Microsoft Dynamics CRM® 2013/2015 WCF Adapter for Microsoft BizTalk Server® 2010/2013

User manual

Published on: January 2015

© 2014 Rödl Consulting AG. All rights reserved

Table of Contents

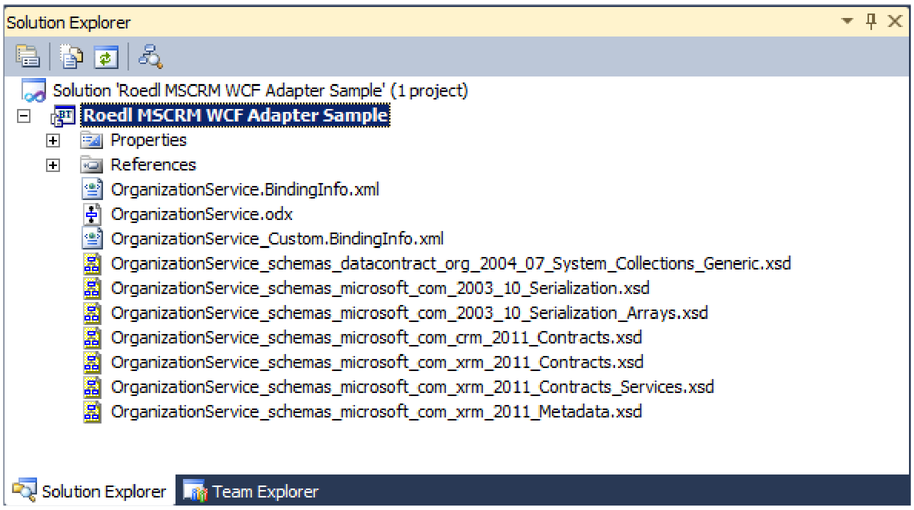
Introduction

Adapter mode of operation

The *R&P MS-CRM WCF Adapter for MS BizTalk* provides a seamless connection between the BizTalk Server and Microsoft Dynamics CRM. The adapter must be installed on the BizTalk Server.
On the MS CRM server, no further installations are required, because the *R&P MS-CRM WCF Adapter for MS BizTalk* communicates with Microsoft Dynamics CRM 2013/2015 via the original WCF service.
The *R&P MS-CRM WCF Adapter for MS BizTalk* is designed generically. This means that the adapter doesn't contain any explicit functionality for processing individual entities or request methods.
The advantage of this generic approach is that entities and attributes created in MS CRM can be used, too.
No customizing of the MS CRM system needed.

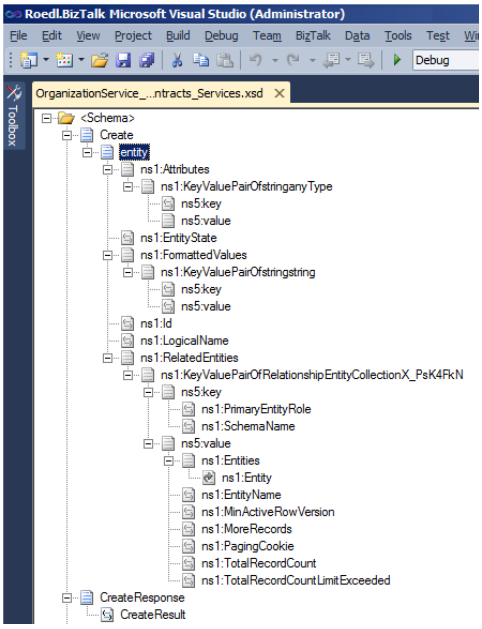
Why an adapter when Microsoft Dynamics CRM 2013/ 2015 provides a WCF interface?

Unlike earlier versions, Microsoft Dynamics CRM 2013/ 2015 provides a native WCF service you can use directly in the MS BizTalk Server.
But dealing with this service is anything but simple.
If you create a service reference for this WCF service, you will receive the following entries in your BizTalk solution:



Solution 'Roedl MSCRM WCF Adapter Sample' (1 project)
- Roedl MSCRM WCF Adapter Sample
  - Properties
  - References
  - OrganizationService.BindingInfo.xml
  - OrganizationService.odx
  - OrganizationService_Custom.BindingInfo.xml
  - OrganizationService_schemas_datacontract_org_2004_07_System_Collections_Generic.xsd
  - OrganizationService_schemas_microsoft_com_2003_10_Serialization.xsd
  - OrganizationService_schemas_microsoft_com_2003_10_Serialization_Arrays.xsd
  - OrganizationService_schemas_microsoft_com_crm_2011_Contracts.xsd
  - OrganizationService_schemas_microsoft_com_xrm_2011_Contracts.xsd
  - OrganizationService_schemas_microsoft_com_xrm_2011_Contracts_Services.xsd
  - OrganizationService_schemas_microsoft_com_xrm_2011_Metadata.xsd

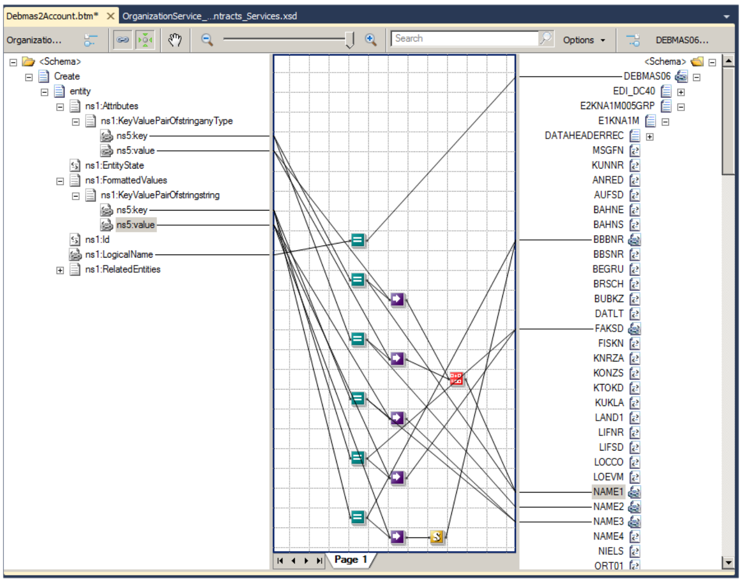
Now let’s consider the generated schema for the entities in BizTalk Server:


Roedl.BizTalk Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team BizTalk Data Tools Wi
Debug
Toolbox
OrganizationService_...ntracts_Services.xsd

```
<Schema>
  Create
    entity
      ns1:Attributes
        ns1:KeyValuePairOfstringanyType
          ns5:key
          ns5:value
      ns1:EntityState
      ns1:FormattedValues
        ns1:KeyValuePairOfstringstring
          ns5:key
          ns5:value
      ns1:Id
      ns1:LogicalName
      ns1:RelatedEntities
        ns1:KeyValuePairOfRelationshipEntityCollectionX_PsK4FkN
          ns5:key
            ns1:PrimaryEntityRole
            ns1:SchemaName
          ns5:value
            ns1:Entities
              ns1:Entity
            ns1:EntityName
            ns1:MinActiveRowVersion
            ns1:MoreRecords
            ns1:PagingCookie
            ns1:TotalRecordCount
            ns1:TotalRecordCountLimitExceeded
  CreateResponse
    CreateResult
```

The schema „Create“ is completely generic and independent from the several entities. The name of the entity to be created has to be entered in the element „LogicalName“. The individual attributes are filled through key/value pairs (e.g. the elements KeyValuePairOfstringstring/key and KeyValuePairOfstringstring/value).

Because of this generic approach, the individual attributes of the several entities do not appear obviously. In managed code, such as .Net assemblies, this approach may be useful. So let’s take a look at what is happening when we are trying to map such a schema:



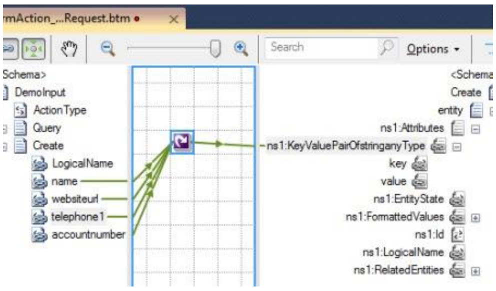
Since the individual attributes of an entity are no longer visible, all content from the key/value pairs have to be extracted. The BizTalk developer needs to know all the attributes and has to enter them into the appropriate functoids.

When reading from the messages it is still practicable. But if you need to fill the messages with the key/value pairs, the complexity increases significantly.

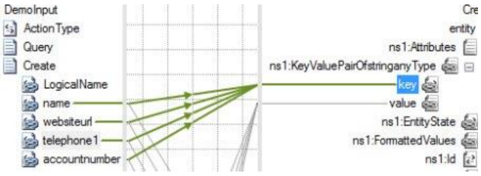
Based on Richard Seroter’s blog, who has dealt extensively with the MS CRM WCF service and how to use it in MS BizTalk, we collected some key points:

(more at: http://seroter.wordpress.com/2011/02/10/the-good-bad-and-ugly-of-integrating-dynamics-crm-2011-and-biztalk-server-2010/)

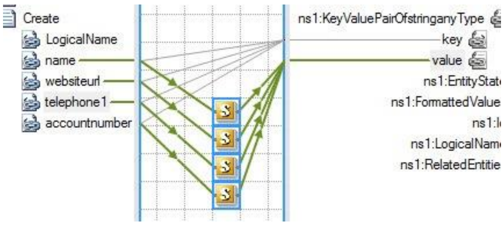
1. First you have to create a loop for each attribute (property of an entity) on the KeyValuePairOfstringanyType node, because a node must be created for each field.



2. Then the fields can be mapped on the key element.



3. When we trying to map the values directly, we will get a serialization exception at runtime, because the data type of the values is “xsd:anytype” and it is impossible to map values of this data type directly to an attribute. So we have to use a script functoid to convert the values to the appropriated data types.



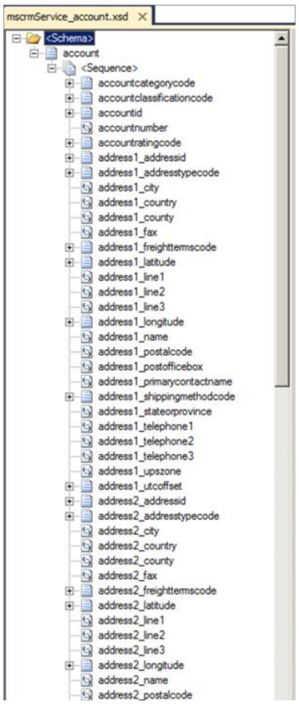
4. The script code looks like this:

```
<xsl:template name="SetNameValue">
<xsl:param name="param1" />
<value xmlns="http://schemas.datacontract.org/2004/07/System.Collections.Generic">
   <xsl:attribute name="xsi:type">
      <xsl:value-of select="'xs:string'" />
   </xsl:attribute>
   <xsl:value-of select="$param1" />
</value>
</xsl:template>
```

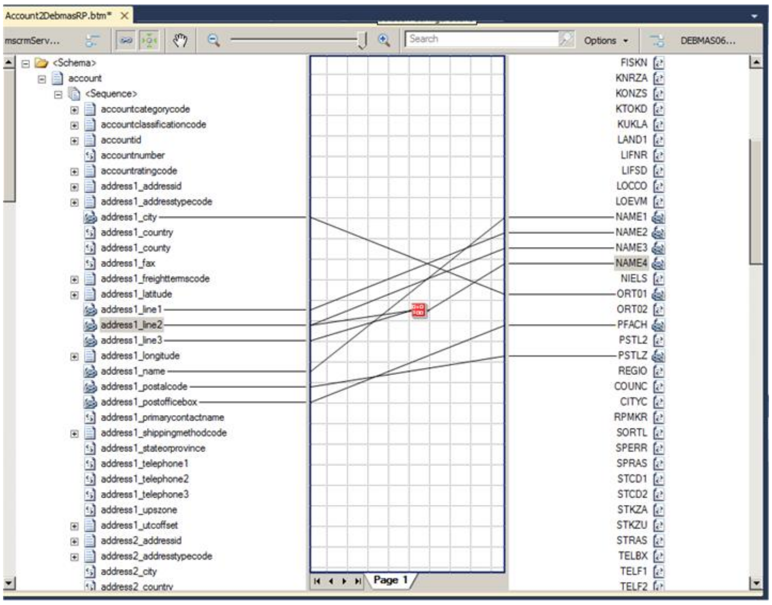
This procedure is highly inefficient.
Therefore, the *R&P MS-CRM WCF Adapter for MS BizTalk* works just like the *R&P MS-CRM WS Adapter for MS BizTalk*. Namely, with typed schemas in which the attributes of the several entities are clearly defined and typed as well.

Another important aspect is the complete compatibility of the new adapter with the old adapter. If you are going to switch your BizTalk Server integration from Microsoft Dynamics CRM 3.0 or 4.0 to Microsoft Dynamics CRM 2013/ 2015, you only need to install the new adapter and switch the ports from the old to the new adapter. The solution will run without any restrictions.

When using the new adapter, a schema of an account may look like this:



Accordingly, the mapping will look like this and not like the mapping created with the original WCF service:



Because the data types of the attributes are determined at designtime, no problems occur during serialization at runtime.

The *R&P MS CRM WCF Adapter for MS BizTalk* will only create the schemas that you actually need.

You can easily add the schemas to your project in Visual Studio in the same way as you have done in our WS adapter. We will discuss this point later in detail.

For each entity a schema will be created which contains only the attributes of the several entities. These individual schemas are well-arranged and fast. Furthermore, the communication with MS CRM has been simplified. There will be no special schemas for request and response required anymore, but everything is focused on the entity you want to work with.

Example Account:

	Schema for Request	Schema for Response
Insert	**account**	**account**
Update	**account**	**account**
Delete	**account**	**account**
Retrieve	**account**	**account**
Fetch	**fetch**	**accountList**
Upsert	**accountCustomAction**	**account**

Security

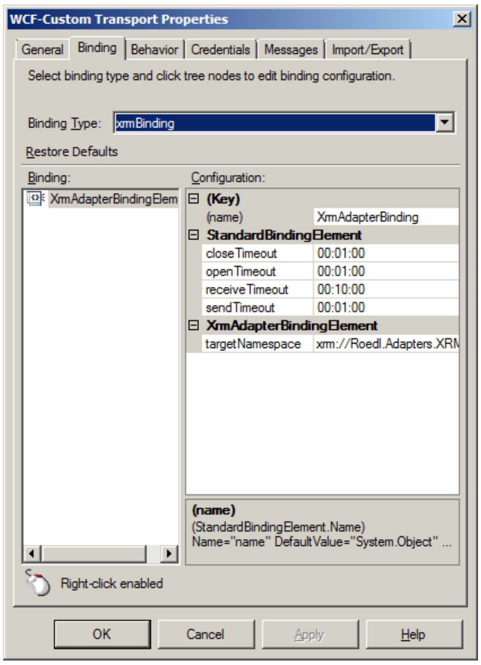
The *R&P MS CRM WCF Adapter for MS BizTalk* communicates directly with the original Microsoft Dynamics CRM 2013/ 2015 WCF service interface.
The same security rules take effect as if you were working with the WCF service itself.

Configuration of the *R&P MS CRM WCF Adapter for MS BizTalk*

Installation

With the distribution of the adapter you will receive an executable installation file.
You can choose between the 32bit and the 64bit installation, which depends on the Windows Server you use for your BizTalk integrations.
For installing the adapter, administrative rights are necessary.
After completing the installation, a new binding for the "WCF-Custom" adapter named "xrmBinding" appears. Because the adapter is based on the Microsoft xRM Framework and provides Microsoft Dynamics CRM 2013/ 2015 functions as well as Microsoft xRM functions, the binding is called "xrmBinding".

You can use the adapter to, e.g., create, modify or delete OptionSet values and publish customizations.
You will even be able to create, modify or delete custom entities and attributes.



Select binding type and click tree nodes to edit binding configuration.

Binding Type: xrmBinding

XrmAdapterBindingElem

(Key)	
(name)	XrmAdapterBinding
StandardBindingElement	
closeTimeout	00:01:00
openTimeout	00:01:00
receiveTimeout	00:10:00
sendTimeout	00:01:00
XrmAdapterBindingElement	
targetNamespace	xrm://Roedl.Adapters.XRM

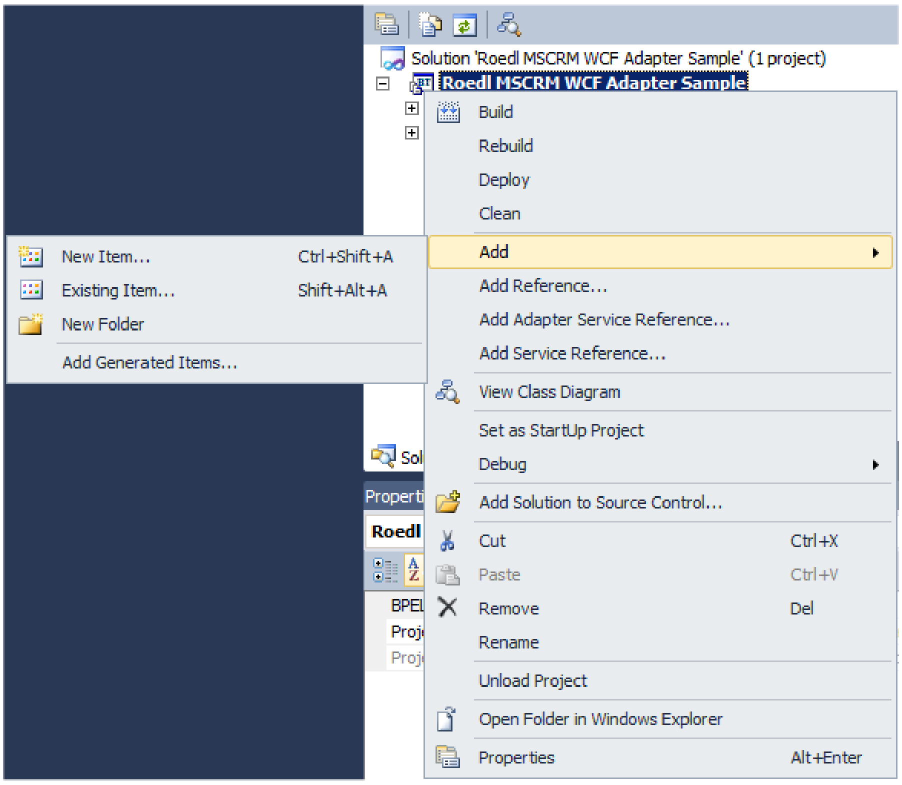
(name)
(StandardBindingElement.Name)
Name="name" DefaultValue="System.Object" ...

Working with the R&P MS CRM WCF Adapter for MS BizTalk

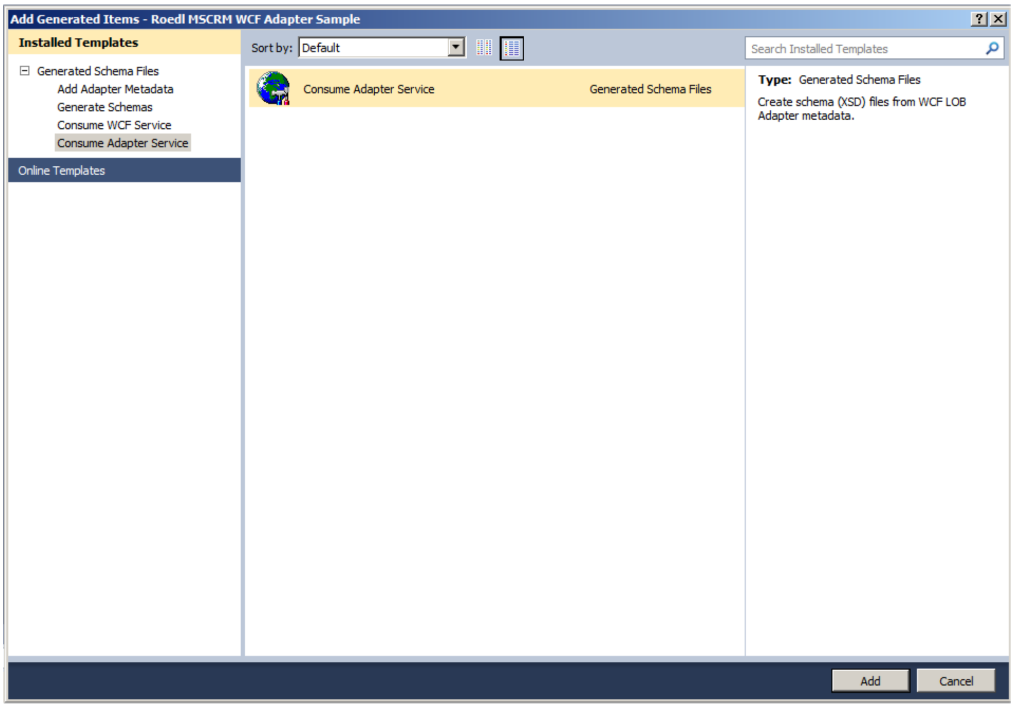
Working with the *R&P MS CRM WCF Adapter for MS BizTalk* is divided into two workspaces. On the one hand send ports will be defined that are physically linked with the adapter. We call this workspace **Runtime Customization**.
On the other hand schemas will be created in Visual Studio, e.g. account schema. These schemas are needed to send and receive messages. We call this workspace **Designtime Customization**.

Working with the adapter in Visual Studio 2010 (Designtime)

1. To start the adapter wizard, right-click on your project and select ***Add*** and ***Add Generated Items***.



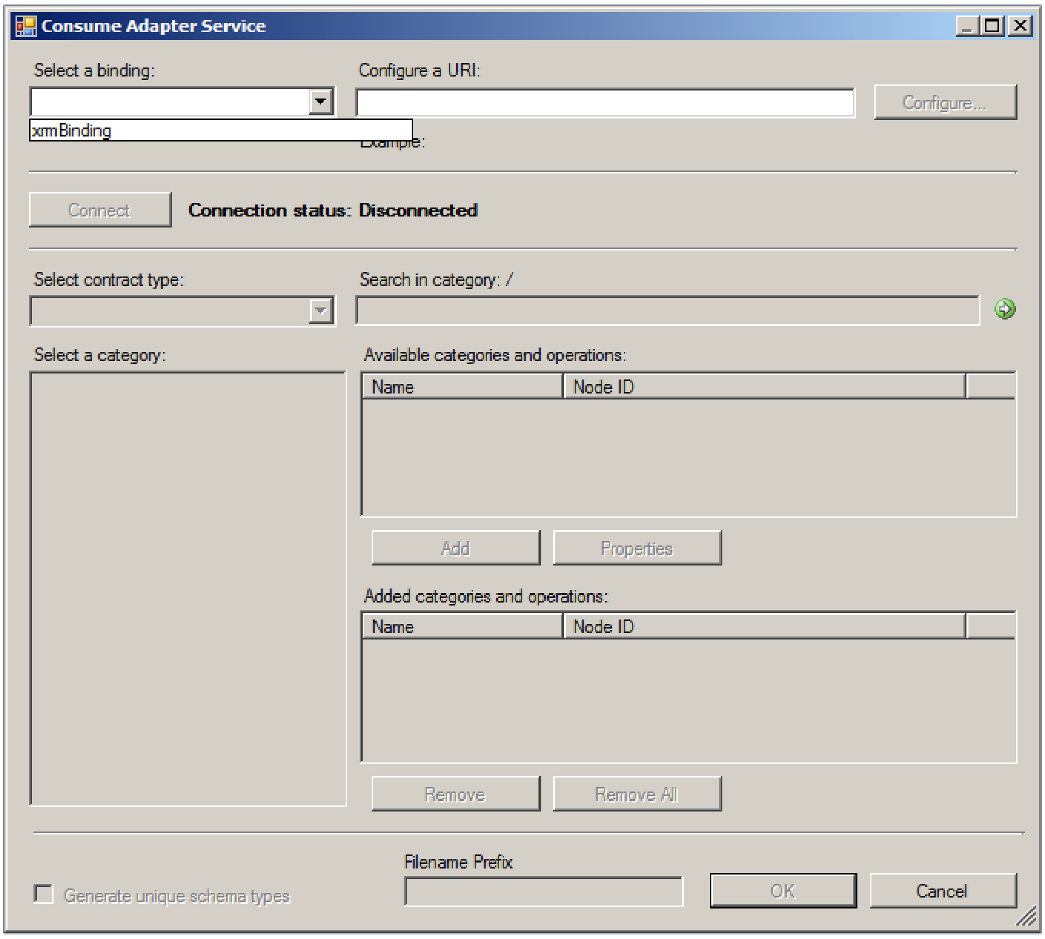
2. In the following screen, select ***Consume Adapter Service*** and click ***Add***.



