WF and WCF Interceptors for BAM

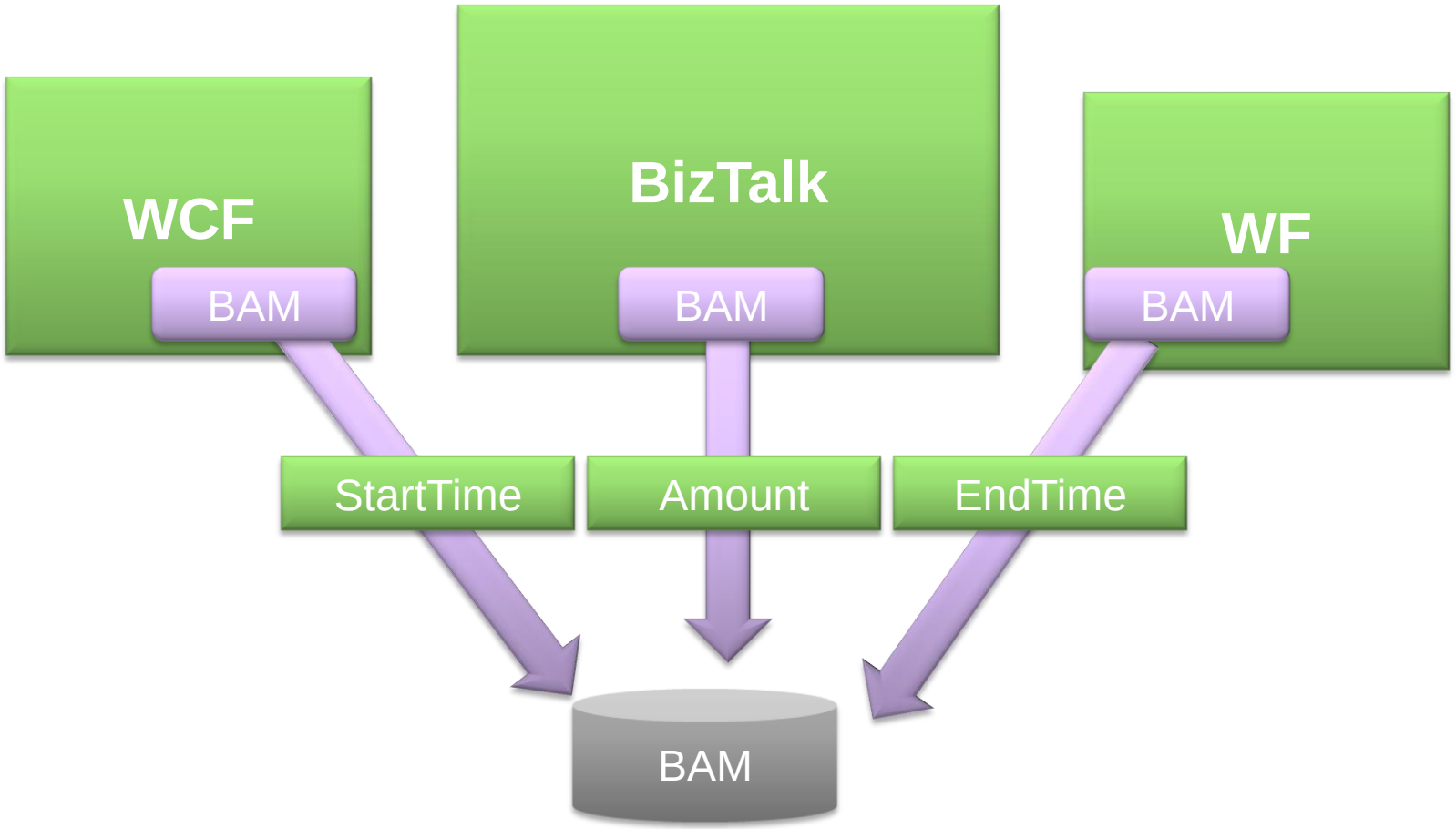
Objectives

- Interceptor structure and basics
- Using the WF interceptor
- Using the WCF interceptor

BAM Interceptors for WF and WCF

- **BizTalk Server ships with two .NET framework interceptors**
  - Windows Workflow Foundation
  - Windows Communication Foundation
- **Both are no-code interceptors**
  - Configure the runtime environment
  - Configure the interceptor
- **Enables declarative end-to-end business level tracking**

