BOS Webfrontend/EAI –TechnischeDokumentation BANKnow

Version: 5.3

Stand: 09.07.2013

Autor: Markus Brüderl, Jürgen Kamleiter

Status: Konzept

Historie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Autor | Version | Änderung | Kapitel | Datum |
| Markus Brüderl | 3.1 | Überarbeitung im Rahmen R5.1 | Alle | 11.01.13 |
| Ralph Berger | 4.0 | Überarbeitung IIS 7.5 | Alle | 16.01.13 |
| Ralph Berger | 4.1 | Ergänzung Oracle Client | Alle | 17.01.13 |
| Gregor Horochowski | 4.2 | Neuer Parameter in web.config | 3.1 | 15.02.13 |
| Ralph Berger | 4.3 | 32 Bit Erweiterungen installieren | 4.3 | 20.02.13 |
| Ralph Berger | 4.4 | Datenbank Anbindung - Fehlercodes | 8.1 | 27.02.13 |
| Jürgen Kamleiter | 4.5 | Anpassung Simple Service Konfiguration | 3.3 | 30.04.13 |
| Ralph Berger | 4.6 | Ergänzung ZEK Batch Service | 3.3 | 02.05.13 |
| Jürgen Kamleiter | 4.7 | Umstellung WebServices auf 64 bit | 3.0 | 07.05.13 |
| Jürgen Kamleiter | 5.0 | Anpassungen aufgrund Änderungen Release 6 | Alle | 17.05.13 |
| Vladimir  Yantsalovskyy | 5.1 | Überarbeitung IIS 7.5 - Konfiguration HTTPS | 2.2.4.4 | 14.06.13 |
| Gregor Horochowski | 5.2 | Anpassung der log4net-Konfiguration | 6.1 | 18.06.13 |
| Gregor Horochowski | 5.3 | IIS-Konfiguration, Parameter uploadReadAheadSize | 2.2.4.5 | 09.07.13 |

Revisionen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revisor | Version | Kommentar | Kapitel | Datum |
|  |  |  |  |  |

Inhalt

[1 Allgemeines 5](#_Toc361132764)

[1.1 Dokumentenbeschreibung 5](#_Toc361132765)

[1.2 Zielsetzung 5](#_Toc361132766)

[2 Softwarevoraussetzungen 5](#_Toc361132767)

[2.1 Oracle Client installieren 6](#_Toc361132768)

[2.1.1 Management Console konfigurieren 7](#_Toc361132769)

[2.1.2 Zertifikate importieren 9](#_Toc361132770)

[2.1.3 Zertifikat nutzen 10](#_Toc361132771)

[2.2 Windows Komponenten prüfen / installieren 12](#_Toc361132772)

[2.2.1 .NET 4.0 Framework installieren 18](#_Toc361132773)

[2.2.2 Windows Installationstest 19](#_Toc361132774)

[2.2.3 OFiSWebServices Dateien & Freigaben 20](#_Toc361132775)

[2.2.4 IIS 7.5 Konfiguration 21](#_Toc361132776)

[3 Konfigurationseinstellungen web.config 33](#_Toc361132777)

[3.1.1 Aufbau des bindingConfiguration Attributs 34](#_Toc361132778)

[3.1.2 Aufbau des contract Attributs 35](#_Toc361132779)

[3.1.3 Aufbau des name Attributs 35](#_Toc361132780)

[3.2 ApplicationSettings (Dienstbeschreibung) 36](#_Toc361132781)

[3.2.1 Parameter Liste ApplicationSettings 38](#_Toc361132782)

[3.2.2 Weitere Konfigurationen im web.config Abschnitt ApplicationSettings: 39](#_Toc361132783)

[3.3 Bindings Settings 40](#_Toc361132784)

[3.3.1 Parameter Liste Bindings 42](#_Toc361132785)

[3.4 Endpoint Tag (Client-Tag) 44](#_Toc361132786)

[3.4.1 Parameterliste Endpoint Tag (Client-Tag) 45](#_Toc361132787)

[3.5 Service Definition (Service Tag) 46](#_Toc361132788)

[3.5.1 Parameterliste Service Definition 48](#_Toc361132789)

[3.6 WebService Verhalten (Behaviors Tag) 49](#_Toc361132790)

[3.7 DotConnect Dataprovider (DbProviderFactories Tag) 49](#_Toc361132791)

[3.8 2way-SSL-Konfiguration der Service-Hosts für BOS Intern / BOS Extern 50](#_Toc361132792)

[4 Konfigurationseinstellungen OpenLease 51](#_Toc361132793)

[4.1 Konfiguration des Endpoints vom BOS-EAI 51](#_Toc361132794)

[4.1.1 Test der Verbindung 51](#_Toc361132795)

[4.2 Auskunftstypen 52](#_Toc361132796)

[4.3 Pre Shared Keys 54](#_Toc361132797)

[4.4 Tabelle Auskunftcfg 55](#_Toc361132798)

[4.5 Tabelle Auskunfttyp 55](#_Toc361132799)

[5 TestUI 56](#_Toc361132800)

[5.1 Aufruf der TestUI 56](#_Toc361132801)

[5.2 Test-UI Zertifikate 57](#_Toc361132802)

[6 Logging- Log4Net 58](#_Toc361132803)

[6.1 Log-Level Einstellung in der Admin-Box 58](#_Toc361132804)

[6.1.1 Konfiguration Log-File – Appender Tag 59](#_Toc361132805)

[6.1.2 Log-Level – Root Tag 59](#_Toc361132806)

[6.1.3 Schreibrechte im Log-Ordner 59](#_Toc361132807)

[7 Anhang 60](#_Toc361132808)

[7.1 Datenbank Anbindung -Fehlercodes 60](#_Toc361132809)

# Allgemeines

## Dokumentenbeschreibung

Das Dokument beschreibt die IIS 7.5 Konfiguration und die Installation der Webservices unter IIS 7.5.

TEST

## Zielsetzung

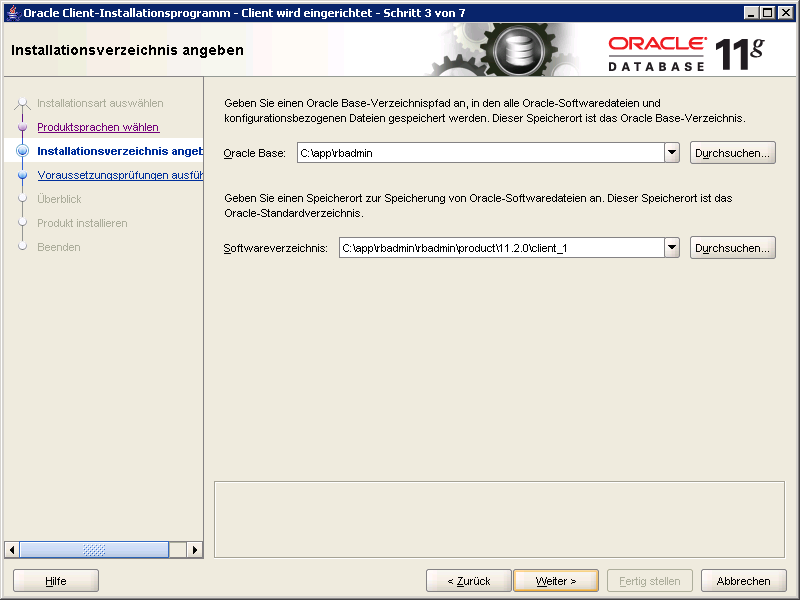
Ziel dieses Dokumentes ist es, die Konfiguration ausgehender Verbindungen der Webservices sowie sämtlicher dazu notwendigen Zertifikate und Konfigurationseinstellungen zu beschreiben.

# Softwarevoraussetzungen

* .NET Framework 4
* SMTP Server mit Fax und SMS Gateway (für Notification Gateway)
* IIS 7.5
* CIC WebServices
* Zertifikate von Drittanbietern für Webservices

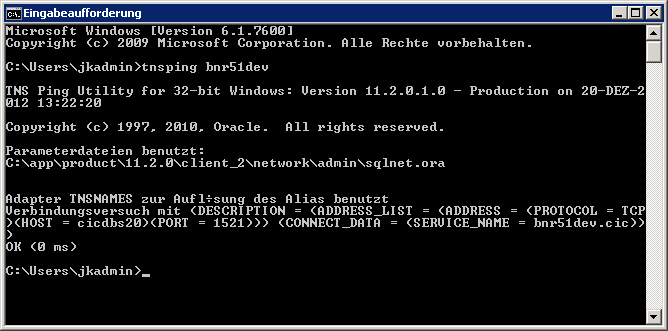
## Oracle Client installieren

Starten Sie das Oracle Client Installationsprogramm als Admin. Wählen Sie Umfang, Sprache und Verzeichnis:



Bei einer dualen Installation beginnen Sie mit dem 64 Bit Client. Denken Sie daran TNS Names sowohl für den 32 bit Client als auch für den 64 Bit Client zu erstellen.

Öffnen Sie die DOS Box und geben Sie folgendes ein: tnsping TNSNAME



Der installierte Oracle Client und die TNSNAME Parameter werden angezeigt. Client Zertifikate für IIS (u.a. Deltavista)

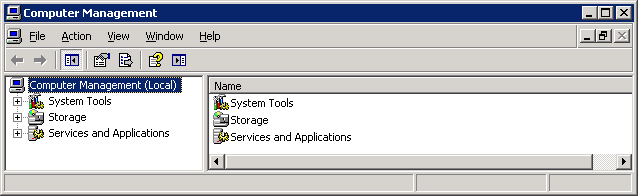
Für ausgehende https Abfragen benötigen die WebServices Client Zertifikate von Drittanbietern. Wir nutzen in unserer Testinstallation die Windows Option „Zertifikate für Computer“. Windows Server 2008 verwaltet Zertifikate für

* Benutzer Konten
* Computer
* Services

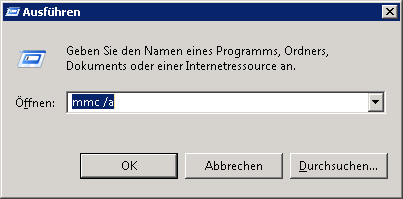
### Management Console konfigurieren

Der IIS Server nutzt für die WebServices ein eigenes IIS\_User Konto. Von den WebServices ausgehende https Verbindungen greifen auf einen eigenen „Certifikate Store“ zu. Der Import von Zertifikaten mit Hilfe des Import Wizard (Rechtsklick Zertifikatsdatei installiert ein Zertifikat für das aktuelle Benutzer Konto (z.B. Administrator)).

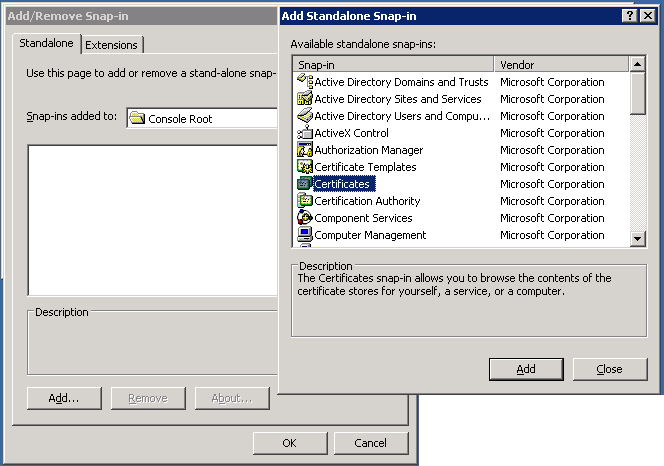
Um Zertifikate für alle Konten (incl. IIS\_User Konto) zu erstellen ist ein Computer Zertifikat über die Computer Management Console ( Start -> Administrative Tools -> Computer Management) zu installieren. Eine Standard Windows Server Installation (vgl. Abb. Unten) muss mit dem Snap In „Certificates“ ergänzt werden.



Gehen Sie bitte wie folgt vor: Start -> Run

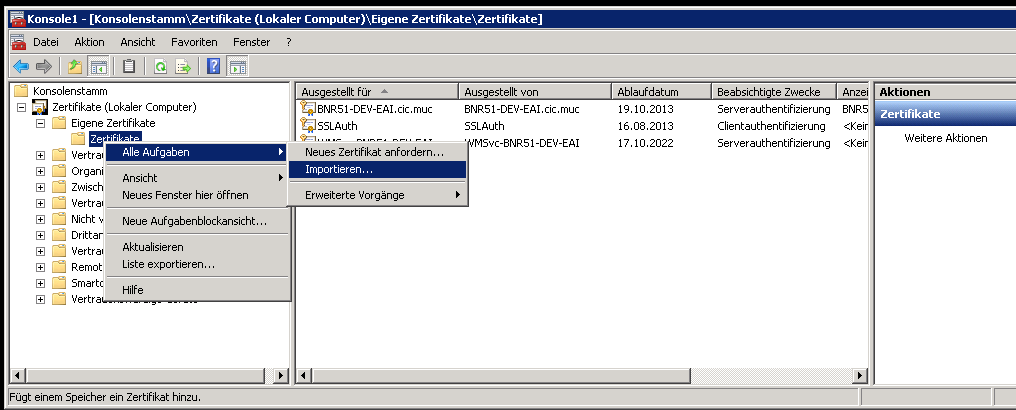


In Fenster Konsole wählen Sie im Datei Menü -> Add/Remove Snap-in -> Add. Im Dialog Snap-In wählen Sie Certificates dann Computer account und klicken Sie Next.



Um das Snap In für den lokalen Computer zu nutzen, klicken Sie auf local Computer und beenden Sie den Dialog mit Finish. Beenden Sie den Snap In Dialog. Die Management Console enthält nun den Knoten Certificates.

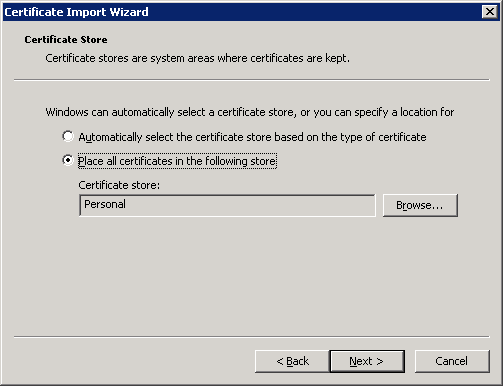
### Zertifikate importieren



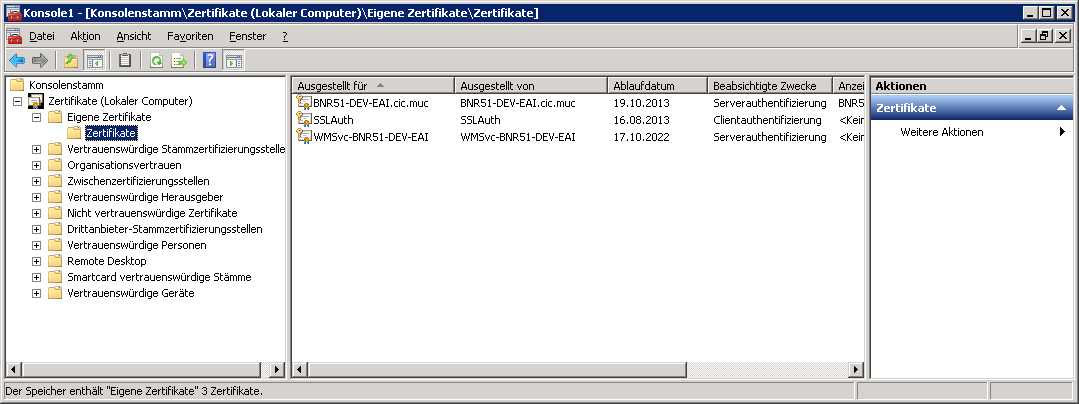
Öffnen Sie das Options Menü für Certificates (local Computer)/Personal/Certificates (Rechtsklick).

Wählen Sie unter All Tasks die Option Import. Der Import Wizard startet.

Wählen Sie als Certficate Store:



Das Zertifikat wird importiert und in der MMC Console angezeigt:



### Zertifikat nutzen

Sie müssen nun dem Netzwerk Service oder IIS\_User den Lesezugriff auf das importierte Zertifikat ermöglichen. Wir benutzen das Tool *FindPrivateKey.exe* um den Dateinamen zu ermitteln unter dem das Zertifikat gespeichert wurde und *icacls.exe* um die Dateifreigabe zu setzen.