Type: Generated Schema Files

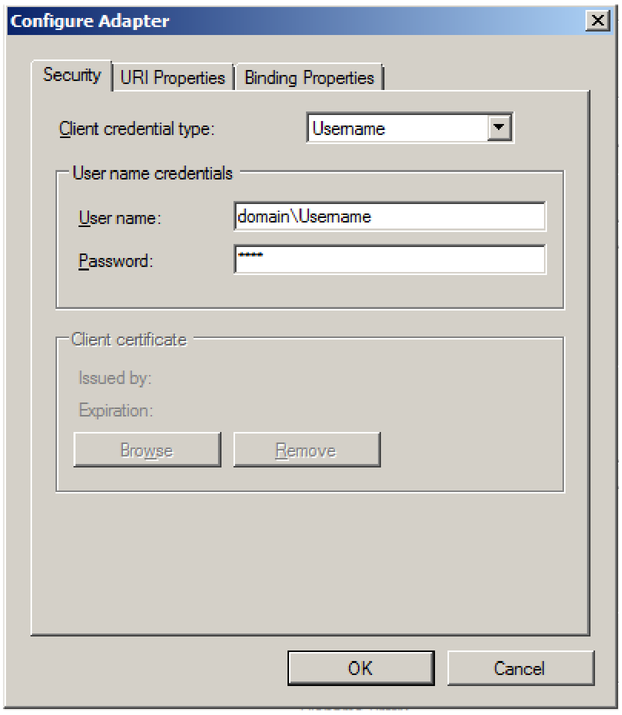
Create schema (XSD) files from WCF LOB Adapter metadata.

3. Now choose the binding and click ***Configure…***



Connection status: Disconnected

4. Here, you can enter the configuration for the Microsoft Dynamics CRM 2013/ 2015 system you want to connect with. On the tab “Security” you have to enter the credentials which are used to authenticate against Microsoft Dynamics CRM 2013/ 2015.



Configure Adapter

Security | URI Properties | Binding Properties

Client credential type: Username

User name credentials

User name: domain\Username

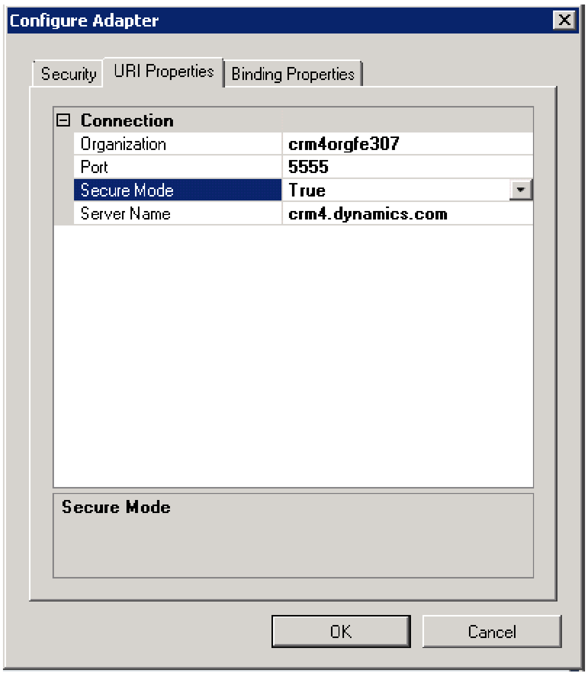
Password: ****

Client certificate

Issued by:

Expiration:

5. On the tab “URI Properties” you have to select the **Authentication Type**. Choose **Active Directory** if you are going to connect to an on-premise system, which uses the active directory to authenticate users. Choose **Live ID** if the target system is hosted in the Cloud and uses Windows Live ID for authentication. Furthermore, the organization, the port, transport security (http or https) and the server name have to be entered.



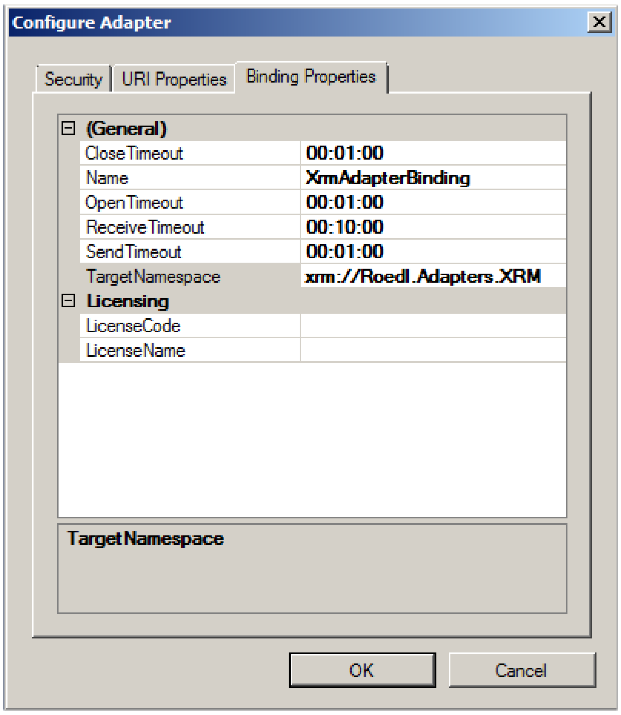
Configure Adapter

Security | URI Properties | Binding Properties

Connection	
Organization	crm4orgfe307
Port	5555
Secure Mode	True
Server Name	crm4.dynamics.com

Secure Mode

On the tab “Binding Properties” you can choose some general timeout options and the target namespace. In the section **licensing** you have to enter the license code and license name you have received upon your purchase.



(General)

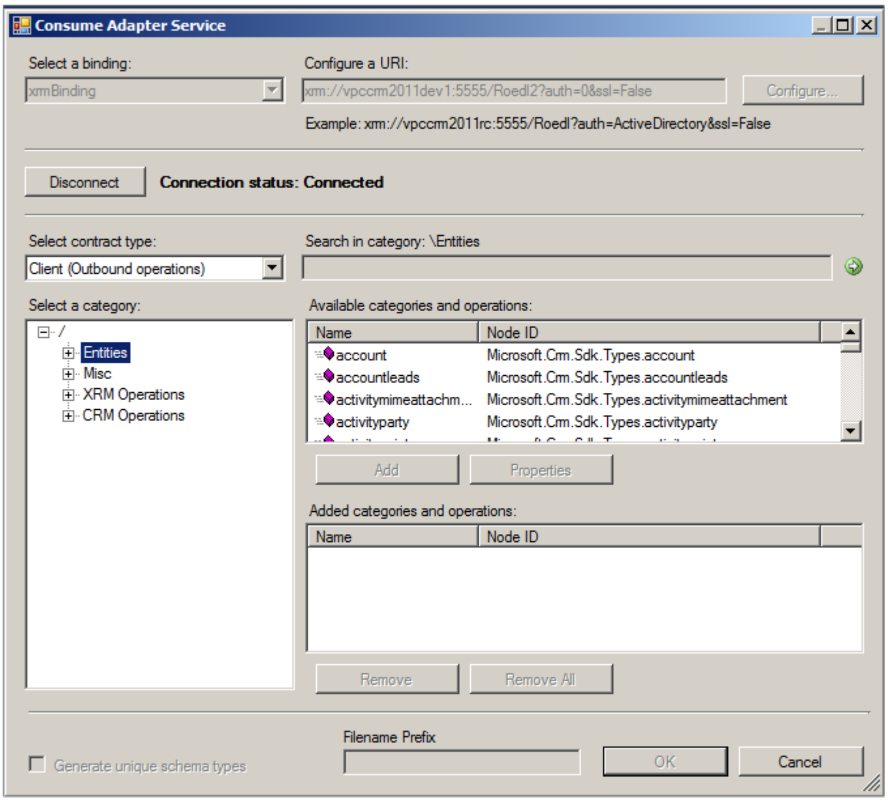
CloseTimeout	00:01:00
Name	XrmAdapterBinding
OpenTimeout	00:01:00
ReceiveTimeout	00:10:00
SendTimeout	00:01:00
TargetNamespace	xrm://Roedl.Adapters.XRM

Licensing

LicenseCode	
LicenseName	

TargetNamespace

6. Click Ok to save the configuration and close the window. If you do not wish to define your own target namespace, the adapter will use the default target namespace “xrm://Roedl.Adapters.XRM”. Of course, you can make changes to this setting at any time. By clicking Connect, the adapter authenticates against Microsoft Dynamics CRM 2013/ 2015, collects some further information from the system and finally provides you a set of categories to work with.



Select a binding: xrmBinding

Configure a URI: xrm://vpccrm2011dev1:5555/Roedl2?auth=0&ssl=False

Example: xrm://vpccrm2011rc:5555/Roedl?auth=ActiveDirectory&ssl=False

Connection status: Connected

Select contract type: Client (Outbound operations)

Search in category: \Entities

Select a category:

- /
  - Entities
  - Misc
  - XRM Operations
  - CRM Operations

Available categories and operations:

Name	Node ID
account	Microsoft.Crm.Sdk.Types.account
accountleads	Microsoft.Crm.Sdk.Types.accountleads
activitymimeattachm...	Microsoft.Crm.Sdk.Types.activitymimeattachment
activityparty	Microsoft.Crm.Sdk.Types.activityparty

Added categories and operations:

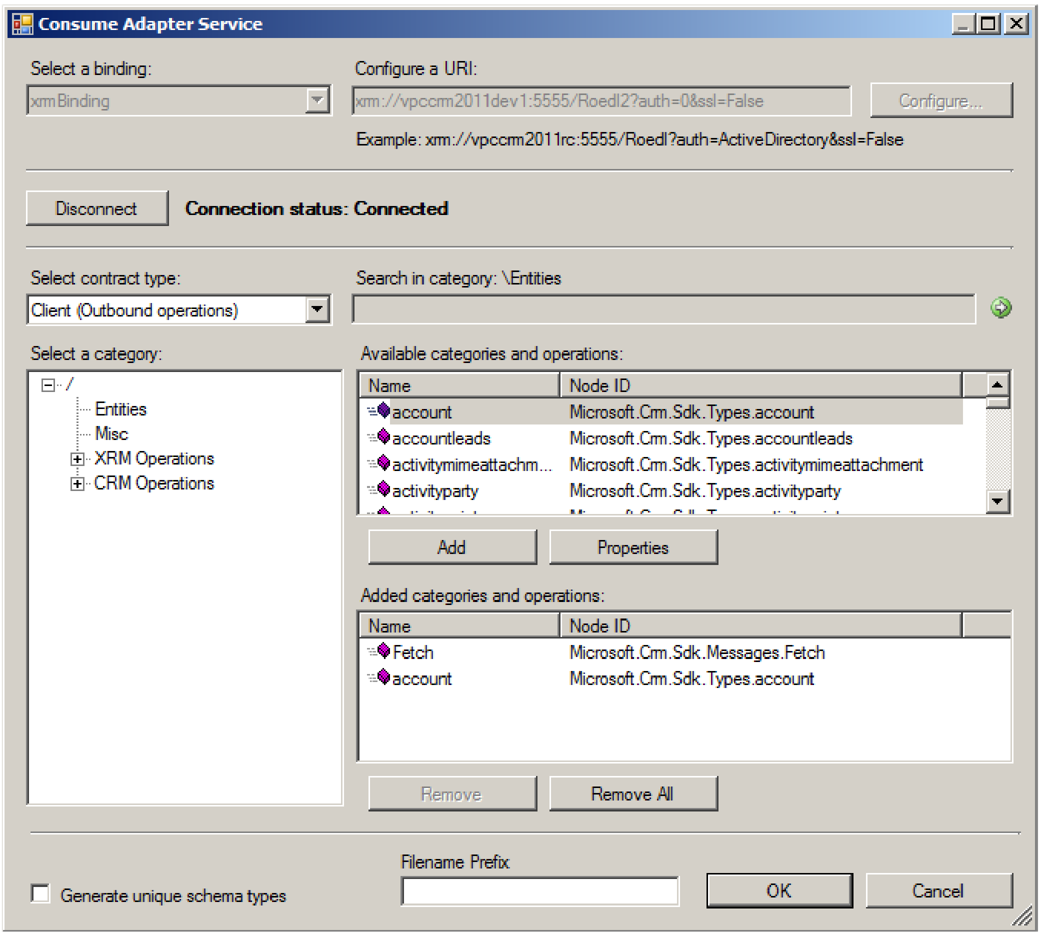
Name	Node ID

□ Generate unique schema types

Filename Prefix

7. In the treeview, four groups appear
   - Entities (contains all entities found in the target system, even custom entities)
   - Misc (special functions such as Fetch)
   - XRM Operations (all possible WCF operations of the XRM object model, especially metadata operations)
   - CRM Operations (all possible WCF operations of the CRM object model, such as assign owner)

8. Select the desired entities and functions and add them with the Add button.



Consume Adapter Service

Select a binding: xrmBinding

Configure a URI: xrm://vpccrm2011dev1:5555/Roedl2?auth=0&ssl=False

Example: xrm://vpccrm2011rc:5555/Roedl?auth=ActiveDirectory&ssl=False

Connection status: Connected

Select contract type: Client (Outbound operations)

Search in category: \Entities

Select a category:

- /
  - Entities
  - Misc
  - XRM Operations
  - CRM Operations

Available categories and operations:

Name	Node ID
account	Microsoft.Crm.Sdk.Types.account
accountleads	Microsoft.Crm.Sdk.Types.accountleads
activitymimeattachm...	Microsoft.Crm.Sdk.Types.activitymimeattachment
activityparty	Microsoft.Crm.Sdk.Types.activityparty

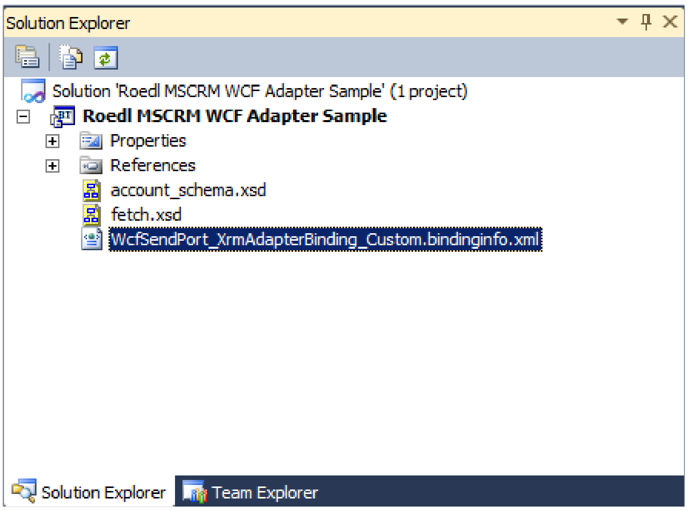
Added categories and operations:

Name	Node ID
Fetch	Microsoft.Crm.Sdk.Messages.Fetch
account	Microsoft.Crm.Sdk.Types.account

☐ Generate unique schema types

Filename Prefix

9. Click OK to close the window and generate the chosen schemas. The solution explorer may appear like this if you have selected the Fetch operation and the account entity:

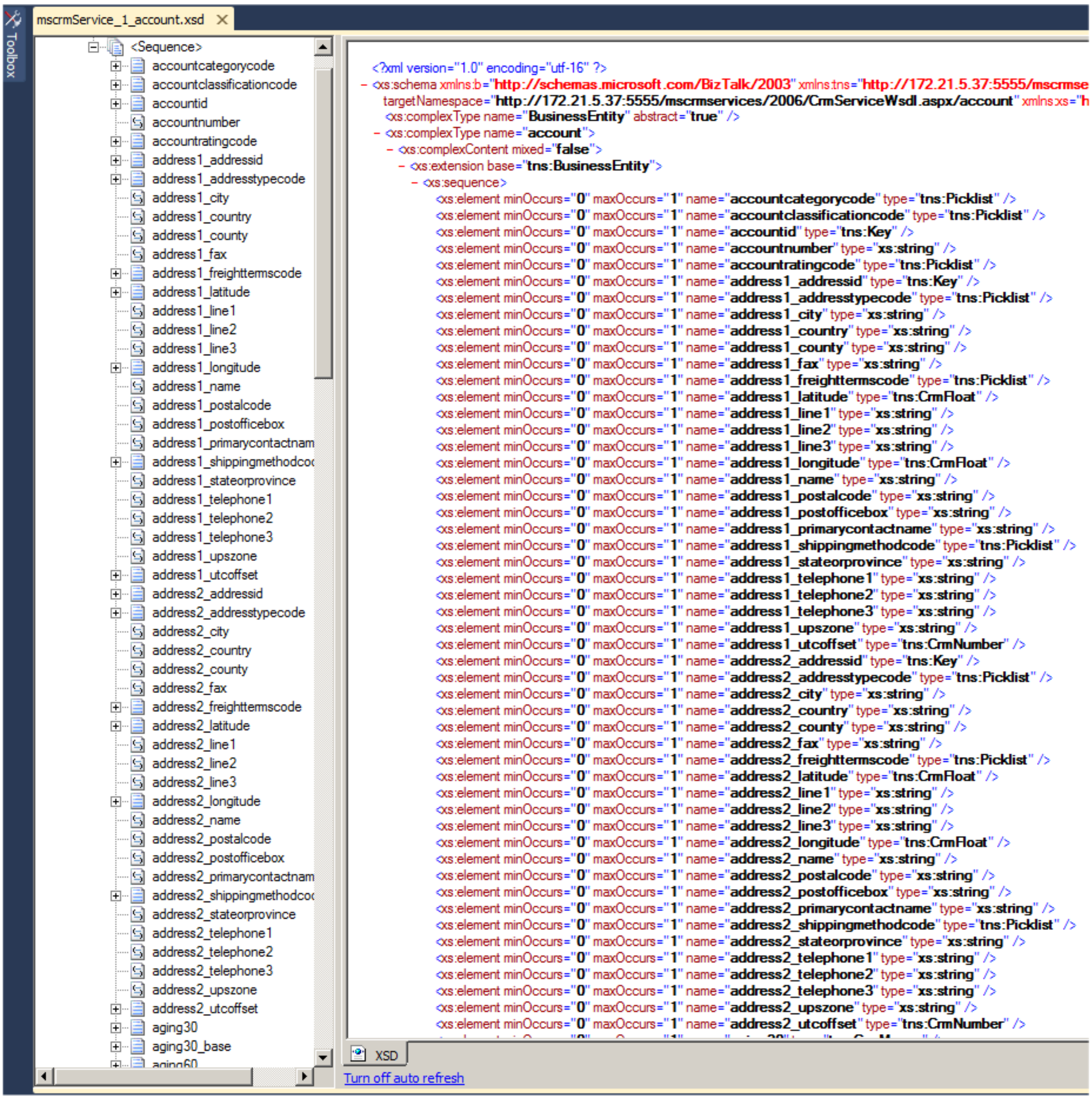

Solution Explorer
Solution 'Roedl MSCRM WCF Adapter Sample' (1 project)
Roedl MSCRM WCF Adapter Sample
Properties
References
account_schema.xsd
fetch.xsd
WcfSendPort_XrmAdapterBinding_Custom.bindinginfo.xml
Solution Explorer
Team Explorer

10. If other files are added, you may already use the beta version of the adapter. Feel free to delete the unnecessary files.

11. Use the generated file named WcfSendPort_Adaptername_Custom.bindinginfo.xml to easily create and configure a send port, without entering the whole configuration in the BizTalk Administration Console again.

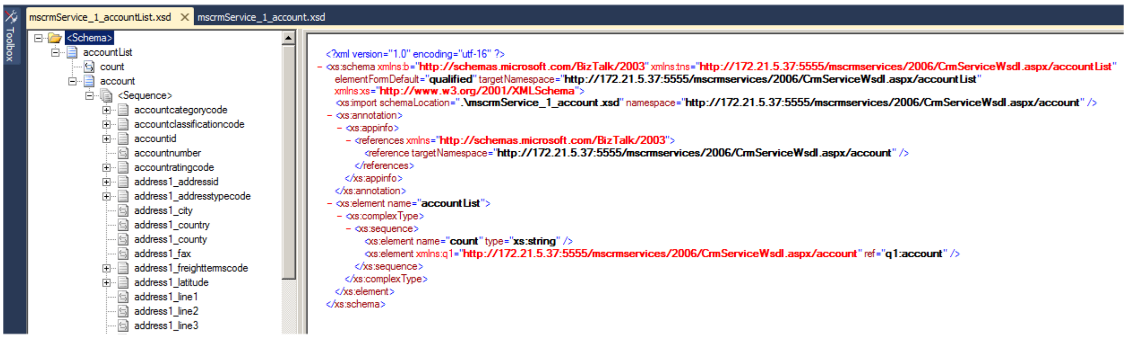
⚠ **Attention:**
In the XSD file of an entity, a schema for the entity itself and the appropriated entity collection will be generated. The name of this collection is a combination of the entity name and the word *List*. For example, by generating the schemas for the entity “**Account**” a XSD file named *entity_Account.xsd* containing the schemas *Account* and *AccountList* will be created. The collection schema will be needed for the Fetch response of the several entities.

