BizTalk Server 2006 R2: WCF Send Adapters

Objectives

WCF Send Adapter Architecture

WCF Consumption Wizard

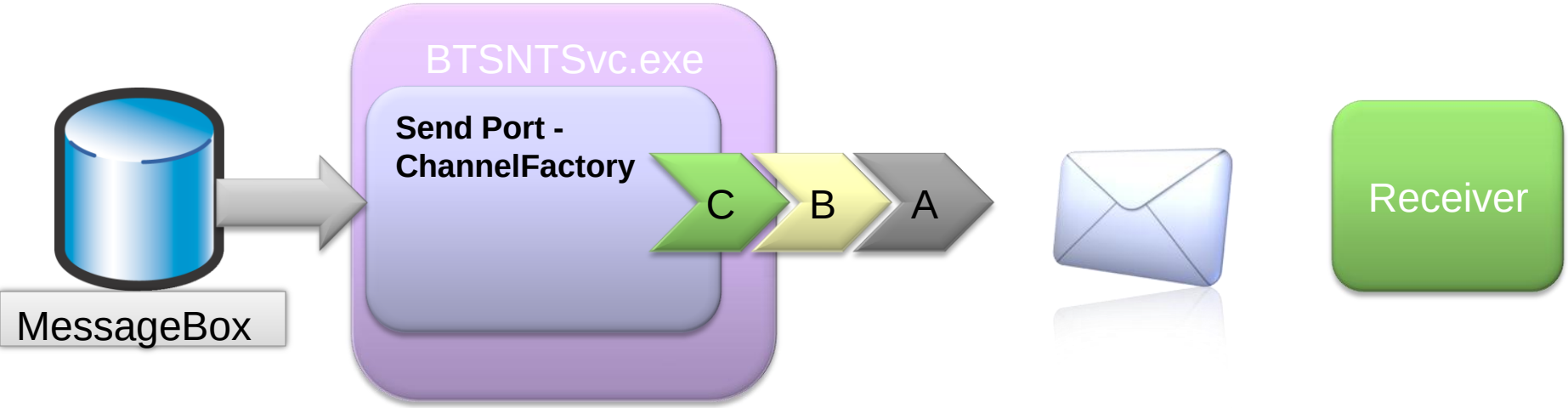
Programming WCF Headers

WCF Send Adapters

- **Each receive adapter has a corresponding Send Adapter**
  - Except WCF-CustomIsolated
- **Send adapters build up a channel stack dynamically**
  - Using generic contract definitions and ChannelFactory
- **Channel stack built based on Send Port configuration**
- **No way (or need) to specify contract**
  - Outbound message just has to match remote endpoint
  - Policies need to match

Configuring Send Port

- **WCF adapter Send Port configuration almost identical to Receive Location**
  - WCF's programming mode l is symmetrical - same programming model for both senders and receivers



Action Mapping

General Tab

- **Send Adapter needs correction Action header for outgoing call**
- **Operation name must be found in Message Context**
  - Automatic for messages from Orchestration
- **Mapping generated by WCF consuming wizard (more on this later)**
  - Must generate by hand if not using wizard

Propagate fault message

- **Each Send Adapter has this setting**
  - For Solicit-Response Send Ports only
- **If checked (the default) fault messages are routed to the subscriber**
- **If unchecked – normal BizTalk Nack message is sent**
  - Orchestration Port delivery notification can kick in

MessagesTab

Outbound credentials

- **All WCF Send Adapters (except WCF-NetNamedPipes) support using SSO for UserName credentials**
  - UI either on Credentials tab (WCF-WsHttp, WCF-BasicHttp, WCF-Custom) or Security (WCF-NetTcp, WCF-NetMsmq) tab
  - Can configure a static username/password combination as well
- **Host instance credentials used when Windows security is used**

Proxy configuration

- **Both Http protocol adapters allow for configuration of Proxy**
  - Make sure to use hostname instead of “localhost” for local proxies

