BizTalk Server 2006 R2: Sending EDI messages

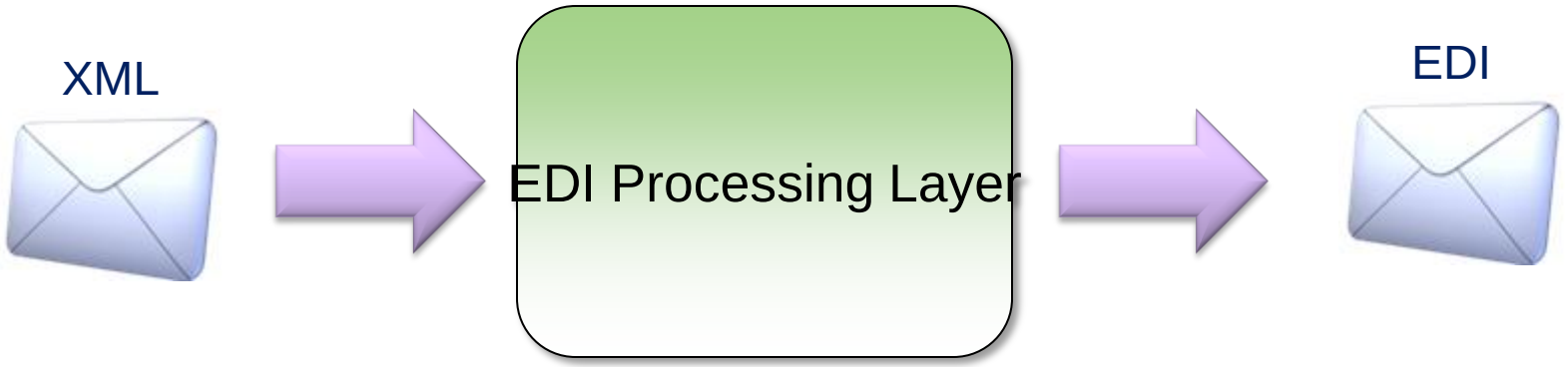
Objectives

Understand how R2 sends EDI messages

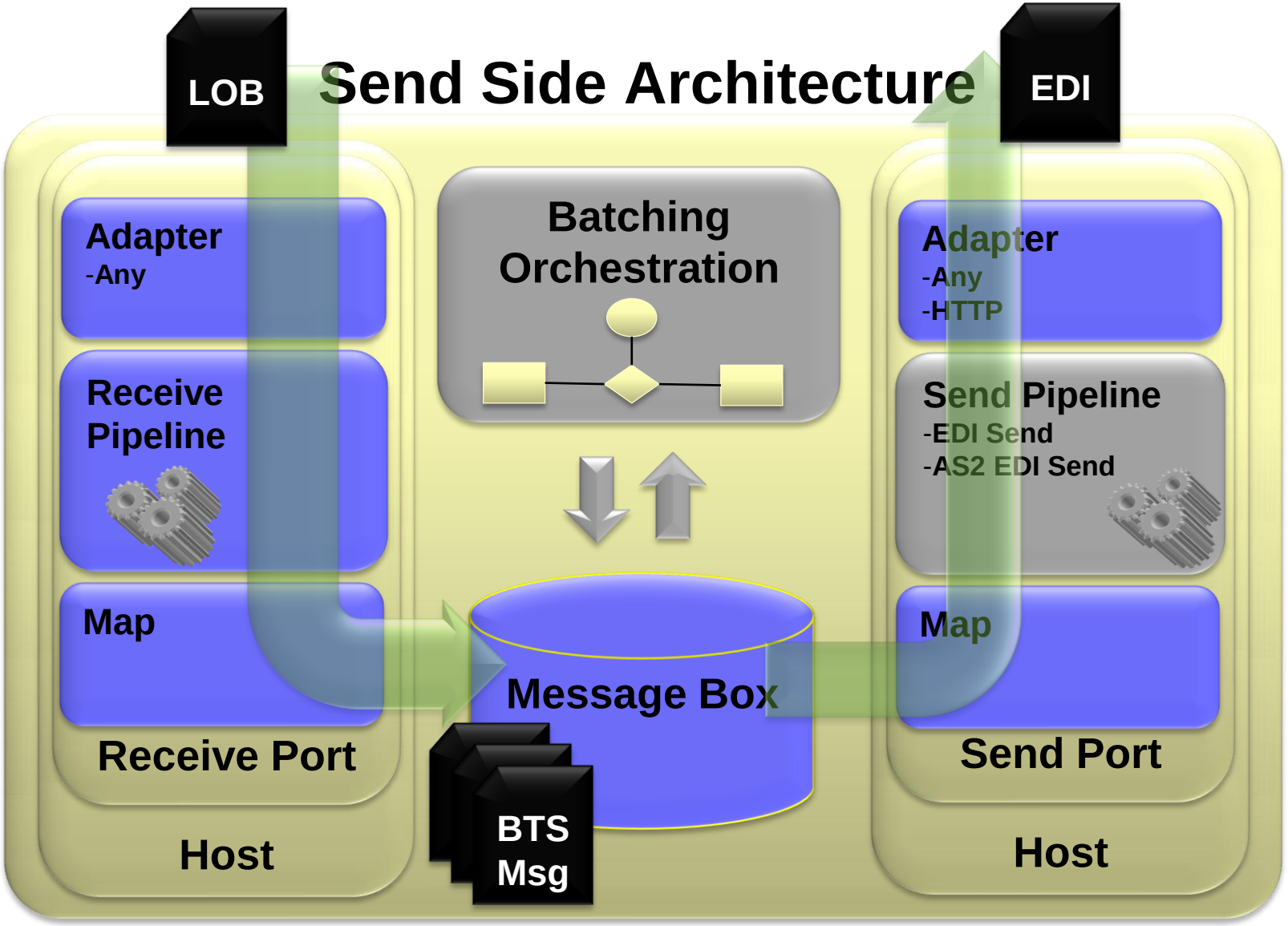
See the EDI batching capabilities of R2

Sending EDI Messages

- **EDI capabilities in R2 enable sending EDI formatted messages**
  - Highly dependent on Party settings
  - Requires XML be formatted as an EDI Schema instance (transaction set)