Sample schema for an entity (**Account**):



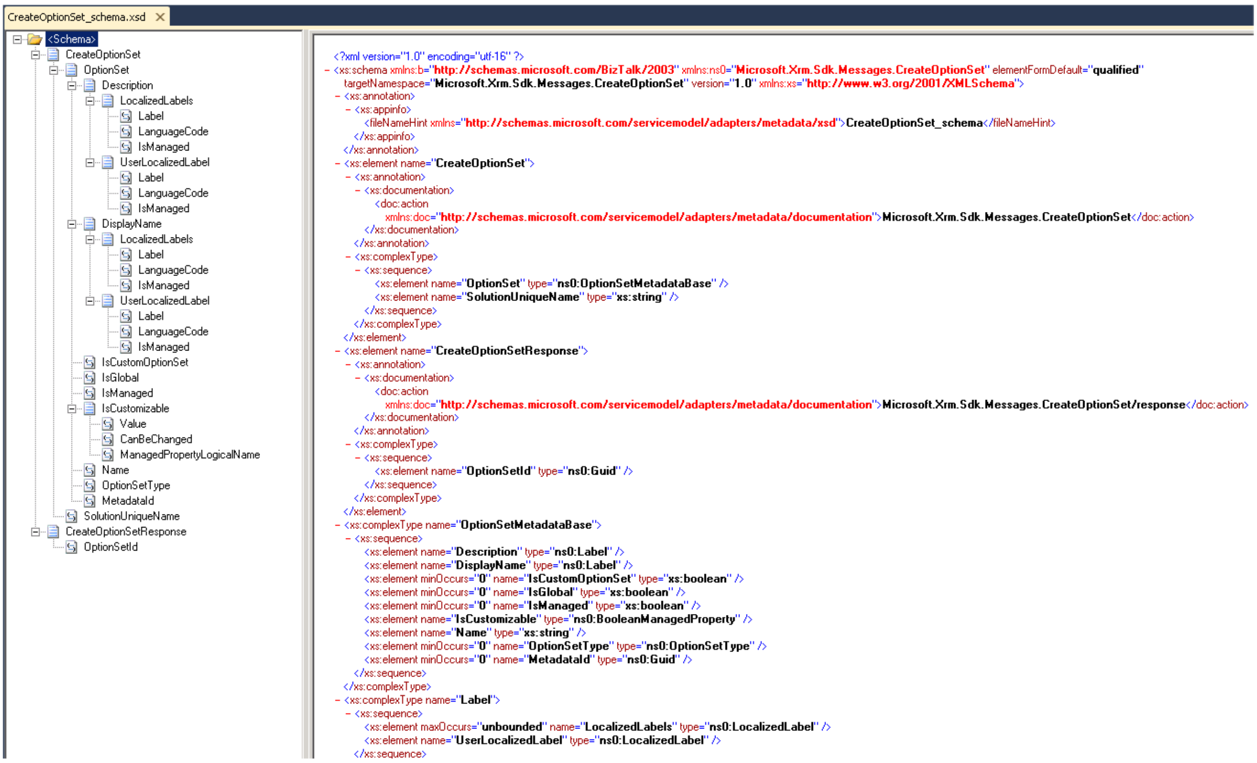
```
  <?xml version="1.0" encoding="utf-16" ?>
- <xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003" xmlns:tns="http://172.21.5.37:5555/mscrmse
   targetNamespace="http://172.21.5.37:5555/mscrmservices/2006/CrmServiceWsdl.aspx/account" xmlns:xs="h
   <xs:complexType name="BusinessEntity" abstract="true" />
 - <xs:complexType name="account">
  - <xs:complexContent mixed="false">
   - <xs:extension base="tns:BusinessEntity">
    - <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="accountcategorycode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="accountclassificationcode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="accountid" type="tns:Key" />
        <xs:element minOccurs="0" maxOccurs="1" name="accountnumber" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="accountratingcode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_addressid" type="tns:Key" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_addresstypecode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_city" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_country" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_county" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_fax" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_freighttermscode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_latitude" type="tns:CrmFloat" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_line1" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_line2" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_line3" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_longitude" type="tns:CrmFloat" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_name" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_postalcode" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_postofficebox" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_primarycontactname" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_shippingmethodcode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_stateorprovince" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_telephone1" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_telephone2" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_telephone3" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_upszone" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address1_utcoffset" type="tns:CrmNumber" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_addressid" type="tns:Key" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_addresstypecode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_city" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_country" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_county" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_fax" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_freighttermscode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_latitude" type="tns:CrmFloat" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_line1" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_line2" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_line3" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_longitude" type="tns:CrmFloat" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_name" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_postalcode" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_postofficebox" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_primarycontactname" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_shippingmethodcode" type="tns:Picklist" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_stateorprovince" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_telephone1" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_telephone2" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_telephone3" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_upszone" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="address2_utcoffset" type="tns:CrmNumber" />
```

Sample schema for an entity collection (**AccountList**):



```
  <?xml version="1.0" encoding="utf-16" ?>
- <xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003" xmlns:tns="http://172.21.5.37:5555/mscrmservices/2006/CrmServiceWsdl.aspx/accountList"
   elementFormDefault="qualified" targetNamespace="http://172.21.5.37:5555/mscrmservices/2006/CrmServiceWsdl.aspx/accountList"
   xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:import schemaLocation=".\mscrmService_1_account.xsd" namespace="http://172.21.5.37:5555/mscrmservices/2006/CrmServiceWsdl.aspx/account" />
 - <xs:annotation>
  - <xs:appinfo>
   - <references xmlns="http://schemas.microsoft.com/BizTalk/2003">
       <reference targetNamespace="http://172.21.5.37:5555/mscrmservices/2006/CrmServiceWsdl.aspx/account" />
     </references>
    </xs:appinfo>
   </xs:annotation>
 - <xs:element name="accountList">
  - <xs:complexType>
   - <xs:sequence>
       <xs:element name="count" type="xs:string" />
       <xs:element xmlns:q1="http://172.21.5.37:5555/mscrmservices/2006/CrmServiceWsdl.aspx/account" ref="q1:account" />
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:schema>
```

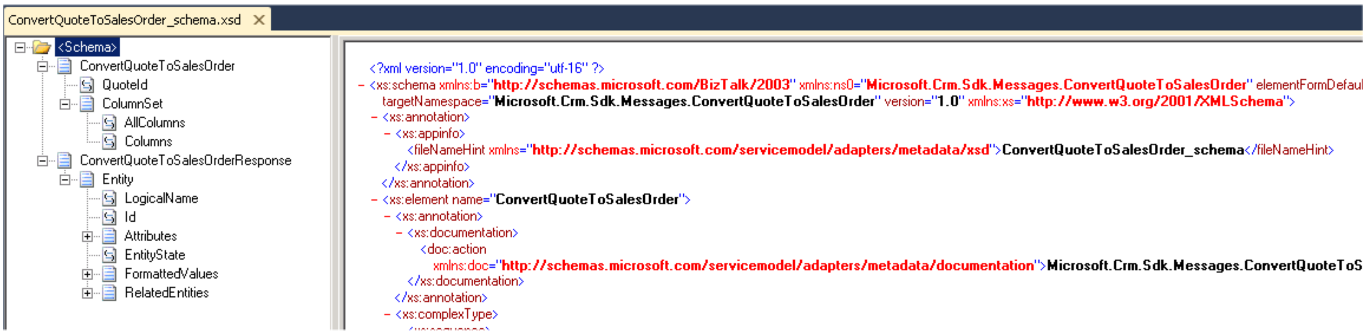
Sample schema for an XRM function (**CreateOptionSet**)



```
<Schema>
  CreateOptionSet
    OptionSet
      Description
        LocalizedLabels
          Label
          LanguageCode
          IsManaged
        UserLocalizedLabel
          Label
          LanguageCode
          IsManaged
      DisplayName
        LocalizedLabels
          Label
          LanguageCode
          IsManaged
        UserLocalizedLabel
          Label
          LanguageCode
          IsManaged
      IsCustomOptionSet
      IsGlobal
      IsManaged
      IsCustomizable
        Value
        CanBeChanged
        ManagedPropertyLogicalName
      Name
      OptionSetType
      MetadataId
    SolutionUniqueName
  CreateOptionSetResponse
    OptionSetId
```

```
  <?xml version="1.0" encoding="utf-16" ?>
- <xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003" xmlns:ns0="Microsoft.Xrm.Sdk.Messages.CreateOptionSet" elementFormDefault="qualified"
    targetNamespace="Microsoft.Xrm.Sdk.Messages.CreateOptionSet" version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  - <xs:annotation>
    - <xs:appinfo>
        <fileNameHint xmlns="http://schemas.microsoft.com/servicemodel/adapters/metadata/xsd">CreateOptionSet_schema</fileNameHint>
      </xs:appinfo>
    </xs:annotation>
  - <xs:element name="CreateOptionSet">
    - <xs:annotation>
      - <xs:documentation>
          <doc:action
            xmlns:doc="http://schemas.microsoft.com/servicemodel/adapters/metadata/documentation">Microsoft.Xrm.Sdk.Messages.CreateOptionSet</doc:action>
        </xs:documentation>
      </xs:annotation>
    - <xs:complexType>
      - <xs:sequence>
          <xs:element name="OptionSet" type="ns0:OptionSetMetadataBase" />
          <xs:element name="SolutionUniqueName" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  - <xs:element name="CreateOptionSetResponse">
    - <xs:annotation>
      - <xs:documentation>
          <doc:action
            xmlns:doc="http://schemas.microsoft.com/servicemodel/adapters/metadata/documentation">Microsoft.Xrm.Sdk.Messages.CreateOptionSet/response</doc:action>
        </xs:documentation>
      </xs:annotation>
    - <xs:complexType>
      - <xs:sequence>
          <xs:element name="OptionSetId" type="ns0:Guid" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  - <xs:complexType name="OptionSetMetadataBase">
    - <xs:sequence>
        <xs:element name="Description" type="ns0:Label" />
        <xs:element name="DisplayName" type="ns0:Label" />
        <xs:element minOccurs="0" name="IsCustomOptionSet" type="xs:boolean" />
        <xs:element minOccurs="0" name="IsGlobal" type="xs:boolean" />
        <xs:element minOccurs="0" name="IsManaged" type="xs:boolean" />
        <xs:element name="IsCustomizable" type="ns0:BooleanManagedProperty" />
        <xs:element name="Name" type="xs:string" />
        <xs:element minOccurs="0" name="OptionSetType" type="ns0:OptionSetType" />
        <xs:element minOccurs="0" name="MetadataId" type="ns0:Guid" />
      </xs:sequence>
    </xs:complexType>
  - <xs:complexType name="Label">
    - <xs:sequence>
        <xs:element maxOccurs="unbounded" name="LocalizedLabels" type="ns0:LocalizedLabel" />
        <xs:element name="UserLocalizedLabel" type="ns0:LocalizedLabel" />
      </xs:sequence>
```

Sample schema for a CRM function (**ConvertQuoteToSalesOrder**)



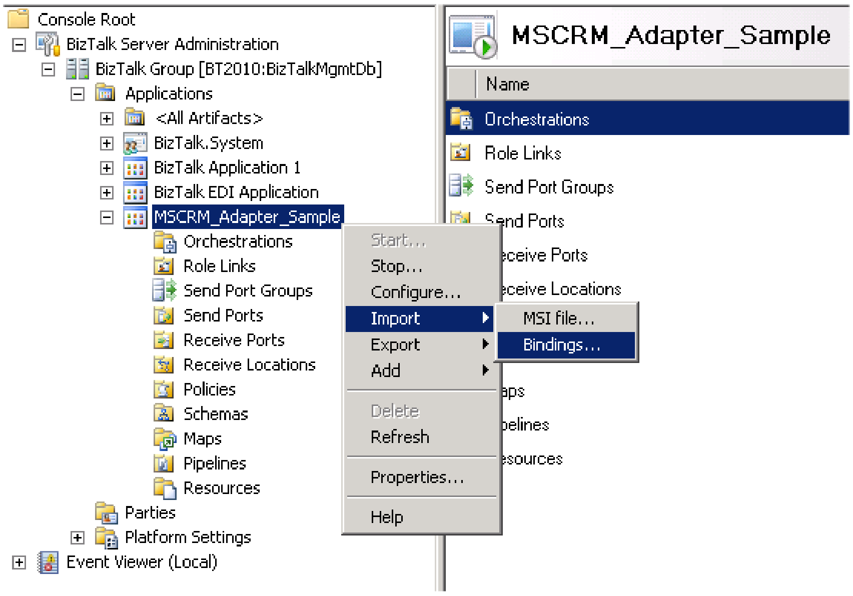
```
<Schema>
  ConvertQuoteToSalesOrder
    QuoteId
    ColumnSet
      AllColumns
      Columns
  ConvertQuoteToSalesOrderResponse
    Entity
      LogicalName
      Id
      Attributes
      EntityState
      FormattedValues
      RelatedEntities
```

```
  <?xml version="1.0" encoding="utf-16" ?>
- <xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003" xmlns:ns0="Microsoft.Crm.Sdk.Messages.ConvertQuoteToSalesOrder" elementFormDefaul
    targetNamespace="Microsoft.Crm.Sdk.Messages.ConvertQuoteToSalesOrder" version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  - <xs:annotation>
    - <xs:appinfo>
        <fileNameHint xmlns="http://schemas.microsoft.com/servicemodel/adapters/metadata/xsd">ConvertQuoteToSalesOrder_schema</fileNameHint>
      </xs:appinfo>
    </xs:annotation>
  - <xs:element name="ConvertQuoteToSalesOrder">
    - <xs:annotation>
      - <xs:documentation>
          <doc:action
            xmlns:doc="http://schemas.microsoft.com/servicemodel/adapters/metadata/documentation">Microsoft.Crm.Sdk.Messages.ConvertQuoteToS
        </xs:documentation>
      </xs:annotation>
    - <xs:complexType>
```

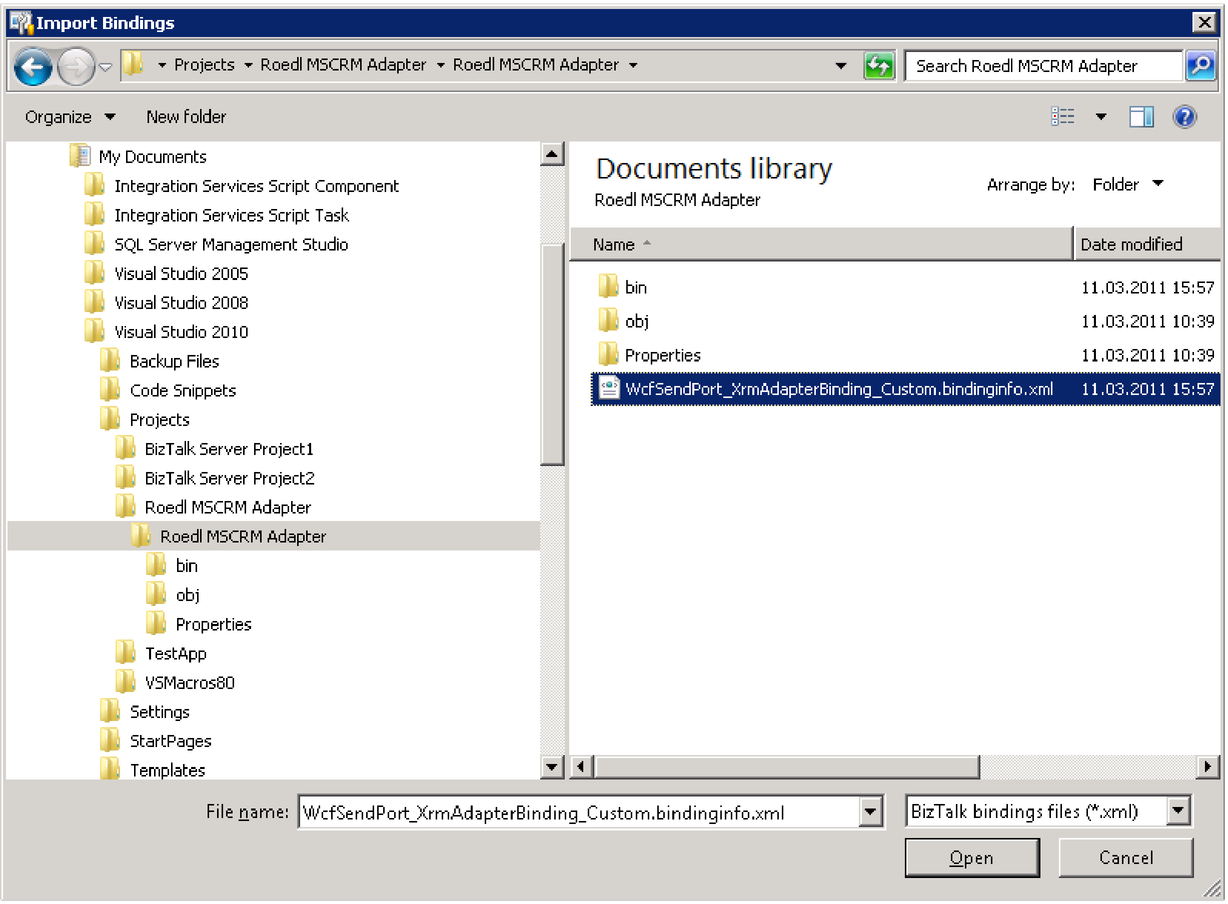
Configuring send ports for the adapter (Runtime)

1. When creating the schema in the designtime, an XML file is created and added to your project. Start the ***BizTalk Server Administration Console*** und navigate to your application, e.g. ***„MSCRM_Adapter_Sample“.***

2. Right-click on ***Send Ports***, select ***Import*** and click ***Bindings…***



3. A file selection window appears



Import Bindings

Documents library
Roedl MSCRM Adapter

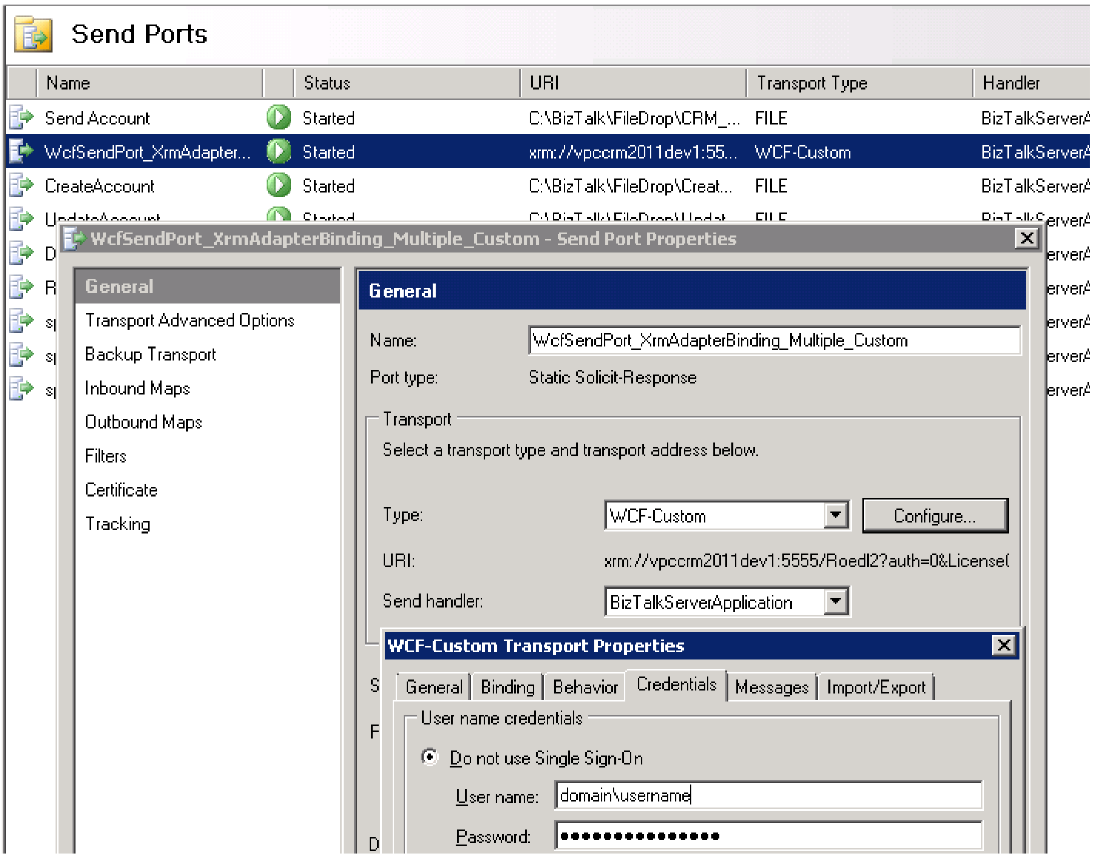
Name	Date modified
bin	11.03.2011 15:57
obj	11.03.2011 10:39
Properties	11.03.2011 10:39
WcfSendPort_XrmAdapterBinding_Custom.bindinginfo.xml	11.03.2011 15:57

File name: WcfSendPort_XrmAdapterBinding_Custom.bindinginfo.xml

BizTalk bindings files (*.xml)

Navigate to your VS Solution and select the appropriated XML file for the binding configuration. Confirm the import and the send receive port will be created automatically.

4. You must enter your credentials again, usually only the password, and the send port to Microsoft Dynamics CRM 2013/ 2015 is almost fully configured.



Send Ports

Name	Status	URI	Transport Type	Handler
Send Account	Started	C:\BizTalk\FileDrop\CRM_...	FILE	BizTalkServerA
WcfSendPort_XrmAdapter...	Started	xrm://vpccrm2011dev1:55...	WCF-Custom	BizTalkServerA
CreateAccount	Started	C:\BizTalk\FileDrop\Creat...	FILE	BizTalkServerA
UpdateAccount	Started	C:\BizTalk\FileDrop\Updat...	FILE	BizTalkServerA

WcfSendPort_XrmAdapterBinding_Multiple_Custom - Send Port Properties

General

Name: WcfSendPort_XrmAdapterBinding_Multiple_Custom

Port type: Static Solicit-Response

Transport

Select a transport type and transport address below.

Type: WCF-Custom

URI: xrm://vpccrm2011dev1:5555/Roedl2?auth=0&License(

Send handler: BizTalkServerApplication

WCF-Custom Transport Properties

User name credentials

● Do not use Single Sign-On

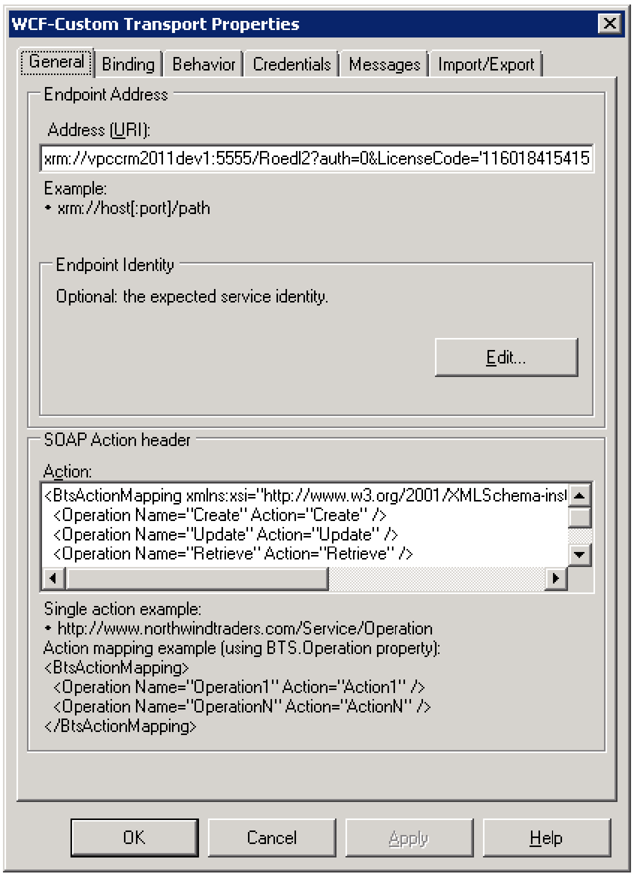
User name: domain\username

Password: ●●●●●●●●●●●●●●●

5. Enter the following XML string into the text field in the section SOAP Action Header:

```
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="Create" Action="Create" />
  <Operation Name="Update" Action="Update" />
  <Operation Name="Retrieve" Action="Retrieve" />
  <Operation Name="Fetch" Action="Fetch" />
  <Operation Name="Delete" Action="Delete" />
  <Operation Name="Upsert" Action="Upsert" />
</BtsActionMapping>
```

Attention:
The Upsert statement is a new functionality for much faster development!