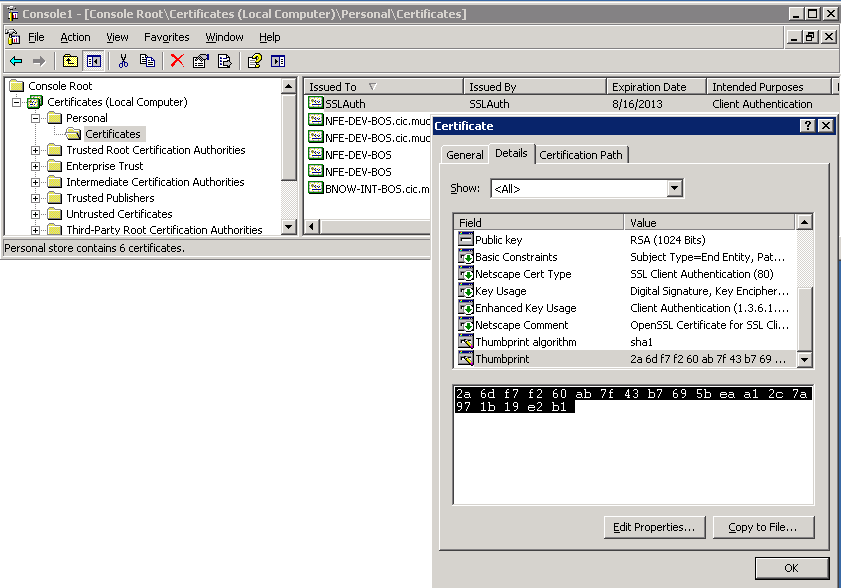
C:\findprivatekey My LocalMachine -t "2a 6d f7 f2 60 ab 7f 43 b7 69 5b ea a1 2c 7a 97 1b 19 e2 b1" -a

C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA\Machin

eKeys\6578b7610e0831774c00a4935f64f757\_1acfb165-e7f5-4ec2-a728-4cb14adf3041

C:\temp>

Der grün markierte Text kann unverändert übernommen werden. Der rot markierte Text entspricht dem Thumbprint unseres importierten Zertifikates. Diesen ermitteln Sie durch aufrufen des Certficate Context Menüs. Klicken Sie dann auf den Menüpunkt Open und im Dialogfenster Certificate auf den Reiter Details. Und übernehmen Sie die Werte, die unter Thumbprint stehen, in die Kommandozeile (s.o.)

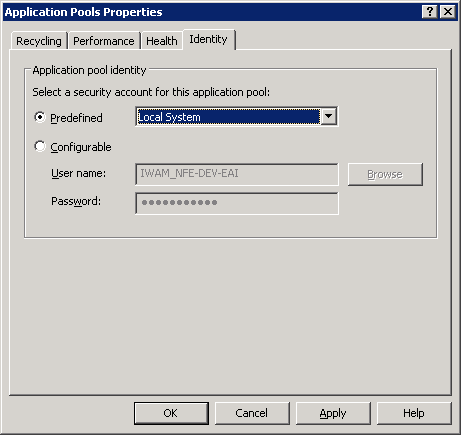


FindPrivateKey liefert als Ergebnis den Dateinamen unter dem die Zertifikatsinfos abgespeichert wurden (blauer Text). Rufen Sie in der DOS BOX das Programm icacls.exe mit folgenden Parametern auf:

icacls C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA\ MachineKeys\6578b7610e0831774c00a4935f64f757\_1acfb165-e7f5-4ec2-a728-4cb14adf3041 /grant IHRNAME:r

Ersetzen Sie IHRNAME durch EVERYONE oder ein anderes Benutzerkonto.

Klicken Sie im IIS Manager auf Application Pools mit der rechten Maustaste und wählen im Kontext Menü Properties. Auf dem Reiter Identity ändern Sie den security account für Ihre WebServices wie folgt:

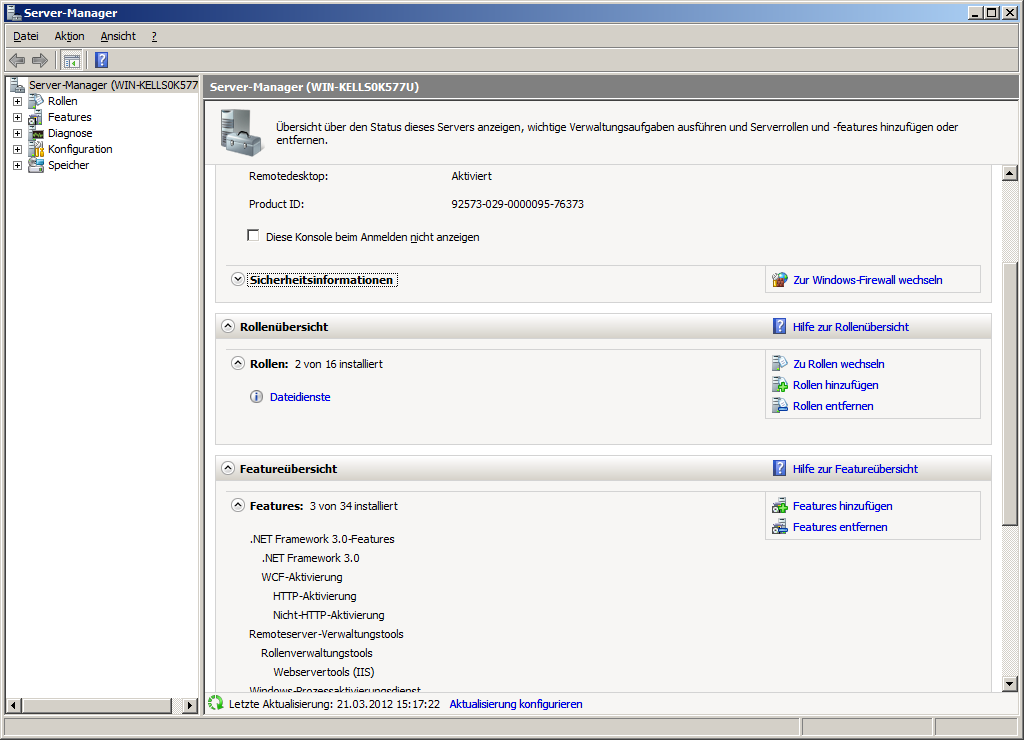


Achten Sie darauf, dass der hier gewählte Benutzer schreibrechte auf den Log-Ordner besitzt.

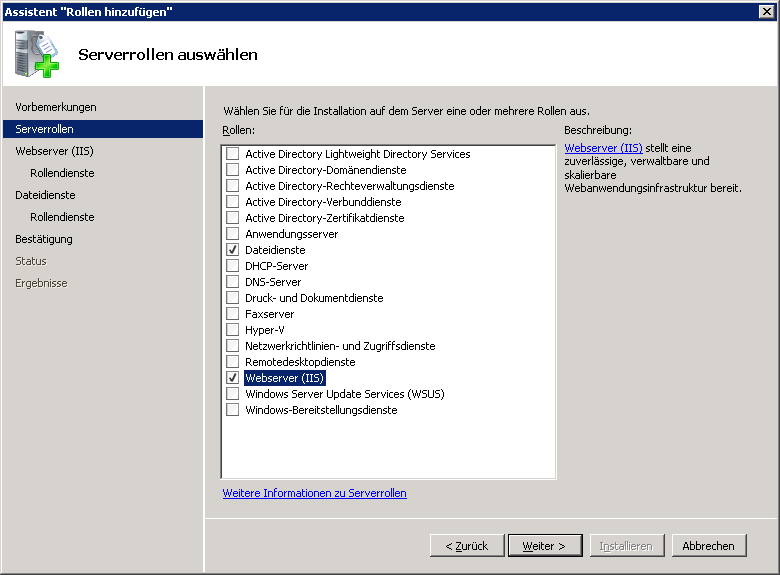
Hierzu ist in der Konfigurationsdatei web.config im Abschnitt behaviors der Eintrag *storeLocation auf LocalMachine* zu setzten, bzw. dies zu kontrollieren.

## Windows Komponenten prüfen / installieren

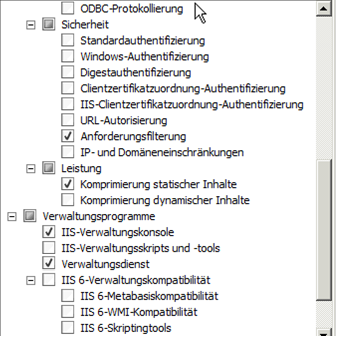
Die OFiSWebServices nutzen den Devart Client oder den Oracle Client 64 bit, den IIS 7.5 und die WebService Erweiterung ASP.Net 4.0 zum Ausführen der Web Services. Prüfen Sie die installierten Server Rollen und fügen Sie bei Bedarf die Rolle IIS hinzu. Starten Sie den Server Manager und klicken Sie auf Rollen hinzufügen:



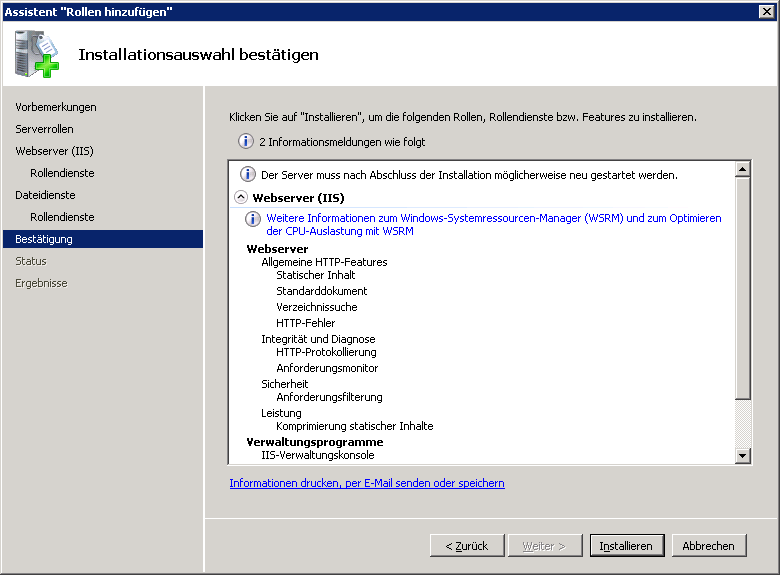
Wählen Sie in folgendem Dialog die Rolle WebServer IIS und klicken Sie auf Weiter:



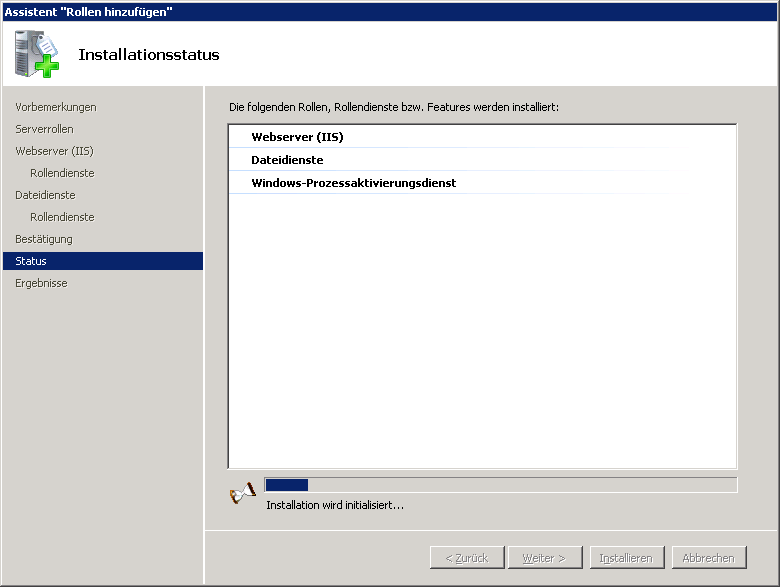
Wählen Sie die zu installierenden Komponenten aus. Wählen Sie zusätzlich zu den vorgeschlagenen Komponenten die Option Verwaltungsdienst aus:



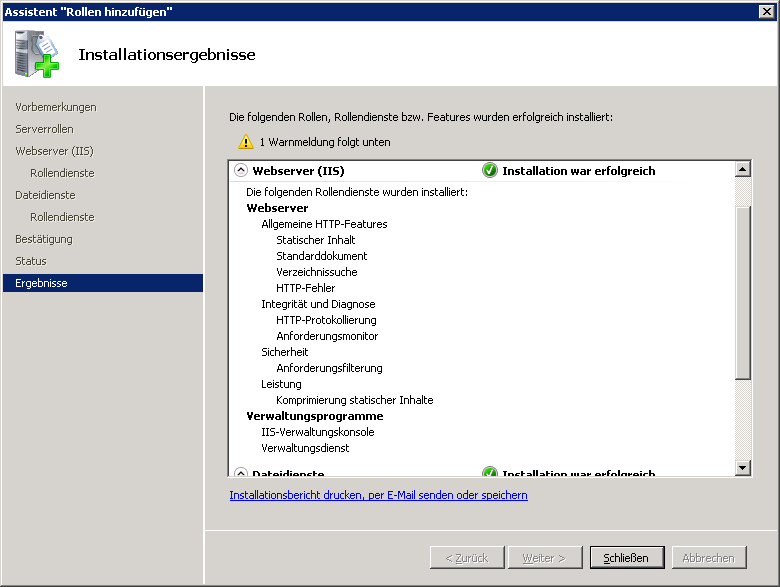
Bestätigen Sie die Installationsauswahl durch Klick auf den Installieren Button:



Die neuen Rollen IIS und Dateidienste werden installiert:

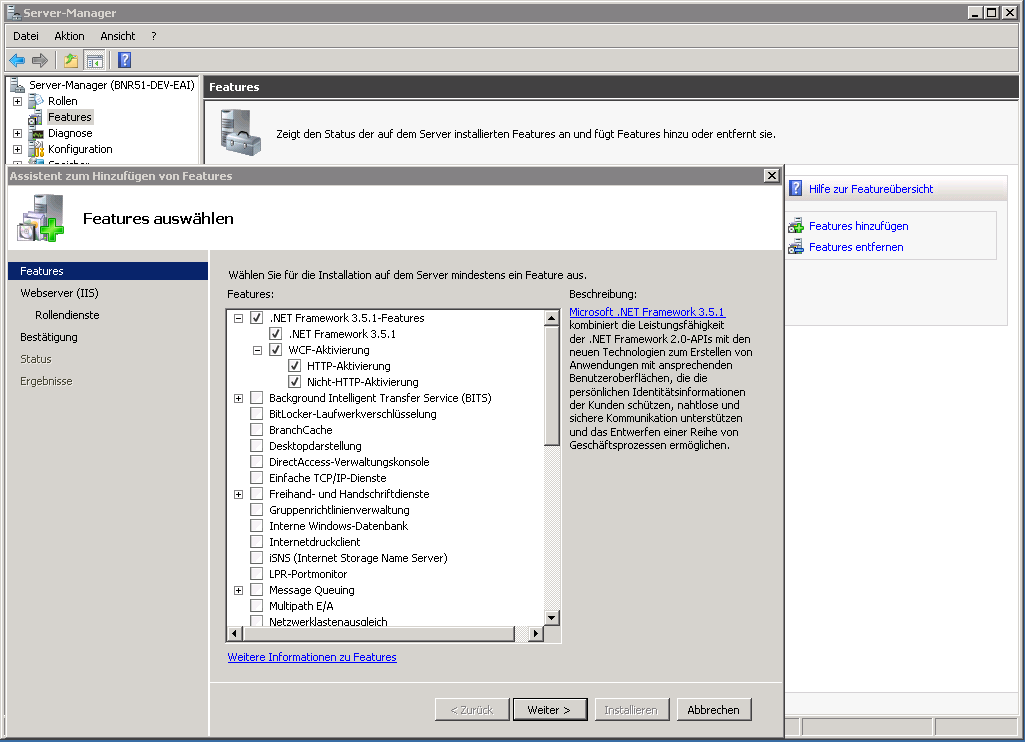


Der Installationsverlauf wird zusammengefasst:

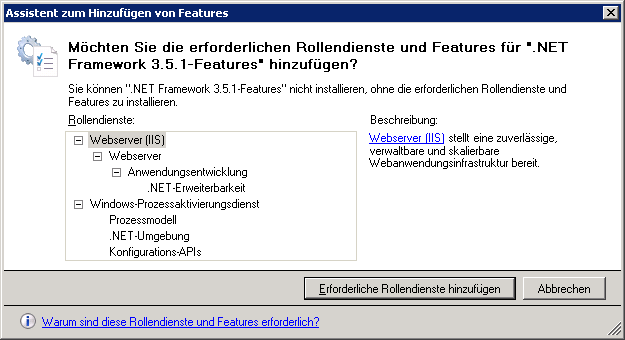


Durch Klick auf den Schließen Butten gelangen sie zurück zum Server Manager.