- **How messages get routed to Send Port is application specific**
  - Can use Batching (more on this later)

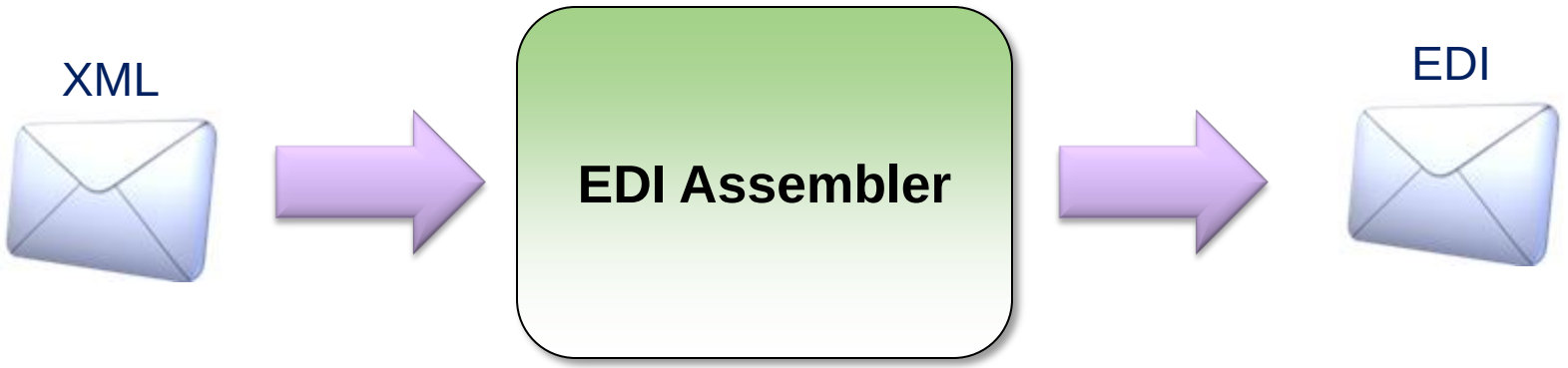


Send Side Architecture



EDI Assembler

- **The EDI Assembler does all the hard work**
  - Turns the XML into an EDI formatted message
  - Generates the EDI Envelope
- **AS2EDISend and EDISend pipelines contain this component**