Endpoint Address

Address (URI): xrm://vpccrm2011dev1:5555/Roedl2?auth=0&LicenseCode='116018415415

Example:
- xrm://host[:port]/path

Endpoint Identity

Optional: the expected service identity.

SOAP Action header

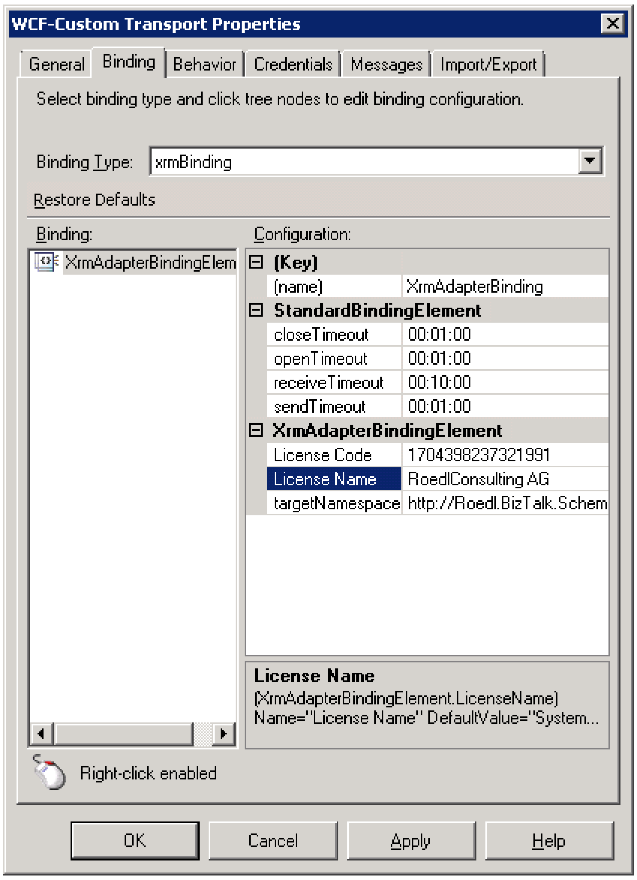
Action:

```
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-inst
  <Operation Name="Create" Action="Create" />
  <Operation Name="Update" Action="Update" />
  <Operation Name="Retrieve" Action="Retrieve" />
```

Single action example:
- http://www.northwindtraders.com/Service/Operation

Action mapping example (using BTS.Operation property):

```
<BtsActionMapping>
  <Operation Name="Operation1" Action="Action1" />
  <Operation Name="OperationN" Action="ActionN" />
</BtsActionMapping>
```



Select binding type and click tree nodes to edit binding configuration.

Binding Type: xrmBinding

Binding: XrmAdapterBindingElem

Configuration:

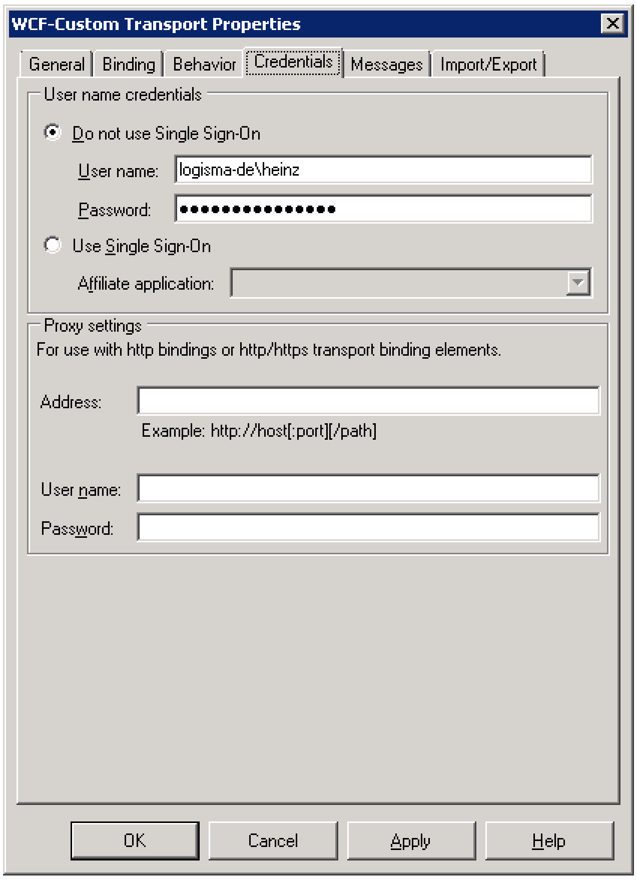
(Key)	
(name)	XrmAdapterBinding
StandardBindingElement	
closeTimeout	00:01:00
openTimeout	00:01:00
receiveTimeout	00:10:00
sendTimeout	00:01:00
XrmAdapterBindingElement	
License Code	1704398237321991
License Name	RoedlConsulting AG
targetNamespace	http://Roedl.BizTalk.Schem

License Name
(XrmAdapterBindingElement.LicenseName)
Name="License Name" DefaultValue="System...

Right-click enabled

6. Enter the License Name you've received upon your purchase.

7. Enter the License Code you've received upon your purchase.
8. You should take the default value for "Target Namespace", because your schemas are already generated with this namespace.



User name credentials

● Do not use Single Sign-On

User name: logisma-de\heinz

Password: •••••••••••••••

○ Use Single Sign-On

Affiliate application:

Proxy settings

For use with http bindings or http/https transport binding elements.

Address:

Example: http://host[:port][/path]

User name:

Password:

9. Enter user name and password on the "Credentials" tab. Depending on the CRM system it could be a domain account or a Windows Live ID account. If you are using a proxy, enter the data into the fields *Address*, *User name* and *Password*.

10. No modifications needed on the tabs "Messages" und "Import/Export".

11. Now the send port is ready for use.

Example: Call a CRM entity and process the response

In this example, an account record will be created and the response will be saved in an XML file.
Unlike the original Microsoft Dynamics CRM 2013/ 2015 WCF service, the adapter always works with entity objects (schemas). For both request and response.
This reduces the effort of the needed schemas and the user does not have to choose which attributes for the several entities are needed.

Prerequisites (Runtime)

Before starting the implementation of the example, some preparations have to be made:

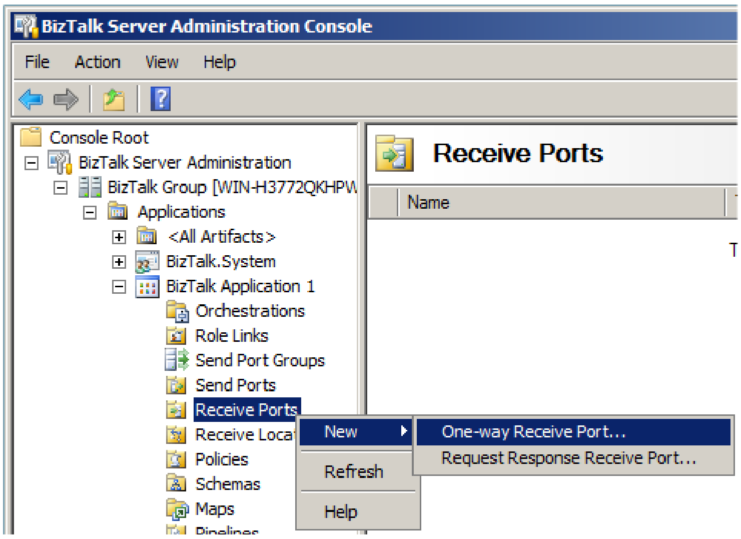
- Create the following folder structure on the BizTalk Server:
  c:\
  - □ BizTalk
    - □ Filedrop
      - □ CreateAccount
        - □ Request
        - □ Response

- We will need the following informations of the CRM system:
  - URL
  - Login credentials
  - Proxy, if necessary

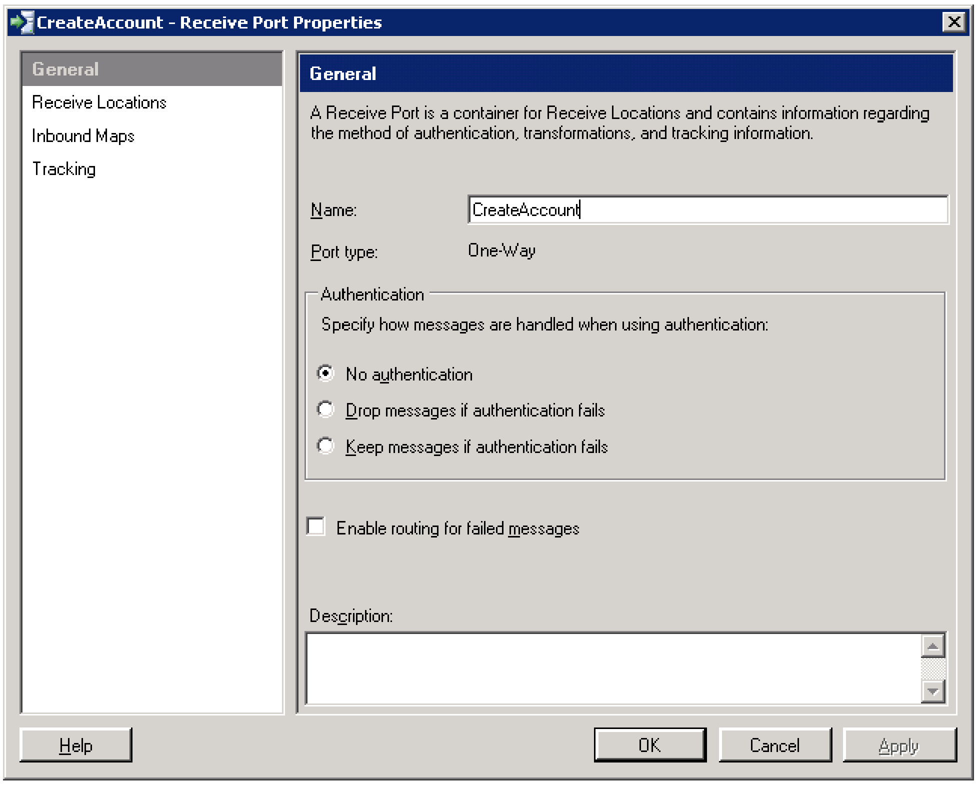
Step 1: Creating the receive port “Account“

1. Open the ***BizTalk Server Administration Console***

2. Right-click on ***Receive Ports,*** select ***New*** and click ***One-way Receive Port…***.



3. Enter **CreateAccount** in the *Name* field and click OK.



A Receive Port is a container for Receive Locations and contains information regarding the method of authentication, transformations, and tracking information.

Name: CreateAccount

Port type: One-Way

Authentication

Specify how messages are handled when using authentication:

● No authentication

○ Drop messages if authentication fails

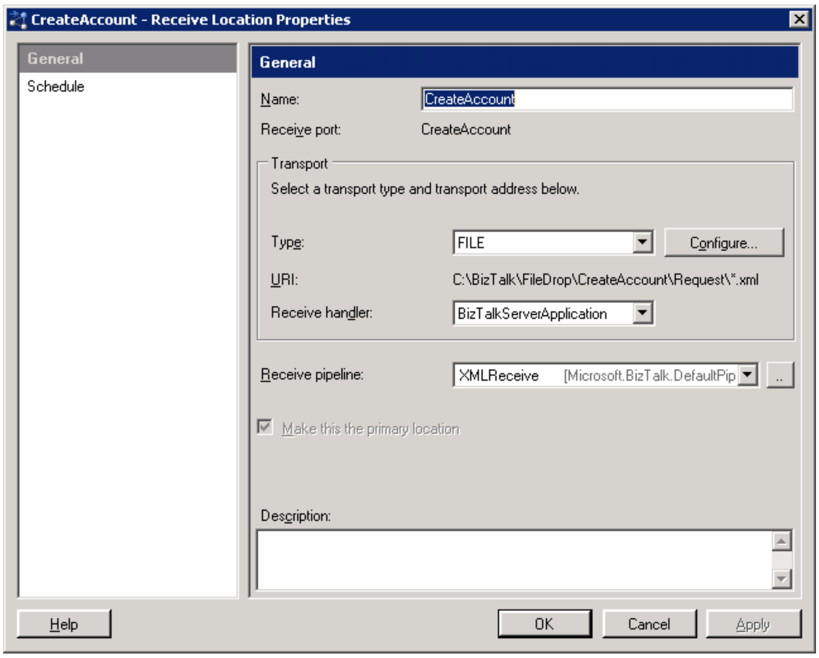
○ Keep messages if authentication fails

□ Enable routing for failed messages

Description:

4. Right-click on the created receive port, select ***New*** and click ***Receive Location…*** .

5. In the following window select **File** in the field ***Transport Type***.



CreateAccount - Receive Location Properties

General

Name: CreateAccount

Receive port: CreateAccount

Transport

Select a transport type and transport address below.

Type: FILE

URI: C:\BizTalk\FileDrop\CreateAccount\Request\*.xml

Receive handler: BizTalkServerApplication

Receive pipeline: XMLReceive [Microsoft.BizTalk.DefaultPip

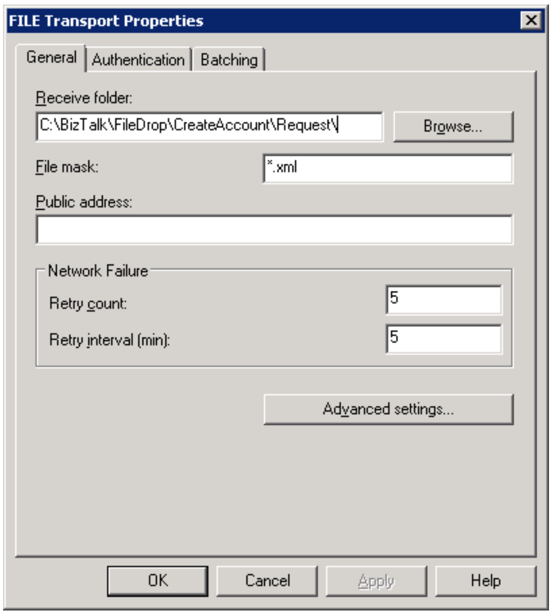
☑ Make this the primary location

Description:

6. Select your BizTalk application instance in the field ***Receive handler***. If you didn't install a BizTalk Cluster, you can only choose the main instance.

7. Select **XMLReceive** in the field ***Receive pipeline***.

8. Click Configure....



FILE Transport Properties

Receive folder: C:\BizTalk\FileDrop\CreateAccount\Request\

File mask: *.xml

Public address:

Network Failure

Retry count: 5

Retry interval (min): 5

9. Enter the folder “C:\BizTalk\FileDrop\CreateAccount\Request\” in the field ***Receive Folder.*** In the other fields, leave the default values.

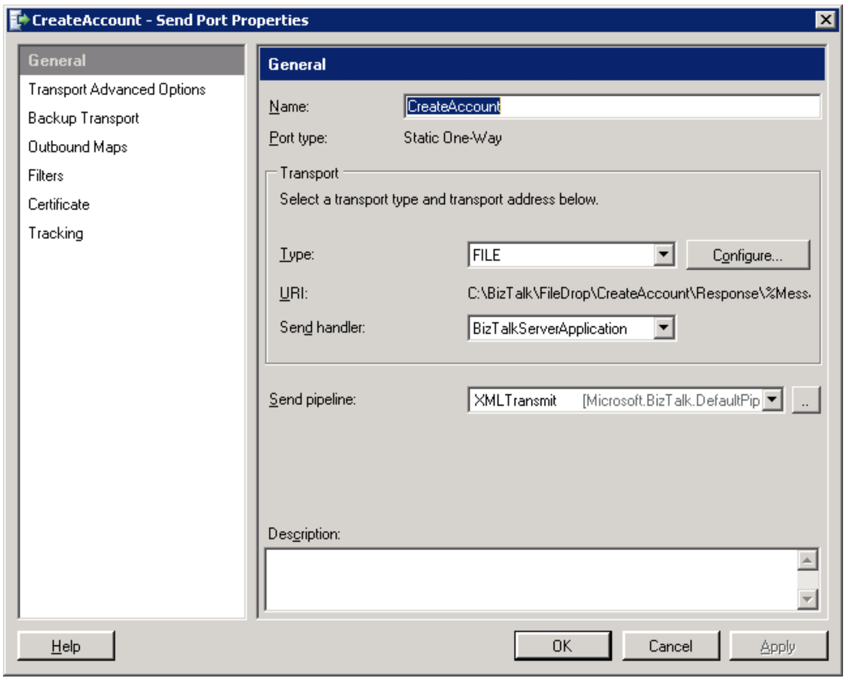
10. Click OK.

11. On the window ***Receive Location Properties*** click OK.

Step 2: Creating the send port “Account“

1. In the ***BizTalk Server Administration Console*** right-click on ***Send Ports***, select ***New*** and click ***Static One-way Send Port…*** .

2. In the field *Name* enter **CreateAccount**.



CreateAccount - Send Port Properties

General

Name: CreateAccount

Port type: Static One-Way

Transport

Select a transport type and transport address below.

Type: FILE

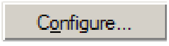
URI: C:\BizTalk\FileDrop\CreateAccount\Response\%Mess

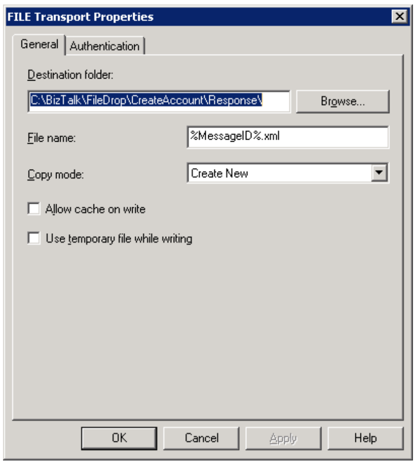
Send handler: BizTalkServerApplication

Send pipeline: XMLTransmit [Microsoft.BizTalk.DefaultPip

Description:

3. Select **File** in the field *Transport Type*.

4. Click .



FILE Transport Properties

Destination folder: C:\BizTalk\FileDrop\CreateAccount\Response\

File name: %MessageID%.xml

Copy mode: Create New

□ Allow cache on write

□ Use temporary file while writing

5. Enter the folder “C:\BizTalk\FileDrop\CreateAccount\Response\” in the field ***Destination Folder.*** In the other fields, leave the default values.

6. Click .

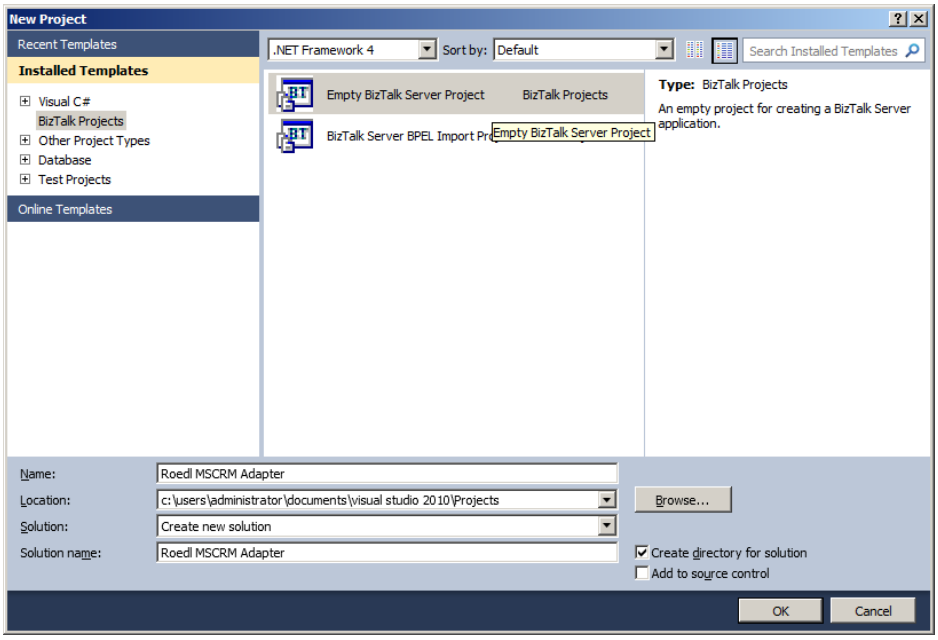
7. Select **XMLTransmit** in the field *Send Pipeline*.

8. Click .

Step 3: Creating the schema for „Account“

1. Open ***Microsoft Visual Studio .NET***

2. Create a new BizTalk project named **Roedl MSCRM Adapter**



Type: BizTalk Projects

An empty project for creating a BizTalk Server application.

Name: Roedl MSCRM Adapter

Location: c:\users\administrator\documents\visual studio 2010\Projects

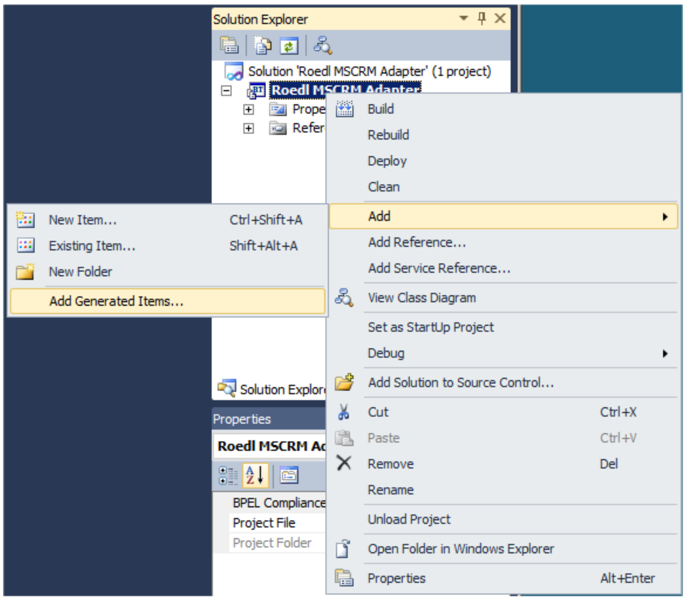
Solution: Create new solution

Solution name: Roedl MSCRM Adapter

☑ Create directory for solution

□ Add to source control

3. In the Solution Explorer right-click on your BizTalk project, select ***Add*** and click ***Add Generated Items…***



4. Proceed as described in the section <u>**Working with the adapter in Visual Studio 2010 (Designtime)**</u> and create the schema for the entity **Account** and the function **Fetch** (**Misc**).

5. ***R&P MS CRM WCF Adapter for MS BizTalk*** will create three files.

6. In the file “entity_Account.xsd” is one schema for the account (**Account)** and one schema for the account collection (**AccountList**). In the file “crm_Fetch.xsd” is the schema for the Fetch message in the file “crm_Fetch.xsd”. And an XML file containing the binding configuration for the send port.