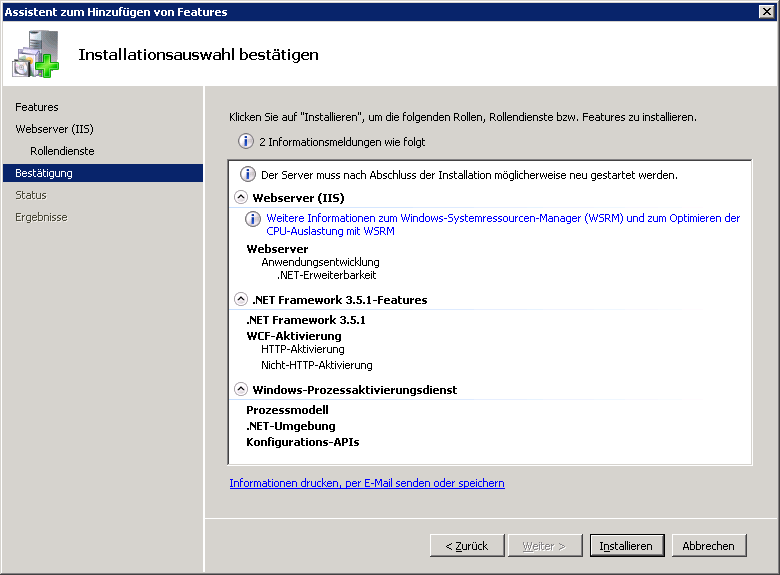
Wählen Sie im Server Manager (vgl. Abb. o.) die Option Features hinzufügen. Klicken Sie im Assistent auf die Option .Net Framework und WCF-Aktivierung. Klicken Sie dann auf weiter um die ausgewählten Komponenten installieren zu lassen.



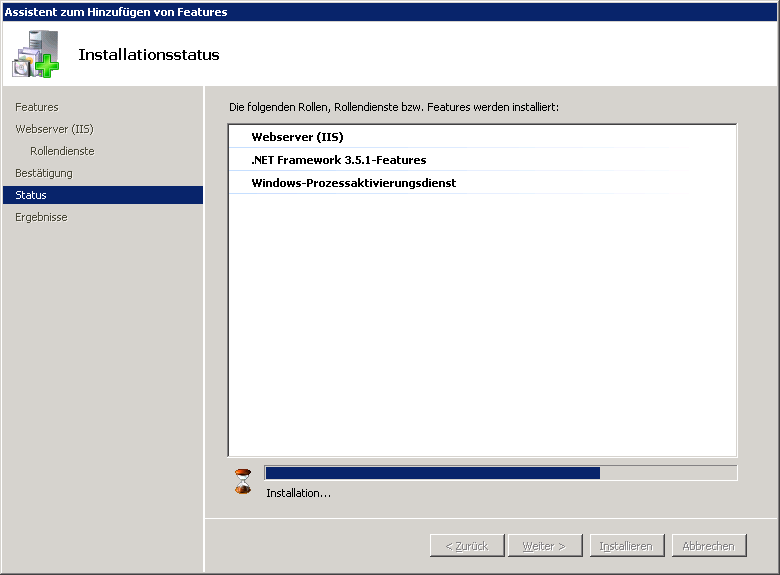
Bestätigen Sie die Installation zusätzlicher Rollendienste durch klick auf den Button:



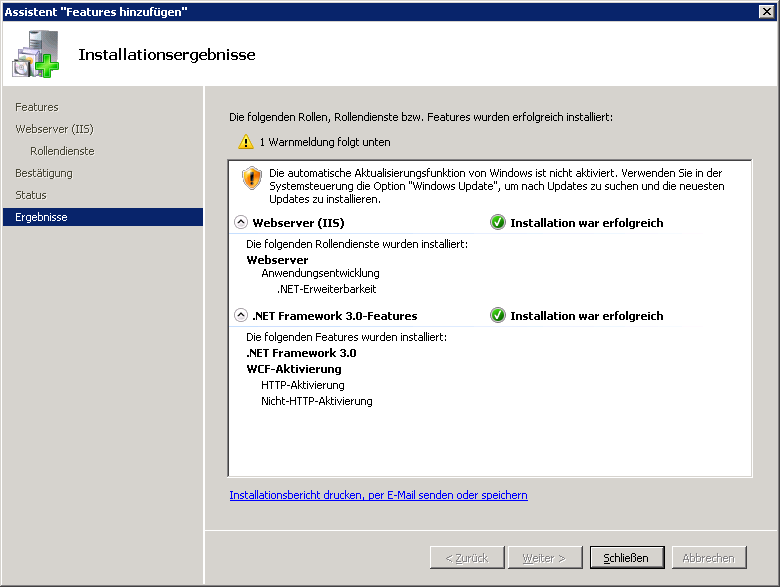
Klicken Sie auf mehrmals auf Weiter bis der Assistent Features hinzufügen Sie auffordert die Installationsauswahl zu bestätigen und dann auf Installieren:



Der Installationsverlauf wird angezeigt:



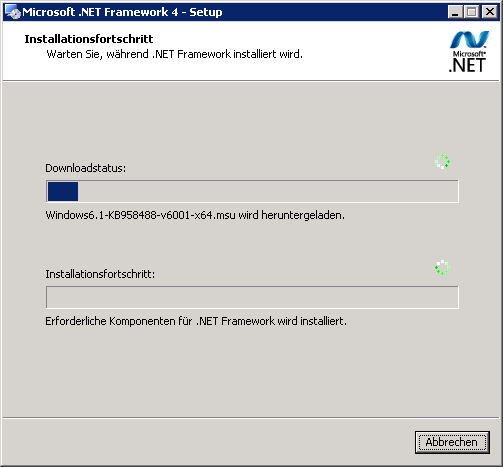
Das Ergebnis wird angezeigt:



### .NET 4.0 Framework installieren

Stellen sie sicher dass sie mit dem Internet verbunden sind. Starten Sie das .Net 4.0 Installationsprogramm 

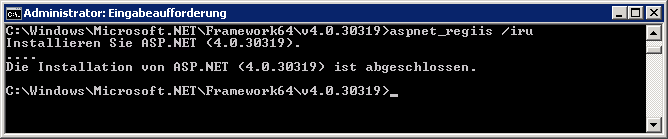
Download und Installation der Komponenten laufen ohne weitere Benutzerangaben durch:

Im Anschluss muss der Server neu gestartet werden.

Registrieren Sie nun die ASP.Net 4.0 Erweiterungen mit dem Programm aspnet\_regiis.exe im Verzeichnis (64bit Version) %SystemRoot%\Microsoft.NET\Framework64\v4.0.30319.

Die Directory Versionsnummer kann abweichen – Stand April 2011. Öffnen Sie eine CMD-Box als Administrator und wechseln Sie in das o.g. Verzeichnis:

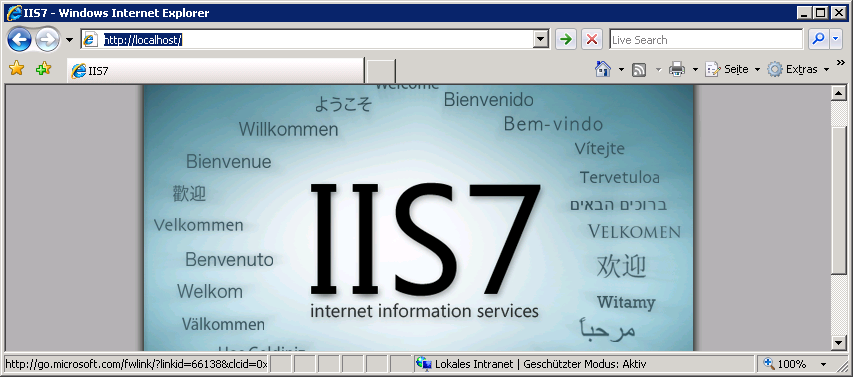


C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319>aspnet\_regiis.exe /iru

Wenn das Verzeichnis nicht existiert sollten Sie das .Net FW 4.0 per **dotNetFx40\_Full\_setup.exe** erneut installieren. Sie finden das Programm auf der Microsoft Homepage. Die Installation der Windows Komponenten ist damit abgeschlossen.

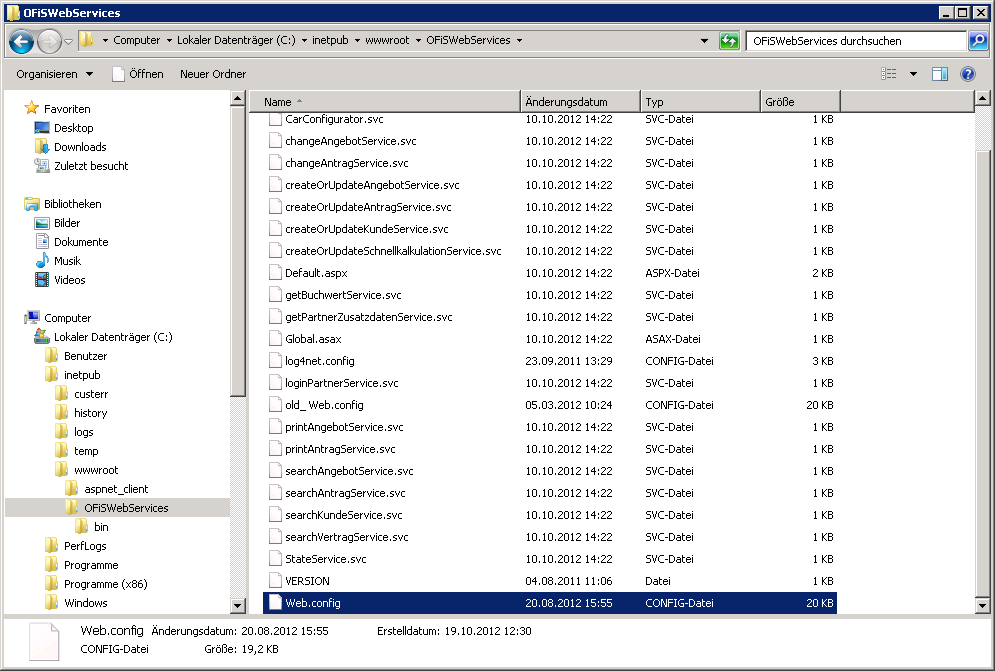
### Windows Installationstest

Unser Server kann auf das Internet zugreifen, befindet sich in einer Domäne und der IIS ist mit den benötigten Features installiert. Prüfen Sie die erfolgreiche Installation des IIS per Aufruf der Seite <http://localhost/> in Ihrem Browser:

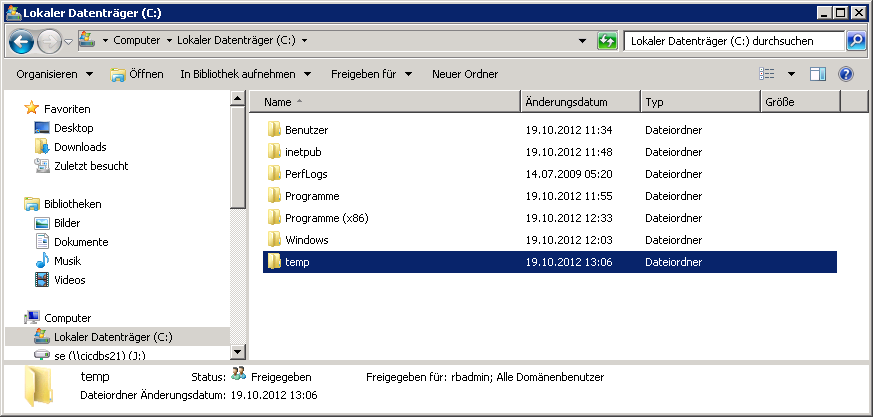


### OFiSWebServices Dateien & Freigaben

Kopieren Sie das Verzeichnis OFisWebServices nach **\Inetpub\wwwroot:**

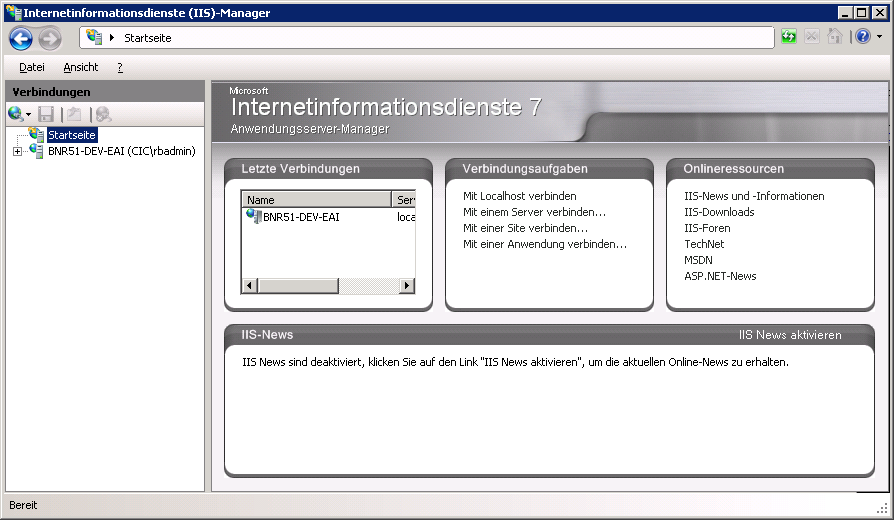


Überprüfen Sie die in der web.config Datei eingetragene Datenbankverbindung. Erstellen Sie ein Temp Verzeichnis unter C:\temp auf das alle Domänenbenutzer zugreifen können:

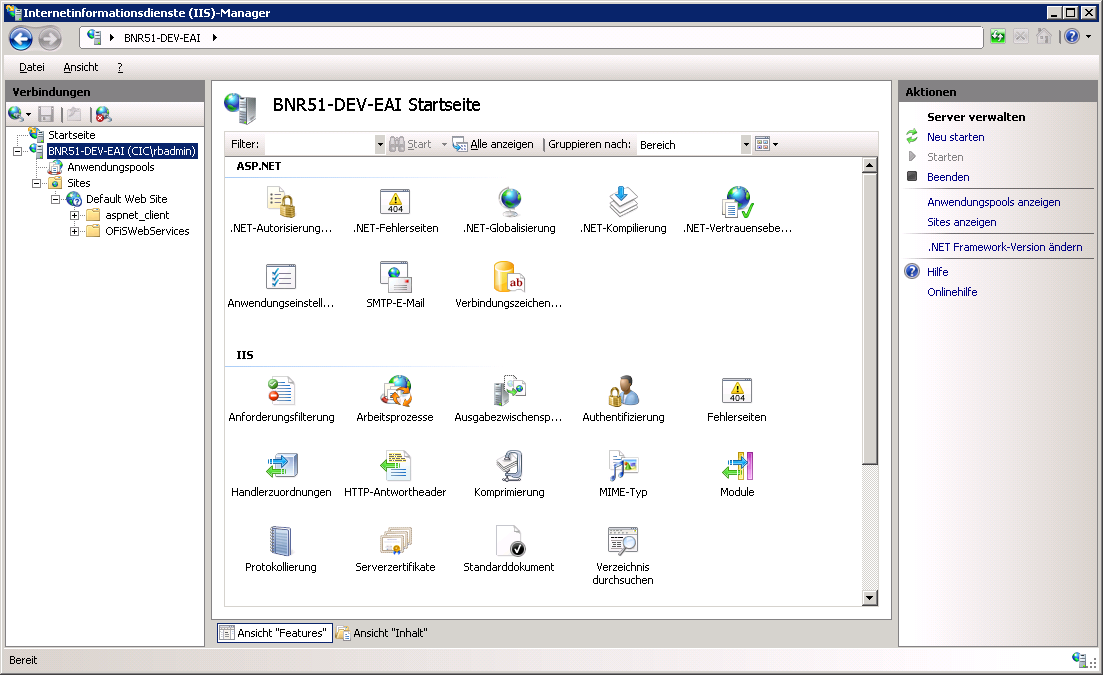


### IIS 7.5 Konfiguration

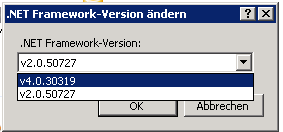
Starten Sie den IIS Manager per Start -> Verwaltung -> IIS Manager. Folgendes Fenster wird geöffnet:



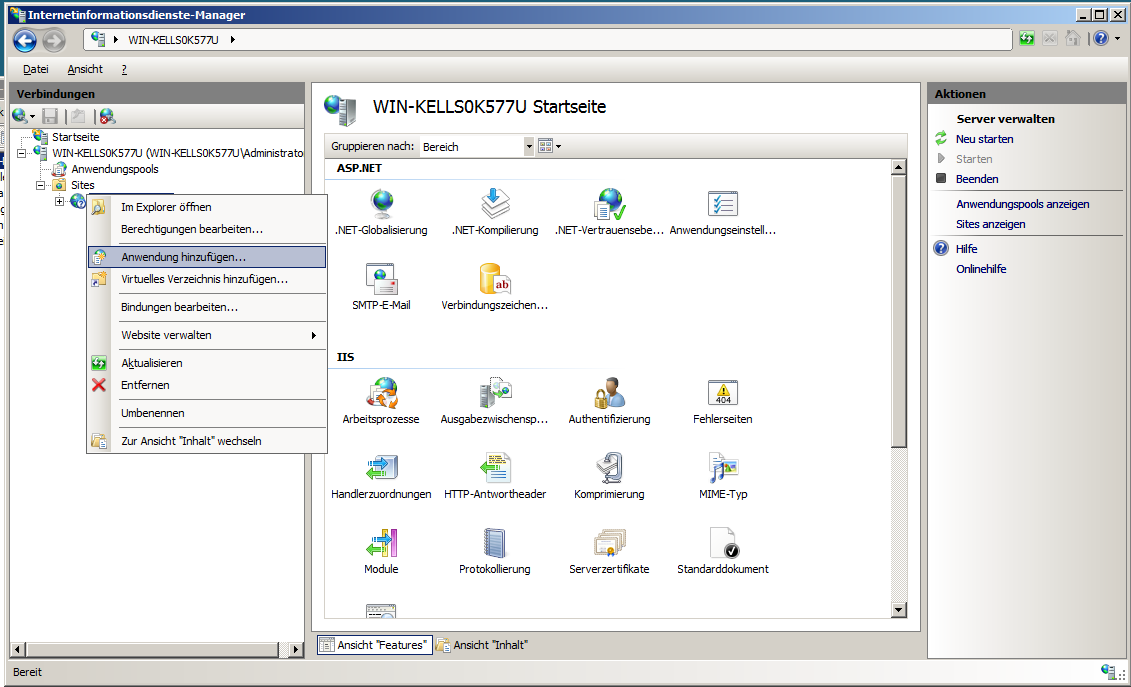
Klicken Sie unter Verbindungen auf den Knoten Ihres Servers. Folgendes Fenster wird geöffnet:



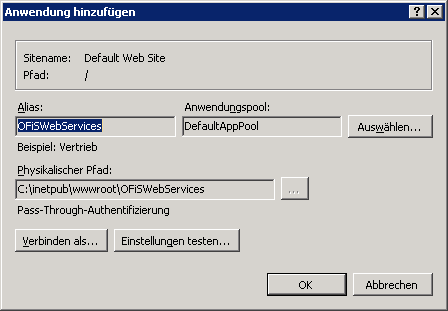
Klicken Sie unter Aktionen auf „.NET Framework-Version ändern“ und ändern Sie .Net Framework Version global von 2 auf 4:



Unter Verbindungen, per Rechtsklick auf den Eintrag DefaultWebSite klicken um das Kontext Menü zu öffnen.



Wählen Sie dann die Option in Anwendung konvertieren. Folgender Dialog wird geöffnet:

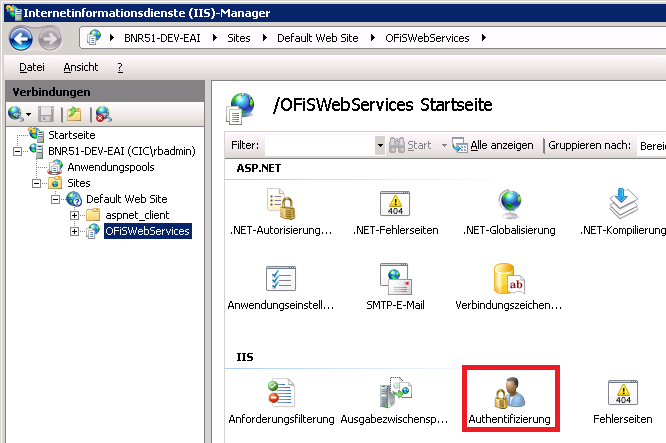


Ändern Sie den Anwendungspool durch Klick auf den Auswählen Button wie folgt:

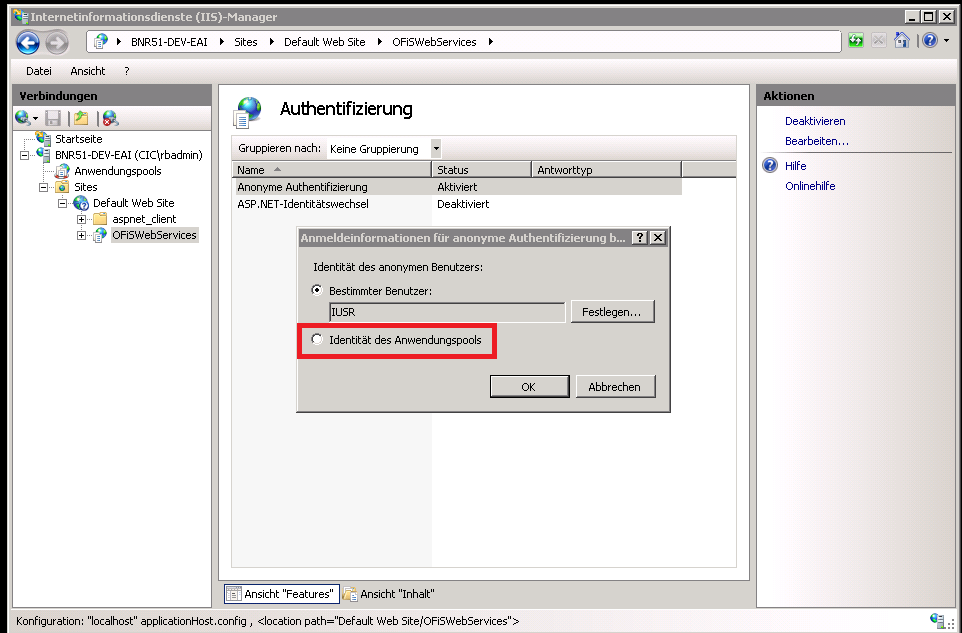


Bestätigen Sie mit OK. Sie gelangen zurück zum Anwendung hinzufügen Dialog den Sie bitte ebenfalls mit OK bestätigen. Sie gelangen zurück zur Startseite. Ihre WebServices nutzen den ASP.NET v4.0 Anwendungspool.

Der IIS Manager zeigt die neue Anwendung unterhalb des Knotens Default Web Site:

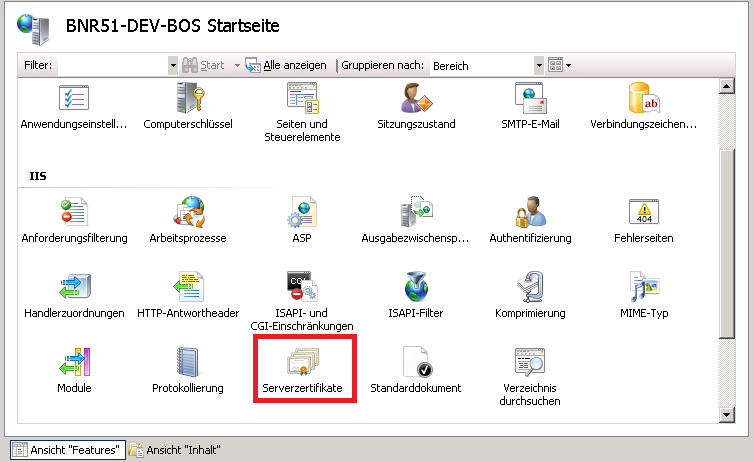


Klicken Sie auf Authentifizierung. Wählen Sie den Eintrag anonyme Authentifizierung und klicken Sie unter Aktionen auf Bearbeiten. Ändern Sie im Dialog Anmeldeinformationen die Auswahl von „bestimmter Benutzer“ zu „Identität des Anwendungspools“:

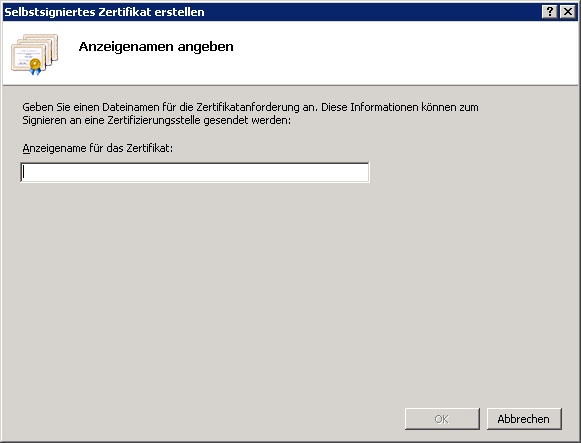


#### Erstellen eigener Zertifikate

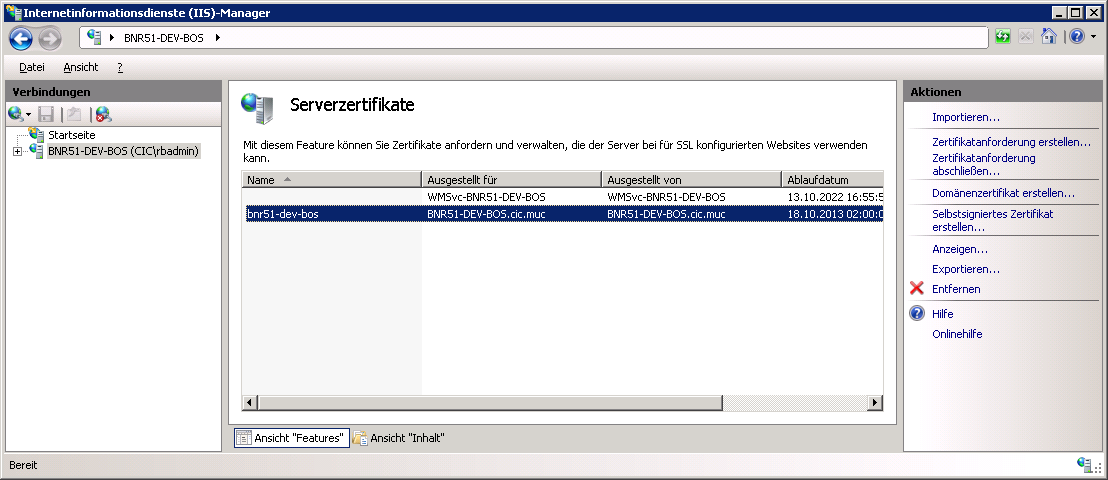
Die WebServices benötigen eine https Verbindung. Klicken Sie auf den Root Knoten Ihres IIS Servers. Im Dialog Startseite wird die Zertifikatsverwaltung durch Klick auf Serverzertifikate gestartet:



Wählen Sie im Fenster Serverzertifikate unter Aktionen den Punkt „Selbstsigniertes Zertifikat erstellen.. “. Geben sie den Namen im folgenden Dialog an:

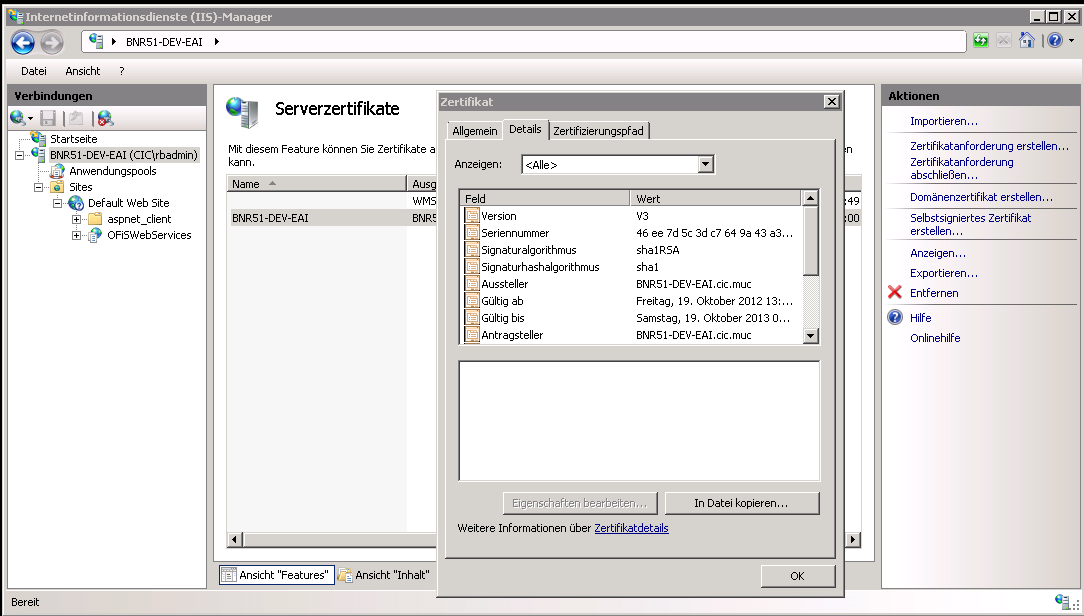


Das neu erstellte Zertifikat wird angezeigt:

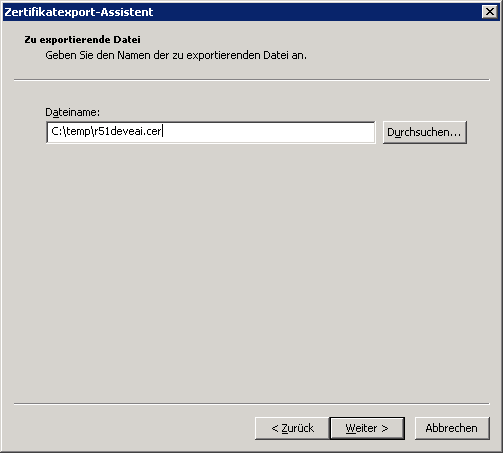


Exportieren sie das Zertifikat in eine \*.cer Datei per Doppelklick auf den Listeneintrag.

Folgender Dialog wird geöffnet:



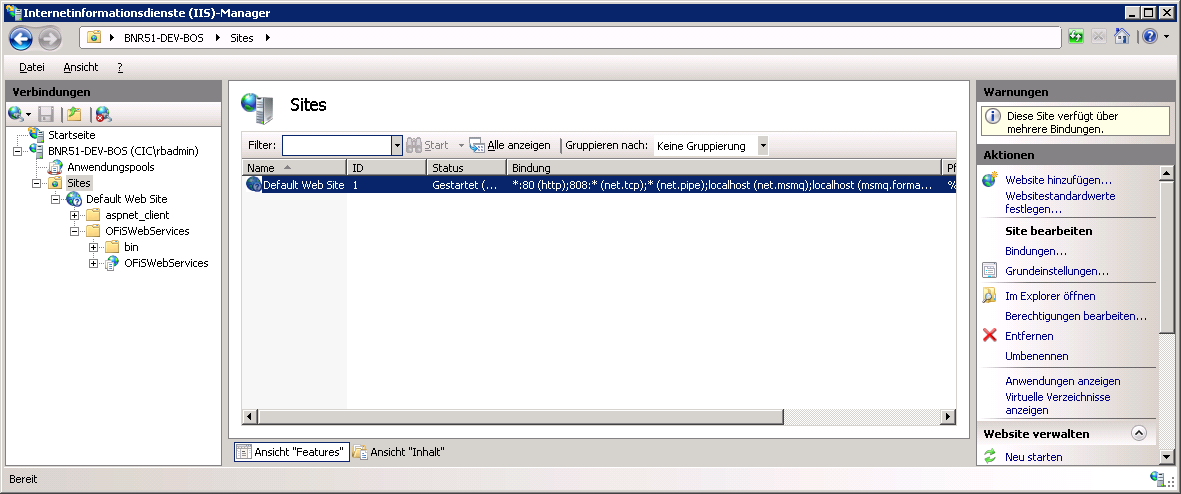
Klicken Sie im Dialog Zertifikat auf den Reiter Details und dann auf den Button „in Datei kopieren“. Der Zertfikatexport-Assistent wird gestartet. Übernehmen sie alle Default- Einstellungen und wählen Sie zuletzt den Namen der Zertifikats-Datei aus:

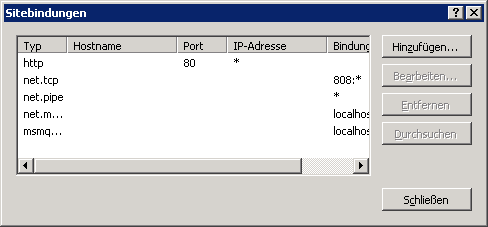
Beenden Sie den Assistenten durch Klick auf „Fertig stellen“. Kopieren Sie das erstellte Zertifikat auf die Client Rechner.

