

BizTalk Server 2006 R2: WCF Adapter Receiving



Objectives

Understand the basic WCF Adapter architecture

See the list of OOB Adapters

How to use the NetTcp, NetNamedPipes, and NetMSMQ receive adapters

How to use Transactions with the WCF Adapter

BizTalk and WCF

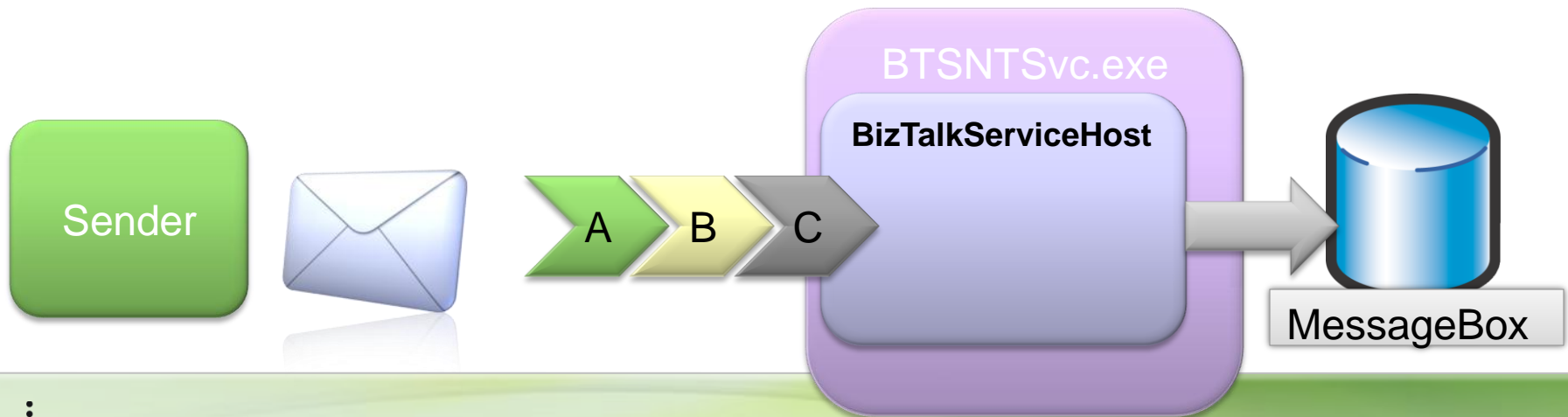
- **WCF fits nicely into the BizTalk architecture**
 - Both have pluggable protocol stacks
- **WCF Receive Adapters expose WCF endpoints**
 - Messages are routed to BizTalk through BizTalk Adapter API
 - HTTP based binding are isolated (run in W3WP.exe)
- **Send adapters build a dynamic Channel Stack**
 - No proxy needed in Orchestration or Send Port ala SOAP Adapter
- **Configuration stored with Receive Location or Send Port instead of application configuration file**
 - Uses same WCF configuration for supported features
- **One adapter per supported OOB Binding**
 - Simplifies usage for the most common cases
 - “Strongly Typed” Adapter supports validation in configuration UI

WCF Adapters

| Adapter | Host Type | Description |
|--------------------|------------|--|
| WCF-NetTcp | In-process | Uses the WCF NetTcpBinding, allows integration with WCF clients |
| WCF-WsHttp | Isolated | Uses the WCF WsHttpBinding, allows integration with compatible WS-* stacks |
| WCF-NetNamedPipe | In-process | Uses WCF NetNamedPipes binding – single machine binding |
| WCF-BasicHttp | Isolated | Uses the WCF BasicHttpBinding, allows integration with WS_BaseProfile1 compatible stacks |
| WCF-NetMsmq | In-process | Uses the WCF NetMsmqBinding, allows integration with WCF MSMQ clients (MSMQ adapter for legacy - msmqIntegration Binding disallowed) |
| WCF-Custom | In-process | Allows you to use any Binding |
| WCF-CustomIsolated | Isolated | Allows you to use any Binding in an Isolated Host (only receive) |