Interceptor integration



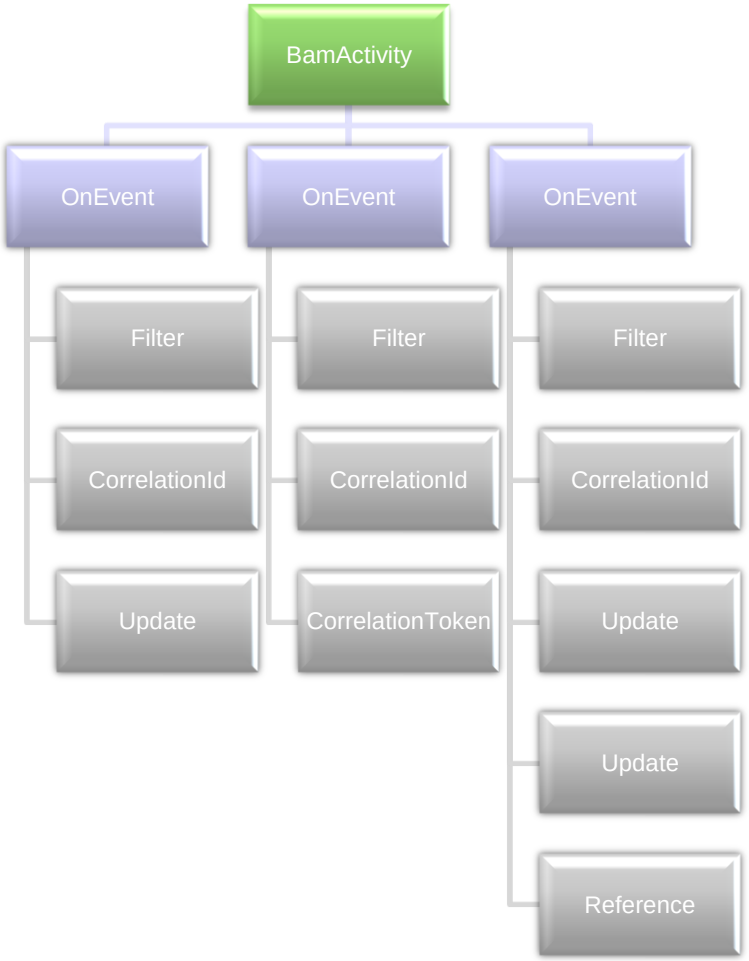
ActivityId	StartTime	Amount	EndTime
Af22f4e11272	1/9/2007 3:50:34 AM	255.55	1/9/2007 3:50:34 AM

Interceptor Configuration (IC)

- **Interceptors are controlled via XML configuration files**
  - Common Schema for all interceptors
  - Specific technology interceptors may add extra elements
  - InterceptorConfiguration is document element
- **No Tracking Profile Editor for these interceptors**
  - Must handwrite IC
- **Can get intelli-sense in Visual Studio by copying schemas**
  - Copy to <Drive>\Program Files\Microsoft Visual Studio 9\Xml\Schemas

BamActivity IC Tree

```
<BamActivity Name="Test">
          <OnEvent Name="EOne">
                    <Filter>
                    </Filter>
                    <CorrelationId>
                    </CorrelationId>
                    <Upate></Upate>
          </OnEvent>
          <OnEvent Name="ETwo">
                    <Filter>
                    </Filter>
                    <CorrelationId>
                    </CorrelationId>
                    <CorrelationToken>
                    </CorrelationToken>
          </OnEvent>
          <OnEvent Name="EThree">
                    <Filter>
                    </Filter>
                    <CorrelationId>
                    </CorrelationId>
                    <Upate></Upate>
                    <Upate></Upate>
                    <Reference></Reference>
          </OnEvent>
</BamActivity>
```



EventSource

- **EventSource is a required element in the IC**
  - maxOccurs="unbounded" (can have multiple EventSources in a single IC)
- **Represents a source of data for a BAM interceptor**

Attribute	Description
Name	Application specific, unique name for the source
Technology	Indicates interceptor type (WF,WCF)
Manifest	Fully qualified name of the event source

BamActivity

- **Represents data to extracted by the interceptor for a BAM Activity**
  - maxOccurs="unbounded" but ..
  - Only one element per unique name allowed in a single IC (key restricted)

