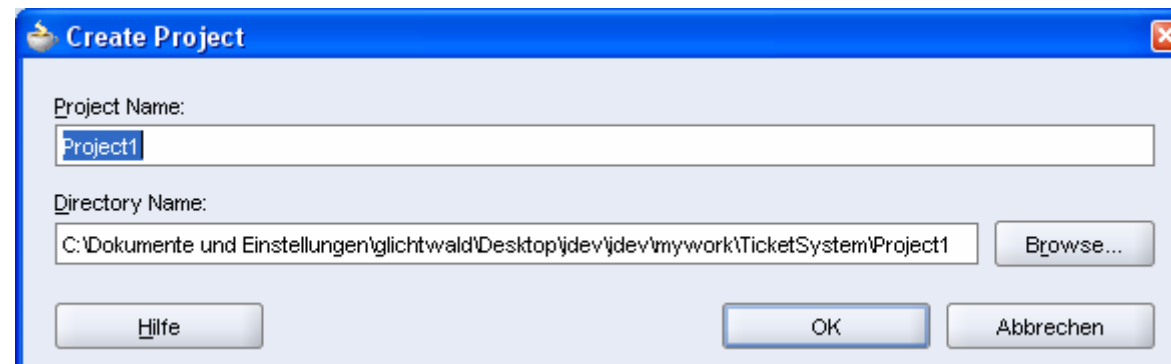
Application Package Prefix:  
de.merlinzwo.demo.ticketsystem

Application Template:  
No Template [All Technologies] [Manage Templates...](#)

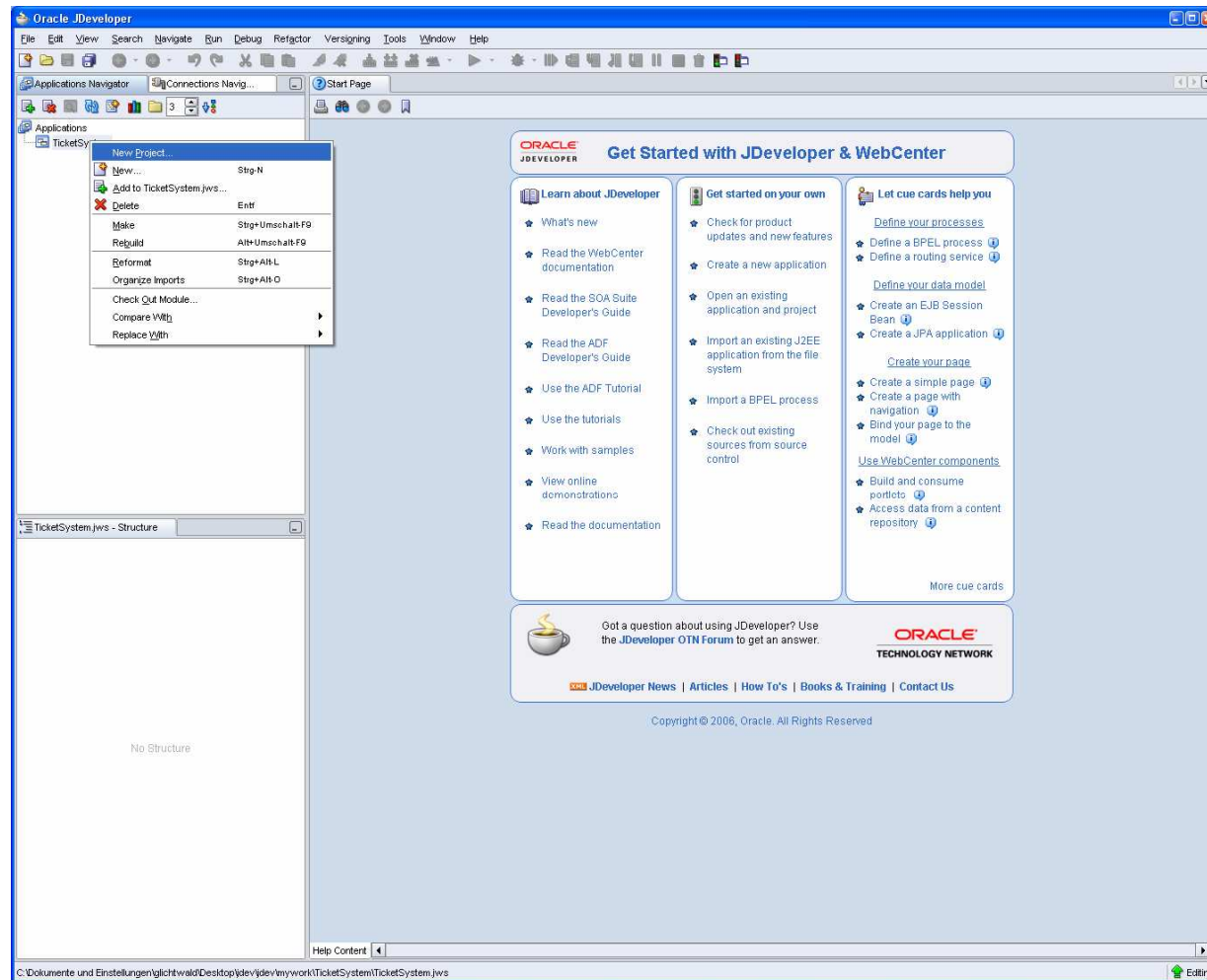
Template Description:  
This application template consists of a single project configured to allow easy access to all JDeveloper technologies.

[Hilfe](#) [OK](#) [Abbrechen](#)

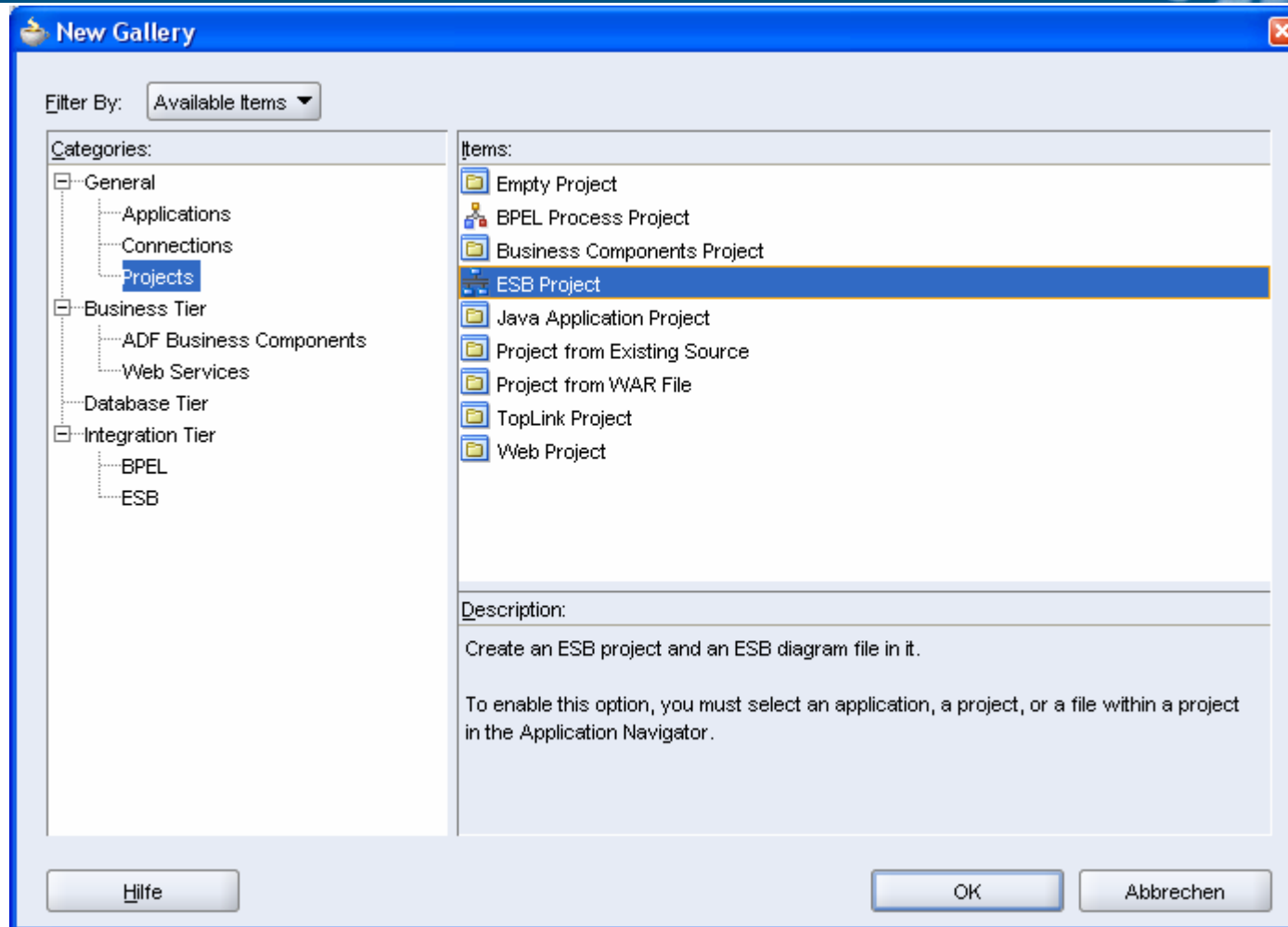
17. Dialogfenster „Create Project“ wird durch „Abbrechen“ Button geschlossen.



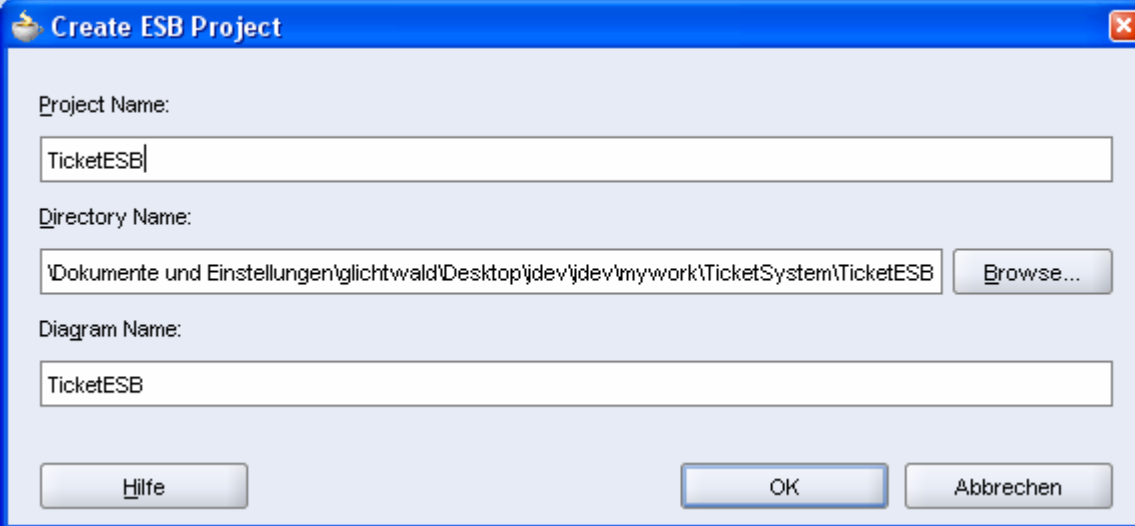
18. Innerhalb der TicketSystem Application wird ein neues Projekt erstellt.



19. „ESB Project“ wird ausgewählt.