#### Fügen Sie eine https Bindung hinzu

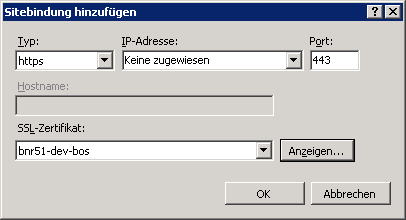
Klicken sie auf Knoten Sites. Wählen Sie Ihre WebSite. Folgendes Fenster wird geöffnet:



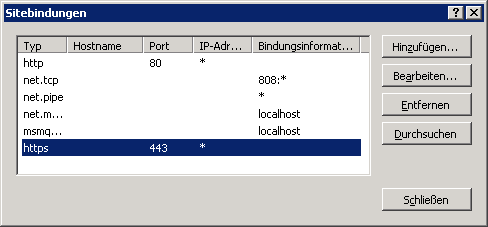
Klicken Sie unter Aktionen auf Bindungen und im folgenden Dialog dann auf den Button hinzufügen:



Tragen sie als Typ https ein, weisen sie keine Ip Adresse zu und benutzen sie den Standard Port 443. Wählen Sie Ihr neu erstelltes SSL-Zertifikat :



Die neue Bindung ( https ) wird im Dialog Sitebindungen angezeigt:



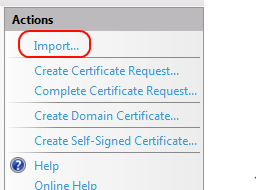
#### 2 way SSL Verbindungen

Die Konfigurationsoberfläche in Windows2008 für IIS7 wurde überarbeitet, so daß die Konfigurationsmasken hier an anderer Stelle zu finden sind.

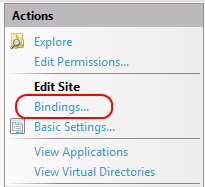
1. In IIS Manager (inetmgr.exe) Server Node Ebene die Server Certificates öffnen:



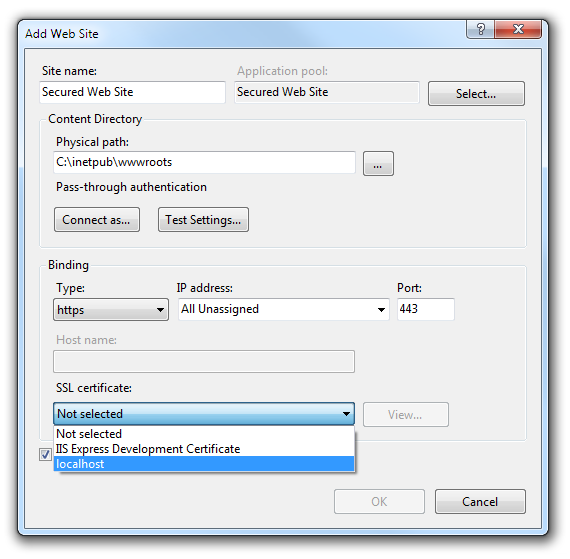
1. Aus Actions den Punkt Import… auswählen und das Server Zertifikat importieren.



1. Das Binding der per SSL zu schützenden Website editieren:



1. Den Bindingtype https und das zugehörige Zertifikat auswählen:

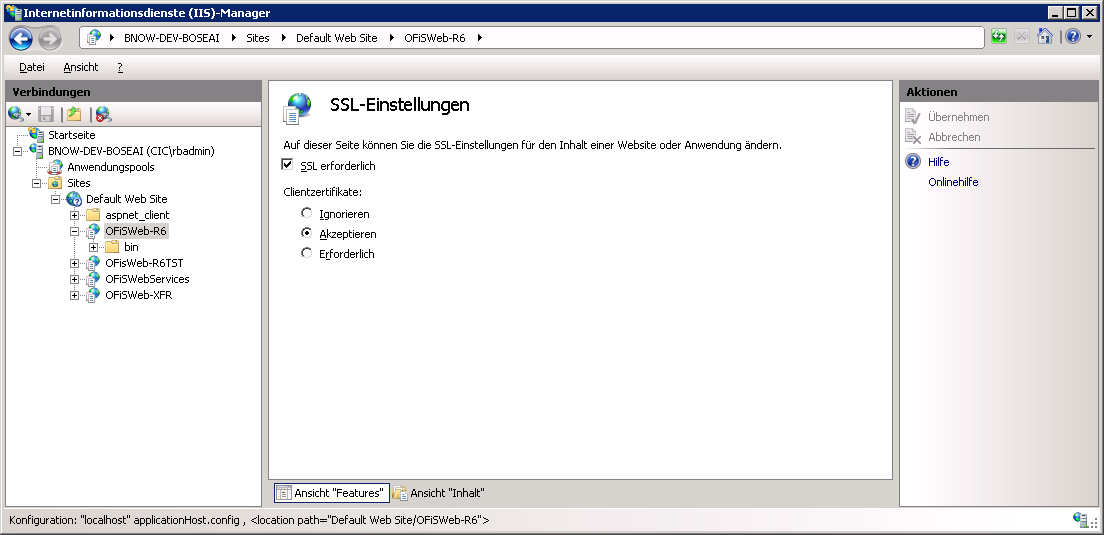


1. SSL-Settings der Website anpassen, um das 2-Way-SSL zu erzwingen:

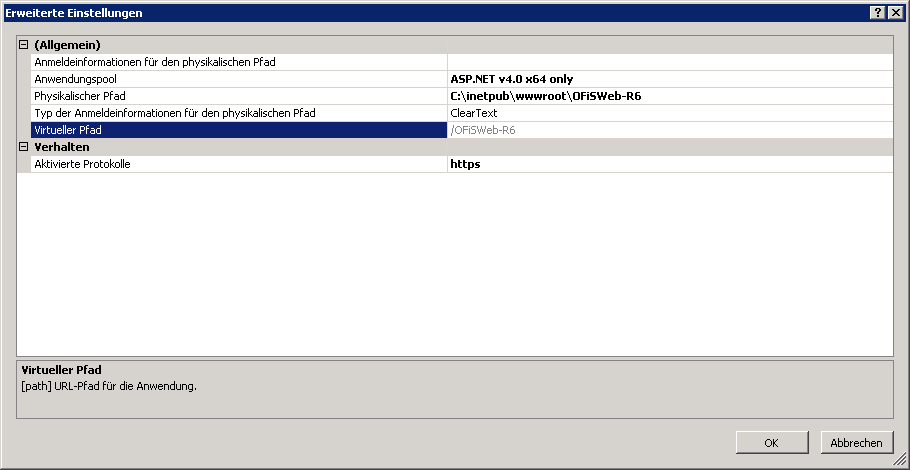


#### Konfiguration HTTPS

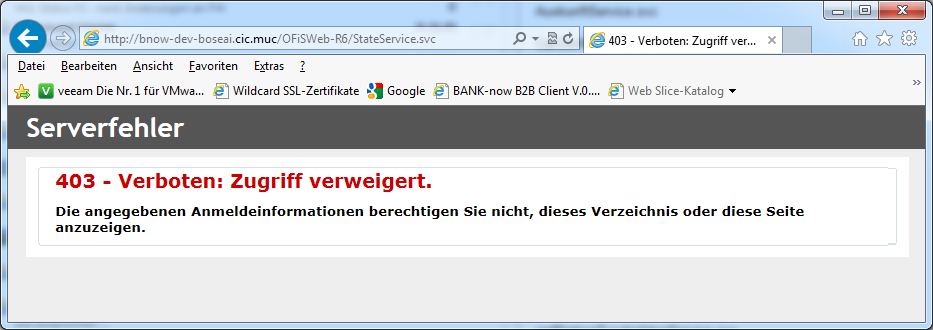
Auf dem BOSEAI Server wurden die SSL Einstellungen für eine WebSeite wie folgt geändert:



Im Anschluss wurde in den erweiterten Einstellungen das aktivierte Protokoll von HTTP nach HTTPS geändert:



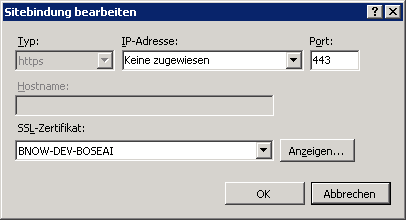
Verboten mit HTTP:



Erlaubt mit HTTPS:



Um eine HTTPS Verbindung zu initiieren, benötigen Sie ein Zertifikat (vgl. Abb. u.). Die Installation wurde oben beschrieben.

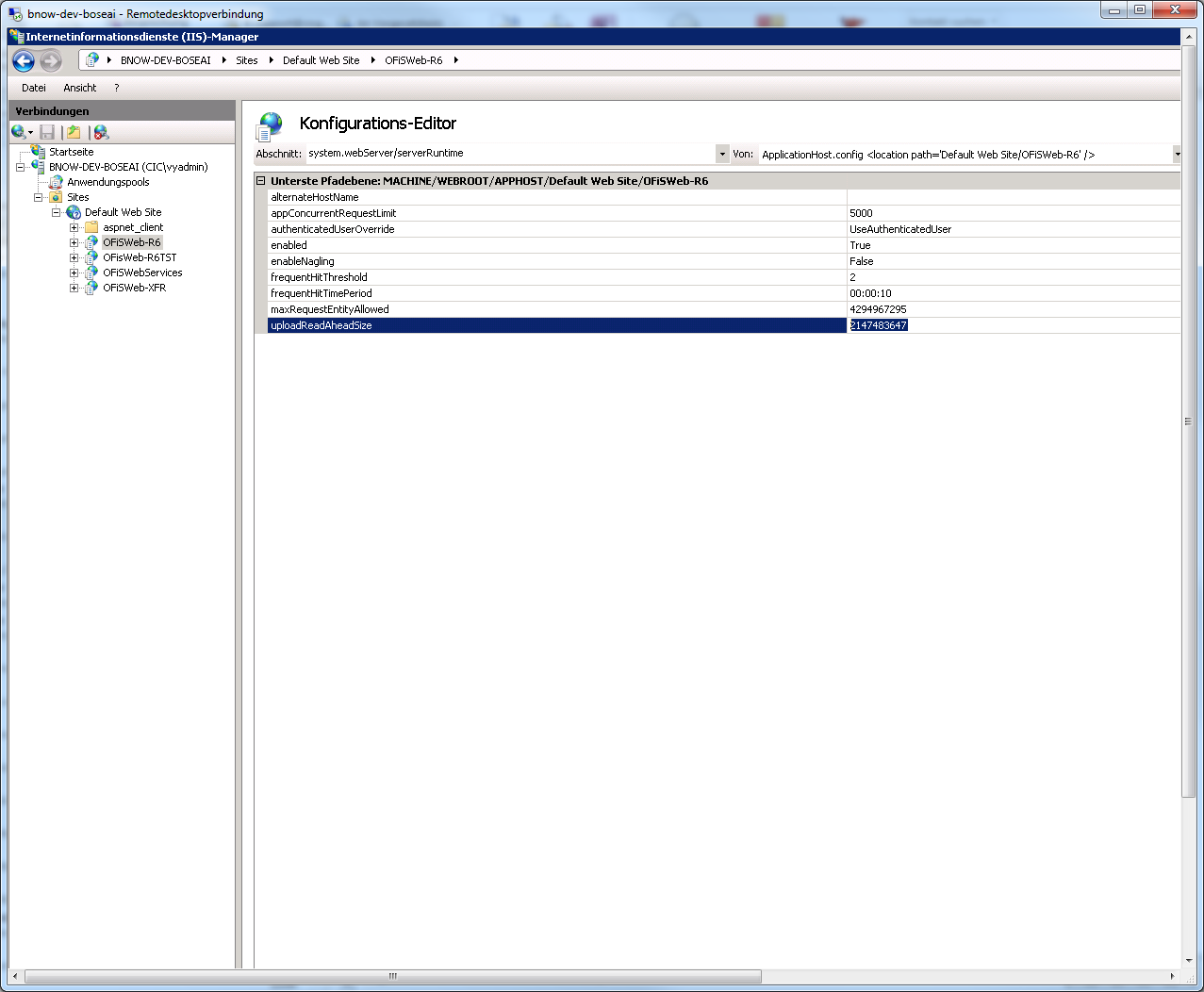


#### uploadReadAheadSize

Auf den BOS-Servern muss die Einstellung für den Parameter uploadReadAheadSize auf einen höheren Wert gesetzt werden, damit die Webservices FibuService und StreamService grössere Datenmengen annehmen können.

Der Parameter wird im GUI vom IIS 7.5 wie folgt gesetzt:

* Im Server-Baum auf die Site (z.B. OFiSWebServices) – Sites – Default WebSite – OFiSWeb-R6 klicken
* Im rechten Panel den Konfigurations-Editor aufrufen
* In der Dropdown-Box oben den Abschnitt system.webServer / serverRuntime auswählen
* In der Parameter -Liste uploadReadAheadSize auf den gewünschten Wert (z.B. 2147483647) setzen
* Änderung speichern
* Den IIS-Server neu starten.



uploadReadAheadSize gibt die Byte-Anzahl an, die vom Webserver in den Buffer eingelesen und an die ISAPI-Erweiterung weitergegeben werden. Das gilt für einen Client-Request. Erlaubte Werte sind zwischen 0 und 2147483647, wobei der Default-Wert 49152 beträgt.

Wenn der Wert zu klein ist, dann liefert der IIS-Server den HTTP-Fehler 413 Request entity too large (Anforderung zu groß).

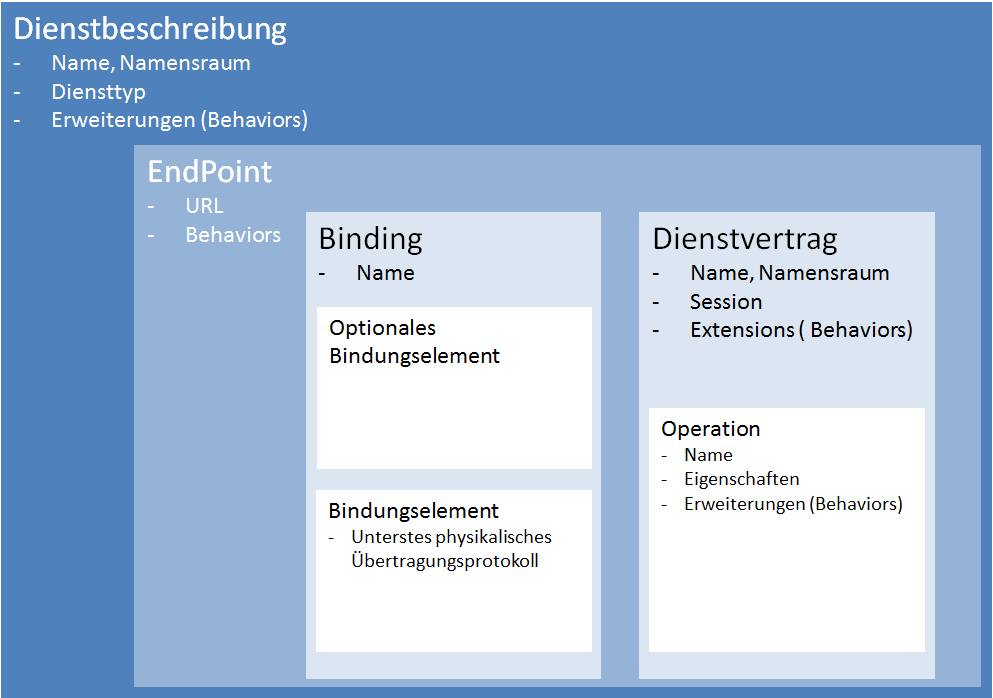
Der Parameter kann auch manuell in der ApplicationHost.config ändern, die sich im Verzeichnis %windir% \system32\inetsrv\config befindet.

# Konfigurationseinstellungen web.config

Die erforderlichen Konfigurationseinstellungen sind in der Datei *web.config* vorzunehmen. Diese finden Sie bei einer Standardkonfiguration auf Ihrem BOS-Server z.B. unter C:\Inetpub\wwwroot\OFiSWebServices. Weitere Informationen zur Dienstmodellen und Service Konfigurationen beschreibt Microsoft unter:

<http://msdn.microsoft.com/de-de/library/ms733086.aspx>

Schematischer Aufbau einer web.config Datei:



### Aufbau des bindingConfiguration Attributs

Das binding Configuration Attribut verweist auf das Binding Element in dem die erweiterte Konfiguration vorgenommen wird.

Müssen die Standardwerte einer Bindung geändert werden, kann dies geschehen, indem das entsprechende **binding**-Element im **bindings**-Element konfiguriert wird. Diesem Attribut sollte derselbe Wert zugewiesen werden, wie dem **name**-Attribut des **binding**-Elements, das verwendet wird, um die Standardwerte zu ändern. Wenn kein Name angegeben wird oder kein **bindingConfiguration** in der Bindung festgelegt ist, wird die Standardbindung des Bindungstyp im Endpunkt verwendet.