Receive Adapter Architecture

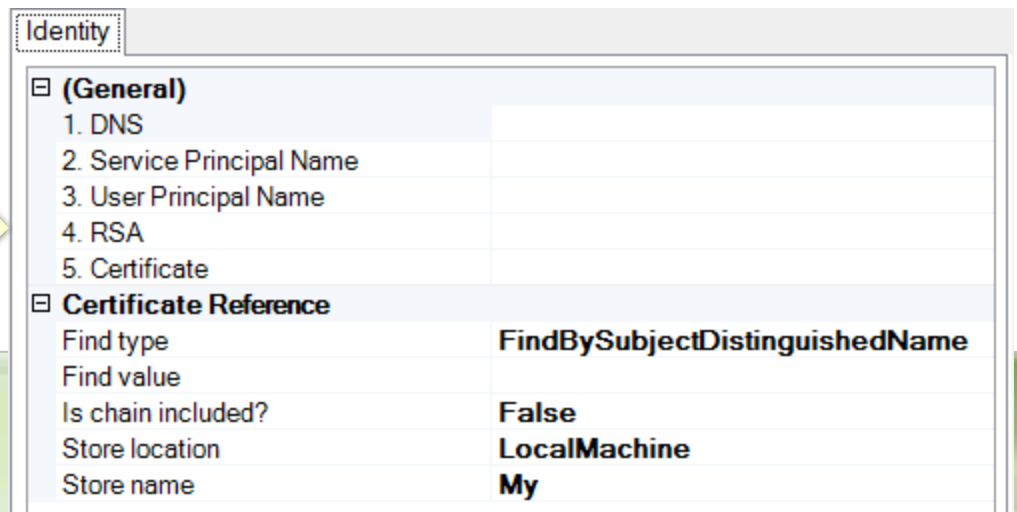
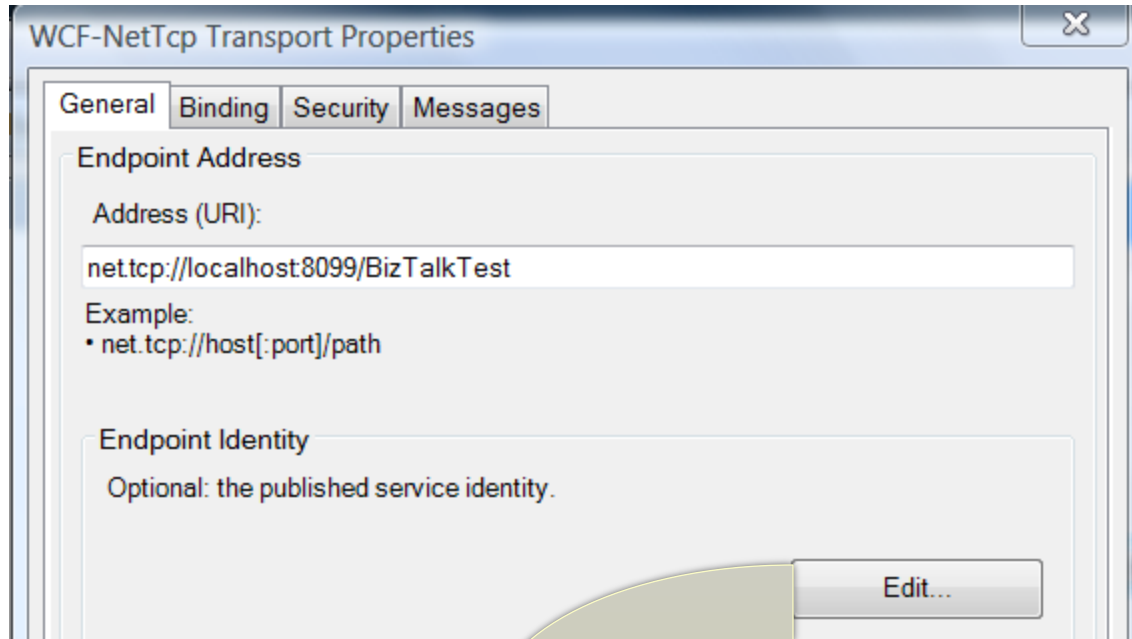
- **BizTalk receive adapter creates a ServiceHost derived instance for each Receive Location**
 - Public WCF extensibility point
- **Rather than implementing particular contracts, endpoints expose loosely typed contracts**
 - Behaves like a SOAP intermediary
 - Avoids double serialization (known issue with SOAP Adapter)