20. Der Name des Projektes ist „TicketESB“.



**Create ESB Project**

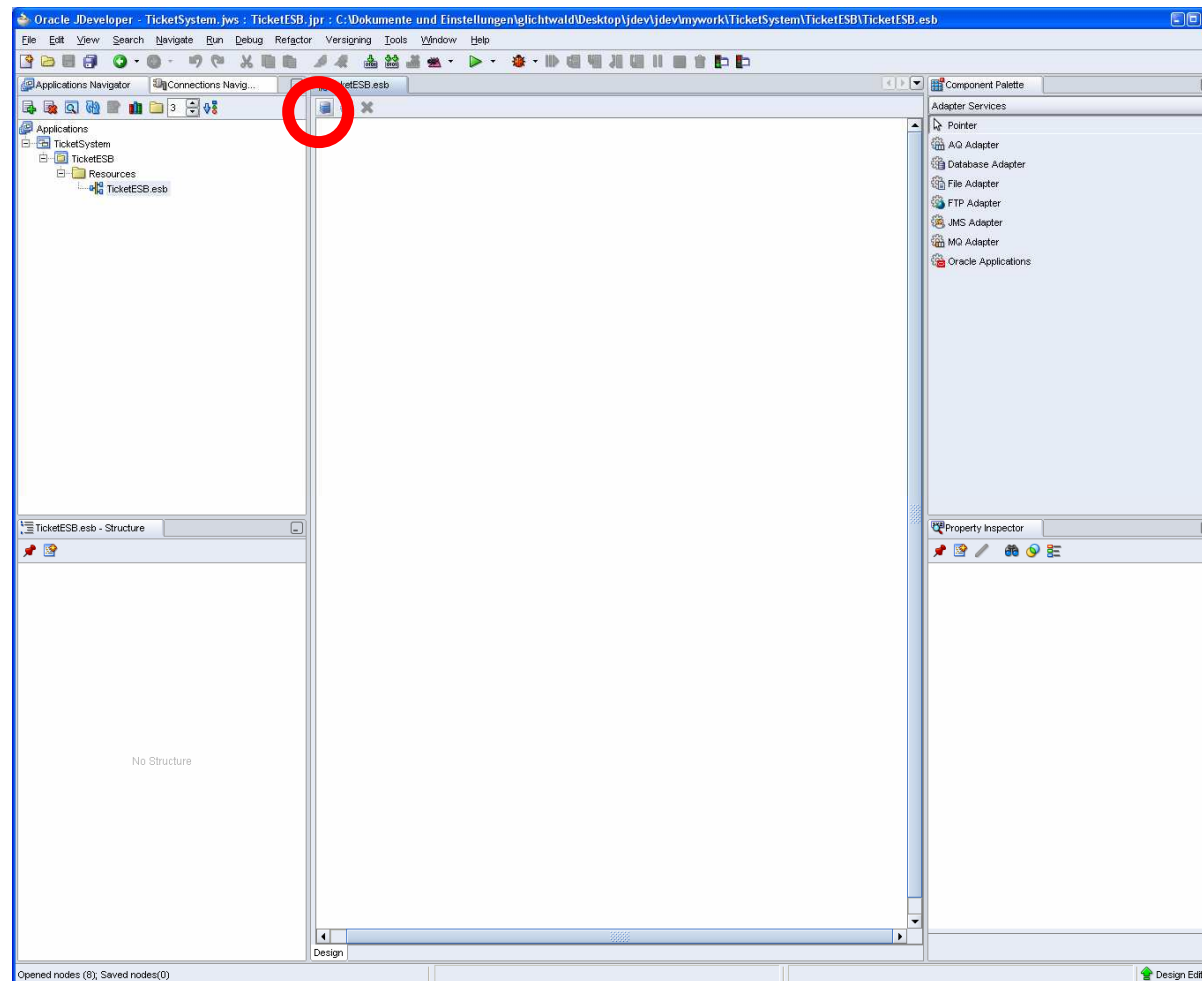
Project Name:  
TicketESB

Directory Name:  
'Dokumente und Einstellungen\glichtwald\Desktop\jdev\jdev\mywork\TicketSystem\TicketESB' Browse...

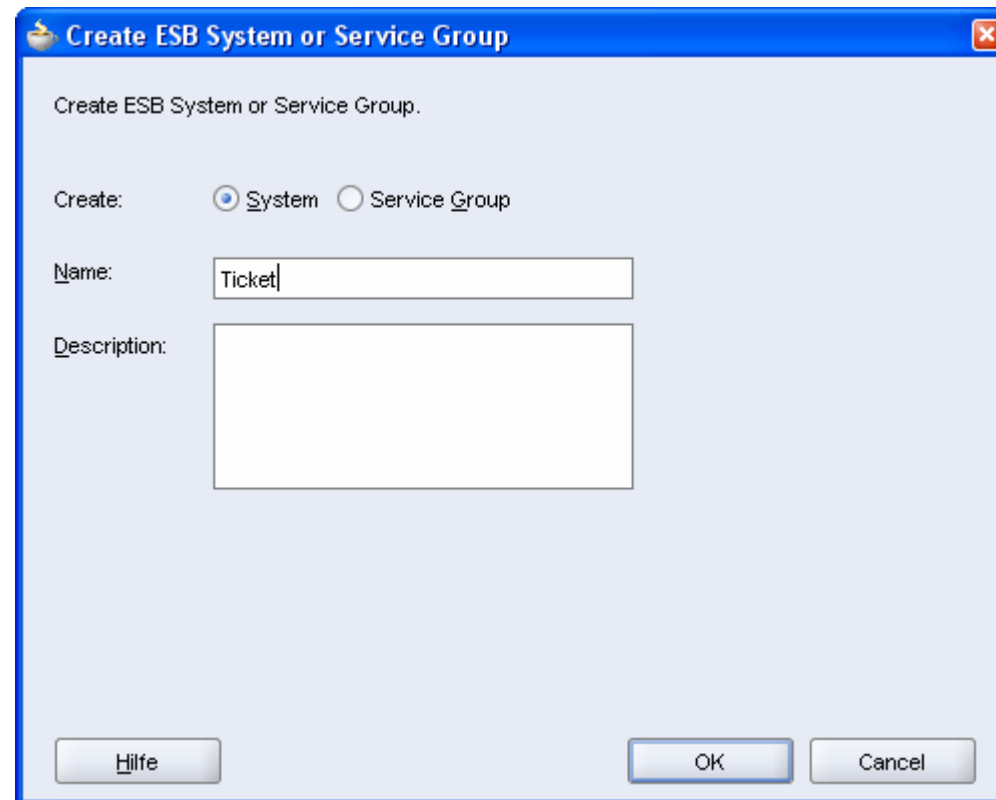
Diagram Name:  
TicketESB

Hilfe OK Abbrechen

21. Beim Klicken auf „Create System/Group“ Button wird ESB System erstellt.



22. Name des ESB System ist „Ticket“.



Create ESB System or Service Group

Create ESB System or Service Group.

Create: ☒ System ☐ Service Group

Name: Ticket

Description:

Hilfe OK Cancel



23. Durch drag and drop vom „Database Adapter“ Element aus der Component Palette ins TicketESB.esb Fenster wird ein Database Adapter erstellt.

**Create Database Adapter Service**

Define the database adapter service

Name:

System/Group:

Description:

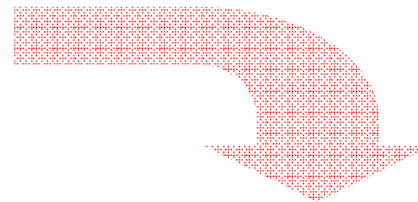
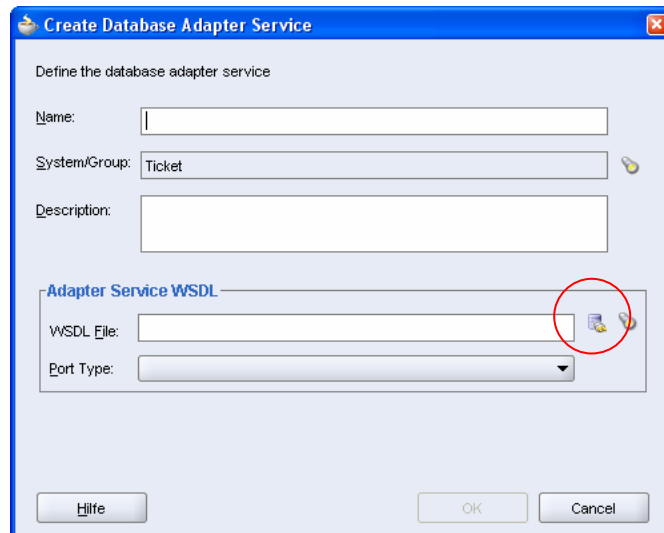
**Adapter Service WSDL**

WSDL File:

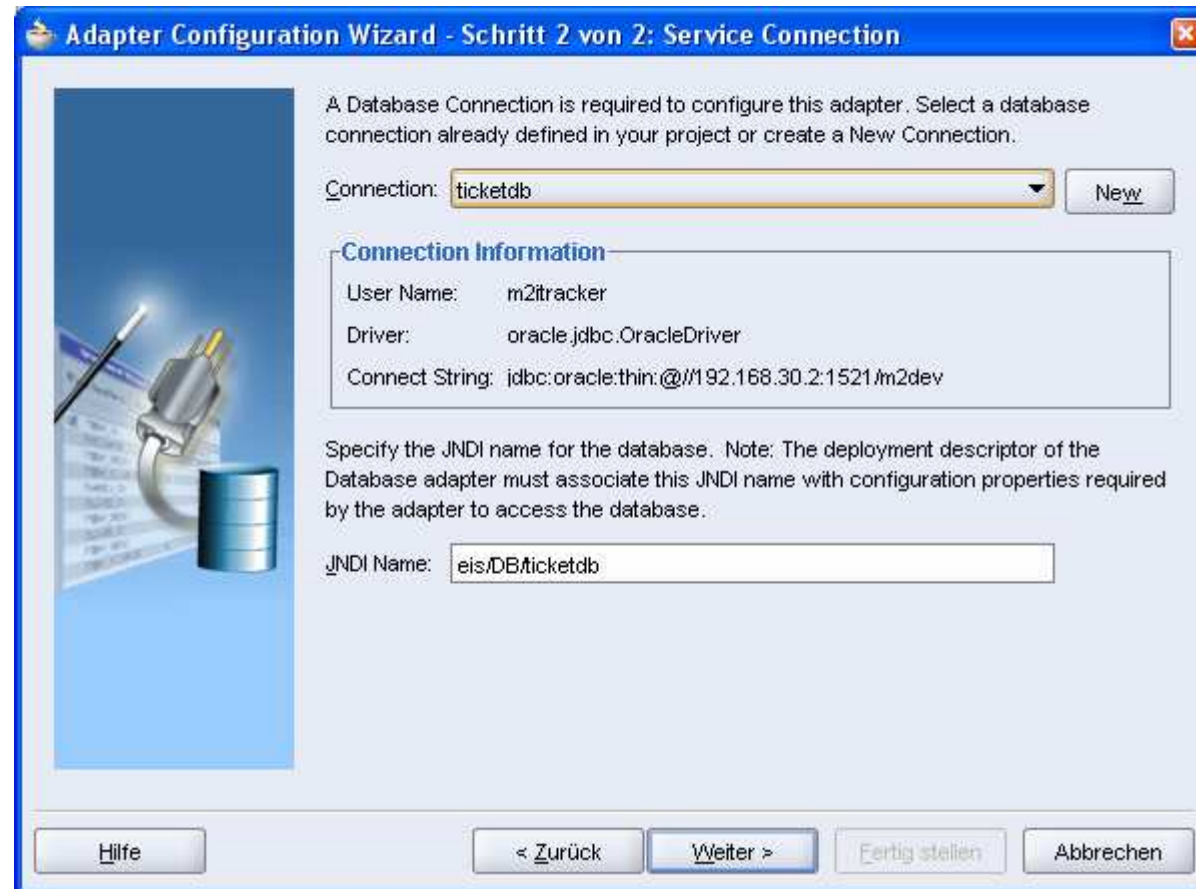
Port Type:

Hilfe OK Cancel

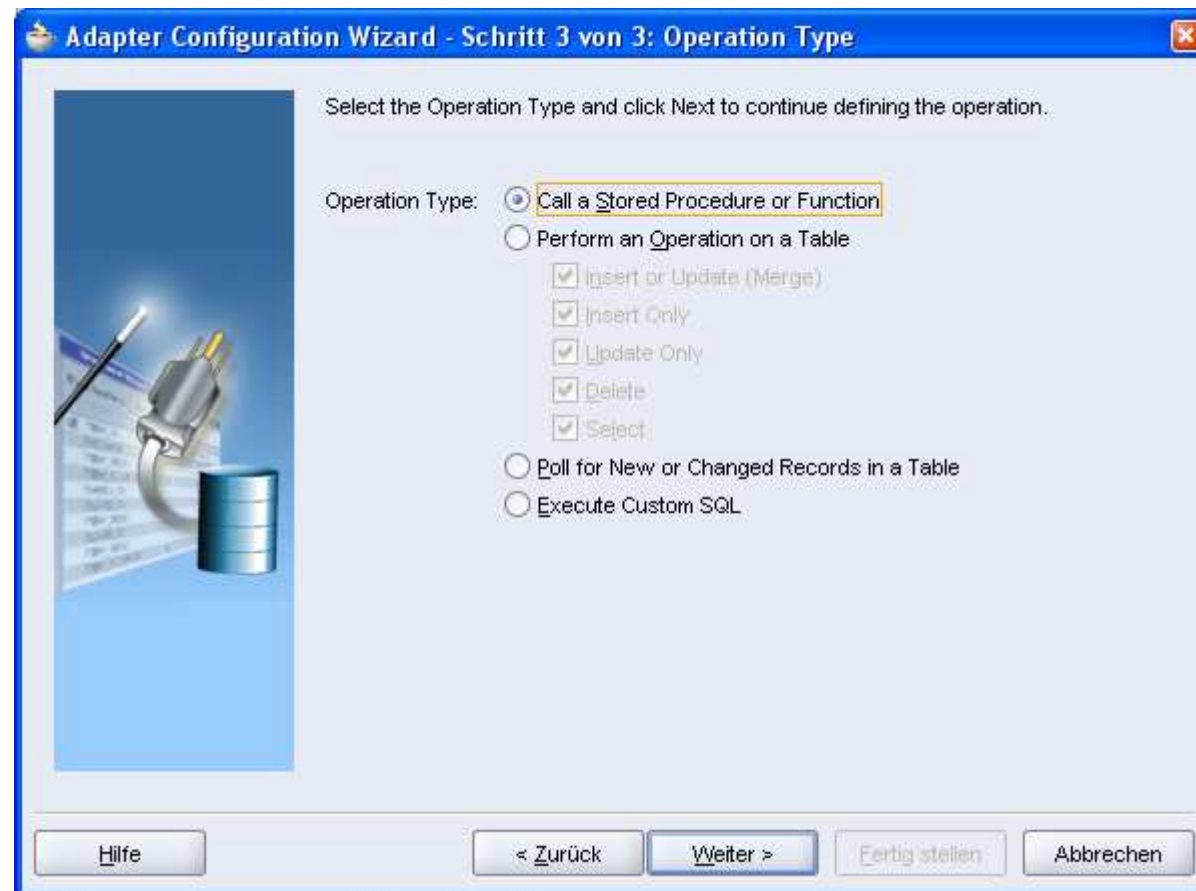
24. Durch Klicken auf Button rechts neben „WSDL File“ Feld wird Database Adapter Configuration Wizard aufgerufen.



25. „ticketdb“ Datenbank Connection wird ausgewählt. JNDI Name wird automatisch generiert.



26. „Call a Stored Procedure or Function“ Operation Type wird ausgewählt.



27. Datenbank Schema („M2ITRACKER“) und Procedure („BUGMAGIC.ISSUE\_CREATE“) Werden angegeben. Im nächsten Schritt wird der Adapter Configuration Wizard abgeschlossen.

**Adapter Configuration Wizard - Schritt 4 von 4: Specify Stored Procedure**

Enter a stored procedure, or a function. The procedure's package name can be included, for example, EMPLOYEE.GET\_NAME, where the package name is EMPLOYEE and the procedure is GET\_NAME. If the procedure does not belong in a package, enter the procedure's name. You can also browse and search for a procedure. The term 'procedure' is used to mean both stored procedures as well as functions.

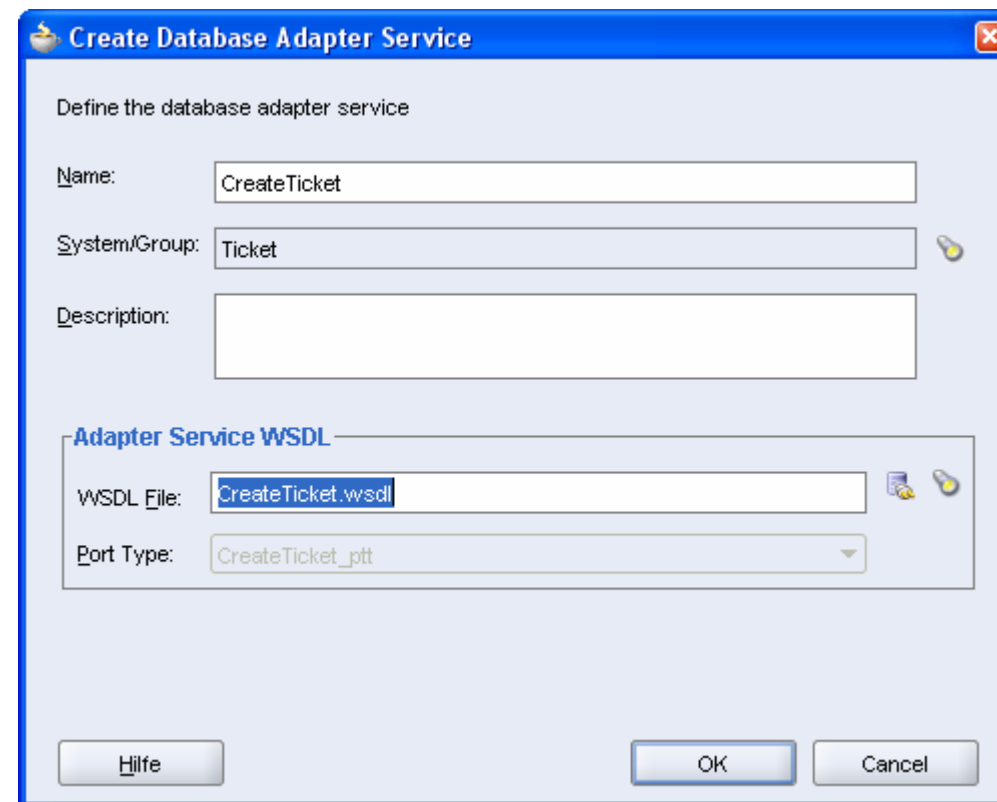
Schema:

Procedure:

Arguments

Name	Type	In/Out	Position
XVERSION	NUMBER	IN	10
XCOMPONENT	NUMBER	IN	9
XSTATUS	NUMBER	IN	8
XPRIOR	NUMBER	IN	7
XSEVERITY	NUMBER	IN	6
XURL	VARCHAR2	IN	5
XDESCRIPTION	VARCHAR2	IN	4
XNAME	VARCHAR2	IN	3
XISSUE_KEY	NUMBER	OUT	2

28. Name des Database Adapter ist „CreateTicket“.



**Create Database Adapter Service**

Define the database adapter service

Name: CreateTicket

System/Group: Ticket

Description:

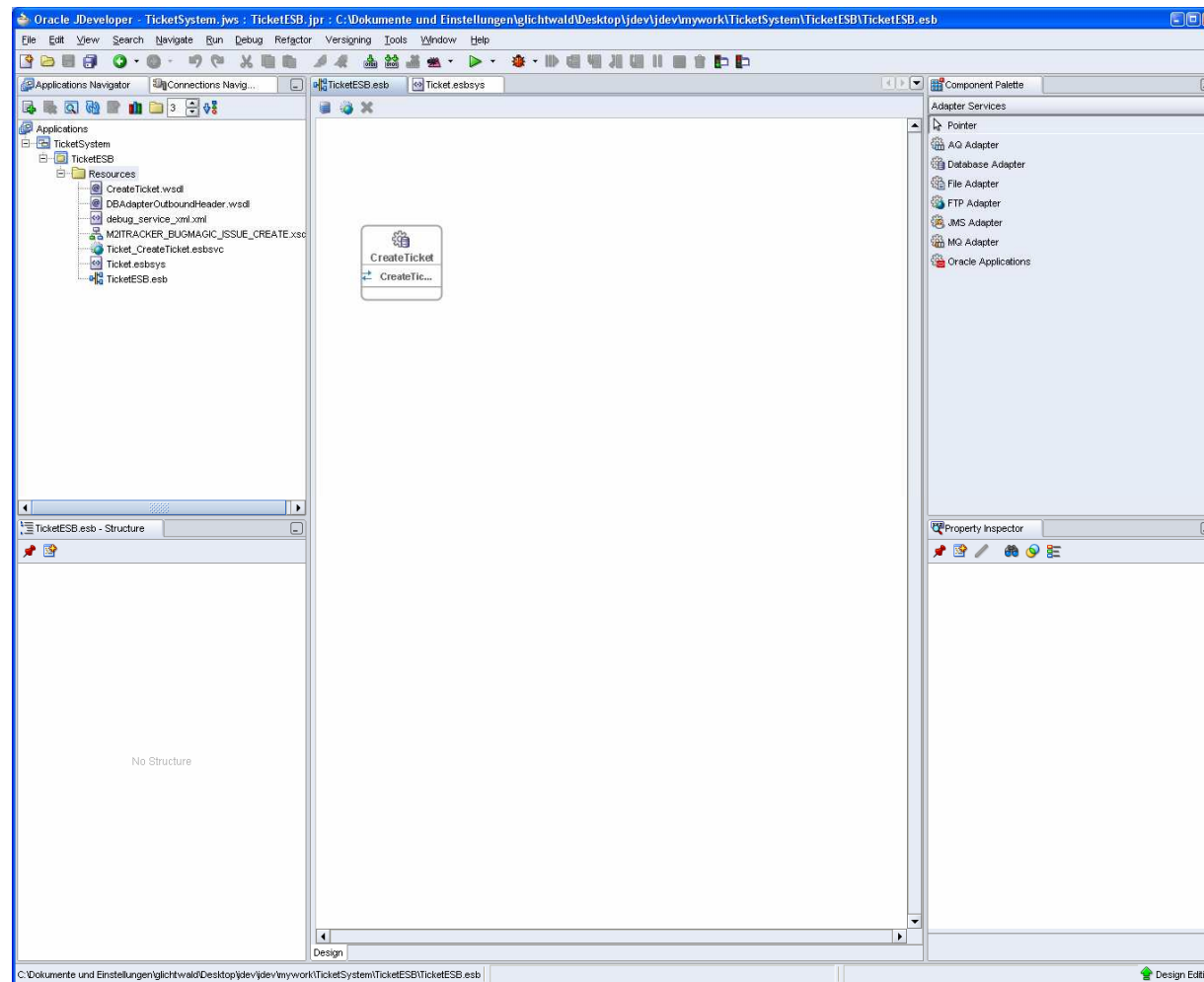
**Adapter Service WSDL**

WSDL File: CreateTicket.wsdl

Port Type: CreateTicket\_ptt

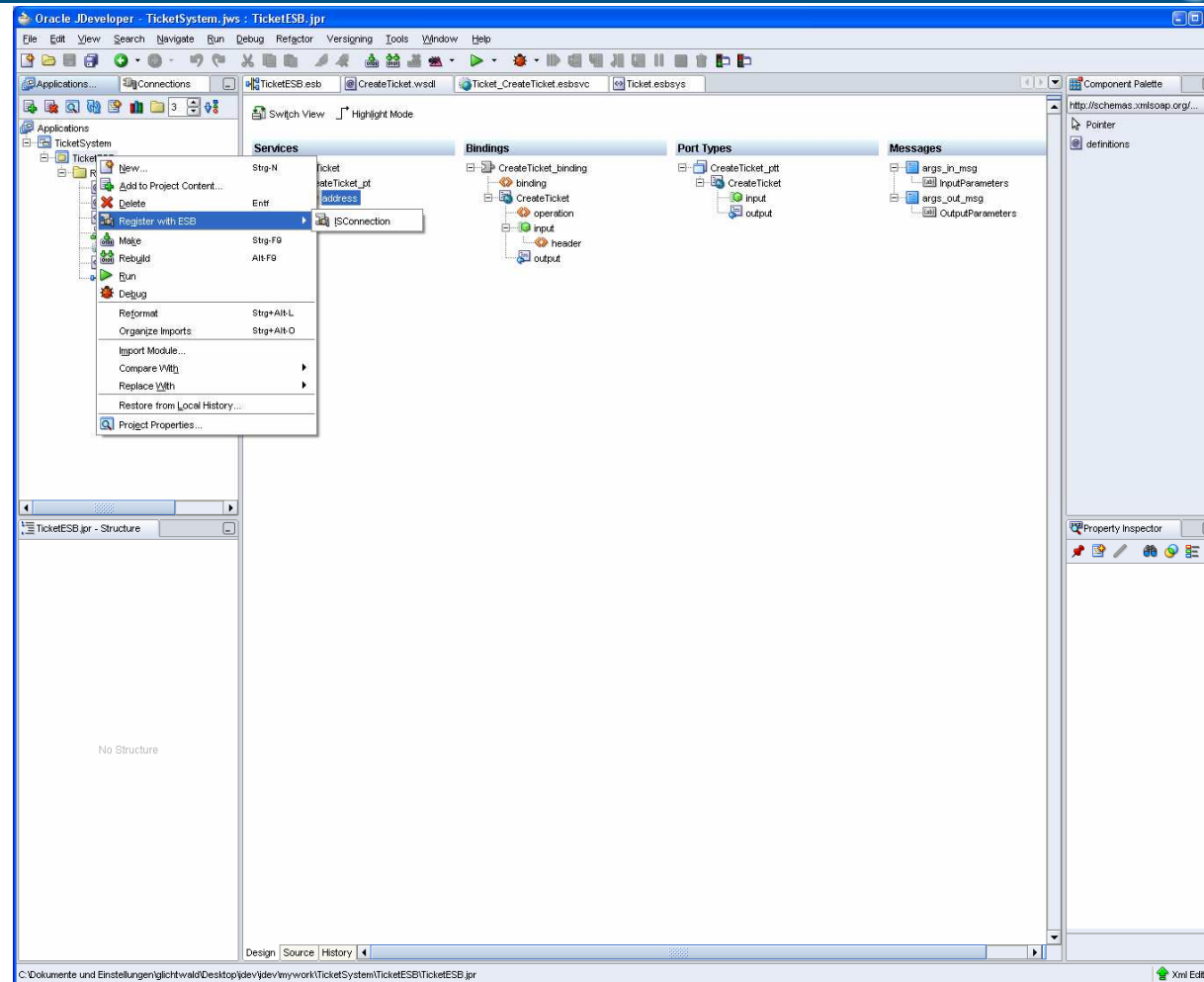
Hilfe OK Cancel

29. In TicketESB.esb Fenster erscheint der Database Adapter in Form eines Rechtecks.



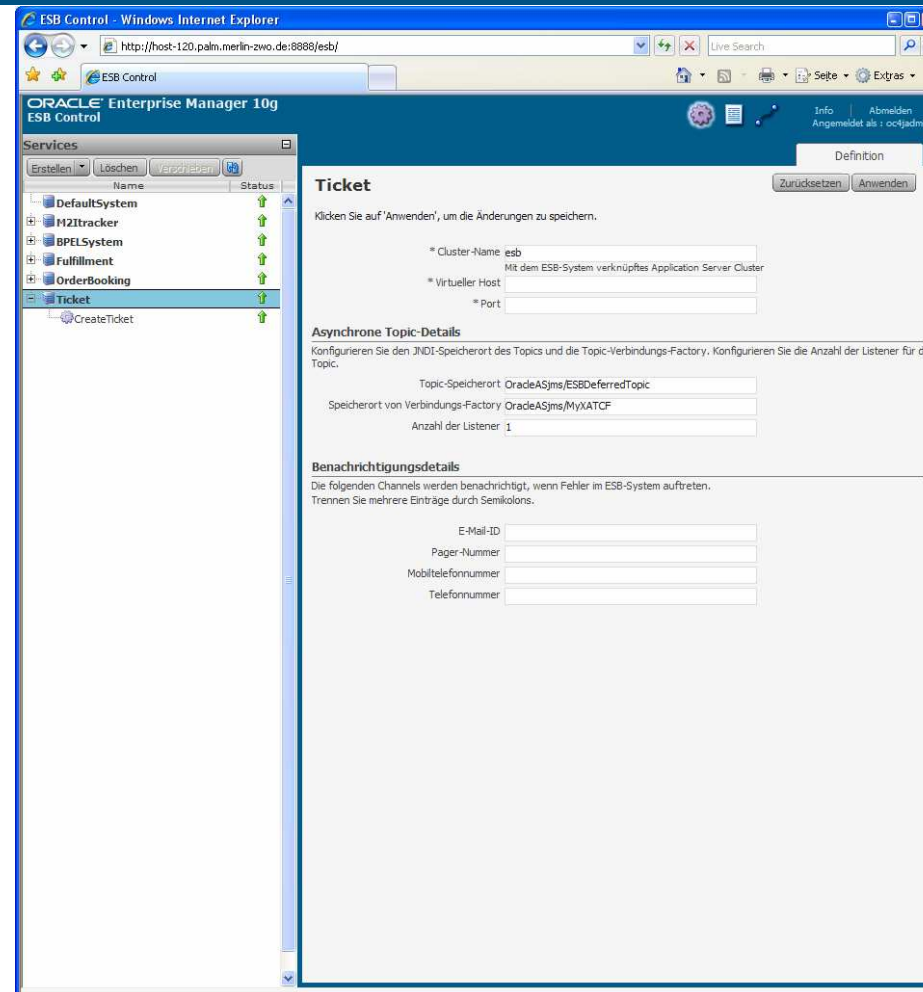
Das Projekt kann jetzt beim Integration Server registriert werden.

30. Mit rechtem Mausklick auf den Projektknoten > Register with ESB > ISConnection wird das Projekt registriert.



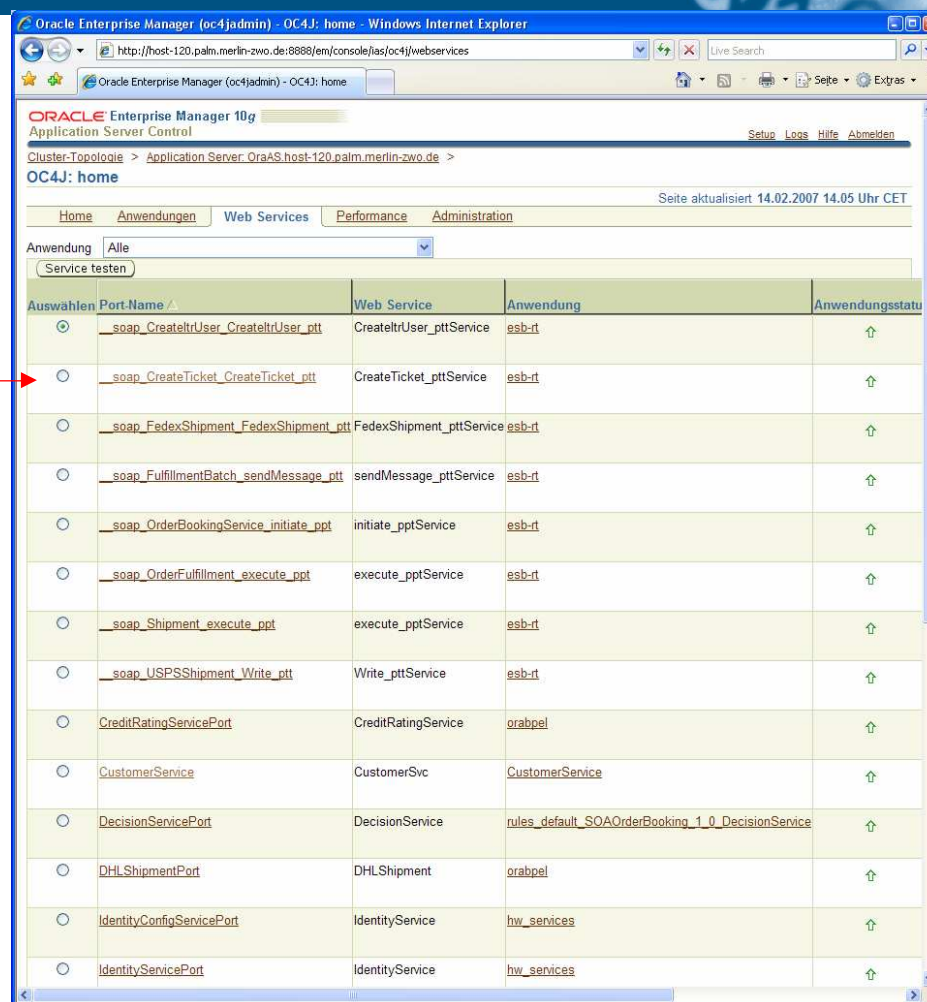


31. Der erfolgreich registrierte ESB ist in ESB Control von Oracle SOA Suite zu sehen.



32. Der Database Adapter ist in Application Server Control als Webservice zu sehen.

„CreateTicket“  
Adapter Service



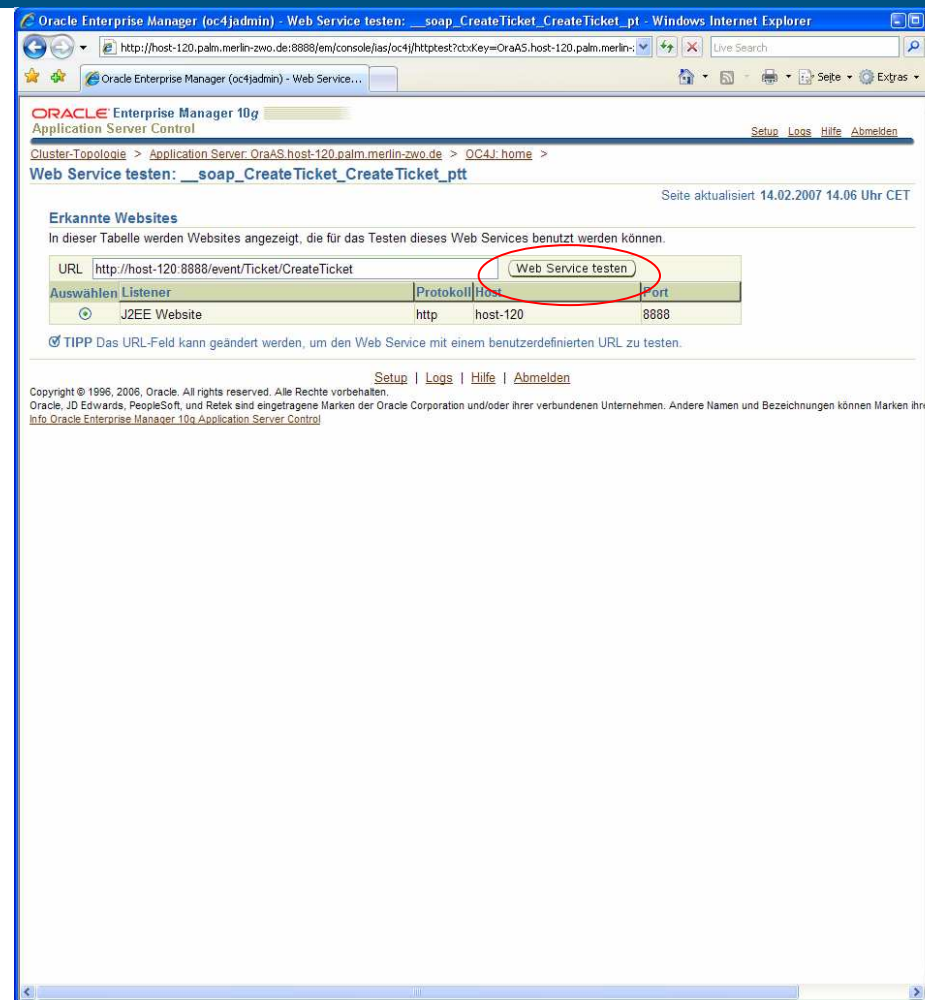
Oracle Enterprise Manager 10g  
Application Server Control  
Cluster-Topologie > Application Server: OraAS host-120.palm.merlin-zwo.de >  
OC4J: home  
Seite aktualisiert 14.02.2007 14:05 Uhr CET