Der Wert ist eine Zeichenfolge, die den Namen der Bindung enthält, die beim Instanziieren des Endpunkts verwendet werden soll. Der Name der Bindung muss sich bei der Endpunktdefinition im Gültigkeitsbereich befinden. Der Standardwert ist eine leere Zeichenfolge.

Dieses Attribut wird zusammen mit **binding** zum Verweisen auf eine spezifische Bindungskonfiguration in der Konfigurationsdatei verwendet. Legen Sie dieses Attribut fest, wenn Sie eine benutzerdefinierte Bindung verwenden möchten. Andernfalls wird unter Umständen eine Ausnahme ausgelöst.

Weitere Informationen unter: <http://msdn.microsoft.com/de-de/library/ms731320.aspx>

### Aufbau des contract Attributs

Das Contract Attribut gibt die Schnittstelle an, die den Vertrag definiert. Das ist die Schnittstelle, die im CLR-Typ (Common Language Runtime) implementiert ist, der vom **name**-Attribut des **service**-Elements angegeben wird.

Der Wert ist eine Zeichenfolge, die angibt, welche Verträge von diesem Endpunkt verfügbar gemacht werden. Die Assembly muss den Vertragstyp implementieren. Wenn eine Dienstimplementierung einen einzelnen Vertragstyp implementiert, kann diese Eigenschaft ausgelassen werden. Der Standardwert ist eine leere Zeichenfolge.

Weitere Informationen unter: <http://msdn.microsoft.com/de-de/library/ms731320.aspx>

### Aufbau des name Attributs

Das Name ist ein Optionales Attribut. Es ist eine Zeichenfolge, die den Namen des Dienstendpunkts angibt. Der Standardwert ist die Verkettung des Bindungsnamen und des Vertragsbeschreibungsnamens. Dienste haben möglicherweise mehrere Endpunkte, sodass das **name**-Attribut des Endpunkts sich vom Namen des Diensts unterscheidet.

Weitere Informationen unter: <http://msdn.microsoft.com/de-de/library/ms731320.aspx>

## ApplicationSettings (Dienstbeschreibung)

Im Abschnitt ApplicationSettings wurden die grundlegenden Datenbank- und Servereinstellungen vorgenommen. Es folgt ein aktueller Auszug bezogen auf die Einstellungen in Horgen.

<applicationSettings>

<Cic.OpenOne.Common.Properties.Config>

<setting name="OpenLeaseConnectionStringDirect" serializeAs="String">

<value>False</value>

</setting>

<setting name="OpenLeaseConnectionStringOraClientHome" serializeAs="String">

<value>OraHome112\_64</value>

</setting>

<setting name="OpenLeaseConnectionStringDataSource" serializeAs="String">

<value>XXX.CH.HEDANI.NET</value>

</setting>

<setting name="OpenLeaseConnectionStringUserId" serializeAs="String">

<value>cic</value>

</setting>

<setting name="OpenLeaseConnectionStringPassword" serializeAs="String">

<value>SKYE</value>

</setting>

<setting name="OpenLeaseConnectionStringPlainPassword" serializeAs="String">

<value>False</value>

</setting>

<setting name="DatasourceProviderName" serializeAs="String">

<value>Devart.Data.Oracle</value>

</setting>

<setting name="DBConnectionTimeout" serializeAs="String">

<value>30</value>

</setting>

<setting name="DBValidateConnection" serializeAs="String">

<value>false</value>

</setting>

<setting name="DBMinPoolSize" serializeAs="String">

<value>10</value>

</setting>

<setting name="DBMaxPoolSize" serializeAs="String">

<value>100</value>

</setting>

<setting name="DBStatementCacheSize" serializeAs="String">

<value>5</value>

</setting>

<setting name="PreSharedKeyType" serializeAs="String">

<value>MA</value>

</setting>

<setting name="PreSharedKeyPath" serializeAs="String">

<value>INTEROP/LOGINKONTEXT/PSK</value>

</setting>

<setting name="SoapLoggingEnabled" serializeAs="String">

<value>True</value>

</setting>

<setting name="SoapLoggingEnabledAuskunft" serializeAs="String">

<value>True</value>

</setting>

</Cic.OpenOne.Common.Properties.Config>

<Cic.OpenOne.GateBANKNOW.Common.Settings>

<setting name="NotificationGatewaySmtpServer" serializeAs="String">

<value>XXX.csintra.net</value>

</setting>

<setting name="NotificationGatewaySmtpFaxConnector" serializeAs="String">

<value>XXX.bank-now.ch</value>

</setting>

<setting name="NotificationGatewaySmtpSmsConnector" serializeAs="String">

<value>XXX.credit-suisse.ch</value>

</setting>

<setting name="SQLRetryCount" serializeAs="String">

<value>1</value>

</setting>

</Cic.OpenOne.GateBANKNOW.Common.Settings>

</applicationSettings>

### Parameter Liste ApplicationSettings

|  |  |  |
| --- | --- | --- |
| Parameter | Beschreibung | Mögliche Werte |
| OpenLeaseConnectionStringDirect | True: Direkte Verbindung ohne Oracle Client (mit Devart dotConnect)  False: Verbindung mit Oracle Client | True / False |
| OpenLeaseConnectionStringDataSource | TNS-Name | z.B. dofis |
| OpenLeaseConnectionStringOraClientHome | OracleClientHome-Name. Er kann aus dem Oracle Universal Installer herausgelesen werden. | z.B. OraHome112\_64 |
| OpenLeaseConnectionStringUserID | UserID | z.B. cic |
| OpenLeaseConnectionStringPassword | DB-User Password |  |
| OpenLeaseConnectionStringPlainPassword | Wird Passwort verschlüsselt | True / False |
| DatasourceProviderName | Name des Oracle Treibers | z.b. Devart.Data.Oracle |
| DBConnectionTimeout | Wartezeit bis Timeout Error; | 15-60, Default 30 |
| DBValidateConnection | Vor der WS-Verbindung wird geprüft, ob Verbindung aufgebaut werden kann | True / False; Default False |
| DBMinPoolSize | Connection String Attribute  Minimum number of connections in a pool | Default 10 |
| DBMaxPoolSize | Connection String Attribute  Maximum number of connections in a pool | Default 100 |
| DBStatementCacheSize | Connection String Attribute | Default 5 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Parameter | Beschreibung | Mögliche Werte |
| PreSharedKeyType | Verschlüsselung für B2B/EAI | B2B /EAI |
| PreSharedKeyPath | Store Location in Database | INTEROP/LOGINKONTEXT/  PSKB2B |
| SoapLoggingEnabled | Logging von Soap Requests und Responses | True / False; Default False |
| SoapLoggingEnabledAuskunft | Logging von Soap Requests von AuskunftService | True / False; Default False |
| OpenLeaseConnectionStringServerName | Datenbank-Servername | z.B. cXXX.csintra.net |
| OpenLeaseConnectionStringSID | Oracle System Identifier | z.B. Dofis |
| OpenLeaseConnectionStringServerPort | Port | z.B. 1523 |

### Weitere Konfigurationen im web.config Abschnitt ApplicationSettings:

Mail/Fax/SMS

Mail, Fax und SMS werden nur vom FAT-Client aus verschickt. Daher sind diese Einstellungen lediglich im BOS-EAI notwendig und zu konfigurieren.

N*otificationGatewaySmtpServer:* IP oder DNS Adresse des SMTP Servers

*NotificationGatewaySmtpFaxConnector:* Adressendung für den Versand der Fax Sendungen

*NotificationGatewaySmtpSmsConnector*: Adressendung für den Versand der SMS Sendungen

<setting name="NotificationGatewaySmtpServer" serializeAs="String">

<value>XXX.csintra.net</value>

</setting>

<setting name="NotificationGatewaySmtpFaxConnector" serializeAs="String">

<value>XXX.bank-now.ch</value>

</setting>

<setting name="NotificationGatewaySmtpSmsConnector" serializeAs="String">

<value>XXX.credit-suisse.ch</value>

</setting>

SQLRetryCount

Der Wert gibt die Anzahl der Wiederholungen an. Default, falls die Einstellung nicht vorhanden ist, ist die 1. Die Einstellung wirkt sich auf folgende die Bereiche Aggregation und Suche (Angebot, Antrag, Vertrag) aus.

      <setting name="SQLRetryCount" serializeAs="String">

        <value>1</value>

      </setting>

## Bindings Settings

Das binding Configuration Attribut verweist auf das Binding Element in dem die erweiterte Konfiguration vorgenommen wird. Es folgt ein aktueller Auszug bezogen auf die Einstellungen in Horgen.

<!-- Eurotax Service -->

<binding proxyAddress="http://intranet-proxy.ch.hedani.net:8080" name="ForecastBindingRPC" closeTimeout="00:01:00" openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard" maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="false">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384"

maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="None">

<transport clientCredentialType="None" proxyCredentialType="None" realm="" />

<message clientCredentialType="UserName" algorithmSuite="Default" />

</security>

</binding>

<!-- Eurotax Service -->

<binding proxyAddress="http://intranet-proxy.ch.hedani.net:8080" name="ForecastBindingDoc" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered"

useDefaultWebProxy="false">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="None">

<transport clientCredentialType="None" proxyCredentialType="None" realm="" />

<message clientCredentialType="UserName" algorithmSuite="Default" />

</security>

</binding>

<!-- Eurotax Service -->

<binding proxyAddress="http://intranet-proxy.ch.hedani.net:8080" name="ValuationBinding" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="false">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="None">

<transport clientCredentialType="None" proxyCredentialType="None" realm="" />

<message clientCredentialType="UserName" algorithmSuite="Default" />

</security>

</binding>

<!-- DecisionEngine Service -->

<binding name="S1PublicServicePortBinding" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="true">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="Transport">

<transport clientCredentialType="None" proxyCredentialType="None" realm="" />

<message clientCredentialType="UserName" algorithmSuite="Default" />

</security>

</binding>

<!-- Deltavista Service -->

<binding proxyAddress="http://intranet-proxy.ch.hedani.net:8080" name="DVSOAPServiceV4SoapBinding" closeTimeout="00:02:04" openTimeout="00:02:05"

receiveTimeout="00:10:00" sendTimeout="00:02:06" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="false">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="Transport">

<transport clientCredentialType="Certificate" />

<message clientCredentialType="UserName" algorithmSuite="Default" />

</security>

</binding>

<!-- Deltavista Service -->

<binding proxyAddress="http://intranet-proxy.ch.hedani.net:8080" name="DVSOAPServiceSoapBinding" closeTimeout="00:02:01" openTimeout="00:02:02"

receiveTimeout="00:10:00" sendTimeout="00:02:03" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="false">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="Transport">

<transport clientCredentialType="Certificate" realm="" />

<message clientCredentialType="UserName" algorithmSuite="Default" />

</security>

</binding>

<!-- ZEK Service -->

<binding proxyAddress="http://intranet-proxy.ch.hedani.net:8080" name="ZEKTransactionServiceSoapBinding" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="false">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="Transport">

<transport clientCredentialType="Certificate" />

</security>

</binding>

<!-- ZEK Batch Service -->

<binding name="ZEKBatchTransactionServiceSoapBinding" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text"

textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="true">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="Transport">

<transport clientCredentialType="Certificate"/>

</security>

</binding<!-- Kremo Service -->

<binding name="ServiceSoap" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"

maxBufferSize="1024000" maxBufferPoolSize="524288" maxReceivedMessageSize="1024000" messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="true">

<readerQuotas maxDepth="32" maxStringContentLength="1024000" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />

<security mode="Transport">

<transport clientCredentialType="None" />

</security>

</binding>

closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:10:00" >

<readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" maxBytesPerRead="2147483647" maxNameTableCharCount="2147483647" />

<security mode="None">

<transport clientCredentialType="None" />

</security>

</binding>

</basicHttpBinding>

</bindings>

<!-- StreamBinding -->

<binding name="StreamBinding" maxReceivedMessageSize="2147483647" transferMode="Streamed" closeTimeout="00:01:00" openTimeout="00:01:00"

receiveTimeout="00:10:00" sendTimeout="00:10:00" >

<readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647" maxBytesPerRead="2147483647" maxNameTableCharCount="2147483647" />

<security mode="None">

<transport clientCredentialType="None" />

</security>

</binding>

</basicHttpBinding>

</bindings>

### Parameter Liste Bindings

Folgend werden kurz die einzelnen Bindings erklärt; Details zu den Services sind den jeweiligen Fachkonzepten zu entnehmen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Binding | Binding name | Beschreibung | https | proxy |
| Eurotax Service | ForecastBindingRPC | Aufruf Eurotax-Forecast Service | Nein | Ja |
| Eurotax Service | ForecastBindingDoc | Aufruf Eurotax-Forecast Service | Nein | Ja |
| Eurotax Service | ValuationBinding |  | Nein | Ja |
| DecisionEngine Service | S1PublicServicePortBinding | Aufruf der DecisionEngine von ARVATO | Ja | Nein/Default |
| Deltavista  Service | DVSOAPServiceV4SoapBinding | Aufruf Deltavista Service | Ja | Ja |
| Deltavista  Service | DVSOAPServiceSoapBinding | Aufruf Deltavista Service | Ja | Ja |
| ZEK Service | ZEKTransactionServiceSoapBinding | Aufruf ZEK Service | Ja | Ja |
| ZEK Batch Service | ZEKBatchTransactionServiceSoapBinding | Aufruf ZEK-Batch Service | Ja | Nein |
| Kremo Service | ServiceSoap | Aufruf Kremo Service | Ja | Nein |
| StreamBinding | StreamBinding | Aufruf des SimpleService auf Desicion Engine | Ja | Nein |

## Endpoint Tag (Client-Tag)

Die WebServices sind vollständig von den benutzten Protokollen getrennt. Die Trennung zwischen Transport und Service Layer erlaubt die einfache Adaption an Ihre IT Infrastruktur. Die Konfigurationsdatei web.config ist hierzu hierarchisch aufgebaut. Die erste Ebene - der Tag Client - beschreibt die grundlegenden Verbindungseinstellungen eines Endpoints und enthält Verweise auf weitergehende Einstellungen.

Folgend wird ein Auszug der web.config aufgelistet.

<client>

<endpoint address="http://webservices.eurotaxglass.com/webservices/forecast.wsa" binding="basicHttpBinding" bindingConfiguration="ForecastBindingRPC" contract="DAO.Auskunft.EurotaxRef.ForecastPortRPC" name="ForecastPortRPC" />

<endpoint address="http://webservices.eurotaxglass.com/webservices/forecast.wsa" binding="basicHttpBinding" bindingConfiguration="ForecastBindingDoc" contract="DAO.Auskunft.EurotaxRef.ForecastPortDocument" name="ForecastPortDoc" />

<endpoint address="http://webservices.eurotaxglass.com/webservices/Valuation.wsa" binding="basicHttpBinding" bindingConfiguration="ValuationBinding" contract="DAO.Auskunft.EurotaxValuationRef.ValuationSoapPort" name="ValuationService" />

<endpoint address="https://chw20026948.ch.ad.hedani.net:8443/strategyone/S1Public" binding="basicHttpBinding" bindingConfiguration="S1PublicServicePortBinding" contract="DAO.Auskunft.DecisionEngineRef.S1Public" name="S1PublicServicePort" />

<endpoint address="https://chw20026948.ch.ad.hedani.net/KREMOWebService/Service.asmx" binding="basicHttpBinding" bindingConfiguration="ServiceSoap" contract="DAO.Auskunft.KREMORef.ServiceSoap" name="ServiceSoap" />

<endpoint address="https://preprodservices.crif-online.ch/ZEKBatch/ZekBatchServiceAnonymous" behaviorConfiguration="myCertEndpointBehavior" binding="basicHttpBinding" bindingConfiguration="ZEKBatchTransactionServiceSoapBinding" contract="DAO.Auskunft.ZEKBatchRef.ZEKBatchService" name="ZEKBatchTransactionService"/>

<endpoint address="https://preprodservices.crif-online.ch/ZEKsoap/ZEKTransactionServiceAnonymousV3" behaviorConfiguration="myCertEndpointBehavior" binding="basicHttpBinding" bindingConfiguration="ZEKTransactionServiceSoapBinding" contract="DAO.Auskunft.ZEKRef.ZEKService" name="ZEKTransactionService" /

>

<endpoint address="https://preprodservices.crif-online.ch/dvss/DVSOAPService" binding="basicHttpBinding" bindingConfiguration="DVSOAPServiceSoapBinding" behaviorConfiguration="myCertEndpointBehavior" contract="DAO.Auskunft.DeltavistaRef2.DVSoapService" name="DVSOAPService" />