Step 4: Creating the Orchestration

1. Create a new Orchestration named “CreateAccount”.

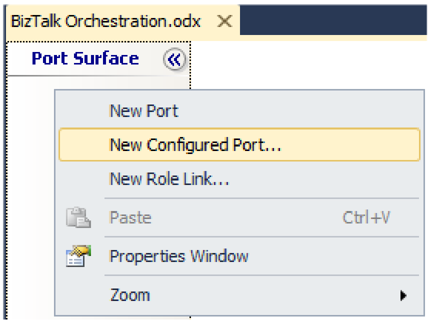
2. In the Orchestration Designer you will need three logical ports (in the Port Surface).

   - A ***One way Receive Port* “rpCreateAccount”**.

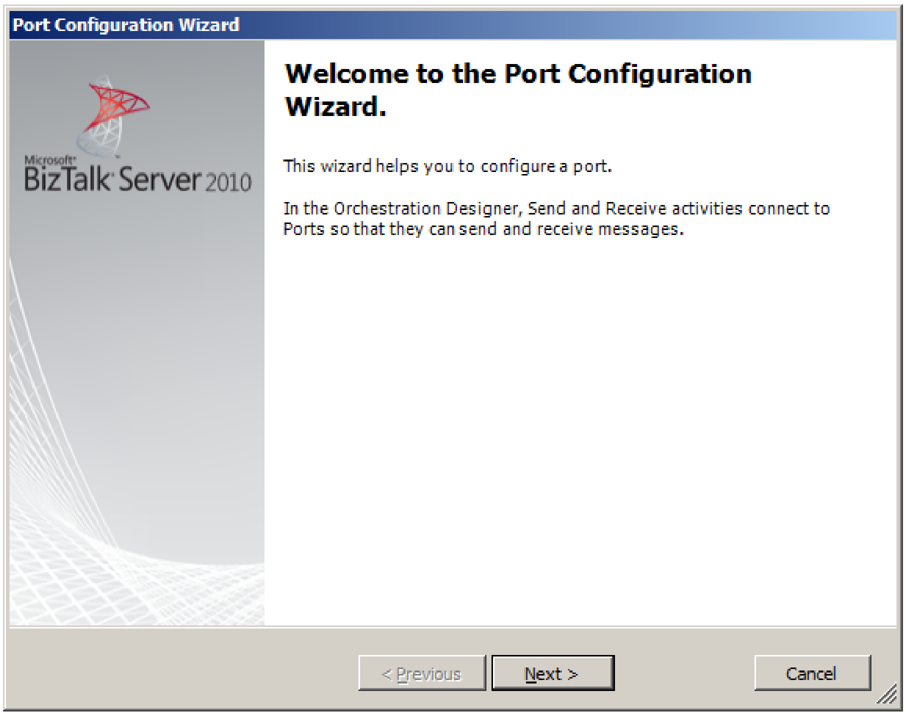
   - A ***One Way Send Port* “spCreateAccount”**.

   - A ***Request-Response Port*** “**MSCRM_WCF”**.

3. Right-click on the left ***Port Surface*** and choose ***New Configured Port…*** .



4. In the appearing wizard, click Next >.

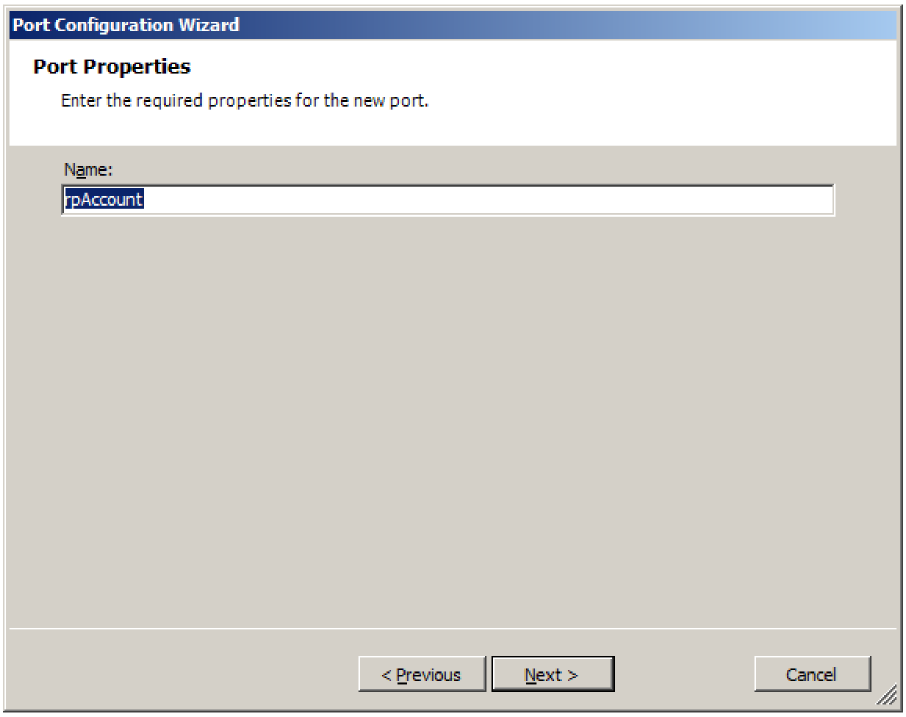


Welcome to the Port Configuration Wizard.

This wizard helps you to configure a port.

In the Orchestration Designer, Send and Receive activities connect to Ports so that they can send and receive messages.

5. Type in **rpAccount** in the field ***Name*** and click Next >.



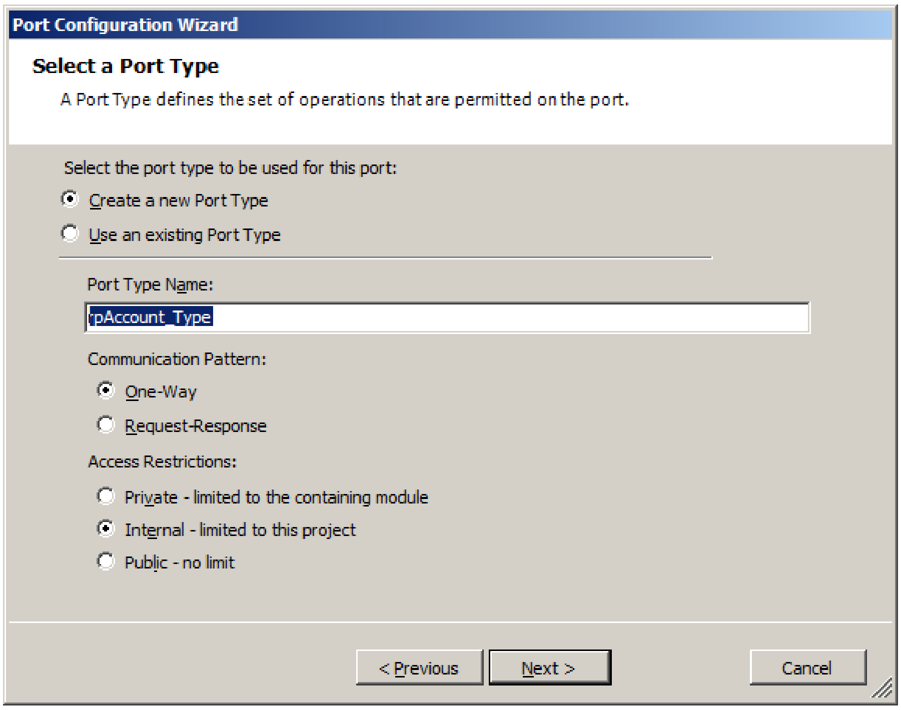
Port Configuration Wizard

Port Properties

Enter the required properties for the new port.

Name: rpAccount

6. Enter **rpAccount_Type** in the field ***Port Type Name***. Leave all other settings as they are. Click Next >.



Port Configuration Wizard

Select a Port Type

A Port Type defines the set of operations that are permitted on the port.

Select the port type to be used for this port:

● Create a new Port Type

○ Use an existing Port Type

Port Type Name: rpAccount_Type

Communication Pattern:

● One-Way

○ Request-Response

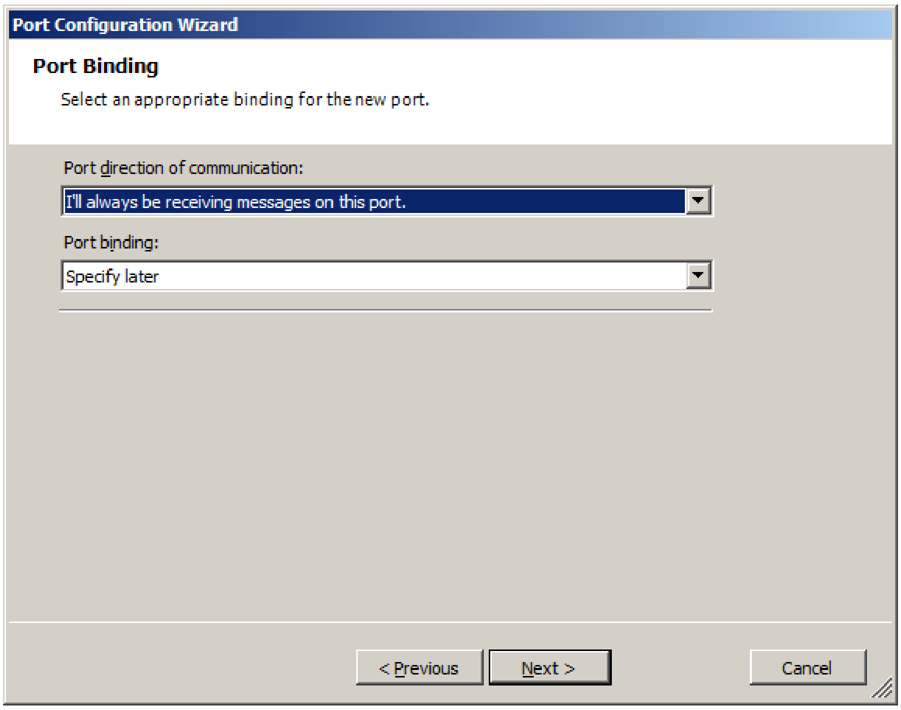
Access Restrictions:

○ Private - limited to the containing module

● Internal - limited to this project

○ Public - no limit

7. On the next form no changes are needed. Click Next >.



Port Configuration Wizard

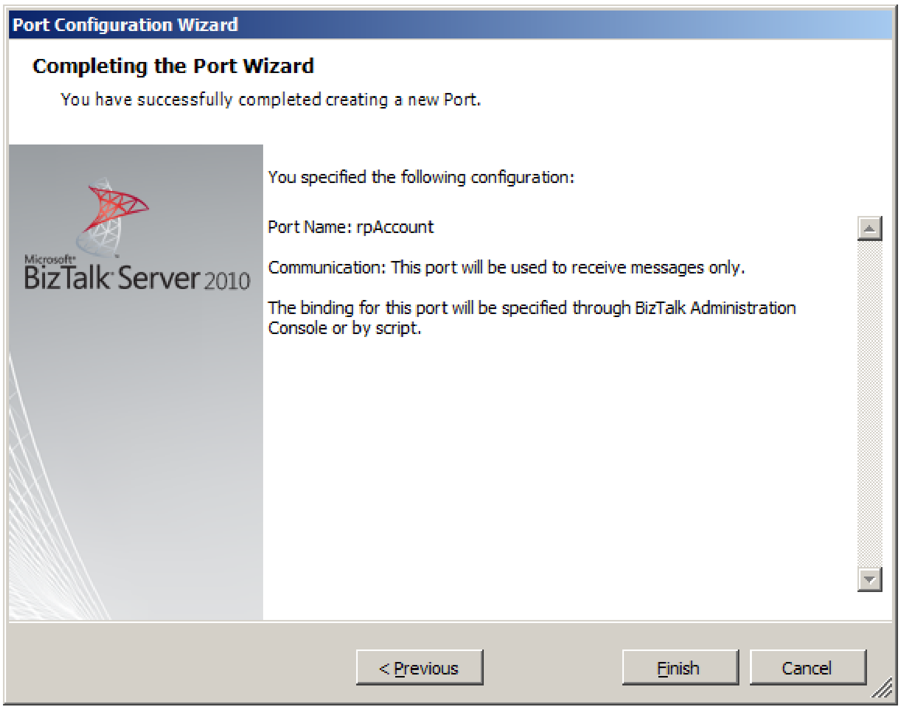
Port Binding

Select an appropriate binding for the new port.

Port direction of communication: I'll always be receiving messages on this port.

Port binding: Specify later

8. Click Finish to finish the wizard.



Port Configuration Wizard

Completing the Port Wizard

You have successfully completed creating a new Port.

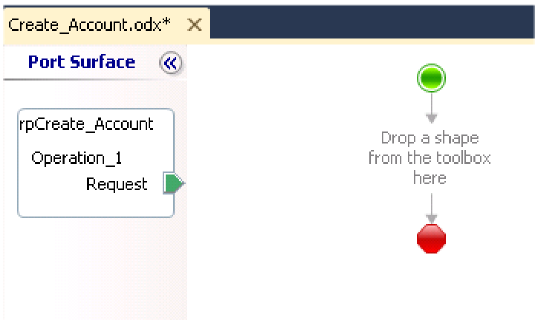
You specified the following configuration:

Port Name: rpAccount

Communication: This port will be used to receive messages only.

The binding for this port will be specified through BizTalk Administration Console or by script.

9. Your orchestration may look like this:

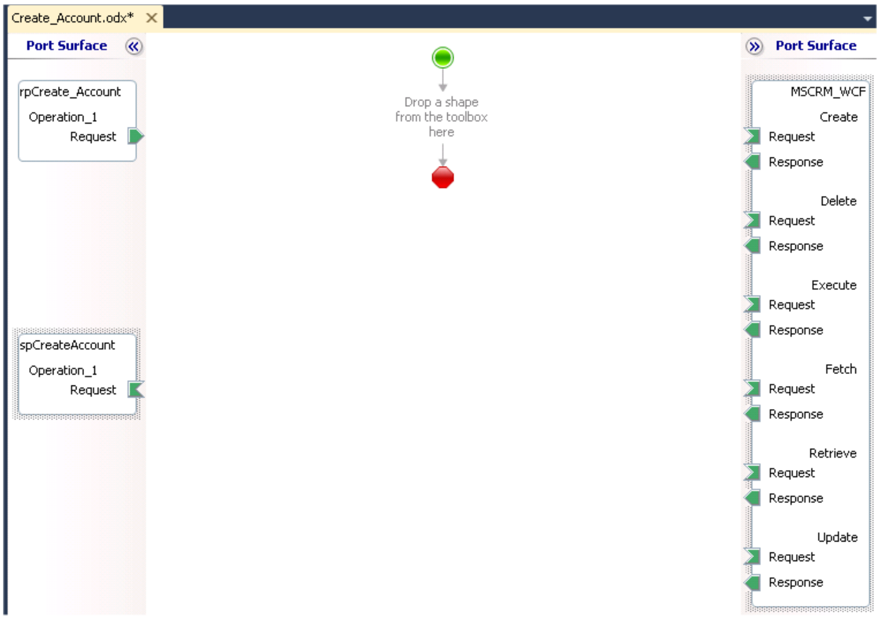


10. Repeat these steps for the send port.

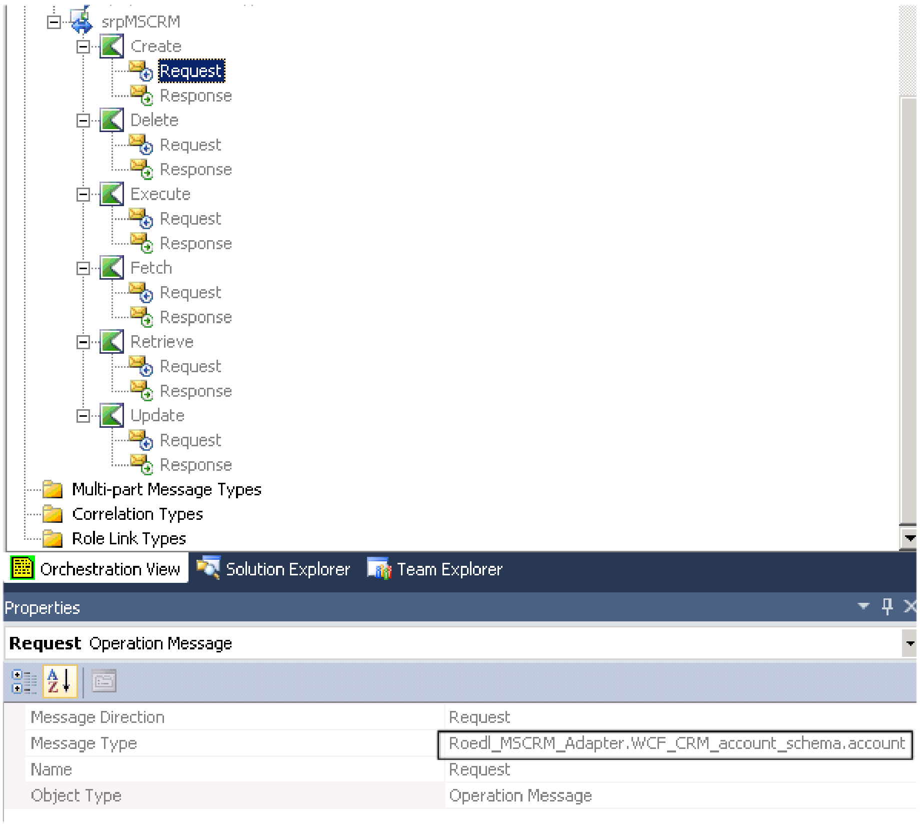
11. While configuring the send port you have to select **I´ll always be sending messages on this port** in the field ***Port direction of communication.*** All other information remains the same.

12. For the port **MSCRM_WCF** you have to create a port with multiple so-called operations, namely: Create, Update, Delete, Retrieve, Fetch and Execute.

13. Now your orchestration should look like this:



14. Select the operation Request of the Create method. In the property section, select the account schema.



Request Operation Message

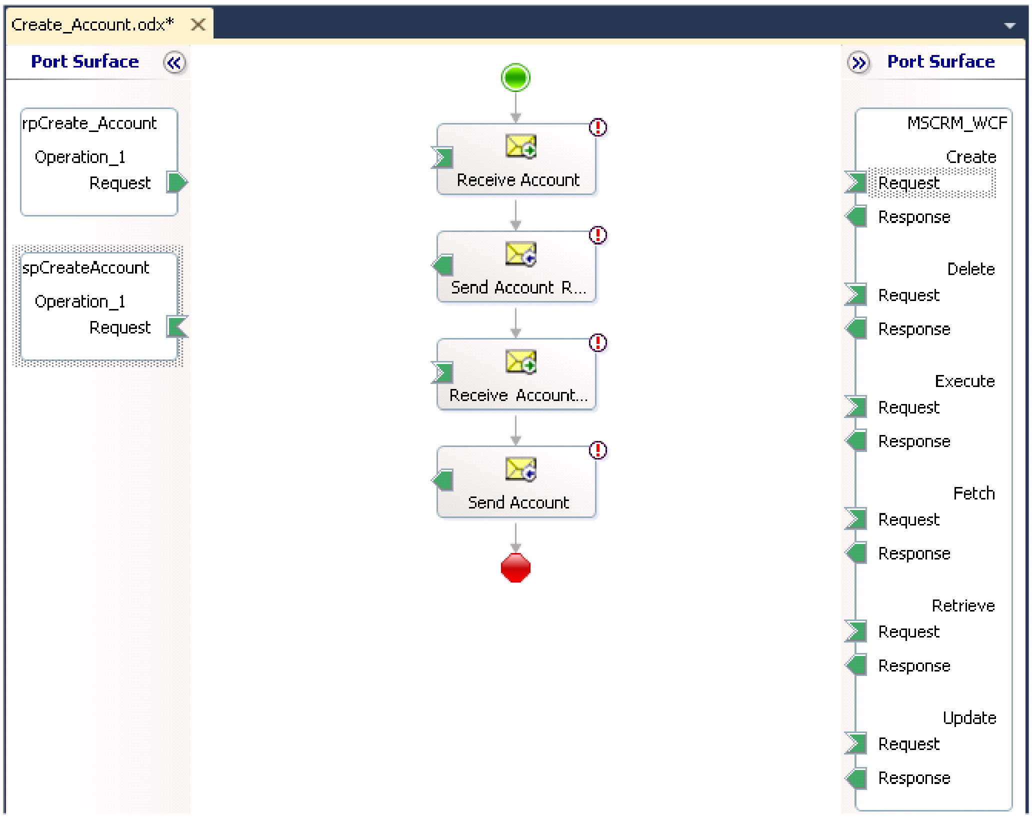
Message Direction	Request
Message Type	Roedl_MSCRM_Adapter.WCF_CRM_account_schema.account
Name	Request
Object Type	Operation Message

15. Repeat this for all methods and enter the following schemas:

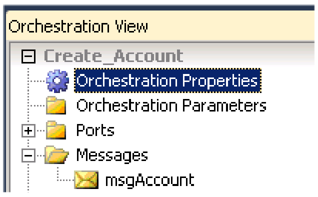
Methode	Request	Response
Insert	Account	Account
Update	Account	Account
Delete	Account	Account
Retrieve	Account	Account
Fetch	Fetch	AccountList
Upsert	accountCustomAction	Account

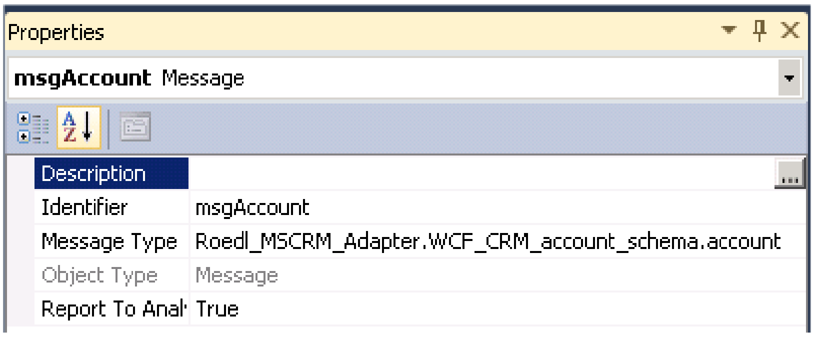
16. Now the port is ready to send the messages to MS CRM.

17. Now you need the send and receive shapes in the orchestration. Choose the appropriated shapes from the toolbox and drag them into the orchestration.