Home Anwendungen Web Services Performance Administration

Anwendung: Alle  
Service testen

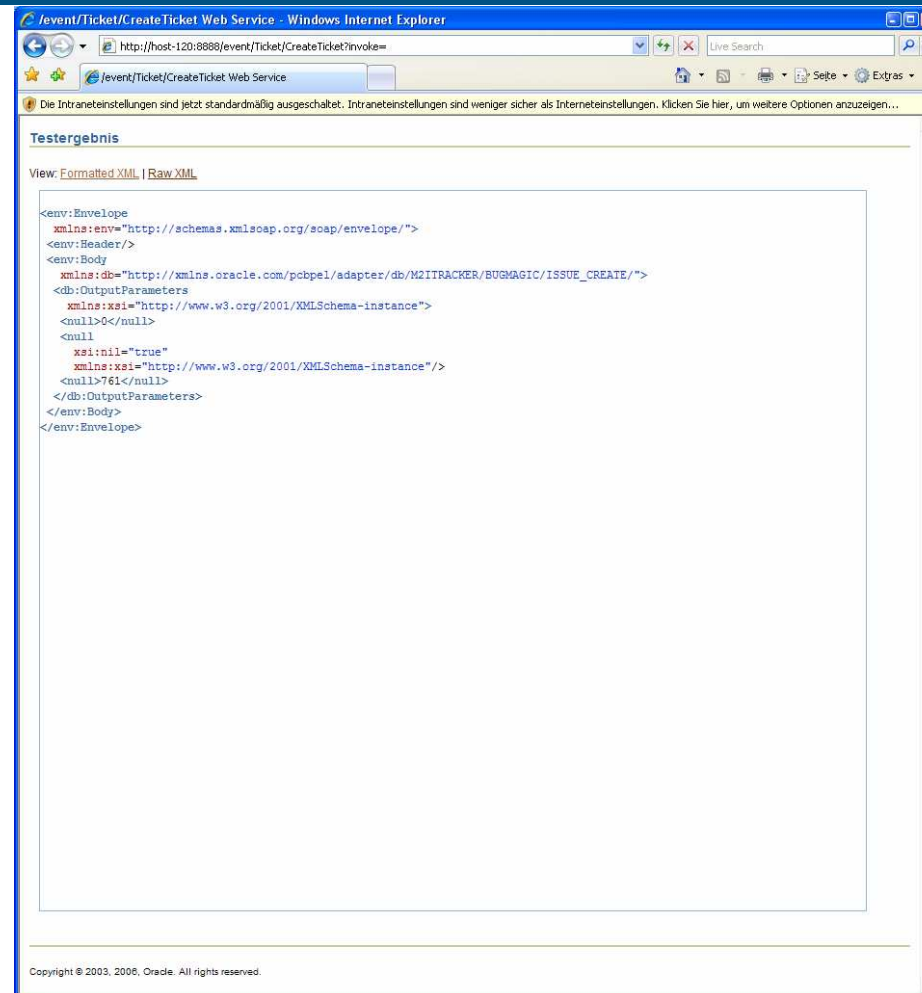
Auswählen	Port-Name	Web Service	Anwendung	Anwendungsstatus
<input checked="" type="radio"/>	soap_CreateUser_CreateUser_ptt	CreateUser_pttService	esb-rt	↑
<input type="radio"/>	soap_CreateTicket_CreateTicket_ptt	CreateTicket_pttService	esb-rt	↑
<input type="radio"/>	soap_FedexShipment_FedexShipment_ptt	FedexShipment_pttService	esb-rt	↑
<input type="radio"/>	soap_FulfillmentBatch_sendMessage_ptt	sendMessage_pttService	esb-rt	↑
<input type="radio"/>	soap_OrderBookingService_initiate_ptt	initiate_pttService	esb-rt	↑
<input type="radio"/>	soap_OrderFulfillment_execute_ptt	execute_pttService	esb-rt	↑
<input type="radio"/>	soap_Shipment_execute_ptt	execute_pttService	esb-rt	↑
<input type="radio"/>	soap_USPSShipment_Write_ptt	Write_pttService	esb-rt	↑
<input type="radio"/>	CreditRatingServicePort	CreditRatingService	orabpel	↑
<input type="radio"/>	CustomerService	CustomerSvc	CustomerService	↑
<input type="radio"/>	DecisionServicePort	DecisionService	rules_default_SOAOOrderBooking_1_0_DecisionService	↑
<input type="radio"/>	DHLShipmentPort	DHLShipment	orabpel	↑
<input type="radio"/>	IdentityConfigServicePort	IdentityService	hw_services	↑
<input type="radio"/>	IdentityServicePort	IdentityService	hw_services	↑

### 33. Der Adapter Service kann in Application Server Control getestet werden.



34. Im Testformular werden Eingabewerte für den Webservice angegeben und anschließend „Invoke“ Button geklickt.

35. Auf der nächsten Seite wird der SOAP-Response Message des Webservice angezeigt.



Testergebnis

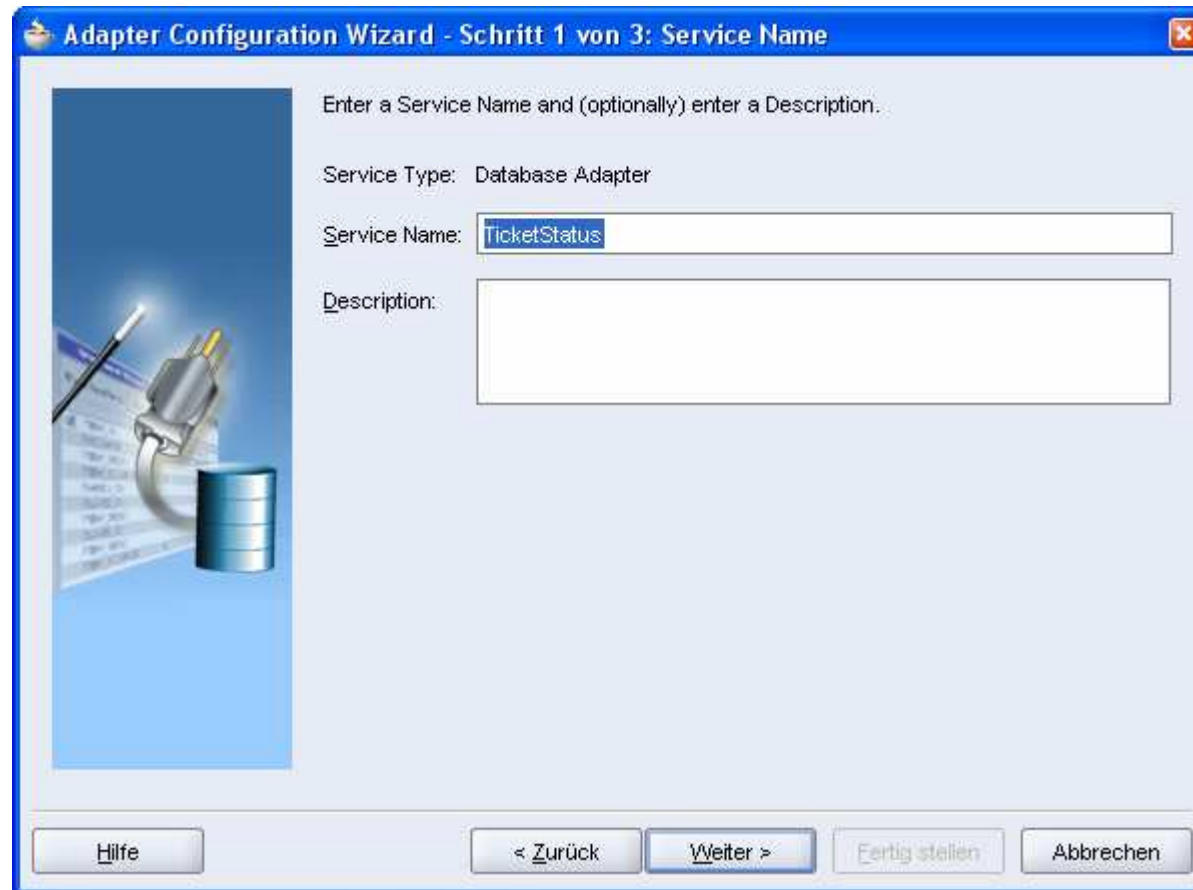
View: [Formatted XML](#) | [Raw XML](#)

```
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
<env:Header/>
<env:Body
  xmlns:db="http://xmlns.oracle.com/pdbpel/adapter/db/M2ITRACKER/BUGMAGIC/ISSUE_CREATE/"
  <db:OutputParameters
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    <null>0</null>
    <null>
      xsi:nil="true"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
    <null>761</null>
  </db:OutputParameters>
</env:Body>
</env:Envelope>
```

Copyright © 2003, 2006, Oracle. All rights reserved.

Als nächstes wird noch ein Database Adapter ins „TicketESB“ Projekt eingefügt, der eine SELECT-Operation aus Datenbank Tabellen präsentiert.

36. Im „TicketESB“ Projekt wird ein neuer Database Adapter hinzugefügt (siehe Schritt 22).  
Der Name des Adapter ist „TicketStatus“.



Adapter Configuration Wizard - Schritt 1 von 3: Service Name

Enter a Service Name and (optionally) enter a Description.