<endpoint address="https://preprodservices.crif-online.ch/dvss/DVSOAPServiceV4" behaviorConfiguration="myCertEndpointBehavior" binding="basicHttpBinding" bindingConfiguration="DVSOAPServiceV4SoapBinding" contract="DAO.Auskunft.DeltavistaRef.DVSoapServiceV4" name="DVSOAPServiceV4" />

<endpoint address="https://chw20026946.ch.ad.hedani.net/SimpleService" binding="basicHttpBinding" bindingConfiguration="StreamBinding" contract="DAO.Auskunft.RISKEWBS1Ref.ICicService" name="RISKEWBS1" />

<endpoint address="https://chw20026948.ch.ad.hedani.net/IBANKernel/IBANKernel.svc" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="IBANService.IBANKernel" name="IBANKernelBinding" />

</client>

### Parameterliste Endpoint Tag (Client-Tag)

Jeder Endpoint-Tag verweist auf ein binding. Die Verbindung wird über bindingConfiguration (Endpoint-Tag) und name (binding-Tag) hergestellt.

|  |  |  |
| --- | --- | --- |
| bindingConfiguration /name | Url | Beschreibung |
| ForecastBindingRP | http://webservices.eurotaxglass.com/webservices/forecast.wsa | URL zu Eurotax ForecastBindingRP |
| ForecastBindingDoc | http://webservices.eurotaxglass.com/webservices/forecast.wsa | URL zu Eurotax ForecastBindingDoc |
| ValuationBinding | http://webservices.eurotaxglass.com/webservices/Valuation.wsa | URL zu Eurotax ValuationBinding |
| S1PublicServicePortBinding | https://chw20026948.ch.ad.hedani.net:8443/strategyone/S1Public | URL zu Decision Engine (strategyone)  S1Public |
| ServiceSoap | https://chw20026948.ch.ad.hedani.net/KREMOWebService/Service.asmx | URL zu den KREMOWebService KREMOWebService/Service.asmx |
| myCertEndpointBehavior | https://preprodservices.crif-online.ch/ZEKBatch/ZekBatchServiceAnonymous | URL zu ZekBatchServiceAnonymous |
| myCertEndpointBehavior | https://preprodservices.crif-online.ch/ZEKsoap/ZEKTransactionServiceAnonymousV3 | URL zu ZEK TransactionService ZEKTransactionServiceAnonymousV3 |
| DVSOAPServiceSoapBinding | https://preprodservices.crif-online.ch/dvss/DVSOAPService | URL zu DVSOAPServiceSoapBinding |
| myCertEndpointBehavior | https://preprodservices.crif-online.ch/dvss/DVSOAPServiceV4 | URL zu VSOAPServiceV4 |
| StreamBinding | https://chw20026946.ch.ad.hedani.net/SimpleService | URL zu der Installation SimpleService |
| myWsHttpBinding | https://chw20026948.ch.ad.hedani.net/IBANKernel/IBANKernel.svc | URL zur Installation IBANKernel-Service (Decision Engine Server) |

## Service Definition (Service Tag)

Im Service Tag werden die angesprochenen WebServices definiert. Folgend ein Auzug aus der web.config.

service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.createOrUpdateAntragService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IcreateOrUpdateAntragService" bindingNamespace="http://cic-software.de/GateBANKNOW/createOrUpdateAntragService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="createOrUpdateAntragServicePortMex1" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW/createOrUpdateAntragService" />

</service>

<service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.createOrUpdateAngebotService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IcreateOrUpdateAngebotService" bindingNamespace="http://cic-software.de/GateBANKNOW/createOrUpdateAngebotService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="createOrUpdateAngebotServicePortMex1" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW/createOrUpdateAngebotService" />

</service>

<service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.changeAngebotService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IchangeAngebotService" bindingNamespace="http://cic-software.de/GateBANKNOW/changeAngebotService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="changeAngebotServicePortMex1" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW/changeAngebotService" />

</service>

<service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.changeAntragService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IchangeAntragService" bindingNamespace="http://cic-software.de/GateBANKNOW/changeAntragService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="changeAntragServicePortMex1" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW/changeAntragService" />

</service>

<service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.AuskunftService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IAuskunftService" bindingNamespace="http://cic-software.de/GateBANKNOW/AuskunftService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="AuskunftServicePortMex1" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW/AuskunftService" />

</service>

<service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.StateService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IStateService" bindingNamespace="http://cic-software.de/GateBANKNOW/StateService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="AuskunftServicePortMex1" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW/StateService" />

</service>

<service name="Cic.OpenOne.GateBANKNOW.Service.StreamService" behaviorConfiguration="StreamBehavior">

<endpoint address="" binding="basicHttpBinding" bindingConfiguration="StreamBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IStreamService" bindingNamespace="http://cic-software.de/GateBANKNOW">

<identity>

<dns value="localhost"/>

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpsBinding" bindingName="StreamServicePortMex2" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW"/>

</service>

<!-- Fibu Binding -->

<service behaviorConfiguration="Cic.OpenOne.Service.ServiceBehavior" name="Cic.OpenOne.GateBANKNOW.Service.FibuService">

<endpoint address="" binding="wsHttpBinding" bindingConfiguration="myWsHttpBinding" contract="Cic.OpenOne.GateBANKNOW.Service.Contract.IFibuService" bindingNamespace="http://cic-software.de/GateBANKNOW">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<!-- UNCOMMENT for SSL -->

<endpoint address="mex" binding="mexHttpBinding" bindingName="FibuServicePortMex2" contract="IMetadataExchange" bindingNamespace="http://cic-software.de/GateBANKNOW" />

</service>

</services>

### Parameterliste Service Definition

|  |  |  |  |
| --- | --- | --- | --- |
| Service Name | Beschreibung | B2B | EAI |
| createOrUpdateAntragService | WebService um Anträge zu Erzeugen | X | X |
| createOrUpdateAngebotService | WebService um Angebote zu Erzeugen, | X | X |
| changeAngebotService | WebService um Änderungen an Angebote zu speichern | X | X |
| changeAntragService | WebService um Änderungen an Anträgen zu speichern | X | X |
| AuskunftService | WebService für die Auskünfte | X | X |
| StateService | WebService für die Abfrage des Status des WebServices |  | X |
| StreamService | WebService für den SimpleService | X | X |
| FibuService | WebService für den FibuService; Aufruf vom EAIJob |  | X |
| CarConfigurator  (nicht in Auszug) | WebService für den Aufruf des CarConfigurator | X |  |
| createOrUpdateKundeService  (nicht in Auszug) | WebService für das Erzeugen eines B2B Kunden | X |  |
| createSchnellkalkulationService  (nicht in Auszug) | WebService für die Schnellkalkulation im B2B | X |  |
| getBuchwertService  (nicht in Auszug) | WebService für die Ermittlung des Buchwertes | X |  |
| getPartnerZusatzdatenService  (nicht in Auszug) | WebService für die Ermittlung der Partner Zusatzdaten | X |  |
| loginPartnerService  (nicht in Auszug) | WebService für das Login im B2B | X |  |
| printAngebotService  (nicht in Auszug) | WebService für das Drucken von Angeboten im B2B | X |  |
| printAntragService  (nicht in Auszug) | WebService für das Drucken von Anträgen im B2B | X |  |
| searchAngebotService  (nicht in Auszug) | WebService für das Suchen von Angeboten im B2B | X |  |
| searchAntragService  (nicht in Auszug) | WebService für das Suchen von Anträgen im B2B | X |  |
| searchKundeService  (nicht in Auszug) | WebService für das Suchen von Kunden im B2B | X |  |
| searchVertragService | WebService für das Suchen von Verträgen im B2B | X |  |

## WebService Verhalten (Behaviors Tag)

Im Behaviors Tag wird das Verhalten der WebServices konfiguriert. Ausgehende https Verbindungen nutzen den hier definierten Zertifikats Store.

<behavior name="Cic.OpenOne.Service.ServiceBehavior">

<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>

<serviceThrottling maxConcurrentCalls="75" maxConcurrentInstances="105" maxConcurrentSessions="30"/>

<!-- //set httpsgetenabled only to true if the server supports https and has a certificate installed! -->

<serviceDebug includeExceptionDetailInFaults="true"/>

</behavior>

<behavior name="StreamBehavior">

<serviceMetadata httpGetEnabled="True" httpsGetEnabled="false"/>

<serviceDebug includeExceptionDetailInFaults="False"/>

</behavior>

</serviceBehaviors>

<endpointBehaviors>

<behavior name="myCertEndpointBehavior">

<clientCredentials>

<clientCertificate findValue="SSLAuth" storeLocation="LocalMachine" x509FindType="FindBySubjectName" />

</clientCredentials>

</behavior>

## DotConnect Dataprovider (DbProviderFactories Tag)

Dieser Eintrag in der web.config ist für die Konfiguration des DevArt dot Connectors definiert.

Für mehr Information kann hier nachgelesen werden: <http://msdn.microsoft.com/de-de/library/system.data.common.dbproviderfactories.aspx>

<!-- dotConnect Dataprovider -->

<system.data>

<DbProviderFactories>

<add name="dotConnect for Oracle" invariant="Devart.Data.Oracle"

description="Devart dotConnect for Oracle"

type="Devart.Data.Oracle.OracleProviderFactory, Devart.Data.Oracle,

Version=5.70.140.0, Culture=neutral, PublicKeyToken=09af7300eec23701" />

</DbProviderFactories>

</system.data>

## 2way-SSL-Konfiguration der Service-Hosts für BOS Intern / BOS Extern

Im Zuge der Umstellung auf 2way-SSL wird die Verbindung zwischen BOS und

Webanwendung entsprechend der DAP-Vorgaben verschlüsselt. Es wird davon ausgegangen, dass BOS Extern dafür auf einer CS-DAP-Plattform installiert ist. Nur dann sind die notwendigen Bibliotheken und das korrekte Zertifikatsmanagement auf dem Server vorhanden. Zur Aktivierung dieser Verschlüsselung auf der DAP-Plattform ist auf Seite des BOS Extern die Service-Host-Factory anzupassen.

In allen .svc-Dateien der Webservice-Anwendung (gleiche Location wie web.config, z.b. searchVertragService.svc) ist sicherzustellen, dass das Attribut Factory für den ServiceHost korrekt eingestellt ist:

<%@ ServiceHost Language="C#" Debug="true"

Service="Cic.OpenOne.GateBANKNOW.Service.searchVertragService"

CodeBehind="searchVertragService.svc.cs"

Factory="CS.Dap.Hosting.DapServiceHostFactory"%>

Die entsprechenden Dateien auf BOS Intern sind von dieser Änderung NICHT betroffen, das Factory-Attribut ist dort nicht vorhanden. Die Lieferungen für die entsprechenden BOS Server beinhalten diese Konfiguration bereits.

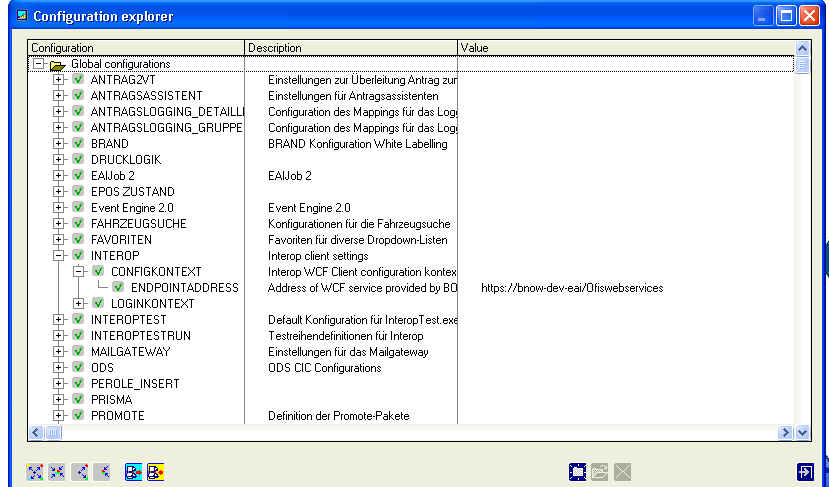
# Konfigurationseinstellungen OpenLease

Neben den Einstellungen in der web.config sind ebenfalls Anpassungen in OpenLease vorzunehmen.

## Konfiguration des Endpoints vom BOS-EAI

Der Endpoint vom BOS-EAI muss in der Adminbox konfiguriert werden. Mit dieser Einstellung wird die Verbindung vom Mitarbeiter-Client zu den WebServices hergestellt.

Hierzu geht man im OpenLease Client in den Configuration Explorer



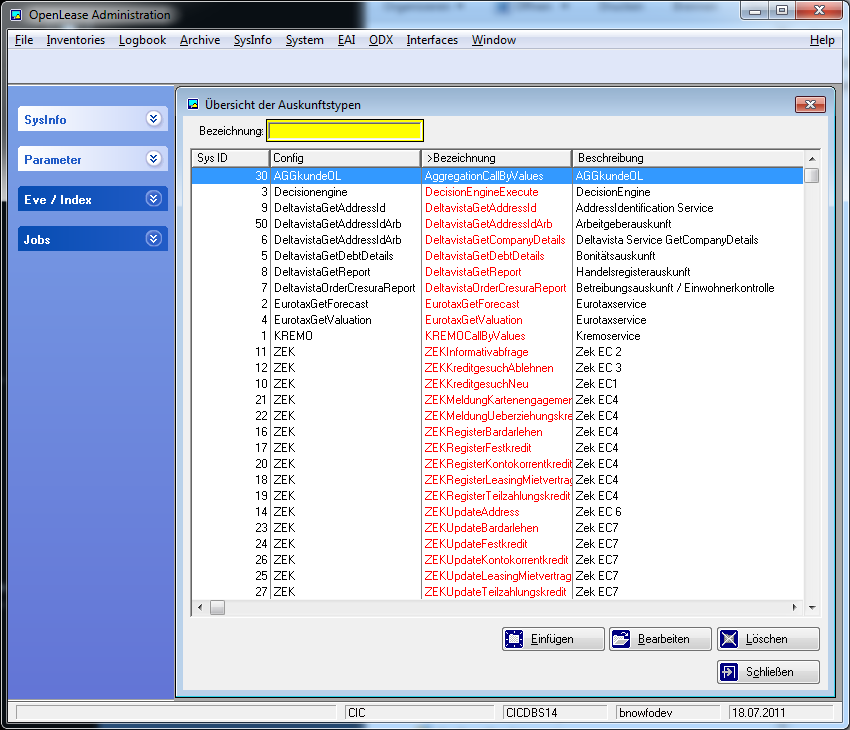
Unter Interop – CONFIGKONTEXT- ENDPOINTADDRESS wird der Pfad zu den WebServices einstellt.

### Test der Verbindung

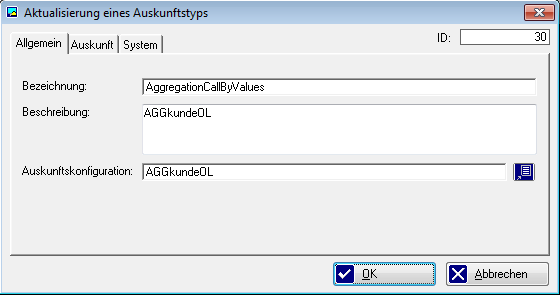
Anhand des Interop-TestClients kann schnell geprüft werden, ob die Einstellung richtig ist und der Mitarbeiter-Client die WebServices aufrufen kann. Dies ist im Dokument „InteropDokumentation“ beschrieben.