Schema and Parsing

- **The EDI Assembler first has to determine the schema for the message being processed**
  - Typical BizTalk approach of parsing the Document Element
  - namespaceURI#RootElementName
- **EDI Assembler doesn't rely on MessageType Context Property**
- **Schema must be deployed to the appropriate Application**

http://schemas.microsoft.com/BizTalk/EDI/EDIFACT/2006#EFACT_D96A_INVOIC_EAN008

http://schemas.microsoft.com/BizTalk/EDI/X12/2006#X12_00401_850

Determining the Party

- **EDI Assembler looks at context properties**
  - EDI.DestinationPartyName checked first
- **If EDI.DestinationPartyName is not found Assembler checks for:**
  - EDI.DestinationPartyReceiverIdentifier
  - EDI.DestinationPartyReceiverQualifier
  - EDI.DestinationPartySenderIdentifier
  - EDI.DestinationPartySenderQualifier
- **If properties or match not found the EDI Global Settings are used**
- **Uses the resolved settings to generate the EDI Envelope**
  - Interchange Header and Group

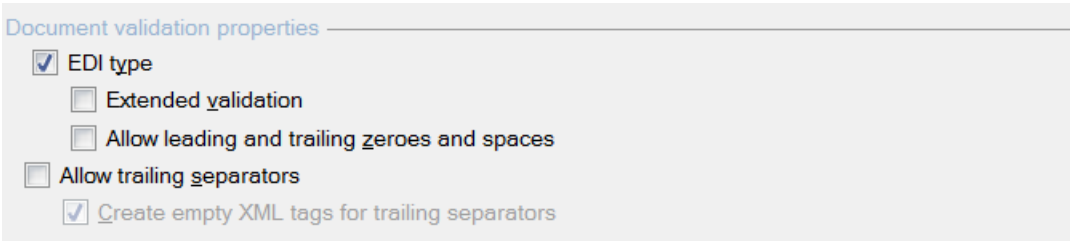
Interchange properties

- **Can set per-party interchange segment settings**
  - ISA or UNA/UNB
- **Headers**
  - GS/ST or UNG/UNH
- **Used by EDI Assembler to create outgoing EDI message**
  - Pushes all transaction sets into headers
  - Can be done per document type