Proxy Tab

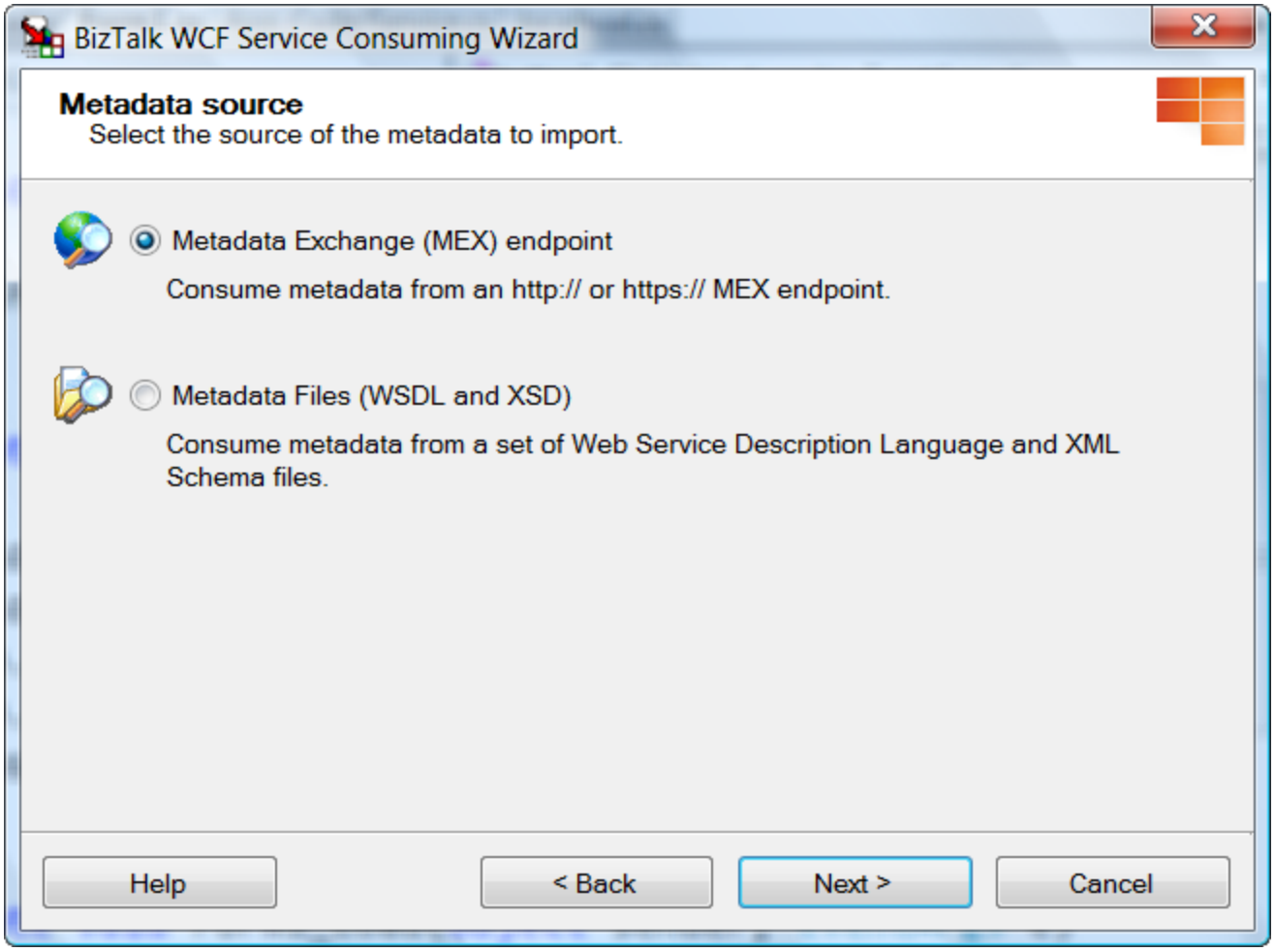
Send Adapter Messaging Tab

- **Same options as Receive Adapter configuration (but reversed)**
- **Outbound message can be**
  - Body (default)
  - Template
- **Response message for two-way ports can send to the MessageBox**
  - Body (default)
  - XPath
  - Envelope

WCF Service Consuming Wizard

- **Generates all the BizTalk artifacts necessary for calling a services from an Orchestration**
  - Can be used against any Web Service definition (WSDL) not just WCF based service endpoints
- **Orchestration could be configured manually**
  - No proxy is generated (different than SOAP Adapter)
- **You can do messaging only with WCF adapter**
  - Schemas from wizard can be used separately