Attribute	Description
Name	The name of the BAM Activity

OnEvent

- **Represents a potential event inside of the interceptor's runtime**
  - maxOccurs="unbounded"

Attribute	Description
Name	Application specific, unique name for the event
Source	Reference to corresponding EventSource element
IsBegin	True indicates a call to EventStream.BeginActivity
IsEnd	True indicates a call to EventStream.EndActivity

Expressions

- **All the potential child nodes of OnEvent contain expressions**
  - BAM interceptors evaluate these expressions
- **Expressions can be conditions or produce data**
- **Single child element of all children of OnEvent**
- **Expressions composed from one or more Operation child elements**

Operation

- **An Operation evaluates its value and places it on internal stack**
  - Some are operators that operate on other values
  - Some are expressions that evaluate to a single value
- **Most Operations are linked to well-known methods called by the interceptors to retrieve data**

Operation
And
Concatenate
Constant
Equals

Reverse Polish Notation

- **A postfix notation where operations follow operands**
  - 2 * (3 + 4) == 2 3 4 + *
- **Alleviates the need for parentheses**
  - Simpler notation for complex operations
- **BAM IC uses RPN to evaluate all Operations**

Operation example

```
<bam:Operation Name="Constant">
        <bam:Argument>Value_</bam:Argument>
</bam:Operation>
<bam:Operation Name="Constant">
        <bam:Argument>_OtherValue</bam:Argument>
</bam:Operation>
<bam:Operation Name="Concatenate"/>
<bam:Operation Name="Constant">
        <bam:Argument>_AndOther</bam:Argument>
</bam:Operation>
<bam:Operation Name="Concatenate"/>
</bam:Expression>
```

“Value__OtherValue_AndOther”

Filter

- **Determines if the current executing code is a match for this event**
  - Single occurring element
- **If Filter's expression evaluates to true, there is a match**
  - Sibling nodes will then execute

CorrelationId

- **Next OnEvent child after Filter**
  - Required
- **The value of this operation will be the value passed to the EventStream as the ActivityId**
  - Must be same value as used to start the activity instance or a valid correlation value
  - Same rules as if programming EventStream
- **If OnEvent IsStart="true" then used as the ActivityId**

Update

- **Represents a call to EventStream.UpdateActivity**
- **Data on the stack after the expression evaluates is passed as the value**
- **MaxOccurs = “unbounded”**

Attribute	Description
Type	BAM Activity checkpoint data type
DataItemName	Reference to corresponding BAM Activity checkpoint

Reference

- **maxOccurs="unbounded"**
- **Expression evaluates and value is used to call EventStream.AddReference**
- **Data child element**
  - Expression specifies data for AddReference
- **LongData optional element**
  - Specify for larger data (more than 1024 bytes)

Attribute	Description
Type	Type of references (e.g. Activity,DocumentUrl)
Name	Name of the reference

CorrelationToken

- **Optional child element of OnEvent**
  - maxOccurs="unbounded"
- **Enables correlation**
  - Effect is same as calling EventStream.EnableContinuation
- **Other events that update the activity may follow that correlation**
  - Same capabilities of EventStream
  - Multiple correlations allowed

Deploying an IC

- **Every IC has to be deployed into the BAM infrastructure for interception to work**
  - bm.exe deploy-interceptor –FileName:<fn>
  - -Force:true to overwrite already deployed version
- **New bm.exe commands for interceptors**

Command	Description
deploy-interceptor	Deploy an IC
remove-interceptor	Remove an IC
get-interceptor	Restrict by –Activity or –EventSource
get-interceptorlist	Gets all ICs

WF Interceptor

- **WF supports a dynamic tracking infrastructure**
  - WorkflowRuntime will use a TrackingService if found in the runtime context
- **IC is converted into Workflow TrackingProfile object**
- **WF actually supports multiple TrackingService instances**
  - Can use BAM interceptor + OOB or custom tracking

WF Operations

Operation	• Evaluates against….
GetActivityEvent	• ActivityExecutionStatus
GetActivityName	• Activity.Name
GetActivityProperty	• an Activity Property
GetActivityType	• typeof Activity instance
GetContextProperty	• workflow context value
GetUserData	• a UserTrackPoint ‘s data
GetUserDataType	• a UserTrackPoint’s type
GetUserKey	• a UserTrackPoint’s key
GetWorkflowEvent	• TrackingWorkflowEvent
GetWorkflowProperty	• A Workflow level property