Extended validation

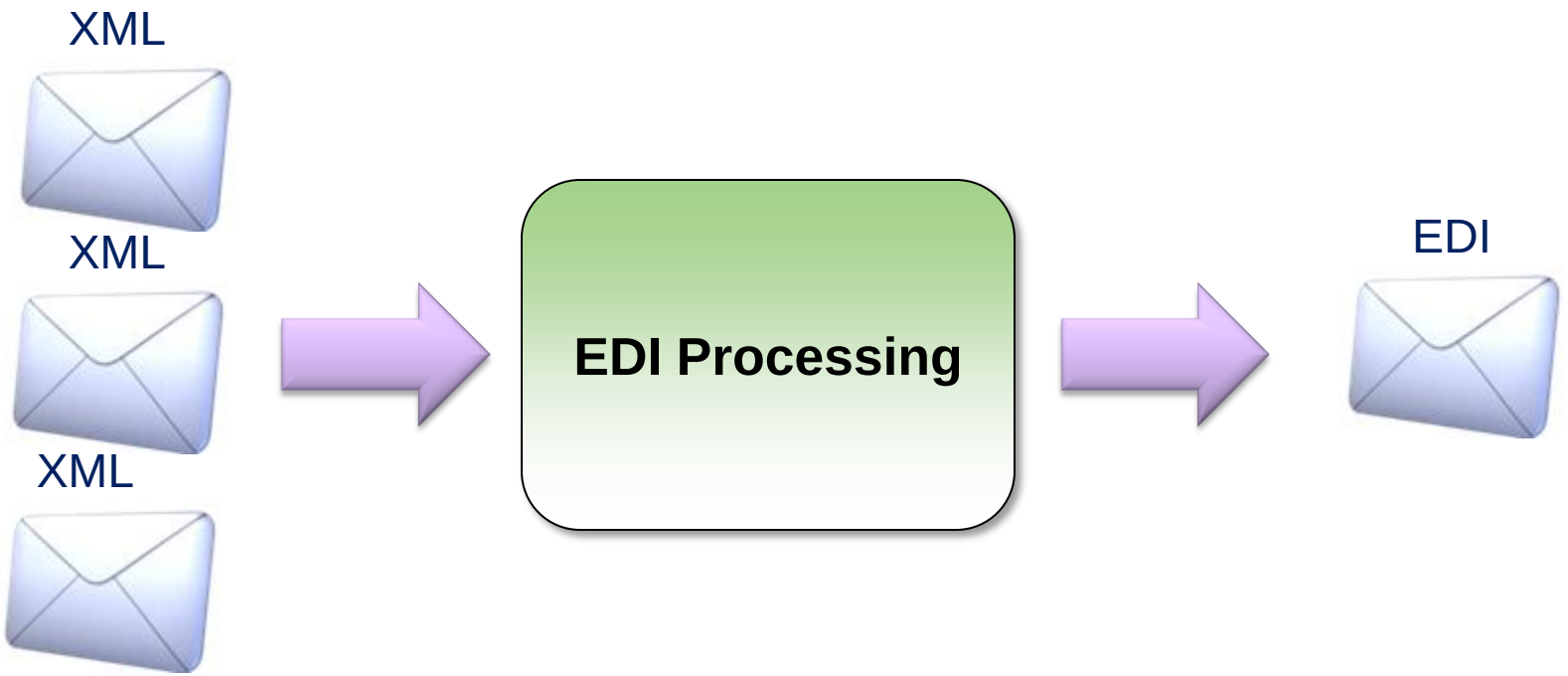
- **EDI Type validation**
  - field length, optionality, and repeat count
- **Extended validation**
  - XSD based validation
- **Allow leading and trailing zeros and spaces**
- **Allow trailing separators**
  - Create empty XML tags