WCF-NetTcp Receive Adapter

- **BizTalkServiceHost will use NetTcpBinding for its endpoint**
- **Use cases:**
 - For Windows based internal services
 - Best performance for WCF across machines
- **Clients must be WCF enabled clients**
 - Not an interoperable binding
- **Arguably the most direct way to access the MessageBox**

Configuration for NetTCP



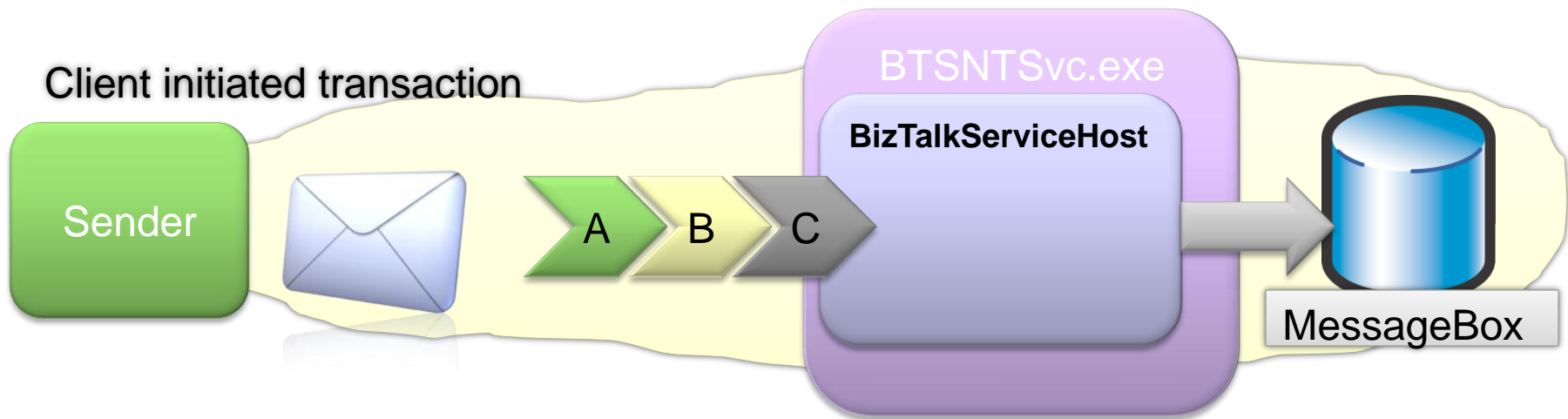
WCF-NetTcp binding config

| | |
|---------------|--|
| Open timeout | <ul style="list-style-type: none">• Timeout for ServiceHost.Open |
| Send timeout | <ul style="list-style-type: none">• Timeout for a send operation (full time if request-response) |
| Close timeout | <ul style="list-style-type: none">• Timeout for closing the ServiceHost |
| Transactions | <ul style="list-style-type: none">• OleTransactions• WSAAtomic Transactions |
| Lease Timeout | <ul style="list-style-type: none">• How long to hold connections (lower is generally better in load-balancing scenarios) |
| Max calls | <ul style="list-style-type: none">• 200 default• Set to 0 for Int32.MaxValue |

| | | | |
|---|---------|-----------------|----------|
| General | Binding | Security | Messages |
| Open timeout (hh:mm:ss): | | 00:01:00 | |
| Send timeout (hh:mm:ss): | | 00:01:00 | |
| Close timeout (hh:mm:ss): | | 00:01:00 | |
| Maximum received message size (bytes): | | 65,536 | |
| Transactions Flow transactions using the specified protocol. | | | |
| <input type="checkbox"/> Enable transactions | | | |
| Transaction protocol: | | OleTransactions | |
| Connection pool settings Use these settings to optimize performance in load-balanced scenarios. | | | |
| Lease timeout (hh:mm:ss): | | 00:05:00 | |
| Service throttling behavior | | | |
| Maximum concurrent calls: | | 200 | |