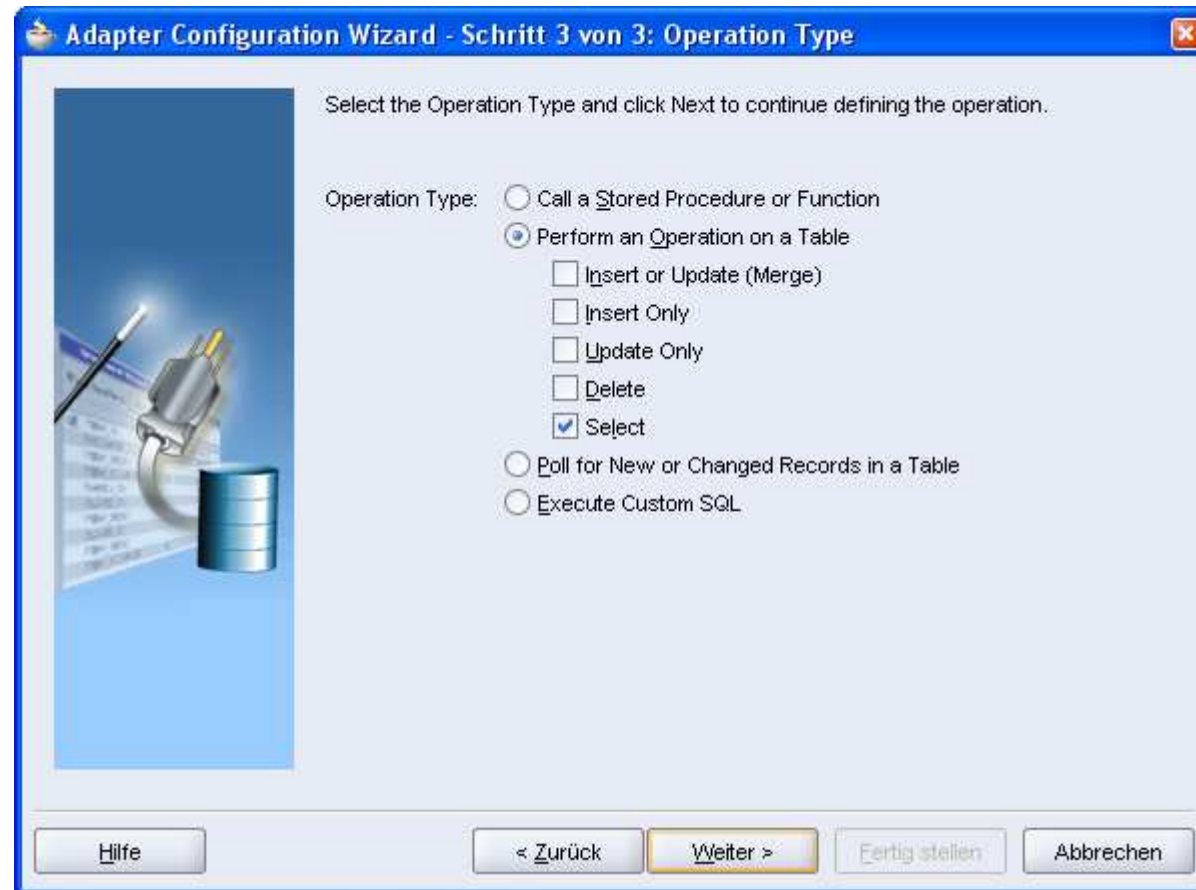
Service Type: Database Adapter

Service Name: TicketStatus

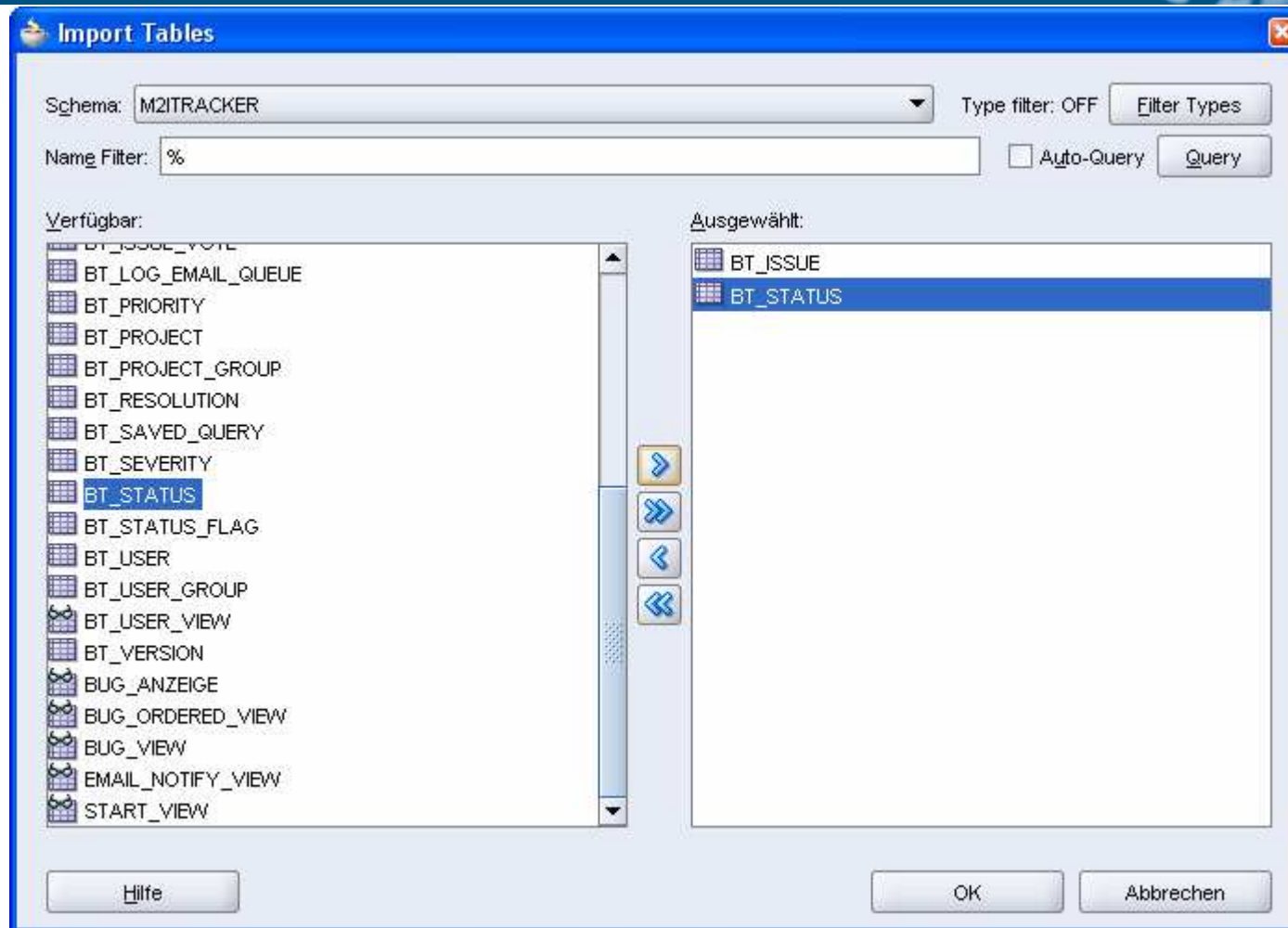
Description:

Hilfe < Zurück Weiter > Fertig stellen Abbrechen

37. In Schritt 3 vom Adapter Configuration Wizard wird „Perform an Operation on a Table“, nur „Select“, Operation Type ausgewählt.