Under Party as Interchange Receiver : Ack Processing and Validation Settings



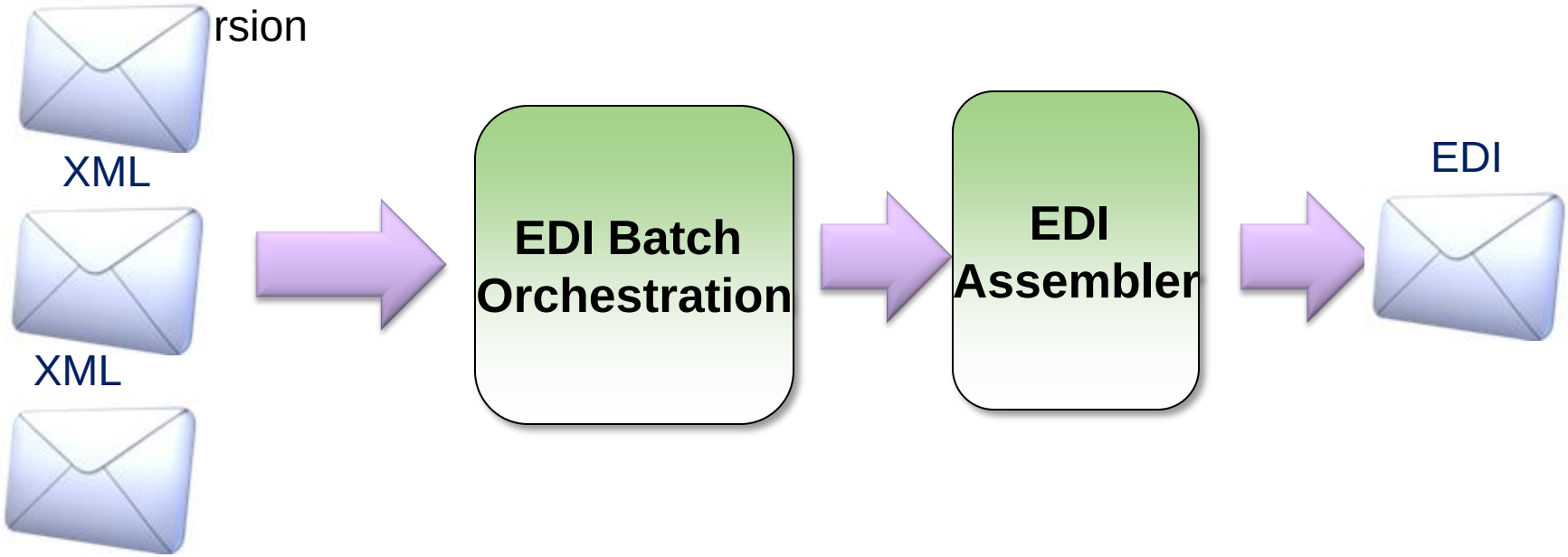
EDI Batching

- **Common scenario to reduce the cost of VANS**
  - Only sent EDI messages when certain conditions occur
  - Now just a common practice



EDI Batch Orchestration

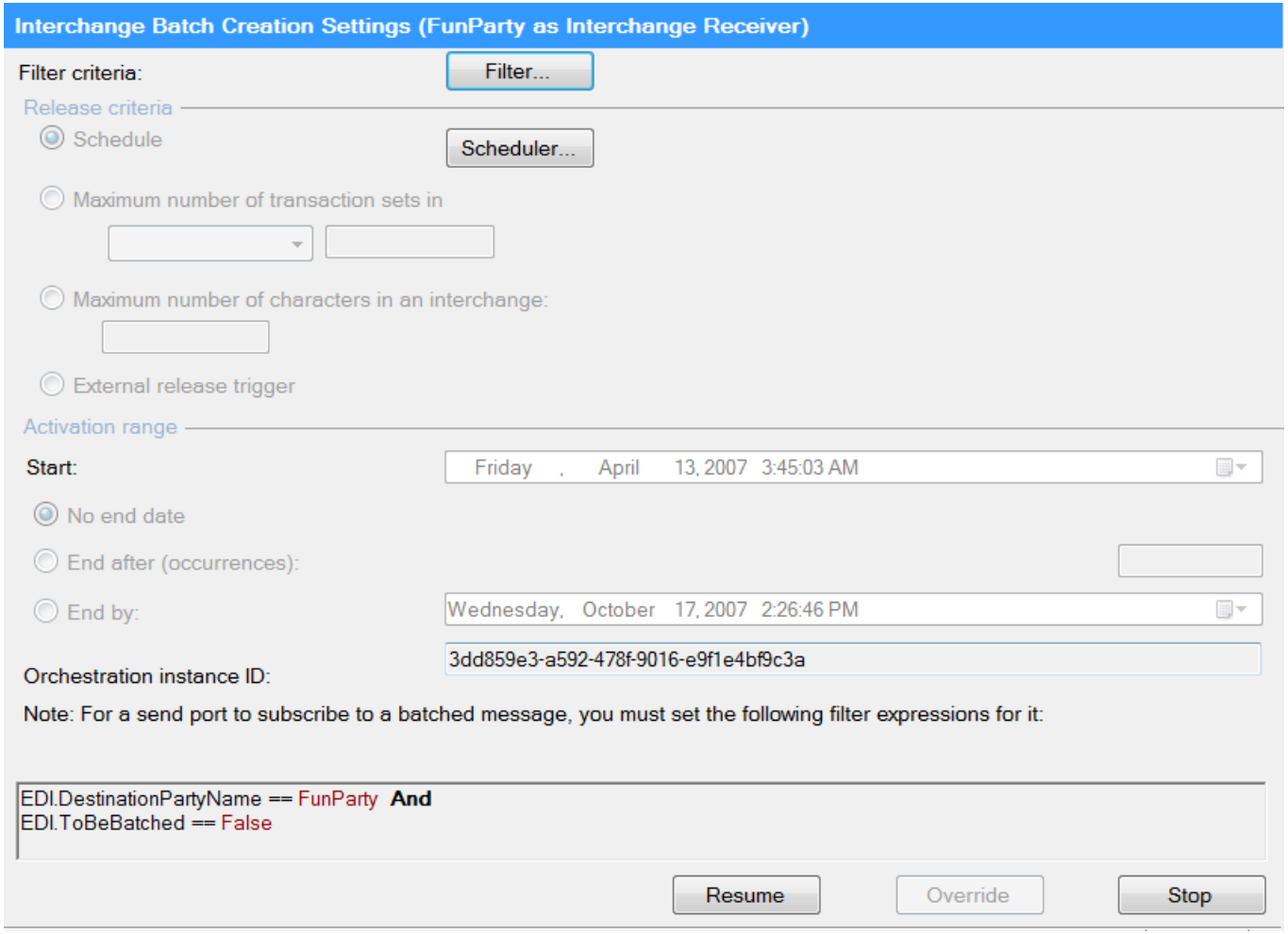
- **EDI functionality in R2 supports batching outgoing EDI messages**
  - Messages are sent to long-running Batch Orchestration
  - One instance per party
  - XML Assembler on Send Port continues to do the XML to EDI conversion