Consuming wizard



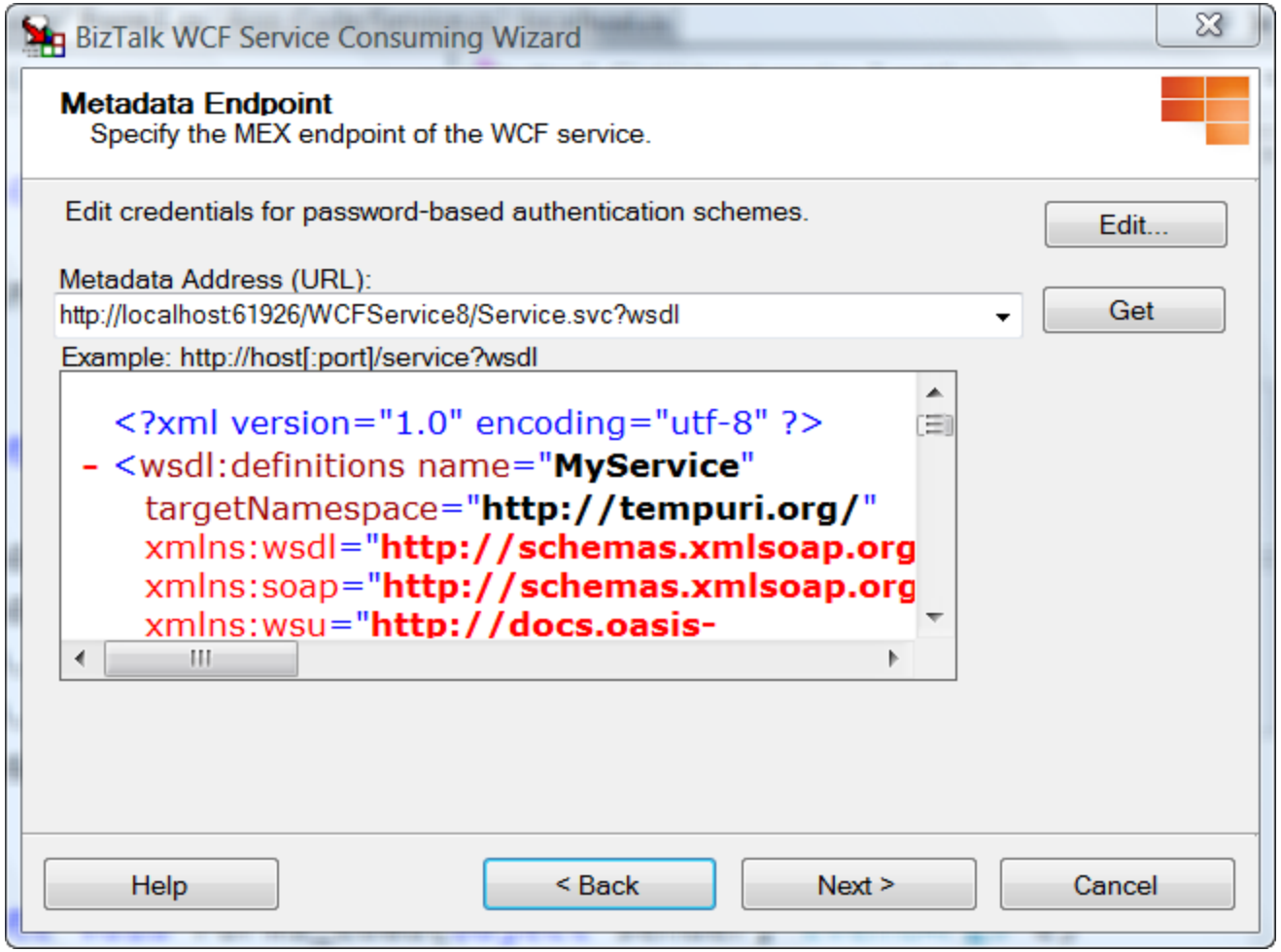
MEX

- Point to mex or ?WDSL Url

Files

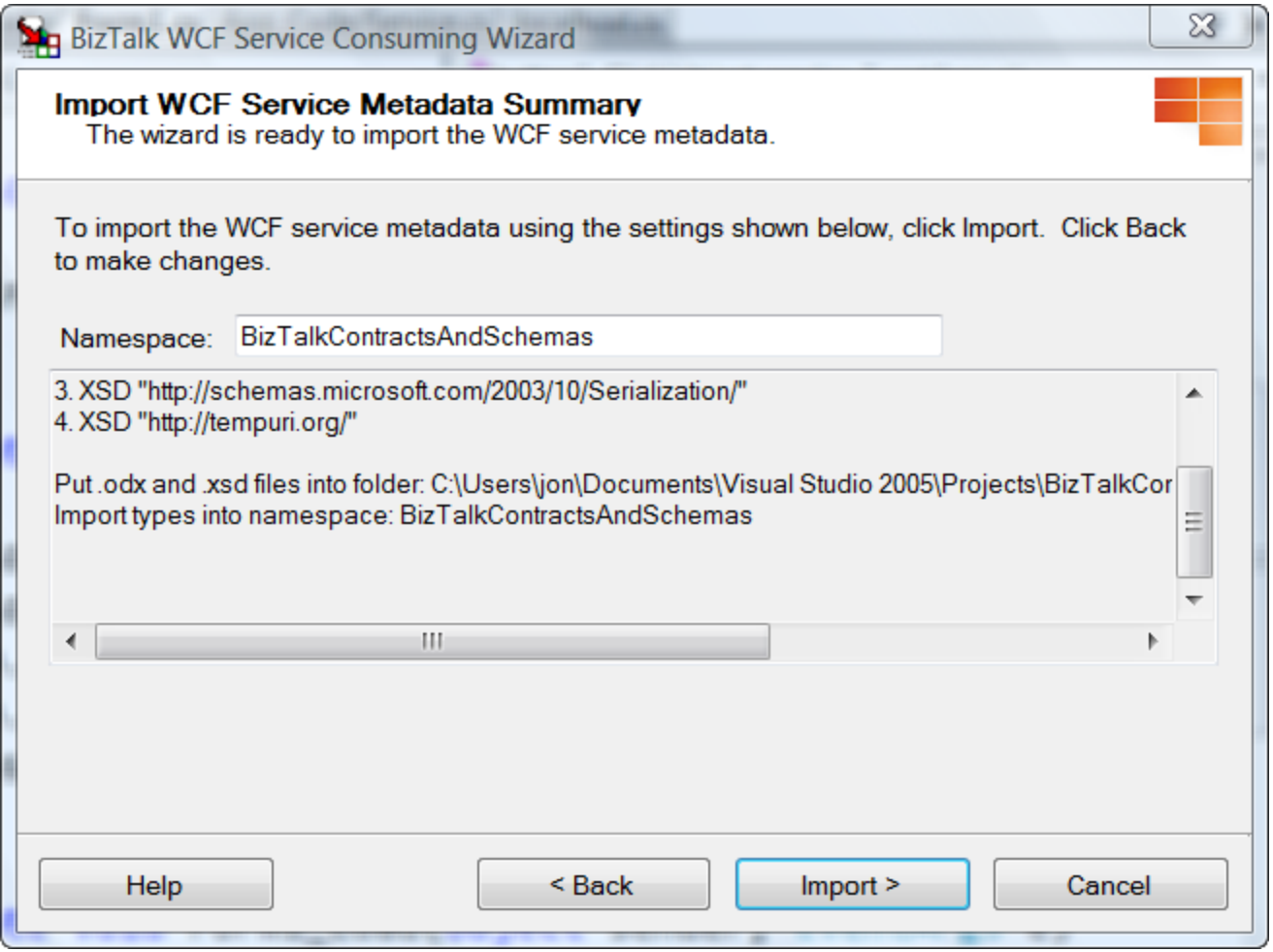
- Point at locally created files

WSDL page



- Enter Url
- Click “Get”
- “Next” to continue

Summary page



- **Change the .NET Namespace which will be used**

Generated artifacts

ODX

- Port Types
- Multipart Message Types

Schemas

- Number depends on imported metadata
- Custom (based on contract)
- WCF specific

Binding files

- Can be one or two
- One if WSDL matches concrete adapter (i.e. WCF-WsHttp)
- One for WCF-Custom

WCF Property Schema*

Name	Description	Promoted
Action	Operation name	Yes
To	WS-Addressing header	Yes
ReplyToAddress	WS-Addressing ReplyTo	Yes
FromAddress	WS-Addressing From	Yes
InboundHeaders	All of the incoming headers (decrypted)	No
OutboundCustomHeaders	All outgoing headers	No

* Not complete

Programming headers

- **All incoming headers are written to InboundHeaders context property**
- **All out bound headers in OutboundCustomHeaders**
  - Only custom headers WS-* headers are ignored
- **Both properties are a string that represents a full Xml document**
  - headers element is document element
- **Can be read in Orchestration (Expression Shape) or pipeline component**
  - Written in Message assignment or pipeline

Header example

```
//inside of a Message Assignment shape
TheMsg(WCF.OutboundCustomHeaders)  = "<headers><CustomToken>" + str +
"</CustomToken></headers>";
```

```
//inside of a pipeline component
public IBaseMessage Execute(IPipelineContext pc,IBaseMessage msg)
{
    string propName  = "OutboundCustomHeaders";
    string propNS =
        "http://schemas.microsoft.com/BizTalk/2006/01/Adapters/
                                    WCF-properties";
    string headerValue = GenerateHeaderXml();
    msg.Context.Write(propName, propNS, headerValue);
}
```

Summary

- WCF Send Adapter(s) enable communication with remote endpoints
- WCF Consuming Wizard simplifies usage of endpoint which expose metadatas
- WCF Adapter headers are programmable