Configuring Transactions

- **Only enabled for One-Way Receive Locations**
 - Holds true for all WCF adapters that support transactions
- **Transaction to MessageBox will be joined with flowed transaction**



Transactional use case effects

- **Client or other RM aborts**

- No fingerprints of transaction in BizTalk (consistent with ACID properties of transactions)

- **BizTalk aborts**

- Routing failure in BizTalk
 - No resumable message (again think ACID)

WCF-NetTcp Security config

Security Mode

- None
- Transport (default)
- Message
- TransportWithMessageCredential

Transport Cred

- None
- Windows (default)
- Certificate

Transport Protection Level

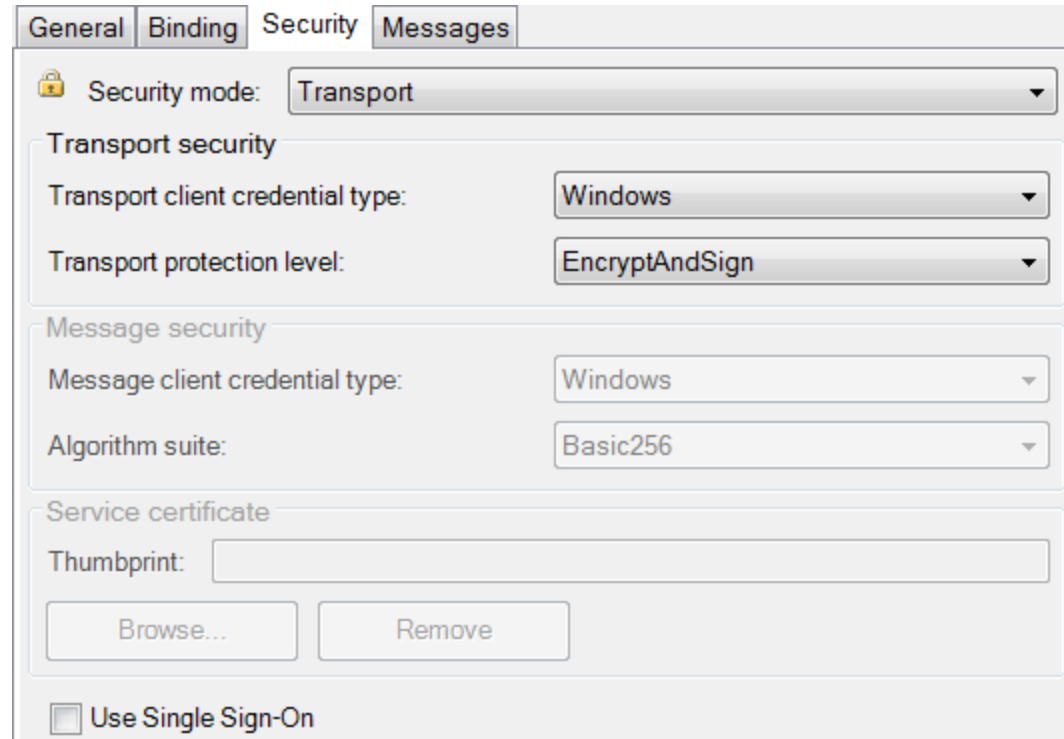
- None
- EncryptAndSign (default)
- Sign

Message – client credential type

- None
- Windows
- UserName
- Certificate

Algorithm

- Basic256 (default)
- <Too many to list>



The screenshot shows the 'Security' tab of the WCF-NetTcp configuration window. The 'Security mode' is set to 'Transport'. Under 'Transport security', the 'Transport client credential type' is 'Windows' and the 'Transport protection level' is 'EncryptAndSign'. Under 'Message security', the 'Message client credential type' is 'Windows' and the 'Algorithm suite' is 'Basic256'. The 'Service certificate' section has an empty 'Thumbprint' field and 'Browse...' and 'Remove' buttons. The 'Use Single Sign-On' checkbox is unchecked.

| Tab | Property | Value |
|----------|----------------------------------|--------------------------|
| General | Security mode | Transport |
| | Use Single Sign-On | <input type="checkbox"/> |
| Binding | Transport client credential type | Windows |
| | Transport protection level | EncryptAndSign |
| Security | Message client credential type | Windows |
| | Algorithm suite | Basic256 |
| Messages | Service certificate thumbprint | |
| | Buttons | Browse... Remove |

Configuring Security

- **Basically the same as raw WCF configuration**
 - UI does enforce validation
- **Transport (the default) is generally the most performant**
- **Can enable SSO if a Windows Principal will be available**

WCF-NetTcp Message config

General Binding Security Messages

Inbound BizTalk message body
Specify the source of the inbound BizTalk message body.