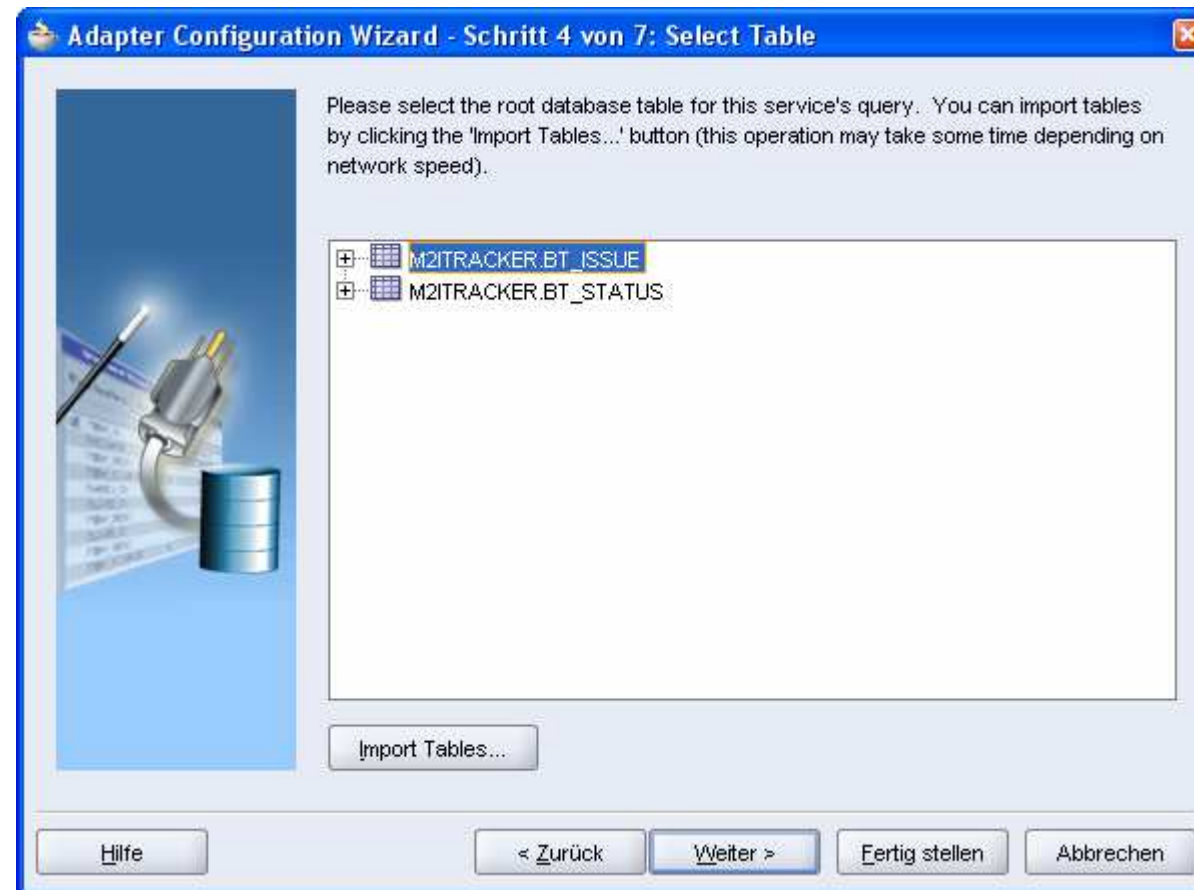


38. BT\_ISSUE und BT\_STATUS Tabellen aus Schema „M2ITRACKER“ werden importiert.

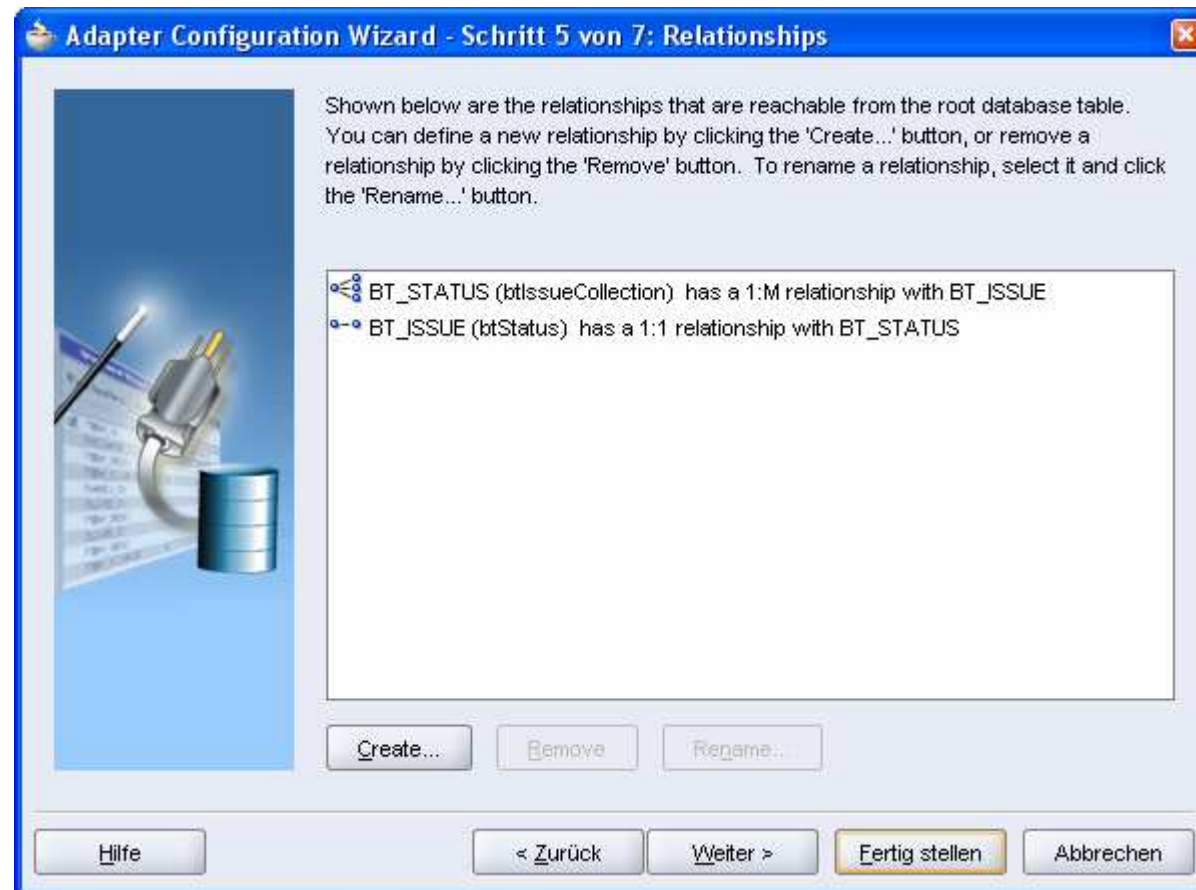




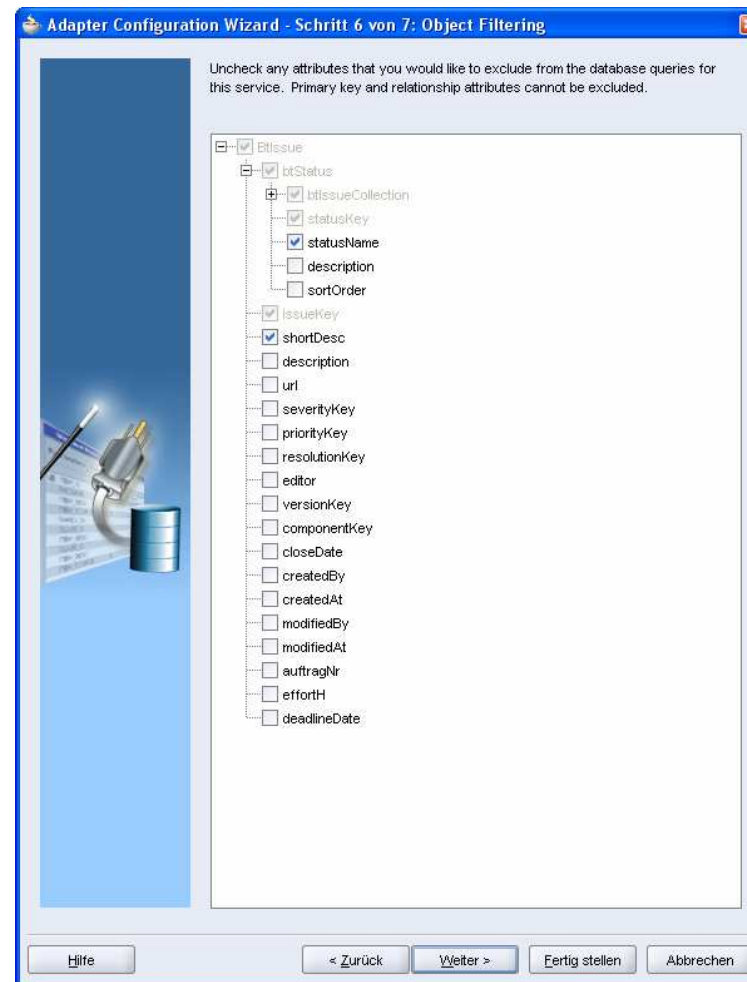
39. Im Schritt 4 vom Adapter Configuration Wizard wird „M2ITRACKER.BT\_ISSUE“ als Master-Tabelle ausgewählt.



40. Beziehungen zwischen den ausgewählten Tabellen werden automatisch erkannt und angezeigt.



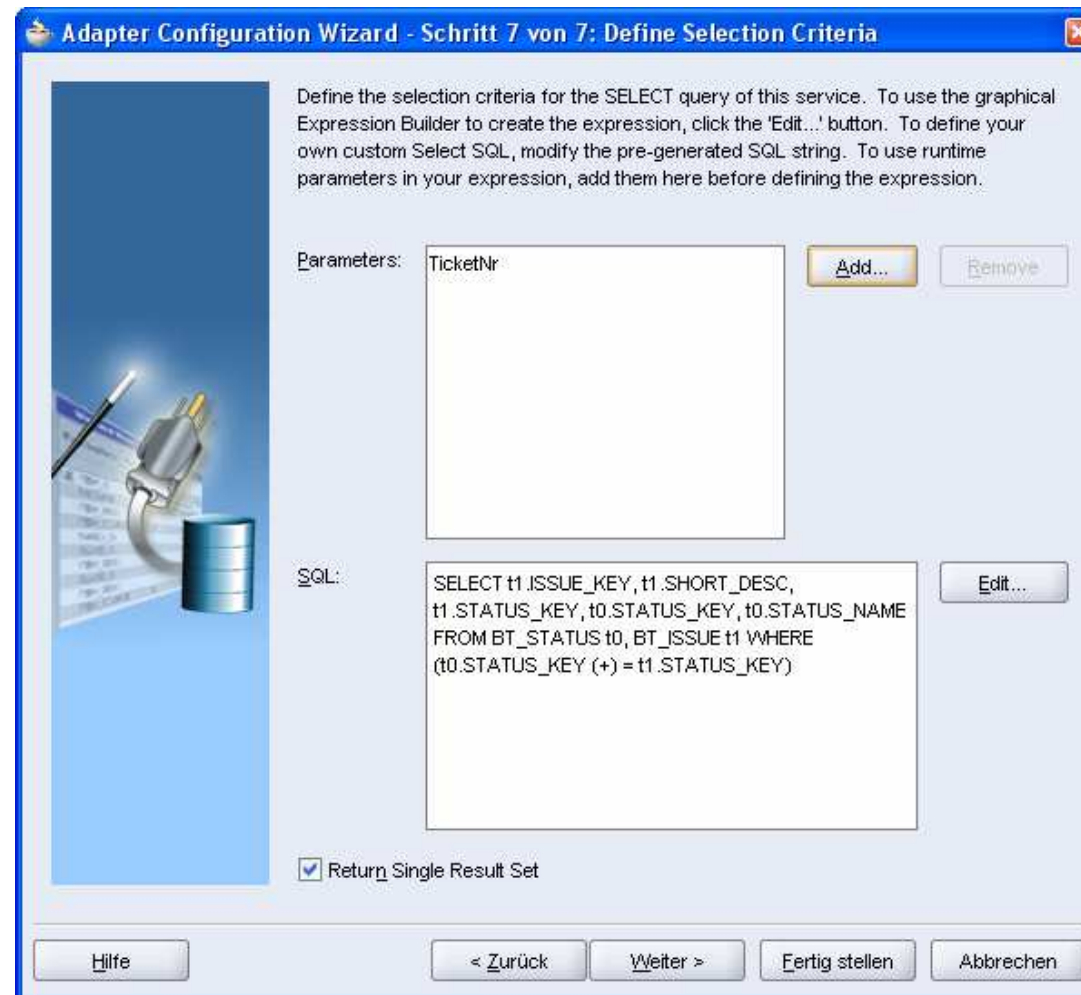
41. Zu selektierende Felder werden ausgewählt.



42. Der SELECT wird anhand des Parameters (Primärschlüsselwertes) ausgeführt, dafür wird der Parameter „TicketNr“ hinzugefügt.

„Return Single Result Set“ Checkbox muss ausgewählt werden, damit die zwei Tabellen „gejoined“ werden.

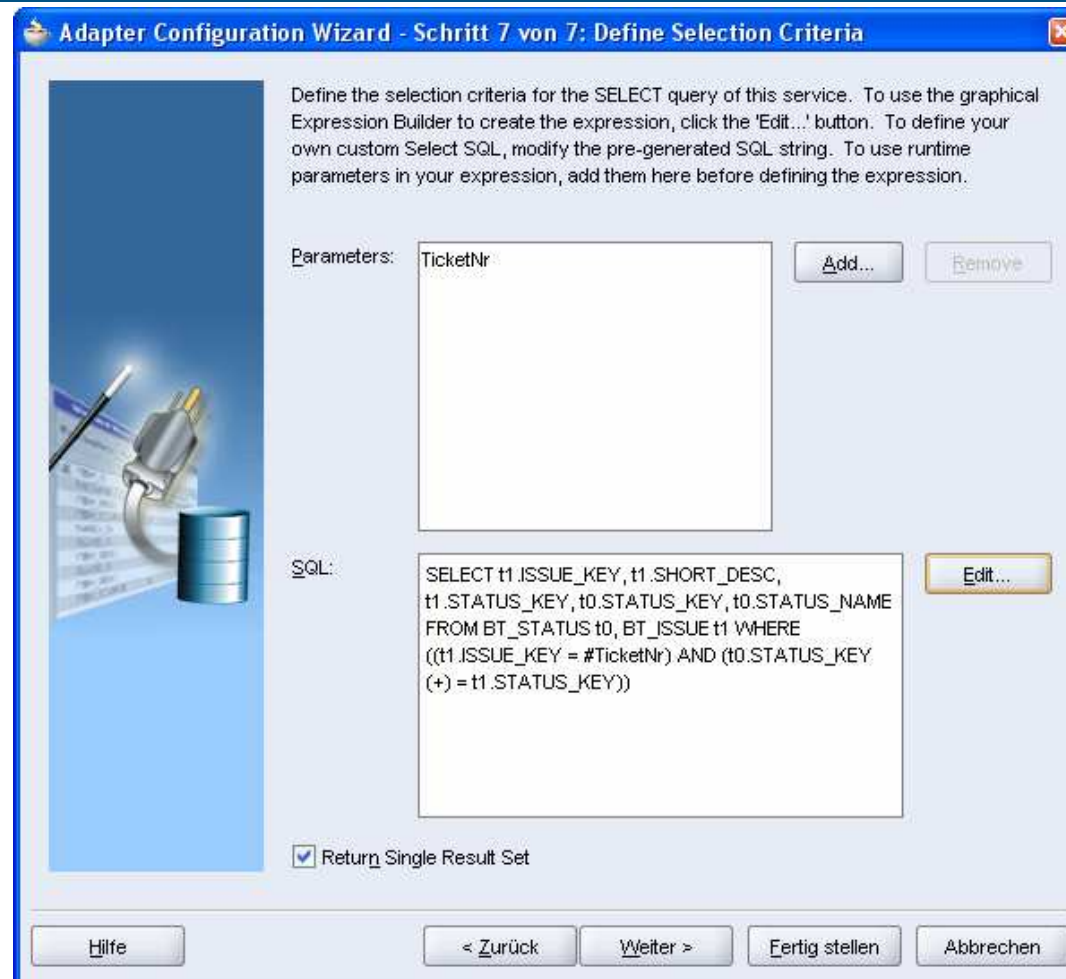
Durch den Klick auf „Edit...“ Button wird der Parameter „TicketNr“ in den SQL-Befehl hinzugefügt.



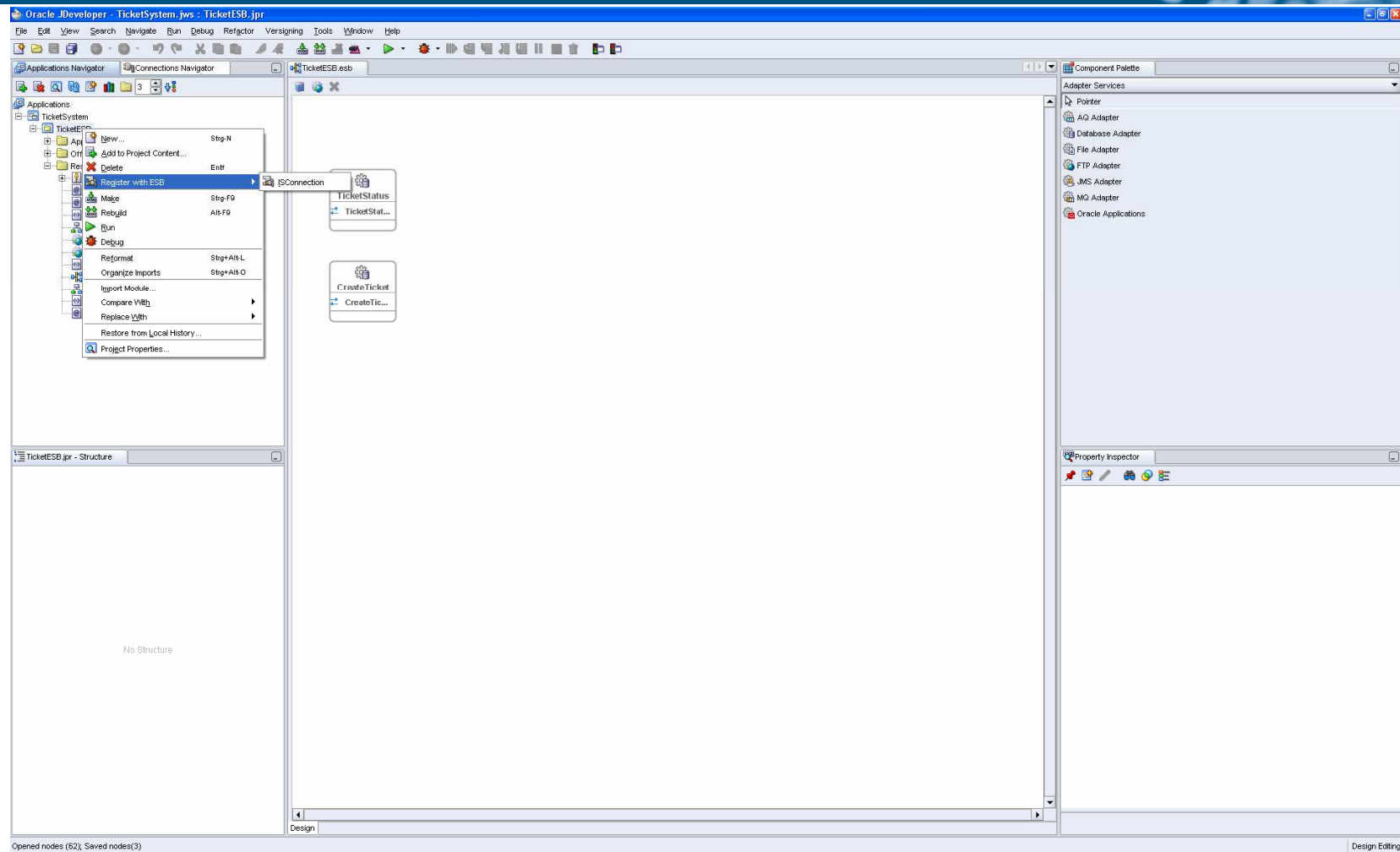
43. Im Expression Builder wird der Parameter „TicketNr“ mit Primärschlüsselspalte „IssueKey“ verknüpft.