18. Now the message for the orchestration will be created.

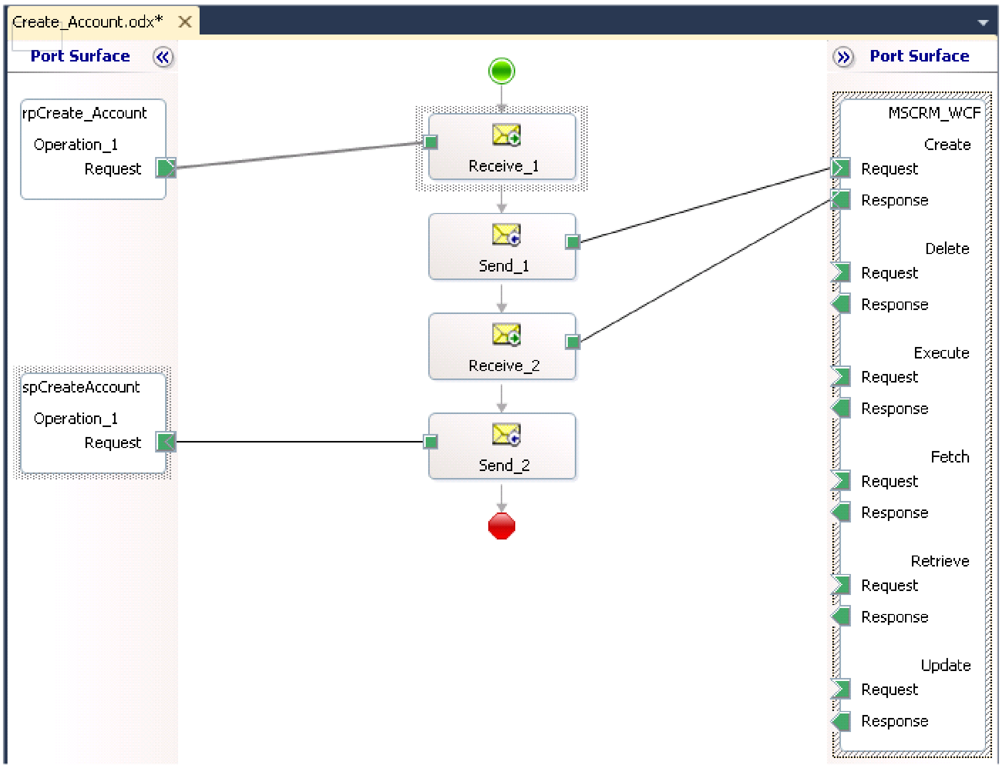




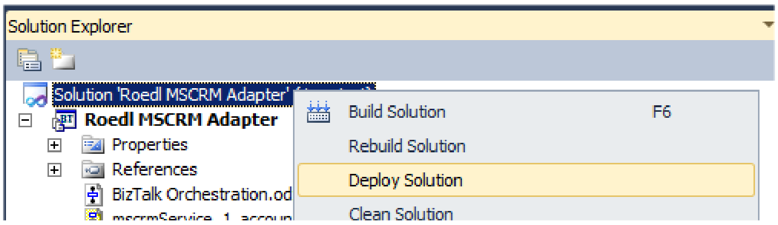
msgAccount Message

Description	
Identifier	msgAccount
Message Type	Roedl_MSCRM_Adapter.WCF_CRM_account_schema.account
Object Type	Message
Report To Anal	True

19. Assign the message to the send and receive shapes and connect them with the logical ports. After that, your orchestration should look like this:

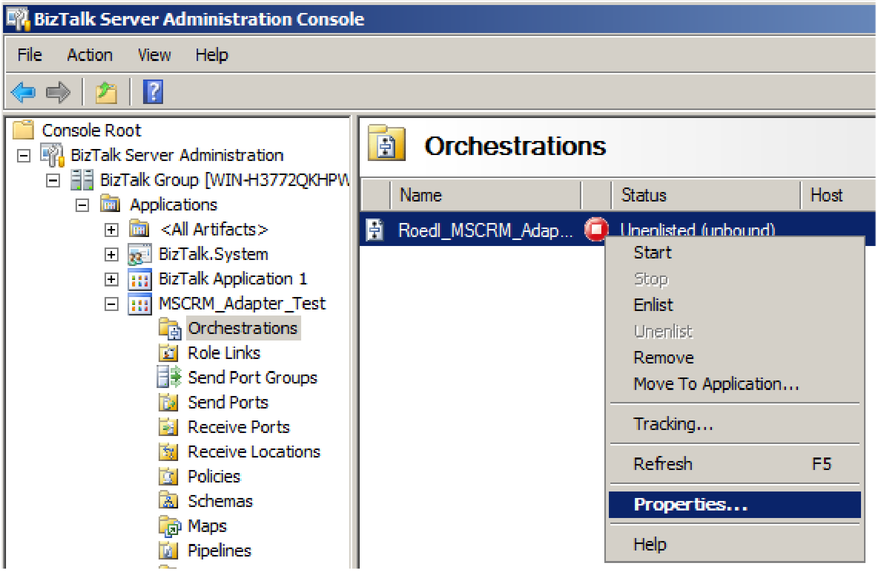


20. The orchestration is now ready to be deployed.



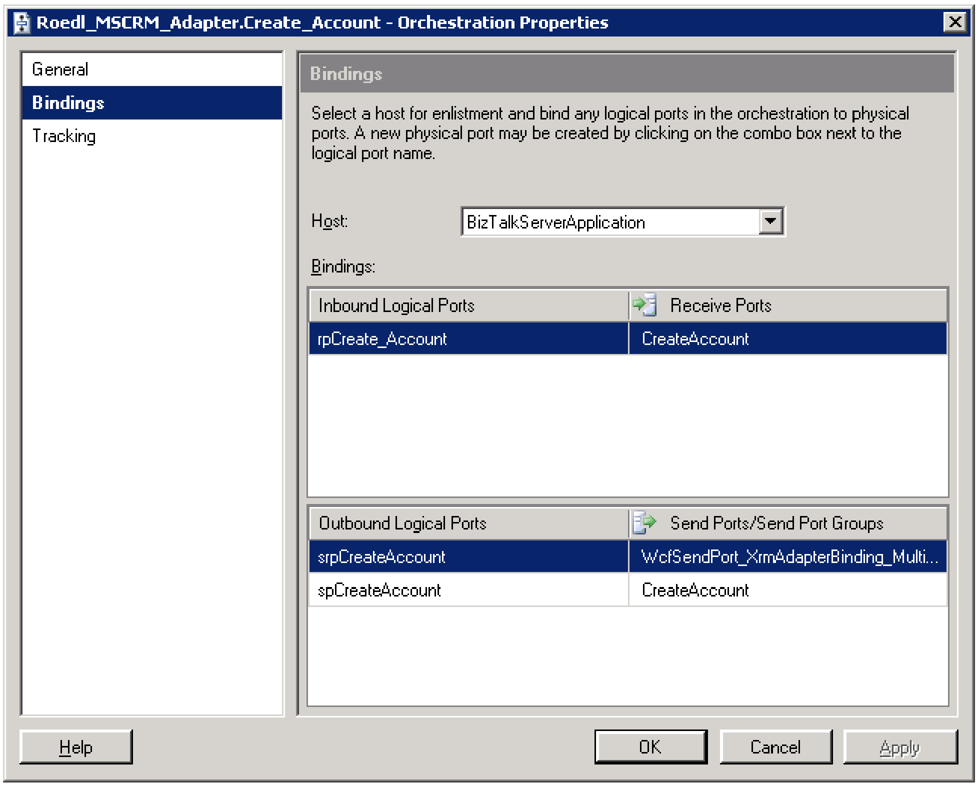
Step 5: Set up the orchestration in the BizTalk Server Administration Console

1. After successful deployment, the **BizTalk Server Administration Console** should look like this:



2. Expand the application node **MSCRM_Adapter_Test** and click *Orchestrations*.

3. Right-click on the orchestration **Create_Account** and select *Properties…*


Roedl_MSCRM_Adapter.Create_Account - Orchestration Properties
General
Bindings
Tracking
Help
OK
Cancel
Apply

Bindings

Select a host for enlistment and bind any logical ports in the orchestration to physical ports. A new physical port may be created by clicking on the combo box next to the logical port name.

Host: BizTalkServerApplication

Bindings:

Inbound Logical Ports	Receive Ports
rpCreate_Account	CreateAccount

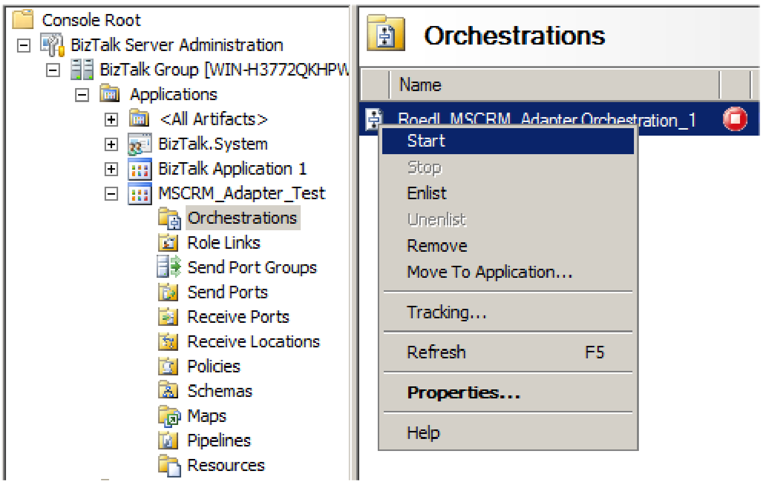
Outbound Logical Ports	Send Ports/Send Port Groups
srpCreateAccount	WcfSendPort_XrmAdapterBinding_Multi...
spCreateAccount	CreateAccount

4. As shown in the screenshot above, bind the three physical ports to the virtual ports of the orchestration.

5. Connect the orchestration to the host.

6. Click .

7. Start the orchestration.



8. The BizTalk Server is now ready to execute the orchestration

Step 6: Testing the Orchestration

1. Copy a XML file with the following structure into the folder *C:\...\CreateAccount\Request*.

   Attention:
   Take care of the namespace. It has to be the same as given in the schema of **Account**!

```
<?xml version="1.0" encoding="utf-8"?>
  <account xmlns="http://Roedl.BizTalk.Schemas.MSCRM/account">
    <name>Roedl Consulting AG</name>
    <address1_name> Roedl Consulting AG</address1_name>
    <address1_line1>Laubanger</address1_line1>
    <address1_line2>23</address1_line2>
    <address1_city>Bamberg</address1_city>
    <address1_country>D</address1_country>
    <address1_postalcode>96052</address1_postalcode>
  </account>
```

2. The BizTalk Server reads the file and sends the data to the adapter.

3. After a short time an XML file will appear in the folder *C:\...\CreateAccount\Response*. It will be named something like: **{DE15B859-B2AD-4B5D-9A46-5A2CDBE2281C}.xml**

4. The content of this file shows the data of the created account. Some system attributes created by the CRM system will appear too.

```
<?xml version="1.0" encoding="utf-8"?>
  <account xmlns="http://Roedl.BizTalk.Schemas.MSCRM/account">
    <accountid>{A61F690E-FB22-DB11-92AE-0050DA4615A6}</accountid>
    <owningbusinessunit>{C4991D79-06EB-DA11-A201-0050DA4615A6}</owningbusinessunit>
    <participatesinworkflow name="No">0</participatesinworkflow>
    <name>Roedl Consulting AG</name>
    <createdon time="4:19 PM" date="03/08/2007">2011-03-08T16:19:24+02:00</createdon>
    <createdby name="BizTalk" dsc="0">{51ACBF89-91EC-DA11-900D-0050DA4615A6}</createdby>
    <modifiedon time="4:19 PM" date="03/08/2007">2011-03-08T16:19:24+02:00</modifiedon>
    <modifiedby name="BizTalk" dsc="0">{51ACBF89-91EC-DA11-900D-0050DA4615A6}</modifiedby>
    <statecode name="Active" formattedvalue="0">0</statecode>
    <statuscode name="Active" formattedvalue="1">1</statuscode>
    <address1_addressid>{A71F690E-FB22-DB11-92AE-0050DA4615A6}</address1_addressid>
    <address1_name>Roedl Consulting AG</address1_name>
    <address1_line1>Laubanger</address1_line1>
    <address1_line2>23</address1_line2>
    <address1_city>Bamberg</address1_city>
    <address1_country>D</address1_country>
    <address1_postalcode>96052</address1_postalcode>
    <address2_addressid>{A81F690E-FB22-DB11-92AE-0050DA4615A6}</address2_addressid>
    <ownerid type="8" name="BizTalk" dsc="0">{51ACBF89-91EC-DA11-900D-0050DA4615A6}</ownerid>
    <donotsendmm name="Yes">0</donotsendmm>
    <merged name="No">0</merged>
  </account>
```

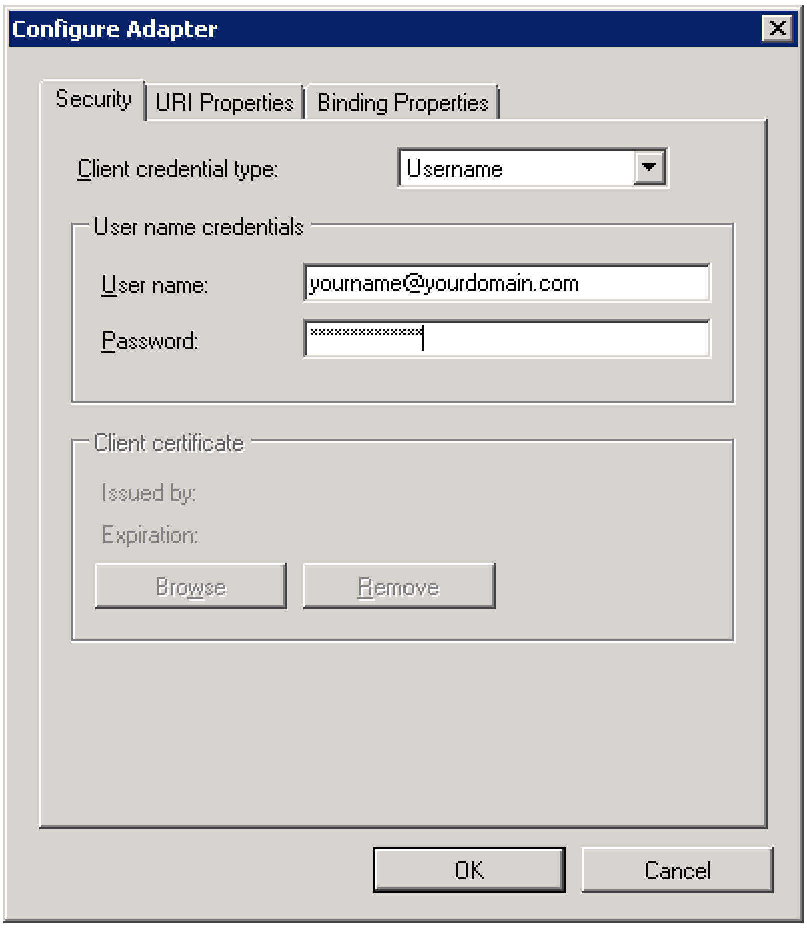
5. The data can now be processed.

   Advantage:
   Note that the response message contains all attributes of the entity, even the attributes provided in the request message. You can directly continue to work with this data.

Connecting to CRM Online

If you want to connect to CRM Online, please follow the steps 1-3 from
“**Working with the adapter in Visual Studio 2010 (Designtime)**” and continue with the steps below.

4. On the Security tab you have to choose the client credential type Username and provide a valid Windows Live Id account. This account has to have enough rights on the CRM Online system.



Configure Adapter

Client credential type: Username

User name credentials

User name: yourname@yourdomain.com

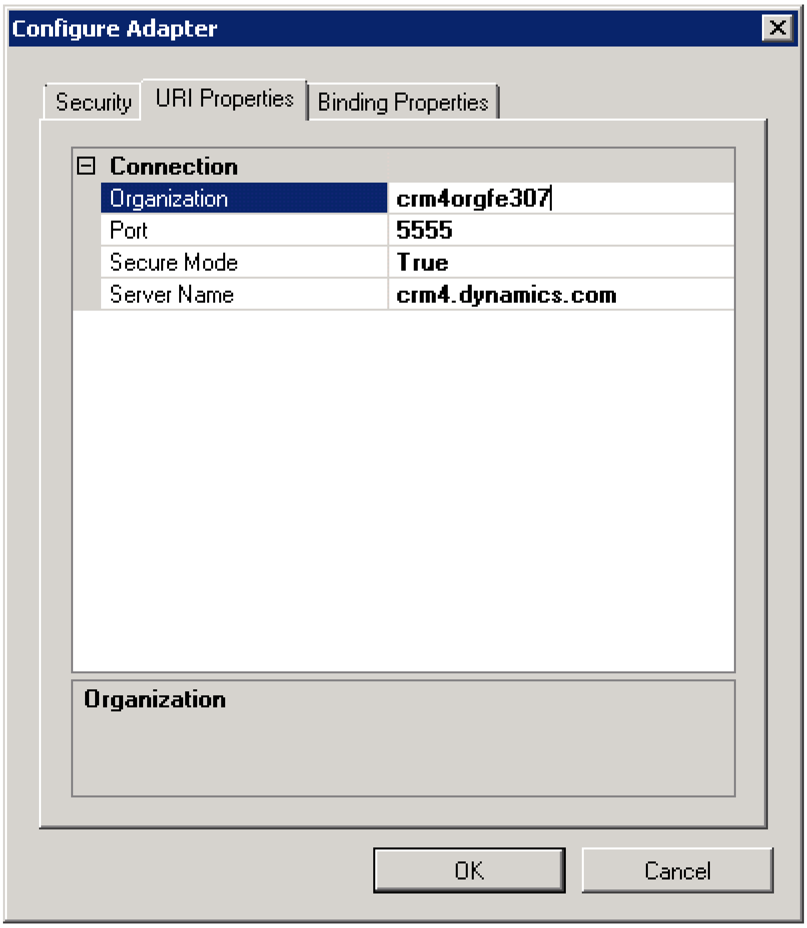
Password: **************

Client certificate

Issued by:

Expiration:

5. On the URI Properties tab you have to provide system related information.
   Please make sure that you type in the correct organization name. This is not the prefix of your CRM Online url, but you can find it in CRM Online when you go to Settings -> Customizations -> Developer Resources. It's the unique name of the organization.
   Server Name should be crm4.dynamics.com for Europe. Secure Mode = True. Port doesn't matter in this case, the Adapter will use 443 anyway.



Connection

Organization	crm4orgfe307
Port	5555
Secure Mode	True
Server Name	crm4.dynamics.com

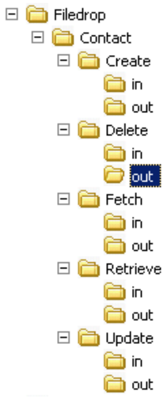
Organization

On the tab "Binding Properties" you can choose some general timeout options and the target namespace. In the section **licensing** you have to enter the license code and license name you have received upon your purchase.
Afterwards you can continue with Step 6 from "**Working with the adapter in Visual Studio 2010 (Designtime)**" and select your entities and messages.

Further examples

In the following, all the methods of the adapter will be presented once more individually in more detail using the Contact entity.

1. First create a folder structure for processing the messages. This may look like this:

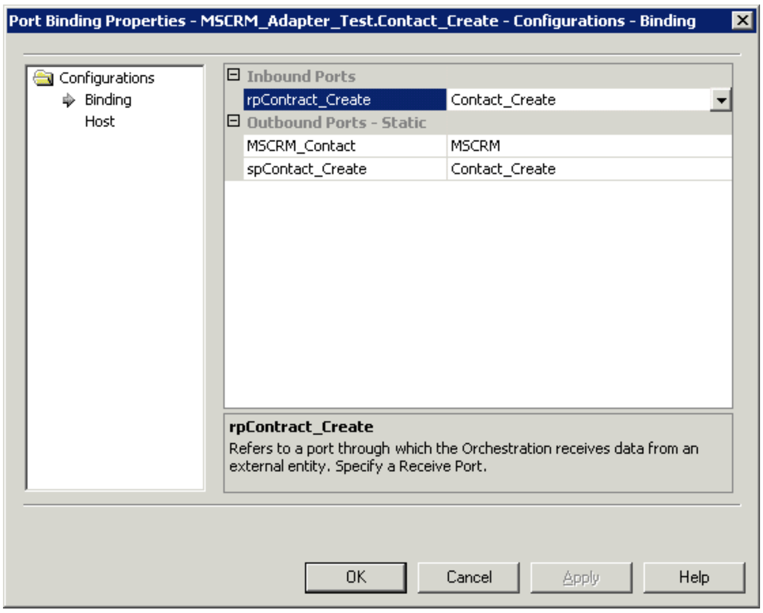


```
Filedrop
  Contact
    Create
      in
      out
    Delete
      in
      out
    Fetch
      in
      out
    Retrieve
      in
      out
    Update
      in
      out
```

2. In the **BizTalk Server Administration Console,** create the appropriate receive and send ports as described earlier in this manual.

3. Deploy the orchestrations described in the following and bind the logical ports to the physical ports in the **BizTalk Server Administration Console**. Right-click on the orchestration **Contact_Create**.

4. In the drop down fields, select the appropriate physical ports and confirm with OK.



Port Binding Properties - MSCRM_Adapter_Test.Contact_Create - Configurations - Binding

Inbound Ports	
rpContract_Create	Contact_Create
Outbound Ports - Static	
MSCRM_Contact	MSCRM
spContact_Create	Contact_Create

rpContract_Create
Refers to a port through which the Orchestration receives data from an external entity. Specify a Receive Port.

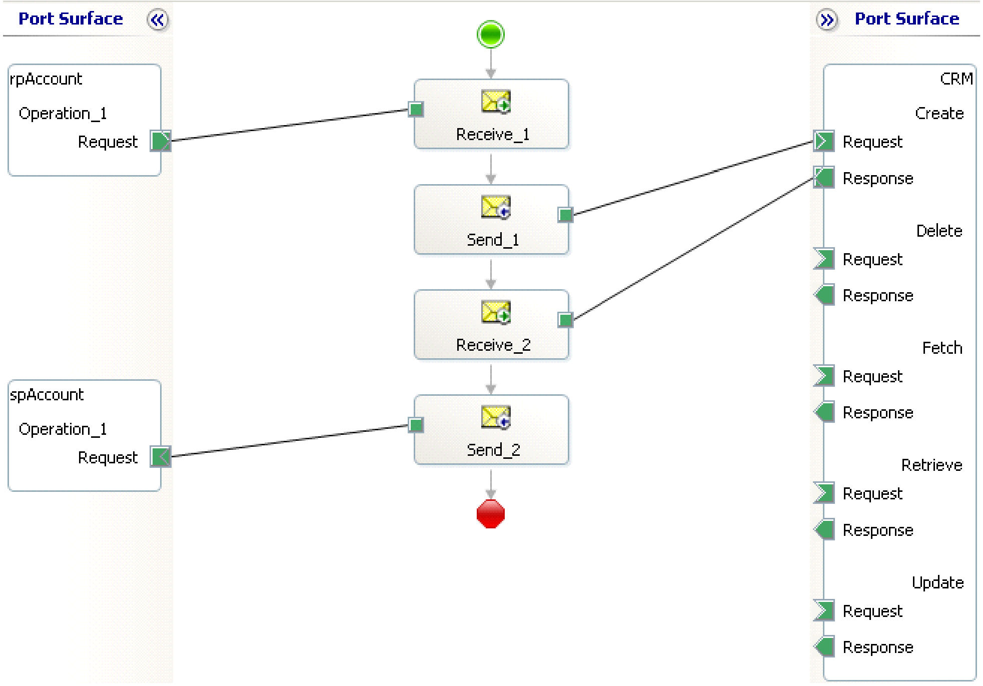
5. Repeat Step 3 and 4 with the orchestrations **Delete_Contact**, **Fetch_Contact**, **Retrieve_Contact** and **Update_Contact**.

Create

The Create method is used to create entity records.

Request Messagetype	Response Messagetype
Contact	Contact

1. Create an orchestration looking like this:



2. Create an XML message file to create a new record.
For example: contact_create.xml

```
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/contact>
  <firstname>Jim</firstname>
  <fullname>Jim T.</fullname>
  <lastname>Kirk</lastname>
  <mobilephone>+49 171 123456</mobilephone>
</contact>
```

Attention:
Take care of the namespace. It has to be the same as given in the schema!