☐ Envelope -- entire <soap:Envelope>
☒ Body -- contents of <soap:Body> element
☐ Path -- content located by body path

Body path expression:

Node encoding:

Outbound WCF message body
Specify the source of the outbound WCF message body.

☒ Body -- BizTalk response message body
☐ Template -- content specified by template

XML:

Body:

Error handling

☐ Suspend request message on failure
☐ Include exception detail in faults

WCF Adapter Message Processing

- **Messages configuration tab exposes functionality common to all WCF adapters**
- **WCF Adapter infrastructure allows fine-grained control over what part of the incoming message gets published to MessageBox**
- **Default is the XML which is the child of the soap:Body element**
 - Same as the SOAP Adapter

Slicing the message

- **Select “Envelope”**
 - Whole soap:Envelope element is published to the MessageBox
 - No processing (if signed/encrypted it will keep those attributes)
- **Combining “Envelope” with XmlReceive pipeline will cancel out functionality**
 - soap:Envelope is registered with BizTalk as a BizTalk Envelope Schema
 - XmlDissassembler strips everything but first child of soap:Body

Specify XPath

- **Enter a valid XPath expression to pull data from the message**
 - Relative to soap:Body only
- **Must not use prefixes**
- **Node encoding controls what processing happens on the node after selection**
 - Node must match encoding or an empty message is published

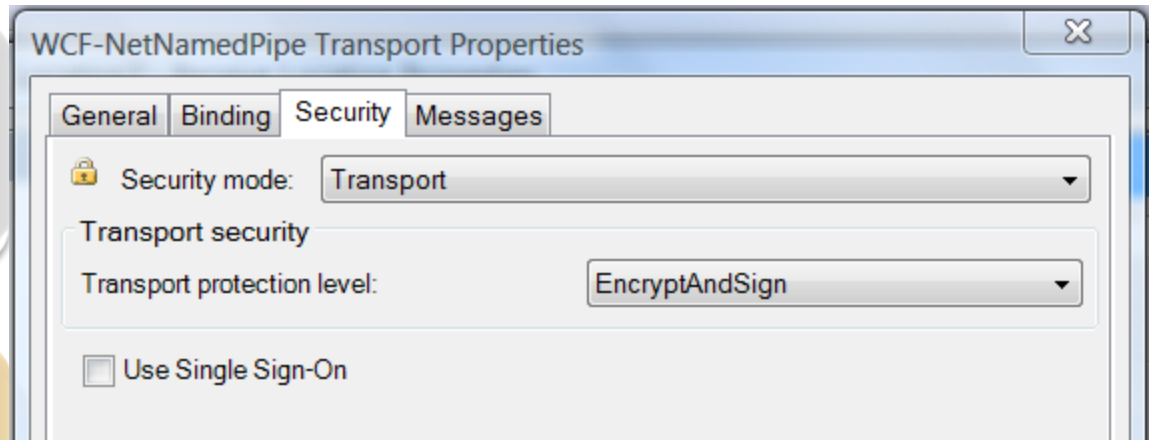
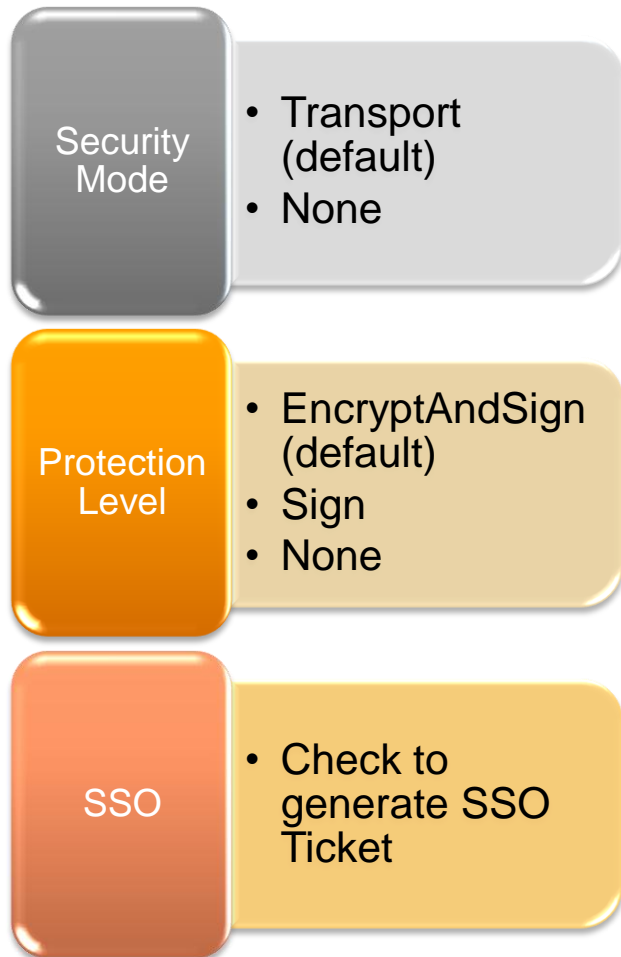
XPath Encoding Values