The screenshot shows the 'Expression Builder' dialog box with the title 'Create a new TopLink Expression'. The main text area contains the expression '1.issueKey EQUAL TicketNr'. To the right of this area are three buttons: 'Add...', 'Add Nested...', and 'Remove'. Below these buttons is a 'Logical Operator:' dropdown menu. The dialog is divided into two main sections: 'First Argument' and 'Second Argument'. In the 'First Argument' section, the 'Query Key:' is set to 'issueKey' with an 'Edit' button. In the 'Second Argument' section, the 'Operator:' dropdown is set to 'EQUAL'. Below the operator, there are three radio buttons: 'Literal', 'Query Key', and 'Parameter', with 'Parameter' being selected. Below the radio buttons is a dropdown menu showing 'TicketNr'. At the bottom of the dialog are 'Help', 'OK', and 'Cancel' buttons.

44. Der „TicketNr“ Parameter ist jetzt im SQL Befehl.

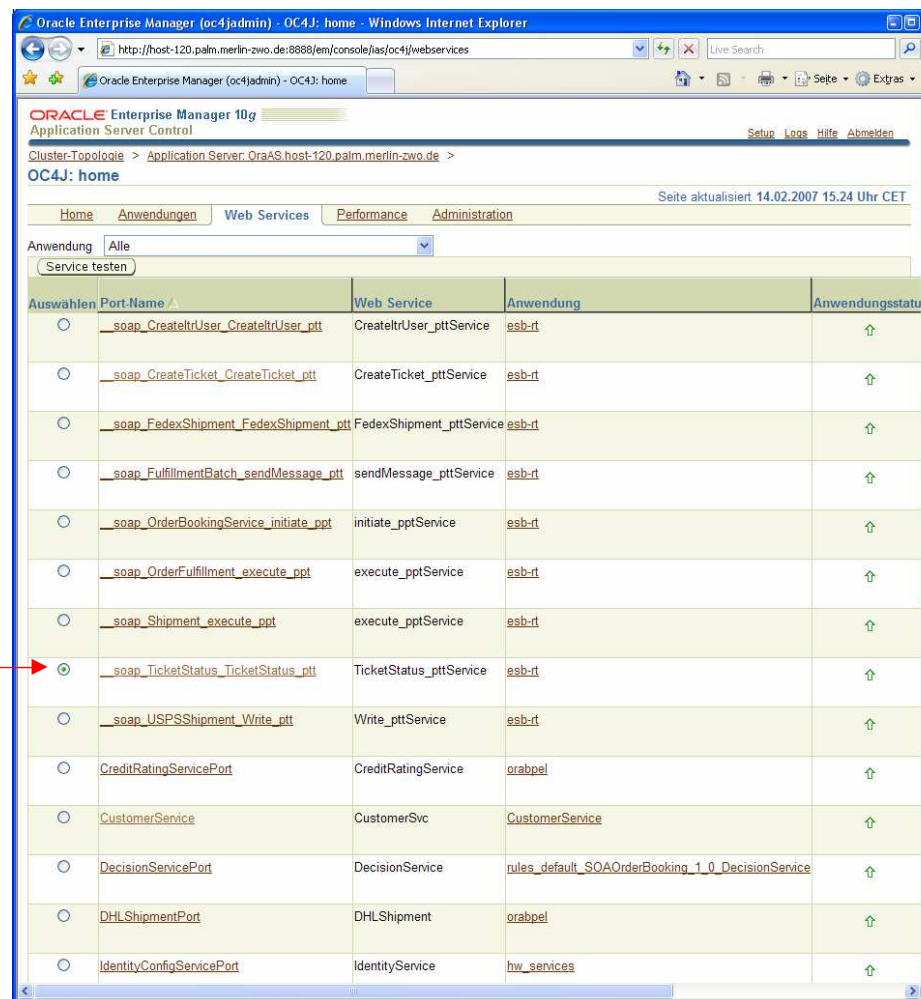


45. Analog dem Schritt 30 wird der ESB Projekt noch mal auf dem ESB Integration Server registriert.





46. Analog den Schritten 32 – 35 kann auch der „TicketStatus“ Database Adapter Service in Application Service Control getestet werden.



Oracle Enterprise Manager (oc4jadmin) - OC4J: home - Windows Internet Explorer

http://host-120.palm.merlin-zwo.de:8888/em/console/ias/oc4j/webservices

Oracle Enterprise Manager (oc4jadmin) - OC4J: home

ORACLE Enterprise Manager 10g

Application Server Control

Cluster-Topologie > Application Server: OraAS.host-120.palm.merlin-zwo.de >

OC4J: home

Seite aktualisiert 14.02.2007 15.24 Uhr CET

Home Anwendungen Web Services Performance Administration

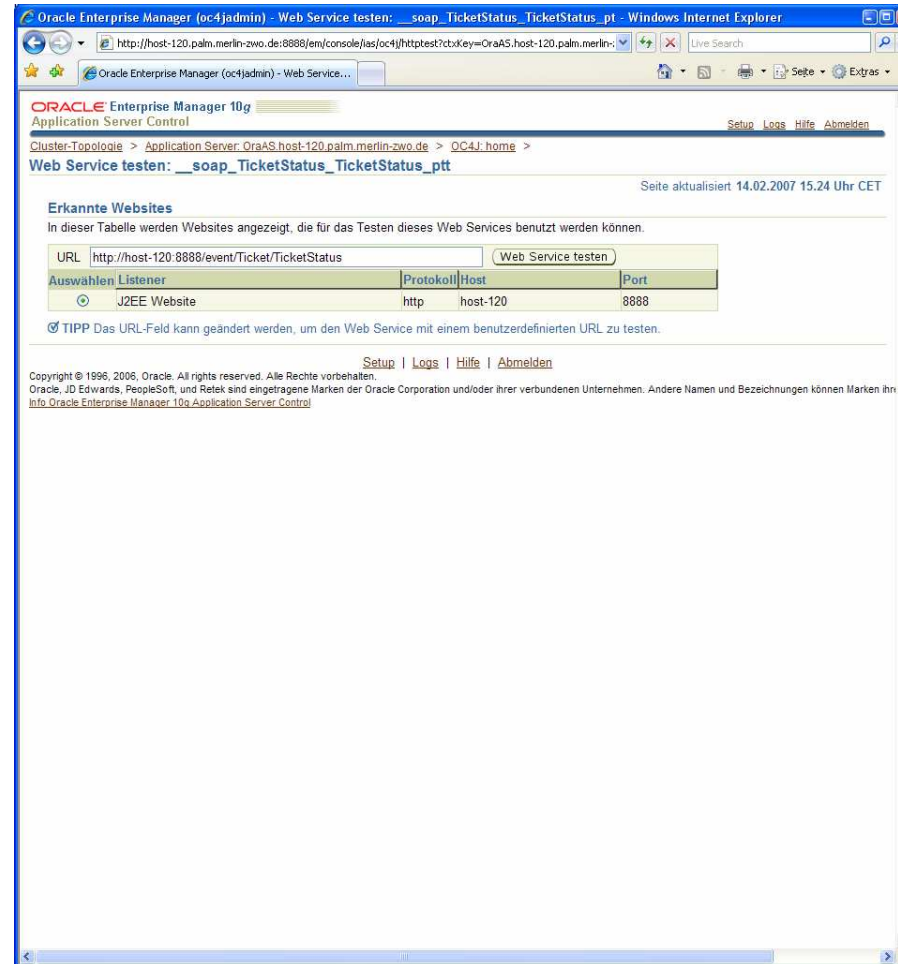
Anwendung: Alle

Service testen

Auswählen	Port-Name	Web Service	Anwendung	Anwendungsstatus
<input type="radio"/>	<a href="#">_soap_CreateUser_CreateUser_ptt</a>	CreateUser_pttService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_CreateTicket_CreateTicket_ptt</a>	CreateTicket_pttService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_FedexShipment_FedexShipment_ptt</a>	FedexShipment_pttService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_FulfillmentBatch_sendMessage_ptt</a>	sendMessage_pttService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_OrderBookingService_initiate_ppt</a>	initiate_pptService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_OrderFulfillment_execute_ppt</a>	execute_pptService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_Shipment_execute_ppt</a>	execute_pptService	esb-rt	↑
<input checked="" type="radio"/>	<a href="#">_soap_TicketStatus_TicketStatus_ptt</a>	TicketStatus_pttService	esb-rt	↑
<input type="radio"/>	<a href="#">_soap_USPSShipment_Write_ptt</a>	Write_pttService	esb-rt	↑
<input type="radio"/>	<a href="#">CreditRatingServicePort</a>	CreditRatingService	orabpel	↑
<input type="radio"/>	<a href="#">CustomerService</a>	CustomerSvc	CustomerService	↑
<input type="radio"/>	<a href="#">DecisionServicePort</a>	DecisionService	rules_default_SQAOOrderBooking_1_0_DecisionService	↑
<input type="radio"/>	<a href="#">DHLShipmentPort</a>	DHLShipment	orabpel	↑
<input type="radio"/>	<a href="#">IdentityConfigServicePort</a>	IdentityService	hw_services	↑

„TicketStatus“  
Adapter Service





Web Service - Windows Internet Explorer

http://host-120:8888/event/Ticket/TicketStatus

Web Service

### TicketStatus\_pttService endpoint

For a formal definition, please review the [Service Description](#).

Download the JavaScript Stub (BETA) for `__soap_TicketStatus_TicketStatus_ptt` and see its [documentation](#).

#### \_\_soap\_TicketStatus\_TicketStatus\_ptt

Vorgang: `TicketStatusSelect_TicketNr` ☒ HTML-Formular ☐ XML-Quelle

☒ Zuverlässiges Messaging ☐ In Header aufnehmen

☒ WS-Sicherheit ☐ In Header aufnehmen

☒ TicketStatusSelect\_TicketNr\_inparameters

TicketNr  xsd:decimal

Hinweis: Der Inhalt der XML-Quellenansicht wird nicht in der Ansicht des HTML-Formulars wiedergegeben

☒ Transport-Info anzeigen

☒ Stresstest ausführen ☐ Enable

Copyright © 2003, 2006, Oracle. All rights reserved.

Testergebnis

View: [Formatted XML](#) | [Raw XML](#)

```
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  <env:Header/>
  <env:Body
    xmlns="http://xmlns.oracle.com/pcbpel/adapter/db/top/TicketStatus">
    <BtIssueCollection
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://xmlns.oracle.com/pcbpel/adapter/db/top/TicketStatus">
      <BtIssue>
        <issueKey>761</issueKey>
        <shortDesc>Test WS</shortDesc>
        <btStatus>
          <statusKey>0</statusKey>
          <statusName>Kein Status</statusName>
        </btStatus>
      </BtIssue>
    </BtIssueCollection>
  </env:Body>
</env:Envelope>
```

Copyright © 2003, 2005, Oracle. All rights reserved.