3. Copy the file into the input folder of the receive port ***Contact_Create***, e.g. *C:\FileDrop\Contact\Create\In*.

4. After a short while, an XML file will be appear in the output folder of the send port ***Contact_Create***, e.g. *C:\FileDrop\Contact\Create\Out*. The content should look something like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/contact>
  <firstname>Jim</firstname>
  <fullname>Jim T.</fullname>
  <lastname>Kirk</lastname>
  <mobilephone>01234</mobilephone>
  <contactid>733ef35b-1947-db11-acc2-0050da4615a6</contactid>
</contact>
```

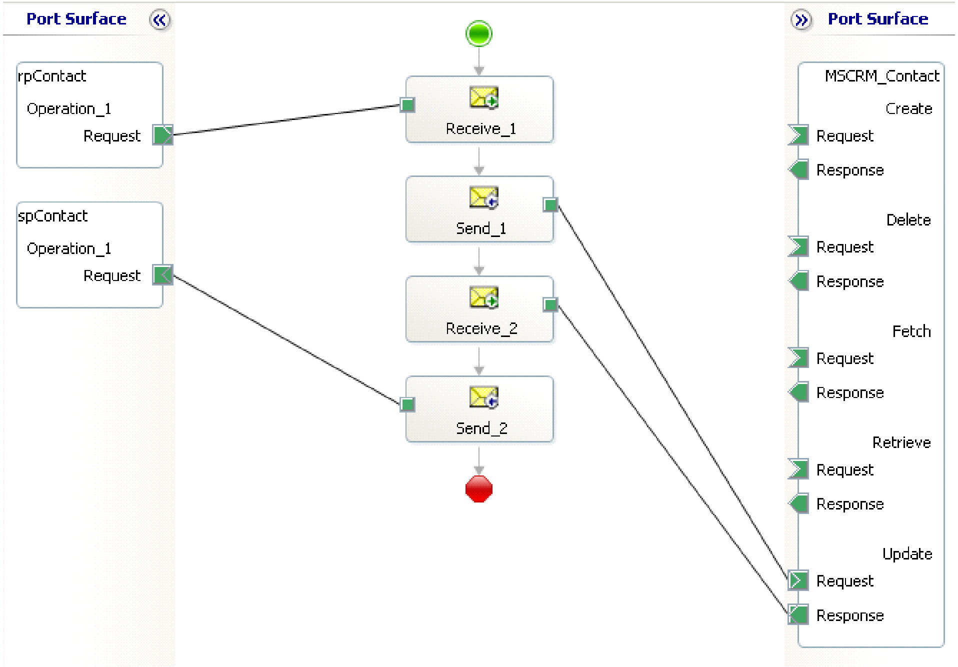
5. The CRM system has created the record and provides the **contactid**. This can be processed directly in the orchestration.

Update

The method Update is used for updating entity records.

Request Messagetype	Response Messagetype
Contact	Contact

1. Create an orchestration looking like this:



2. Create an XML message file to update an existing record.
   For example: contact_update.xml

```
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/contact>
  <firstname>Jim</firstname>
  <fullname>Jim Tiberius</fullname>
  <lastname>Kirk II</lastname>
  <mobilephone>01234</mobilephone>
  <contactid>733ef35b-1947-db11-acc2-0050da4615a6</contactid>
</contact>
```

Attention:
Take care of the **contactid**, so the CRM system can find the desired record.

3. Copy the file into the input folder of the receive port ***Contact_Update***, e.g. *C:\FileDrop\Contact\Update\In*.

4. After a short while, an XML file will appear in the output folder of the send port ***Contact_Update***, e.g. *C:\FileDrop\Contact\Update\Out*. The content should look something like this:

```
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/contact>
  <firstname>Jim</firstname>
  <fullname>Jim Tiberius</fullname>
  <lastname>Kirk II</lastname>
  <mobilephone>01234</mobilephone>
  <contactid>733ef35b-1947-db11-acc2-0050da4615a6</contactid>
</contact>
```

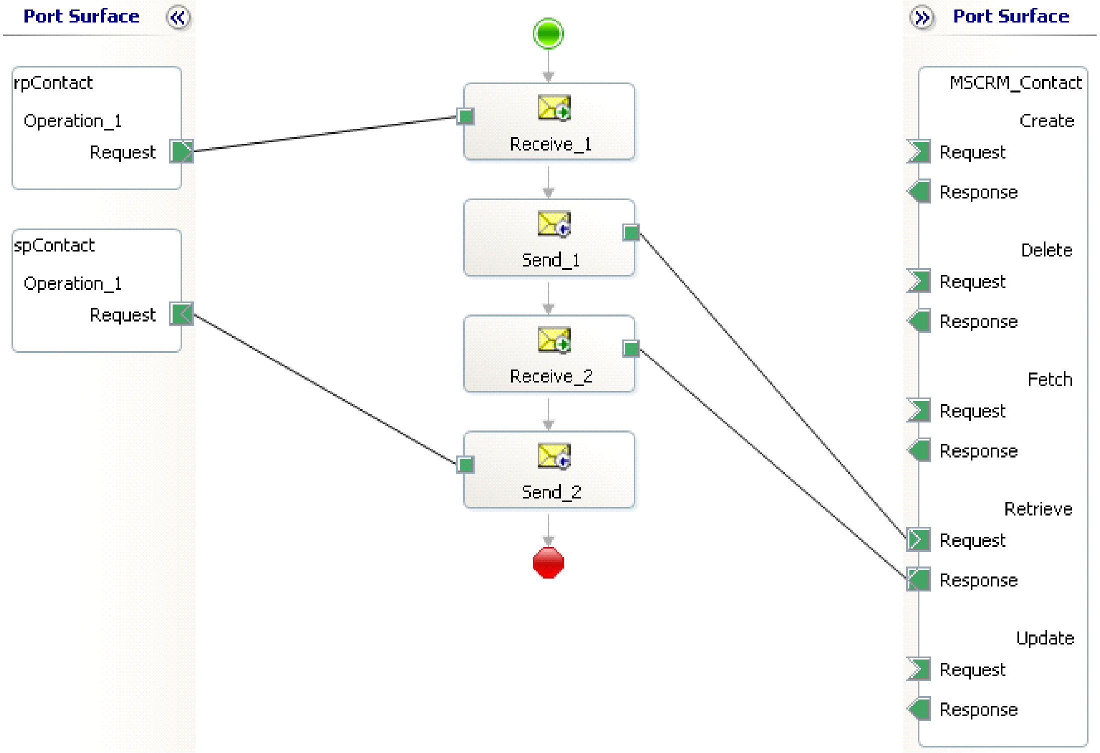
5. The CRM system has updated the record and provides the same message as response.

Retrieve

The Retrieve method is used for retrieving a single entity record based on the unique identifier.

Request Messagetype	Response Messagetype
Contact	Contact

1. Create an orchestration, that looks like this:



2. Create an XML message file to retrieve a single record.
   For example: contact_retrieve.xml

```
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/contact>
  <contactid>733ef35b-1947-db11-acc2-0050da4615a6</contactid>
</contact>
```

Attention:
Take care of the **contactid**, so the CRM system can find the desired record. The message can also contain further attributes, but the adapter will ignore them.

3. Copy the file into the input folder of the receive port ***Contact_Retrieve***, e.g. *C:\FileDrop\Contact\Retrieve\In*.

©2014 Rödl Consulting AG

4. After a short while, an XML file will appear in the output folder of the send port ***Contact_Retrieve***, e.g. *C:\FileDrop\Contact\Retrieve\Out*. The content should look something like this:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <address1_addressid>{607A99B9-AC5A-DB11-ACC2-0050DA4615A6}</address1_addressid>
  <owningbusinessunit>{C4991D79-06EB-DA11-A201-0050DA4615A6}</owningbusinessunit>
  <donotsendmm name="Yes">0</donotsendmm>
  <statecode name="Active" formattedvalue="Active">Active</statecode>
  <statuscode name="Active" formattedvalue="1">1</statuscode>
  <merged name="No">0</merged>
  <createdby dsc="0" name="creator">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</createdby>
  <contactid>{5F7A99B9-AC5A-DB11-ACC2-0050DA4615A6}</contactid>
  <firstname>James</firstname>
  <fullname>James Kirk</fullname>
  <createdon date="03/08/2011" time="1:19 PM">2007-03-08T13:19:44+02:00</createdon>
  <ownerid type="systemuser" dsc="0" name="owner">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</ownerid>
  <modifiedon date="03/08/2011" time="1:19 PM">2011-03-08T13:19:44+02:00</modifiedon>
  <mobilephone>01234</mobilephone>
  <modifiedby dsc="0" name="modifier">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</modifiedby>
  <lastname>Kirk</lastname>
  <address2_addressid>{617A99B9-AC5A-DB11-ACC2-0050DA4615A6}</address2_addressid>
  <participatesinworkflow name="No">0</participatesinworkflow>
  </contact>
```

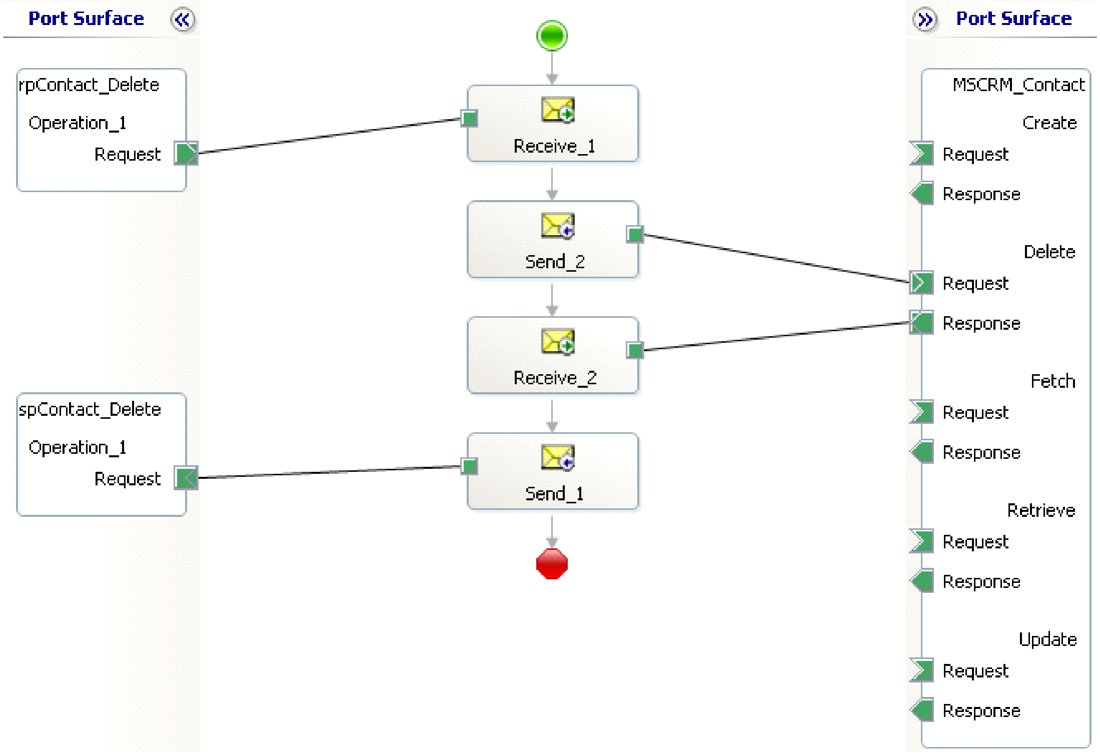
5. The CRM system has provided the record with all filled attributes and the data could now be processed in the orchestration.

Delete

The Delete method is used for deleting an entity record based on the unique identifier.

Request Messagetype	Response Messagetype
Contact	Contact

1. Create an orchestration looking like this:



2. Create an XML message file to delete a single record.
   For example: contact_delete.xml

```
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/contact>
  <contactid>733ef35b-1947-db11-acc2-0050da4615a6</contactid>
</contact>
```

Attention:
Take care of the **contactid**, so the CRM system can find the desired record. The message can also contain further attributes, but the adapter will ignore them.

3. Copy the file into the input folder of the receive port ***Contact_Delete***, e.g. *C:\FileDrop\Contact\Delete\In*.

©2014 Rödl Consulting AG

4. After a short while, an XML file will appear in the output folder of the send port ***Contact_Delete***, e.g. *C:\FileDrop\Contact\Delete\Out*. The content should look something like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<contact xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <address1_addressid>{607A99B9-AC5A-DB11-ACC2-0050DA4615A6}</address1_addressid>
  <owningbusinessunit>{C4991D79-06EB-DA11-A201-0050DA4615A6}</owningbusinessunit>
  <donotsendmm name="Yes">0</donotsendmm>
  <statecode name="Active" formattedvalue="Active">Active</statecode>
  <statuscode name="Active" formattedvalue="1">1</statuscode>
  <merged name="No">0</merged>
  <createdby dsc="0" name="creator">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</createdby>
  <firstname>James</firstname>
  <fullname>James Kirk</fullname>
  <createdon date="03/08/2011" time="1:19 PM">2007-03-08T13:19:44+02:00</createdon>
  <ownerid type="systemuser" dsc="0" name="owner">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</ownerid>
  <modifiedon date="03/08/2011" time="1:19 PM">2011-03-08T13:19:44+02:00</modifiedon>
  <mobilephone>01234</mobilephone>
  <modifiedby dsc="0" name="modifier">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</modifiedby>
  <lastname>Kirk</lastname>
  <address2_addressid>{617A99B9-AC5A-DB11-ACC2-0050DA4615A6}</address2_addressid>
  <participatesinworkflow name="No">0</participatesinworkflow>
  </contact>
```

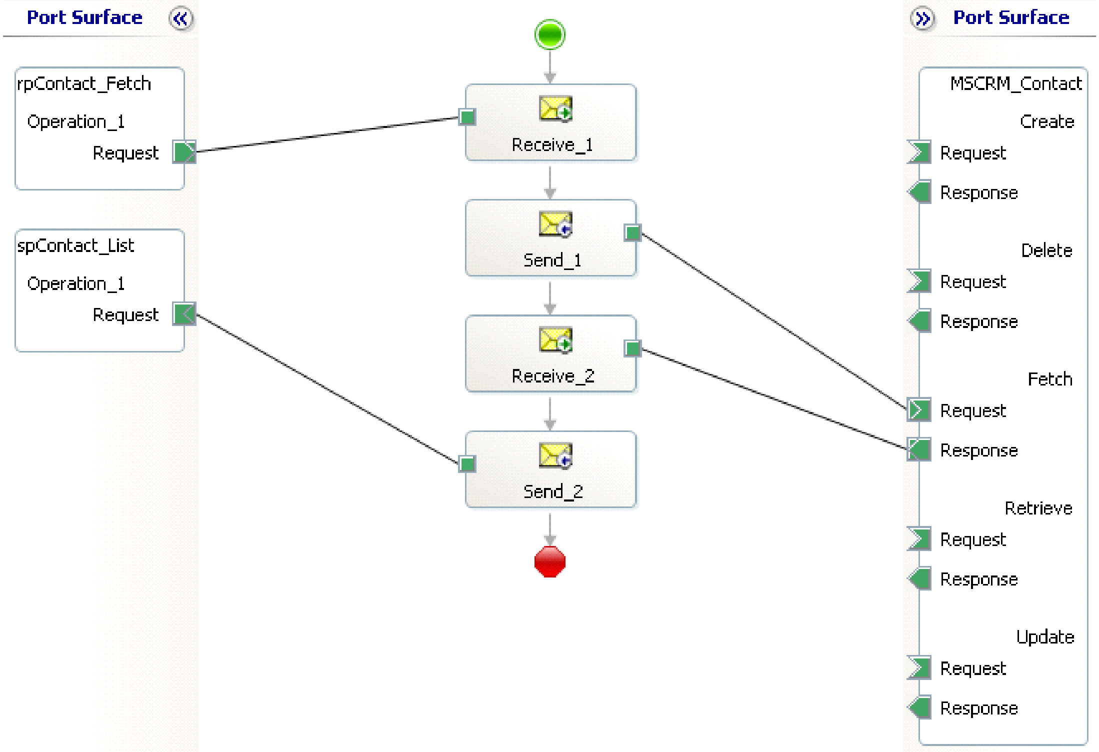
5. The CRM system has deleted the record and provides the complete record data as response. The contactid is not contained, because the record with this unique identifier doesn’t exist anymore.

Fetch

The Fetch method is used to search entity records based on several criterias.

Request Messagetype	Response Messagetype
Fetch	ContactList

1. Create a new orchestration looking like this:



2. Create an XML message file to search for one or more records.
   For example: contact_fetch.xml

```
<fetch mapping="logical" xmlns="http://Roedl.BizTalk.Schemas.MSCRM/fetch">
   <entity name="contact">
      <all-attributes />
      <filter type="or">
         <condition attribute="firstname" operator="eq" value="James" />
      </filter>
   </entity>
</fetch>
```

Attention:
Take care of the namespace. It has to be the same as given in the entity schema, except “contact” instead of “fetch”

3. Copy the file into the input folder of the receive port ***Contact_Fetch***, e.g. *C:\FileDrop\Contact\Fetch\In*.

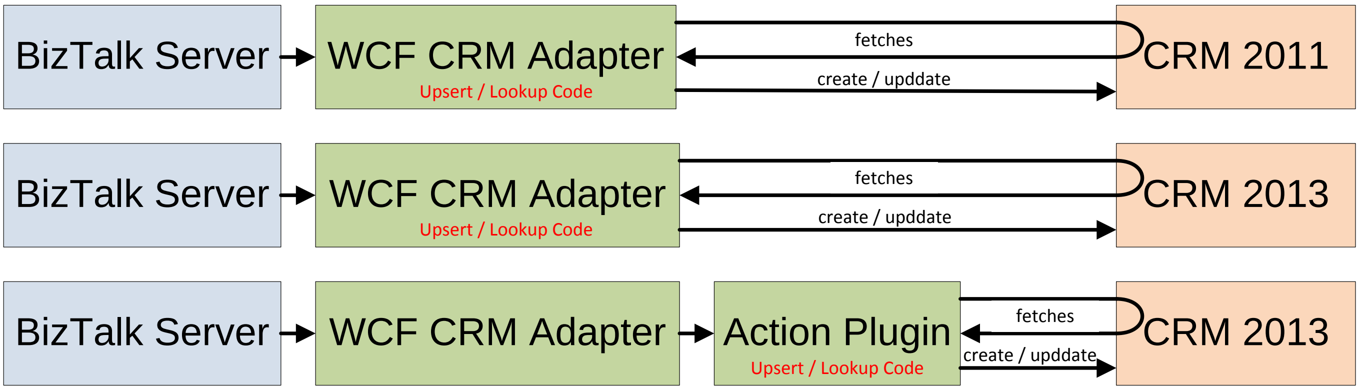
4. After a short while, an XML file will appear in the output folder of the send port ***Contact_Fetch***, e.g. *C:\FileDrop\Contact\Fetch\Out*. The content should look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
<contactList xmlns="http://Roedl.BizTalk.Schemas.MSCRM/contactList">
  <count>2</count>
  <contact xmlns="http://Roedl.BizTalk.Schemas.MSCRM/contact">
    <contactid>{5F7A99B9-AC5A-DB11-ACC2-0050DA4615A6}</contactid>
    <owningbusinessunit>{C4991D79-06EB-DA11-A201-0050DA4615A6}</owningbusinessunit>
    <participatesinworkflow name="No">0</participatesinworkflow>
    <firstname>James</firstname>
    <lastname>Kirk</lastname>
    <fullname>James Kirk</fullname>
    <createdon time="1:19 PM" date="03/08/2011">2011-03-08T13:19:44+02:00</createdon>
    <createdby name="creator" dsc="0">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</createdby>
    <modifiedon time="1:32 PM" date="03/08/2011">2011-03-08T13:32:44+02:00</modifiedon>
    <modifiedby name="modifier" dsc="0">{4CACBF89-91EC-DA11-900D-0050DA4615A6}</modifiedby>
    <mobilephone>01234</mobilephone>
    <statecode name="Active" formattedvalue="0">0</statecode>
    <statuscode name="Active" formattedvalue="1">1</statuscode>
    <address1_addressid>{607A99B9-AC5A-DB11-ACC2-0050DA4615A6}</address1_addressid>
    <address2_addressid>{617A99B9-AC5A-DB11-ACC2-0050DA4615A6}</address2_addressid>
    <ownerid type="8" name="owner" dsc="0">{CD8E9A9A-9D44-DB11-ACC2-0050DA4615A6}</ownerid>
    <merged name="No">0</merged>
    <donotsendmm name="Yes">0</donotsendmm>
  </contact>
  <contact xmlns="http://crm.logisma.de/mscrmservices/2007/crmservice.asmx-contact">
    <contactid>{D9C68592-AE5A-DB11-ACC2-0050DA4615A6}</contactid>
    <owningbusinessunit>{C4991D79-06EB-DA11-A201-0050DA4615A6}</owningbusinessunit>
    <participatesinworkflow name="No">0</participatesinworkflow>
    <firstname>James</firstname>
    <lastname>Belushi</lastname>
    <fullname>James Belushi</fullname>
    <donotphone name="Allow">0</donotphone>
    <donotfax name="Allow">0</donotfax>
    <donotemail name="Allow">0</donotemail>
    <donotpostalmail name="Allow">0</donotpostalmail>
    <donotbulkemail name="Allow">0</donotbulkemail>
    <createdon time="1:32 PM" date="03/08/2011">2011-03-08T13:32:57+02:00</createdon>
    <creditonhold name="No">0</creditonhold>
    <createdby name="creator" dsc="0">{4CACBF89-91EC-DA11-900D-0050DA4615A6}</createdby>
    <modifiedon time="1:32 PM" date="03/08/2011">2011-03-08T13:32:57+02:00</modifiedon>
    <modifiedby name="modifier" dsc="0">{4CACBF89-91EC-DA11-900D-0050DA4615A6}</modifiedby>
    <statecode name="Active" formattedvalue="0">0</statecode>
    <statuscode name="Active" formattedvalue="1">1</statuscode>
    <address1_addressid>{DAC68592-AE5A-DB11-ACC2-0050DA4615A6}</address1_addressid>
    <address2_addressid>{DBC68592-AE5A-DB11-ACC2-0050DA4615A6}</address2_addressid>
    <ownerid type="8" name="owner" dsc="0">{4CACBF89-91EC-DA11-900D-0050DA4615A6}</ownerid>
    <merged name="No">0</merged>
    <donotsendmm name="Yes">0</donotsendmm>
  </contact>
</contactList>
```

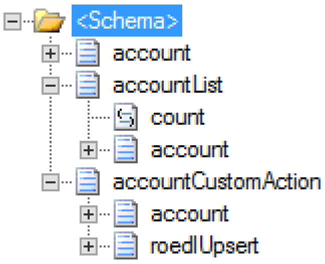
5. The CRM system has founded some records based on the given criteria and provides these data via the ***ContactList***. These contact records may now be transformed to ***Contact*** objects.

New Upsert and Lookup functionalities

The new functionality is running in Microsoft Dynamics CRM 2013/ 2015 and 2013 Systems. For 3013 Systems, we write a special action method, which call al the fetch functions direct on the CRM System.
The Adapter recognizes, if the CRM System is a 2011 or a 2013 System. If the CRM System is a 2013 System, the Adapter also recognize, if the Roedl Action Plugin is installed.
If the Roedl Action Plugin is installed, all fetch statements for Update and Lookups are running on the CRM Server.