| Value | Effect |
|--------|--|
| Xml | OuterXml value of selected node goes to MessageBox |
| Base64 | Binary value is published |
| Hex | Binary value is published |
| String | String value will be published (if Xml encoded it will be converted back to Xml) |

Outbound message

- **If Request-Response Receive Location, option is enabled**
- **Can specify body of BizTalk message (default)**
- **Can alternately specify the an Xml “template”**
 - Specify with bts-msg-body element the placement of the BizTalk message data
 - Also must specify the encoding

WCF-NetNamedPipes Adapter



**Security the only delta between
NetNamedPipes and NetTcp**

WCF-NetMsmq Adapter

- **Creates an endpoint that will pool an MSMQ queue**
 - Same as a custom WCF application using the NetMsmqBinding
- **Usage model is for integrating with a WCF based client using NetMsmqBinding from a client**
- **One-way MEP only**
- **Host process user must have correct permissions**

WCF-NetMsmq Binding

The screenshot shows the 'WCF-NetMsmq Transport Properties' dialog box with the 'Binding' tab selected. The 'General' tab is also visible. The 'Binding' tab contains the following settings:

- Open timeout (hh:mm:ss):** 00:01:00
- Send timeout (hh:mm:ss):** 00:01:00
- Close timeout (hh:mm:ss):** 00:01:00
- Maximum received message size (bytes):** 65,536
- Transactions:** Specify the type of message queue: transactional or non-transactional.
 - ☒ Transactional
- Message order:** Preserve message order when processing messages.
 - ☐ Ordered processing