EDI Batching configuration

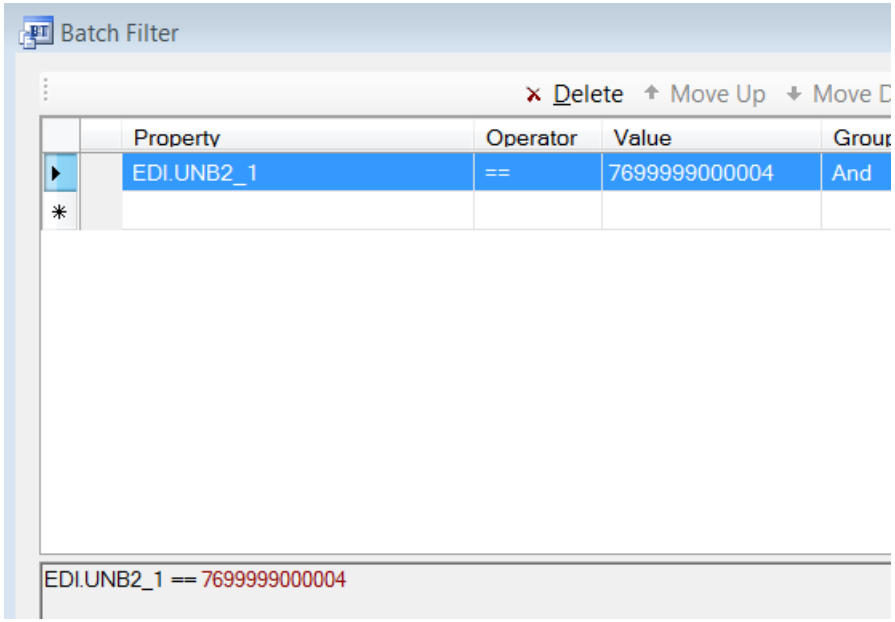
- **Specific to either X12 or EDIFACT party specific configuration**
  - Batch configuration identical
- **Orchestrations in EDI Application needs to be started**
  - Each Party gets a singleton Orchestration instance
  - Instance lifetime controlled via configuration
- **Configure settings**
- **Start Batching**
  - Batching doesn’t start until first message is received