## Auskunftstypen

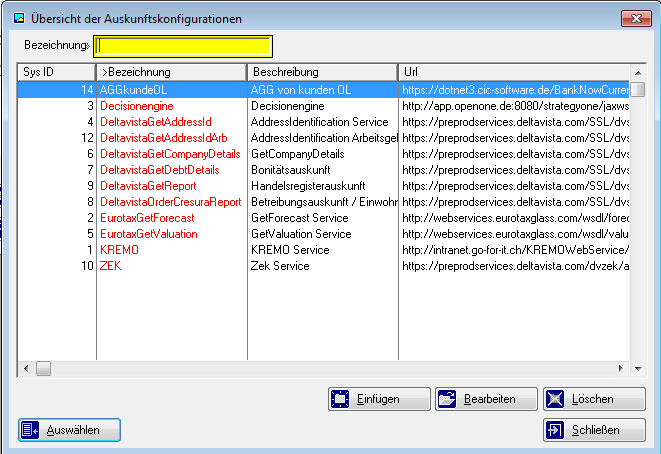
In der Datenbank muss sowohl die Tabelle Auskunftcfg als auch die Tabelle Auskunfttyp genauso gefüllt sein. Änderungen nehmen Sie über die OL4i Adminbox -> Menüzeile EAI -> Auskunfttypen vor. Folgendes Fenster wird geöffnet:



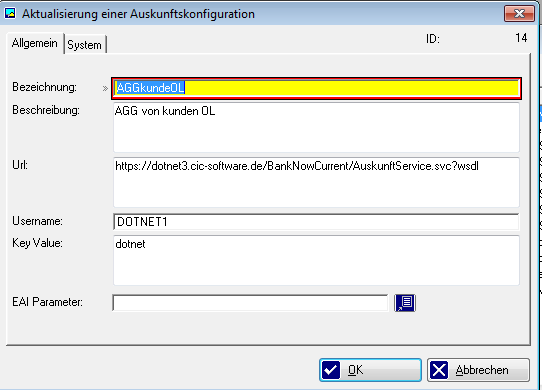
Wählen Sie den zu ändernden Eintrag und klicken Sie auf Bearbeiten. Folgendes Fenster wird geöffnet:



Klicken Sie auf den Display List Button im Feld Auskunftskonfiguration. Folgendes Fenster wird geöffnet:



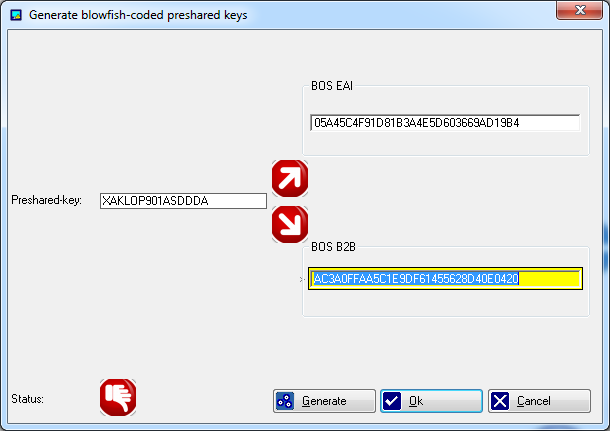
Selektieren Sie den zu bearbeitenden Eintrag und klicken Sie auf den Button Bearbeiten. Folgendes Fenster wird geöffnet:



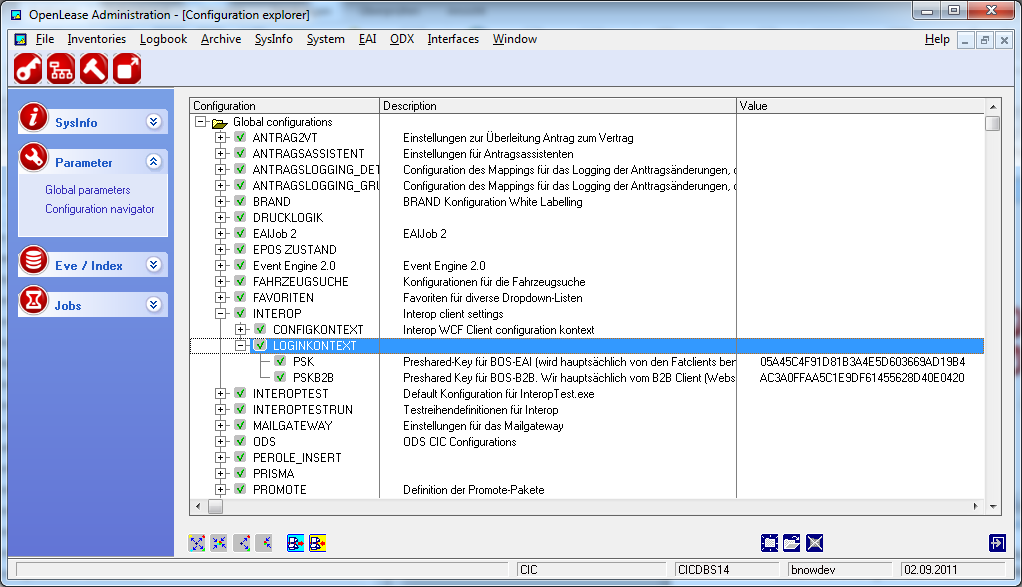
Der Parameter URL ist rein informativ. Die eigentliche Pflege der Endpoint Adresse wird in der web.config vorgenommen. Dagegen müssen die Parameter UserName und KeyValue (Passwort) den Vorgaben des Service Anbieters (Eurotax, KREMO,..) entsprechen.

## Pre Shared Keys

Pre Shared Keys für die Verbindung zwischen WebServices Server und Ol4i Client mit Interop Komponente werden in OL4i Admin Box erzeugt und verwaltet. Starten Sie das Programm durch Klick auf den Button PSK Generator in der Taskleiste. Folgendes Fenster wird geöffnet:



Der Button Generate erzeugt ein neues Schlüsselpaar. Der Button OK speichert dieses ab. Die Schlüssel werden in der Konfiguration gespeichert:



## Tabelle Auskunftcfg

In der Tabelle Auskunftcfg wird unter anderem die URL des Webdienstes, die Bezeichnung der Webservicemethode sowie ein Username und ein Passwort für den Zugang zum Webdienst gespeichert. Die Bezeichnung darf nicht geändert werden, da sonst Username und Passwort nicht mehr geladen werden können. Anbei eine kleine Übersicht über die Felder der Tabelle Auskunftcfg:

|  |  |
| --- | --- |
| Feld | Beschreibung des Feldes |
| SYSAUSKUNFTCFG | ID des Auskunftcfgsatzes |
| BEZEICHNUNG | Bezeichnung der Webservicemethode |
| BESCHREIBUNG | Beschreibung der Webservicemethode |
| URL | URL des Webservices |
| USERNAME | Username |
| KEYVALUE | Passwort |
| DATAOBJECT | Datenobjekt |
| SYSEAIPAR | Verknüpfung zur Tabelle EAIPAR |

## Tabelle Auskunfttyp

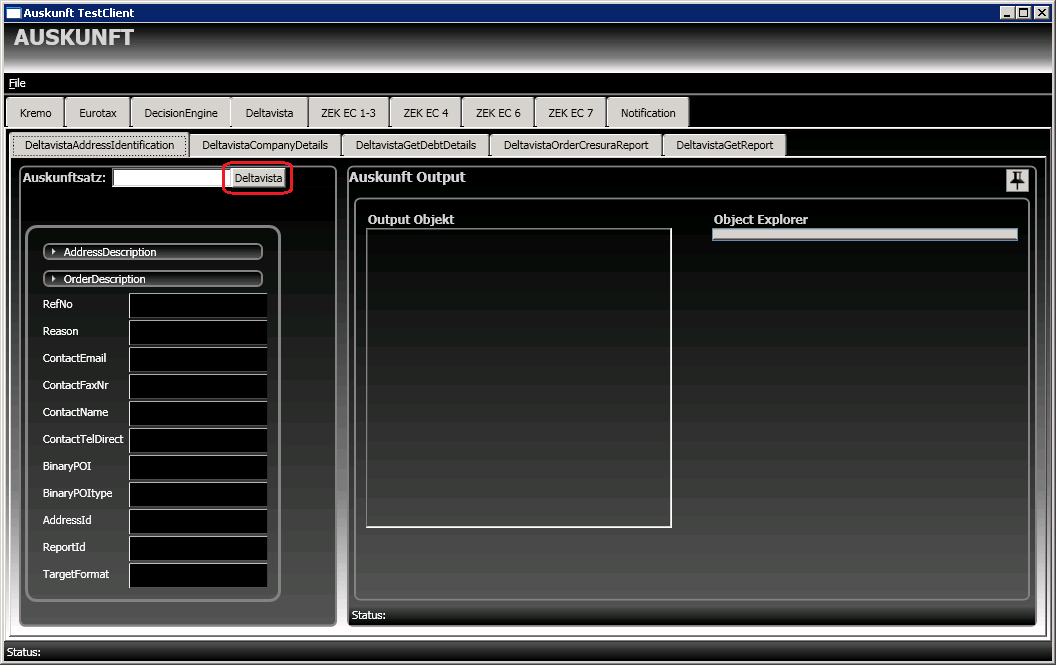
Die Tabelle Auskunfttyp beinhaltet neben einer Bezeichnung und einer Beschreibung der Webservicemethode auch eine Verknüpfung zur Tabelle Auskunftcfg:

|  |  |
| --- | --- |
| Feld | Beschreibung des Feldes |
| SYSAUSKUNFTTYP | ID des Auskunfttypsatzes |
| BEZEICHNUNG | Bezeichnung der Webservicemethode |
| BESCHREIBUNG | Beschreibung der Webservicemethode |
| SYSAUSKUNFTCFG | Verknüpfung zur Tabelle AUSKUNFTCFG |

# TestUI

## Aufruf der TestUI

In Ihrem Transferverzeichnis befindet sich im Order \BOS\Testui ein Testprogramm. Es prüft beim Start die Datenbank Verbindung und die Kommunikation zu den externen Diensten wie Deltavista, Eurotax und anderen. Entpacken Sie die Datei TestUI.zip auf Ihren Server. Wir nutzen z.B. das Zielverzeichnis \programme\cic\testui. Starten Sie Cic.OpenOne.GateBANKNOW.TestUI.exe und wählen Sie einen Dienst (z.B. Deltavista). Folgendes Fenster wird geöffnet:



Klicken Sie auf den rot markierten Button Deltavista.

## Test-UI Zertifikate

Viele Dienste sind nur über eine https Verbindung die mit Zertifikaten gesichert ist erreichbar. Sehen Sie hierzu das mit Ihnen Architekturschaubild, bzw. den Bebauungsplan. Bei fehlenden oder abgelaufenen Zertifikaten wird die https Verbindung nicht aufgebaut und der Fehler in einer MessageBox angezeigt. Im Gegensatz zu selbst erstellten Zertifikaten sind diese in der Regel durch ein Passwort gesichert, welches beim Import angegeben werden muss.

Gehen Sie dazu wie folgt vor:

Klicken Sie mit der rechten Maustaste auf das Zertifikat. Wählen Sie im Optionsmenü Install und folgen Sie den Dialogen des Import Wizards:



Prüfen Sie die Verfügbarkeit des externen Dienstes (z.B. Deltavista) nach erfolgreicher Datenbank Konfiguration und dem Zertifikatsimport erneut um den Kommunikationsweg (Proxy, Adressen) zu testen.

# Logging- Log4Net

Die WebServices unterstützen folgende Log Levels: OFF, FATAL, ERROR, WARN, DEBUG, INFO, ALL

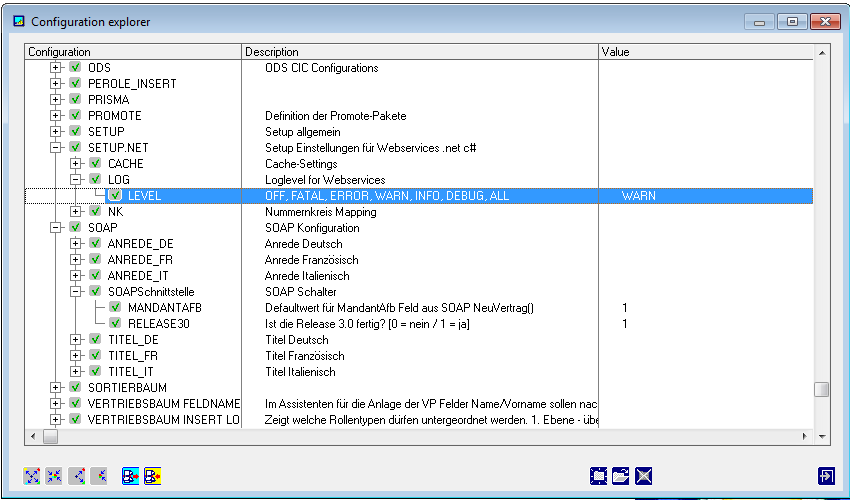
OFF schaltet das Logging ab, ALL loggt alles. Wir schlagen für einzelne Umgebungen folgende Log Levels vor:

|  |  |
| --- | --- |
| Umgebung | Log Level |
| Testsystem | INFO |
| Produktivsystem | WARN |

Um ein System von einer Test- in eine Produktiv Umgebung zu überführen gehen Sie bitte wie folgt vor:

## Log-Level Einstellung in der Admin-Box

Die Einstellung des Log Levels erfolgt in der AdminBox -> Configuration Explorer -> Setup.Net



### Konfiguration Log-File – Appender Tag

Im Appender Tag kann das Log-File an sich definiert werden.

|  |  |
| --- | --- |
| Log4-Config | Beschreibung |
| <file type="log4net.Util.PatternString" value="d:\logs\BOSextern\OFiSWeb-R6-[%processid].log" /> | Das Attribut „value“ definiert den Speicherort des Log-Files; Zusätzlich wird pro Prozess ein Log-File weggeschrieben. Wenn mehrere Webservices in einen Ordner die Logdateien ablegen, sind damit die Dateinamen pro Webservice eindeutig. |
| <param name="MaximumFileSize" value="3000KB" /> | Definiert die Größe des Log-Files. |

<root>

<level value="ALL" />

<appender-ref ref="LogFileAppender" />

</root>

<appender name="LogFileAppender" type="log4net.Appender.RollingFileAppender" >

<file type="log4net.Util.PatternString"

**value=" d:\logs\BOSextern\OFiSWeb-R6-[%processid].log"** />

<param name="AppendToFile" value="true" />

<param name="RollingStyle" value="Size" />

<param name="MaxSizeRollBackups" value="10" />

<param name="MaximumFileSize" value="3000KB" />

<param name="StaticLogFileName" value="true" />

<layout type="log4net.Layout.PatternLayout">

<param name="Header" value="&#13;&#10;Started Logging&#13;&#10;"/>

<param name="Footer" value="Logging stopped&#13;&#10;"/>

<param name="ConversionPattern" value="%n%d [%t] %-5p %c [%x]- %m%n" />

</layout>

</appender>

### Log-Level – Root Tag

Im Root-Tag kann der Log-Level eingestellt werden.

Jedoch wird dieser nur beim Verbindungsaufbau (z.B. nach jedem Neustart der Webservices) herangezogen.

Die Einstellung in der Admin-Box ist für die Log-Level führend und wird nach dem Verbindungsaufbau herangezogen und damit die Einstellung aus der log4net.config übersteuern; siehe hierzu Kapitel **Log-Level Einstellung in der Admin-Box**.

Um die Einstellung des Log-Level aus der Admin-Box ohne Verzögerung heranzuziehen, muss von der IT ein Flush des Cache durchgeführt werden.

### Schreibrechte im Log-Ordner

Achten Sie darauf, dass der User, welcher im IIS eingetragen ist, auf den LogOrdner – definiert im Log4Net- schreibrechte besitzen muss.

# Anhang

## Datenbank Anbindung -Fehlercodes

Aufstellung typischer Konfigurationsfehler im Format

|  |
| --- |
| Konfigurationsfehler |
| Log File Eintrag |

|  |
| --- |
| Falscher TNS-Name bei DataSource oder  falscher HOST in tnsname oder  TNS-Name nicht in tnsnames.ora |
| System.Data.EntityException: The underlying provider failed on Open. --->  Devart.Data.Oracle.OracleException: ORA-12154: TNS:could not resolve the connect identifier  specified  System.Data.EntityException: Fehler beim zugrunde liegenden Anbieter auf Open. --->  Devart.Data.Oracle.OracleException: ORA-12514: TNS: Listener kann in Connect-Deskriptor  angegebenen Service aktuell nicht auflösen |

|  |
| --- |
| Falsches Home |
| System.Data.EntityException: The underlying provider failed on Open. --->  Devart.Data.Oracle.OracleException: Can not find Oracle home. |

|  |
| --- |
| Falscher SERVICE\_NAME in tnsnames.ora: |
| System.Data.EntityException: The underlying provider failed on Open. --->  Devart.Data.Oracle.OracleException: ORA-12514: TNS:listener does not currently know of  service requested in connect descriptor |

|  |
| --- |
| Host im TNS falsch oder  Doppelter Eintrag in tnsnames mit gleichem SERVICE\_NAME |
| System.Data.EntityException: The underlying provider failed on Open. --->  Devart.Data.Oracle.OracleException: Can not find Oracle home. |

|  |
| --- |
| Falscher Port in TNSNames oder  Falscher Port in OpenLeaseConnectionStringServerPort  bei Direct = True |
| System.Data.EntityException: The underlying provider failed on Open. --->  Devart.Data.Oracle.OracleException: Server did not respond within the specified timeout interval |

|  |
| --- |
| Falscher OpenLeaseConnectionStringServerName oder  Falsche OpenLeaseConnectionStringSID  Bei Direct= True |
| System.Data.EntityException: The underlying provider failed on Open. --->  Devart.Data.Oracle.OracleException: NET: Invalid SID  System.Data.EntityException: Fehler beim zugrunde liegenden Anbieter auf Open. --->  Devart.Data.Oracle.OracleException: NET: Invalid SID |