Transactions

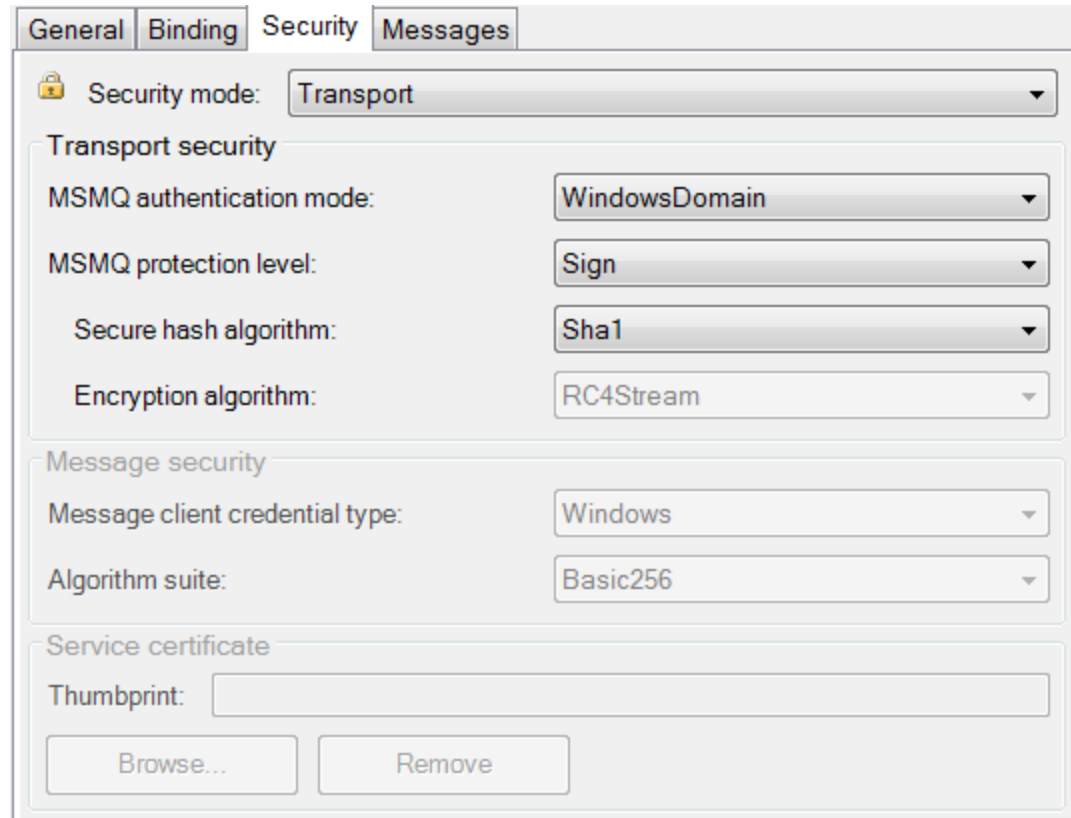
- Checked
- Unchecked - queue not transactional

Message Order

- Check Ordered Processing to support FIFO
- Remember FIFO only applies to the MessageBox

WCF-NetMsmq Security

| | |
|---------------------|---|
| Security Mode | <ul style="list-style-type: none">• Transport (default)• Message• Both• None |
| Transport Security | <ul style="list-style-type: none">• WindowsDomain (default)• Certificate• None |
| Protection Level | <ul style="list-style-type: none">• Sign (default)• EncryptAndSign• None |
| MessageSecurity | <ul style="list-style-type: none">• Windows (default)• UserName• Certificate• None |
| Algorithm Suite | <ul style="list-style-type: none">• Basic256 (default)• <Too Many to List> |
| Service certificate | <ul style="list-style-type: none">• Thumbprint value if using cert for Message |



The screenshot shows the 'Security' tab of the WCF Configuration Wizard. The 'Security mode' is set to 'Transport'. Under 'Transport security', 'MSMQ authentication mode' is 'WindowsDomain', 'MSMQ protection level' is 'Sign', 'Secure hash algorithm' is 'Sha1', and 'Encryption algorithm' is 'RC4Stream'. Under 'Message security', 'Message client credential type' is 'Windows' and 'Algorithm suite' is 'Basic256'. The 'Service certificate' section shows a 'Thumbprint' field and 'Browse...' and 'Remove' buttons.

| | | | |
|---------|---------|----------|----------|
| General | Binding | Security | Messages |
|---------|---------|----------|----------|

Security mode: Transport

Transport security

MSMQ authentication mode: WindowsDomain

MSMQ protection level: Sign

Secure hash algorithm: Sha1

Encryption algorithm: RC4Stream

Message security

Message client credential type: Windows

Algorithm suite: Basic256

Service certificate

Thumbprint:

Browse... Remove

Summary

- **The WCF adapters in BizTalk Server R2 give you powerful control over integrating both WCF and compatible services with BizTalk**
- **WCF Adapters are like all other BizTalk Adapters in that they are message type agnostic**
- **Transactional feature new for WCF/R2 integration**
- **No Contract configuration – use typical BizTalk configuration (pipeline, maps, subscriptions) to manage messages**