Batch settings



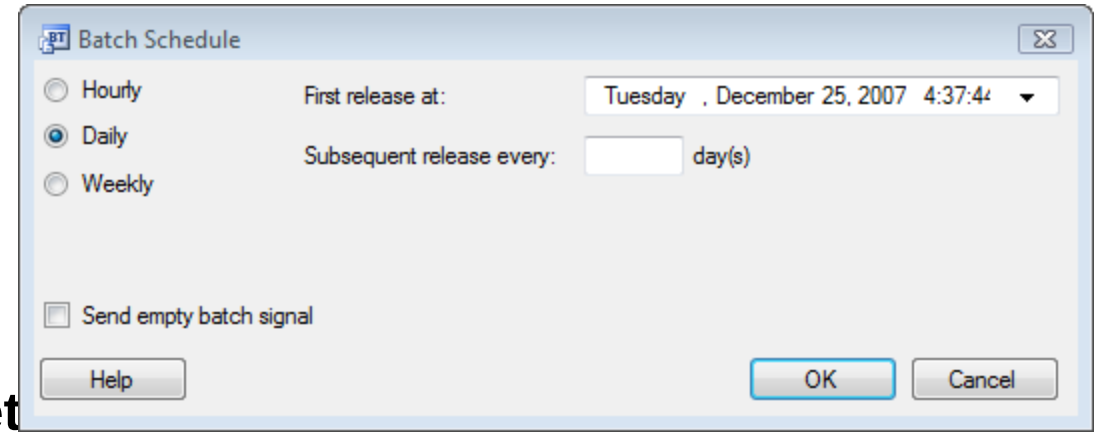
Filter

- **Expression used by Batch Marker Component to filter batch versus non-batch messages**
- **Promotes necessary properties**
  - EDI.DestinationPartyName
  - EDI.ToBeBatched == true
- **All messages that match this filter get routed to Batch Orchestration instance for Party**
  - All messages that match EDI.DestinationPartyName && EDI.ToBeBatched == true get routed to Orchestration
- **Batch Maker component must be used in Receive Pipeline for Filter to work**



Batch release can be based on…

- **Schedule**
  - Hourly
  - Daily
  - Weekly
- **Number of transaction set**
  - In Group or
  - In Interchange
- **Number of characters**
  - In Interchange
- **External release trigger**
  - Just a special message to the Message Box



Releasing a batch explicitly

- **Adapter not important**
  - XMLReceive or XML Disassembler must be used
- **Batch Orchestration is subscribed to this MessageType + DestinationPartyName**

```
<?xml version="1.0" encoding="utf-8"?>
<ControlMessage xmlns="http://SQLControlMessage.IssueSelect">
        <PAM_Control>
                <DestinationParty>nn</DestinationParty>
                <ActionType>EdiBatchActivate</ActionType>
                <ActionSource></ActionSource>
                <ActionSource>EdiBatchActivate</ActionSource>
                <ActionDateTime>yyyy-mm-
ddThh:mm:ss.sss</ActionDateTime>
                <ToBeBatched>n</ToBeBatched>
        </PAM_Control>
</ControlMessage>
```

BatchMarker

- **Determines the encoding type**
  - From EncodingType context property or the root node
  - Set to false if not receiving EDI messages into pipeline
- **Promotes**
  - DestinationPartyId
  - ToBeBatched
  - ToBeRouted
- **Place this component in a custom Receive Pipeline in the Party Resolution stage**
  - Use pipeline in Receive Location that receives messages you want to be sent to Batching Orchestration

Subscription for Batched messages

- **Send Port or Orchestration can subscribe**
- **EDI.DestinationPartyName==<PartyName> and EDI.ToBeBatched==false**
  - If ToBeBatched==true message would route back to Orchestration

Summary

- **EDI functionality in R2 enables XML to EDI conversion**
- **EDI Assembler is the key component**
- **EDI Assembler uses Party to be resolved in order to add Header and Transaction Set information to message**
- **EDI functionality in R2 supports EDI batching**
  - Pre-built batching Orchestration + pipelines provide functionality