For the new functionalities, we need additional information in the schemas.
Especialy for that, we create a new Schema type for the adapter.
The schema type ist ***entity*CustomAction.**
This schema includes the entity schema and the roedlUpsert schema.
For the UPSERT functionality, this new ***entity*CustomAction** have to be send with the adapter.
In the virtual send port, we need for that a new action “upsert”.



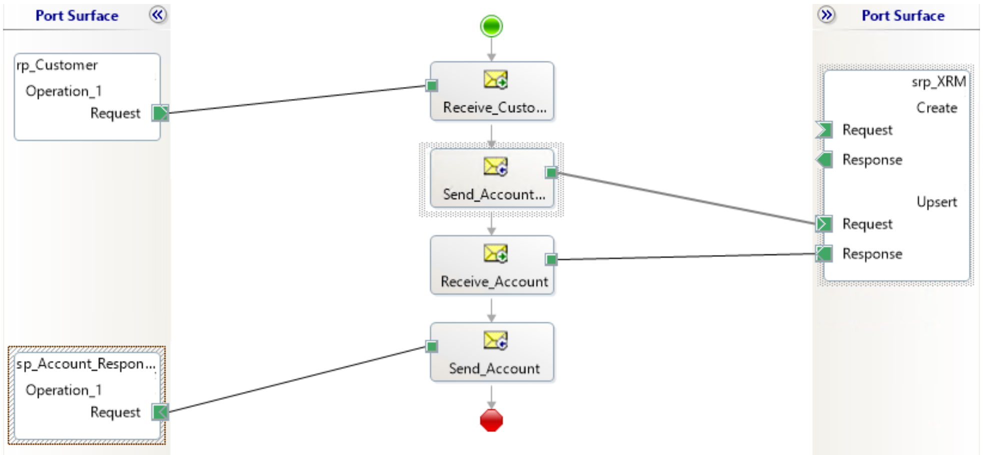
```
<Schema>
    account
    accountList
        count
        account
    accountCustomAction
        account
        roedlUpsert
```

Upsert

The Upsert method is a new feature to create AND update entity records.

Request Messagetype	Response Messagetype
accountCustomAction	account

6. Create an orchestration looking like this:



7. Create an XML message file to create a new record.
For example: contact_create.xml

```
ns0:accountCustomAction xmlns:ns0="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <ns0:account>
    <ns0:accountnumber>123456</ns0:accountnumber>
    <ns0:address1_city>Nuremberg</ns0:address1_city>
    <ns0:address1_country>Germany</ns0:address1_country>
    <ns0:address1_line1>Mainstreet 1</ns0:address1_line1>
    <ns0:address1_postalcode>91000</ns0:address1_postalcode>
    <ns0:address1_telephone1>+49 123456</ns0:address1_telephone1>
    <ns0:emailaddress1>JTK@Space.com</ns0:emailaddress1>
    <ns0:name>James T. Kirk</ns0:name>
  </ns0:account>
  <ns0:roedlUpsert>
    <ns0:condition>accountnumber='123456'</ns0:condition>
  </ns0:roedlUpsert>
</ns0:accountCustomAction>
```

Attention:
Take care of the namespace. It has to be the same as given in the schema!

8. Copy the file into the input folder of the receive port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\In*.

9. After a short while, an XML file will be appear in the output folder of the send port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\Out*. The content should look something like this:

```
<ns0:account xmlns:ns0="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <ns0:accountnumber>123456</ns0:accountnumber>
  <ns0:address1_city>Nuremberg</ns0:address1_city>
  <ns0:address1_country>Germany</ns0:address1_country>
  <ns0:address1_line1>Mainstreet 1</ns0:address1_line1>
  <ns0:address1_postalcode>91000</ns0:address1_postalcode>
  <ns0:address1_telephone1>+49 123456</ns0:address1_telephone1>
  <ns0:emailaddress1>JTK@Space.com</ns0:emailaddress1>
  <ns0:name>James T. Kirk</ns0:name>
  <ns0:accountid>c126574a-8ccd-e311-93f3-00155d1a5a6b</ns0:accountid>
</ns0:account>
```

10. The CRM system has created the record and provides the **accountid**. This can be processed directly in the orchestration.
11. Now change the street in the input message

```
ns0:accountCustomAction xmlns:ns0="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <ns0:account>
    <ns0:accountnumber>123456</ns0:accountnumber>
    <ns0:address1_city>Nuremberg</ns0:address1_city>
    <ns0:address1_country>Germany</ns0:address1_country>
    <ns0:address1_line1>Space Place 1</ns0:address1_line1>
    <ns0:address1_postalcode>91000</ns0:address1_postalcode>
    <ns0:address1_telephone1>+49 123456</ns0:address1_telephone1>
    <ns0:emailaddress1>JTK@Space.com</ns0:emailaddress1>
    <ns0:name>James T. Kirk</ns0:name>
  </ns0:account>
  <ns0:roedlUpsert>
    <ns0:condition>accountnumber='123456'</ns0:condition>
  </ns0:roedlUpsert>
</ns0:accountCustomAction>
```

12. Copy the file into the input folder of the receive port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\In*.

13. After a short while, an XML file will be appear in the output folder of the send port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\Out*. The content should look something like this:

```
<ns0:account xmlns:ns0="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <ns0:accountnumber>123456</ns0:accountnumber>
  <ns0:address1_city>Nuremberg</ns0:address1_city>
  <ns0:address1_country>Germany</ns0:address1_country>
  <ns0:address1_line1>Space Place 1</ns0:address1_line1>
  <ns0:address1_postalcode>91000</ns0:address1_postalcode>
  <ns0:address1_telephone1>+49 123456</ns0:address1_telephone1>
  <ns0:emailaddress1>JTK@Space.com</ns0:emailaddress1>
  <ns0:name>James T. Kirk</ns0:name>
  <ns0:accountid>c126574a-8ccd-e311-93f3-00155d1a5a6b</ns0:accountid>
</ns0:account>
```

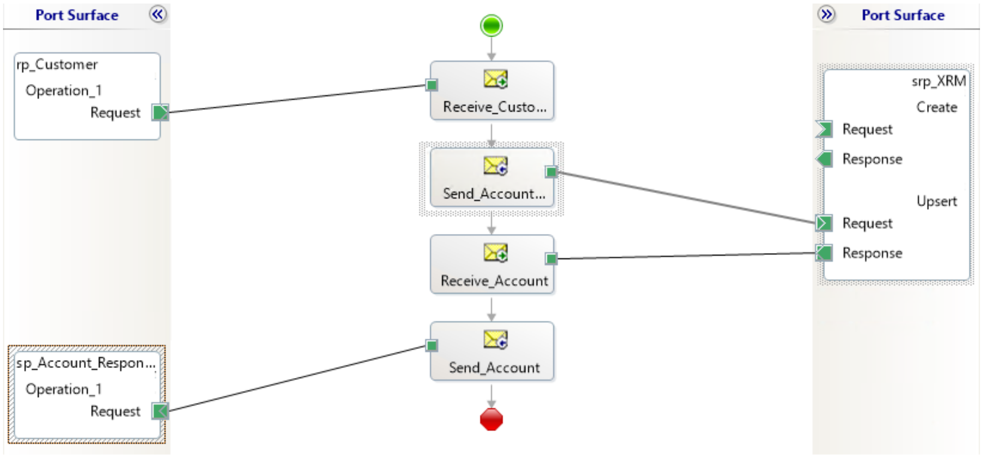
14. The accounted is the same, because there was an update in CRM!

Lookup functionality

The lookup functionality is also a new feature and is only working with the *entity*CustomAction (etc. accountCustomAction).

Request Messagetype	Response Messagetype
accountCustomAction	account

15. Create an orchestration looking like this:



16. Create an XML message file to create a new record.
    For example: contact_create.xml

```
ns0:accountCustomAction xmlns:ns0="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <ns0:account>
    <ns0:accountnumber>123456</ns0:accountnumber>
    <ns0:address1_city>Nuremberg</ns0:address1_city>
    <ns0:address1_country>Germany</ns0:address1_country>
    <ns0:address1_line1>Mainstreet 1</ns0:address1_line1>
    <ns0:address1_postalcode>91000</ns0:address1_postalcode>
    <ns0:address1_telephone1>+49 123456</ns0:address1_telephone1>
    <ns0:emailaddress1>JTK@Space.com</ns0:emailaddress1>
    <ns0:name>James T. Kirk</ns0:name>
    <ns0:parentaccountid type="account" condition="accountnumber='654987'"/>
  </ns0:account>
  <ns0:roedlUpsert>
    <ns0:condition>accountnumber='123456'</ns0:condition>
  </ns0:roedlUpsert>
</ns0:accountCustomAction>
```

Attention:
Take care of the namespace. It has to be the same as given in the schema!

17. Copy the file into the input folder of the receive port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\In*.

18. After a short while, an XML file will be appear in the output folder of the send port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\Out*. The content should look something like this:

```
<ns0:account xmlns:ns0="http://Roedl.BizTalk.Schemas.MSCRM/account">
  <ns0:accountnumber>123456</ns0:accountnumber>
  <ns0:address1_city>Nuremberg</ns0:address1_city>
  <ns0:address1_country>Germany</ns0:address1_country>
  <ns0:address1_line1>Mainstreet 1</ns0:address1_line1>
  <ns0:address1_postalcode>91000</ns0:address1_postalcode>
  <ns0:address1_telephone1>+49 123456</ns0:address1_telephone1>
  <ns0:emailaddress1>JTK@Space.com</ns0:emailaddress1>
  <ns0:name>James T. Kirk</ns0:name>
  <ns0:parentaccountid name="" type="account">c126574a-8ccd-e311-93f3-
00155d1a5a6b</ns0:parentaccountid>
  <ns0:accountid>c126574a-8ccd-e311-93f3-00155d1a5a6b</ns0:accountid>
</ns0:account>
```

19. The CRM system has created the record and provides the **accountid**. This can be processed directly in the orchestration. The parentaccountid is automatically included.

Import Account

A small project that demonstrates the performance of the adapter **before the upsert statement exist**. To be reasonable, it will be assumed, that the customer records from the external application are already available as individual XML files.
After receiving the Account message, a Fetch message will be used to establish whether the Account in the CRM system already exists. The primary key for finding the customer is the customer number. In our case, it is the **accountnumber**.
Therefore, the Account message has to be mapped to a Fetch message. The Fetch Request should look something like this:

```
<ns0:fetch mapping="logical" xmlns ="http://Roedl.BizTalk.Schemas.MSCRM/Fetch">
   <ns0:entity name="account">
      <ns0:all-attributes />
      <ns0:filter type="and">
         <ns0:condition attribute="accountnumber" operator="eq" value="4711" />
      </ns0:filter>
   </ns0:entity>
</ns0:fetch>
```

The response message will be a message of type **AccountList**.
In the subsequent Decide shape, you will have to evaluate the value of the element ***count***:

count value	Consequence
0	No Account with this number was found. A new Account will be created.
1	One Account with this number was found. The Account will be updated.
>1	More than one Account with this number was found. This is possible, because there are no unique indexes for this field in MS CRM. We won't deal with this case in the example.

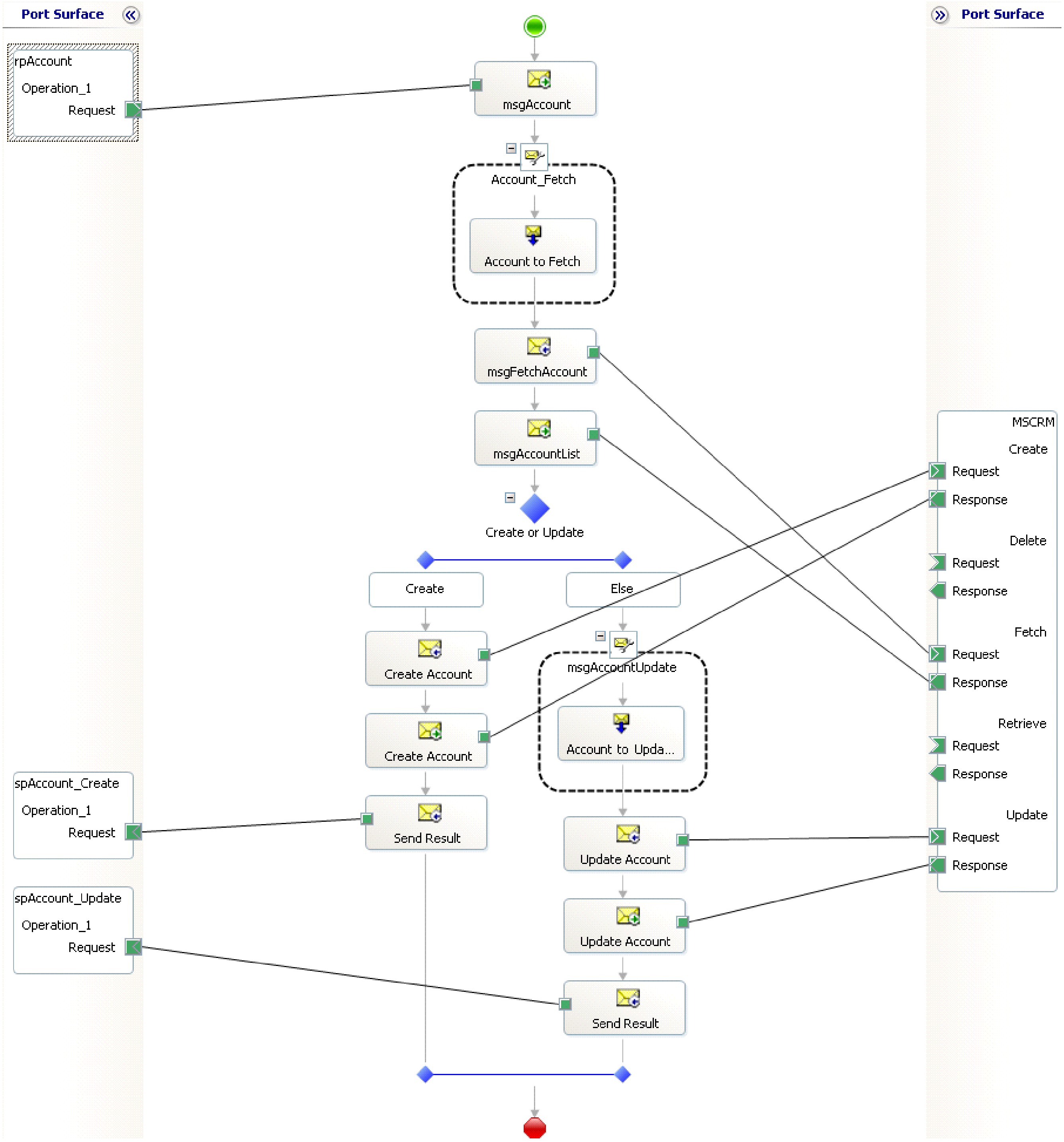
In the Create condition of the Decide shape the following code should be used:

```
System.Convert.ToInt32(msgAccountList.count) == 0
```

In the case of a Create, we can now pass the Account message directly to the Create function. If the record should be updated, we need to include the AccountID in the message. In our example this happens by using a mapping. After that, the Update function can be called with the newly generated Account message.

Since the *R&P MS CRM WCF Adapter for MS BizTalk* is bidirectional, finally, the respective responses of the CRM system will be sent to each send port.

1. Create a new orchestration looking like this:



2. Create an XML message file to create a new record.
   For example: contact_create.xml

```
<?xml version="1.0" encoding="utf-8"?>
  <account xmlns="http://Roedl.BizTalk.Schemas.MSCRM/account">
    <accountnumber>Roedl Consulting AG</accountnumber>
    <name>Roedl Consulting AG</name>
    <address1_name> Roedl Consulting AG</address1_name>
    <address1_line1>Laubanger</address1_line1>
    <address1_line2>23</address1_line2>
    <address1_city>Bamberg</address1_city>
    <address1_country>D</address1_country>
    <address1_postalcode>96052</address1_postalcode>
  </account>
```

Attention:
Take care of the namespace. It has to be the same as given in the Contact schema.

3. Copy the file into the input folder of the receive port ***account_Create***, e.g. *C:\FileDrop\Account\Create\In*.

4. After a short while, an XML file will appear in the output folder of the send port ***account_Create***, e.g. *C:\FileDrop\Account\Create\Out*. The content should look something like this:

```
<?xml version="1.0" encoding="utf-8"?>
  <account xmlns="http://Roedl.BizTalk.Schemas.MSCRM/account">
    <accountnumber>Roedl Consulting AG</accountnumber>
    <name>Roedl Consulting AG</name>
    <address1_name> Roedl Consulting AG</address1_name>
    <address1_line1>Laubanger</address1_line1>
    <address1_line2>23</address1_line2>
    <address1_city>Bamberg</address1_city>
    <address1_country>D</address1_country>
    <address1_postalcode>96052</address1_postalcode>
    <accountid>733ef35b-1947-db11-acc2-0050da4615a6</accountid>
  </account>
```

5. The CRM system has created the record and provides the **contactid**. This can be processed directly in the orchestration.

6. Modify the message, for example as described below, and send it to the input folder of the receive port ***Account_Create***, e.g. *C:\FileDrop\Account\Create\In*.

```
<?xml version="1.0" encoding="utf-8"?>
  <account xmlns="http://Roedl.BizTalk.Schemas.MSCRM/account">
    <accountnumber>Roedl IT Consulting GmbH</accountnumber>
    <name>Roedl Consulting IT Consulting GmbH</name>
    <address1_name> Roedl Consulting AG</address1_name>
    <address1_line1>Laubanger</address1_line1>
    <address1_line2>23</address1_line2>
    <address1_city>Bamberg</address1_city>
    <address1_country>D</address1_country>
    <address1_postalcode>96052</address1_postalcode>
    <accountid>733ef35b-1947-db11-acc2-0050da4615a6</accountid>
  </account>
```

7. After a short while, an XML file will appear in the output folder of the send port ***account_Create***, e.g. *C:\FileDrop\Account\Create\Out*.

WCF XRM Adapter Installation under 64 Bit

Background

Currently the XRM adapter installation process does not support 64 bit. Nevertheless the adapter itself has been compiled for any CPU which means the assembly will JIT to 64 bit code when loaded into a 64 bit process and 32 bit code when loaded into a 32 bit process. The installer only registers the assemblies in the 32 bit machine configuration, which makes it unusable under 64 bit for BizTalk.
However as long as there is no update to fix that issue, there is an easy manual process to fix that.

Manual Registration Instruction

It would be useful to use the entries in the 32 bit machine configuration as an template via copy and paste. Remark: You will need administrative rights to change the machine configuration file!
Copy the template:

1.) Browse to folder: `%Windows Path%\Microsoft.NET\Framework\v.4.0.xxxxx\Config` and open the file `machine.config`

2.) Navigate to the node `configuration\system.serviceModel` and find the entries **`XrmAdapter`** under `<bindingElementExtension>` and **`xrmBinding`** under `<bindingExtension>`

   a. The entry XrmAdapter should look like (depending on installed version):
      ```
      <add name="XrmAdapter"
      type="Roedl.Adapters.XRM.XrmAdapterBindingElementExtensionElement,
      Roedl.Adapters.XRM.XrmAdapter, Version=1.4.xx.xx, Culture=neutral,
      PublicKeyToken=f06782905cde877e" />
      ```
   b. The entry xrmBinding should look like (depending on installed version):
      ```
      <add name="xrmBinding"
      type="Roedl.Adapters.XRM.XrmAdapterBindingCollectionElement,
      Roedl.Adapters.XRM.XrmAdapter, Version=1.4.xx.xx, Culture=neutral,
      PublicKeyToken=f06782905cde877e" />
      ```

Paste to the 64 bit machine configuration:

1.) Browse to folder: `%Windows Path%\Microsoft.NET\Framework64\v.4.0.xxxxx\Config` and open the file `machine.config`

2.) Navigate to the node `configuration\system.serviceModel` and append under `<bindingElementExtension>` the copied **`XrmAdapter`** entry.

3.) Navigate to the node `configuration\system.serviceModel` and append under `< bindingExtension >` the copied **`xrmBinding`** entry.

4.) Save the machine.config file (at this point administrative privileges are needed)

Now you should have the XRM Adapter fully accessible under 64 bit available within the BizTalk Administration Console after the next Restart of the Console.

⚠ **Important Note for Visual Studio:**
Please do not forget also to install the Microsoft BizTalk Adapter Pack for 32 bit (x86), since its crucial for the use under Visual Studio schema generation (the process runs only under 32 bit).

Supported Functions (28.11.2011)

CRM-Methods

- AddItemCampaign
- AddItemCampaignActivity
- AddMemberList
- AddProductToKit
- Assign
- CalculateActualValueOpportunity 1.3.26
- CalculateTotalTimeIncident 1.3.26
- CloseIncident 1.3.26
- CopyCampaign 1.3.26
- CopyDynamicListToStatic 1.3.26
- CopyMembersList 1.3.26
- CancelContract
- CancelSalesOrder
- CloseQuote
- CheckIncomingEmail
- CheckPromoteEmail
- ConvertKitToProduct
- ConvertProductToKit
- DeleteAuditData 1.3.26
- DeprovisionLanguage 1.3.26
- ExecuteWorkflow
- FulfillSalesOrder
- GetInvoiceProductsFromOpportunity 1.3.26
- GetTrackingTokenEmail
- GetQuantityDecimal
- GetReportHistoryLimit 1.3.26
- IsValidStateTransition 1.3.26
- InstallSampleData
- LockInvoicePricing
- LockSalesOrderPricing
- LoseOpportunity
- PublishAllXml
- PublishXml
- PublishDuplicateRule 1.3.26
- ProvisionLanguage 1.3.26
- ReassignObjectsOwner
- ReassignObjectsSystemUser
- Recalculate
- RemoveItemCampaign
- RemoveItemCampaignActivity
- RemoveMemberList
- RemoveParent
- RemovePrivilegeRole
- RemoveProductFromKit
- RemoveSolutionComponent 1.3.26
- RetrieveDeploymentLicenseType 1.3.26
- RetrieveExchangeRate
- RetrieveInstalledLanguagePacks
- RetrieveInstalledLanguagePackVersion
- RetrieveLicenseInfo
- RetrieveVersion
- SendEmail
- SendFax
- SetParentBusinessUnit
- SetParentSystemUser
- SetState
- UninstallSampleData
- UnlockInvoicePricing
- UnlockSalesOrderPricing
- WhoAmI
- WinOpportunity

XRM-Methods

- Associate
- CanBeReferenced
- CanBeReferencing
- CanManyToMany
- DeleteAttribute
- DeleteEntity
- DeleteOptionSet
- DeleteOptionValue
- InsertOptionValue
- UpdateOptionValue
- Disassociate

Contact:
Roedl Consulting AG
Laubanger 23
96052 Bamberg
Germany
fon: ++49 951 96619 0
fax: ++49 951 96619 11
mail: bamberg@roedl.de
web: http://www.roedl.com/integration

Features and specifications are subject to change.

© by Rödl Consulting AG. All rights reserved.

Reproduction in any manner without prior approval of Rödl Consulting AG is strictly forbidden.

All trademarks and trade names used in this document are the property of their respective companies and manufacturers. The Rödl Consulting AG disclaims any proprietary interest in trademarks and trade names other than its own.