GetContextProperty

- **In the WF interceptor context there are two available context properties**
  - EventTime
  - InstanceId (WorkflowInstance.InstanceId)
- **All other operations correspond to TrackPoints and TrackingRecords**

Creating a WF IC

- **OnEvent Filter will correspond to either one of**
  - WorkflowTrackPoint
  - ActivityTrackPoint
  - UserTrackPoint

Simple filter for WF

```
<bam:Filter>
        <bam:Expression>
                <bamwf:Operation Name="GetWorkflowEvent"/>
                <bam:Operation Name="Constant">
                        <bam:Argument>Started</bam:Argument>
                </bam:Operation>
                <bam:Operation Name="Equals"/>
        </bam:Expression>
</bam:Filter>
```

More complex filter for WF

```
<bam:Filter>
<bam:Expression>
<bamwf:Operation Name="GetActivityType"/>
<bam:Operation Name="Constant“>"
          <bam:Argument>CustomActivityLibrary.AmountProcessorActivity,CustomActivityLibrary</
bam:Argument>
</bam:Operation>
<bam:Operation Name="Equals"/>
<bamwf:Operation Name="GetActivityProperty">
          <bamwf:Argument>Name</bamwf:Argument>
</bamwf:Operation>
<bam:Operation Name="Constant">
          <bam:Argument>trackableAmount</bam:Argument>
</bam:Operation>
<bam:Operation Name="Equals"/>
<bamwf:Operation Name="GetActivityEvent"/>
<bam:Operation Name="Constant">
          <bam:Argument>Closed</bam:Argument>
</bam:Operation>
<bam:Operation Name="Equals"/>
<bam:Operation Name="And"/>
<bam:Operation Name="And"/>
</bam:Expression>
</bam:Filter>
```

CorrelationId in WF

- **Typical to use WorkflowInstance.InstanceId**
  - Unique (GUID)
  - Trackable
- **Could use Activity or Workflow Property instead**

```
<bam:CorrelationID>
        <bam:Expression>
                <bamwf:Operation Name="GetContextProperty">
                        <bamwf:Argument>InstanceId</bamwf:Argument>
                </bamwf:Operation>
        </bam:Expression>
</bam:CorrelationID>
```

Pulling data for update

```
<bam:Update Type="FLOAT" DataItemName="Amount">
        <bam:Expression>
                <bamwf:Operation Name="GetActivityProperty">
                        <bamwf:Argument>Amount</bamwf:Argument>
                </bamwf:Operation>
        </bam:Expression>
</bam:Update>
```

```
<bam:Update Type="DATETIME" DataItemName="Start">
        <bam:Expression>
                <bamwf:Operation Name="GetContextProperty">
                        <bamwf:Argument>EventTime</bamwf:Argument>
                </bamwf:Operation>
        </bam:Expression>
</bam:Update>
```

Configuring the WF interceptor

- **The WF Interceptor is a Workflow Runtime Service**
  - Has to be added to the WorkflowRuntime before the Runtime is started
- **Requires a connection string and a timer value**
  - Spins up a thread to check for IC changes based on timer value

Adding it to the Runtime

```
using (WorkflowRuntime wr = new WorkflowRuntime())
{
    string cStr = "server=.;database=BAMPrimaryImport;trusted_connection=yes";
    int timer = 50000;
    BamTrackingService ts = new BamTrackingService(cStr, timer);
    wr.AddService(ts);
    wr.StartRuntime();
}
```

```
<WorkflowRuntime Name="Test">
        <Services>
                <add
type="Microsoft.BizTalk.Bam.Interceptors.Workflow.BamTrackingService,
Microsoft.BizTalk.Bam.Interceptors, Version=3.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"

ConnectionString="server=.;database=BAMPrimaryImport;trusted_connection=yes"
        InterceptorConfigurationPollingInterval="5000"/>
        </Services>
</WorkflowRuntime>
```

WCF Interceptor

- **WCF's Channel infrastructure enables extensibility**
  - Configure components to help process messages
- **WCF Interceptor is configured as a message and parameter inspector**
  - Inbound and outbound messages
  - Both client and service configuration allowed

WCF Operations

Operation	• Evaluates against….
AutoGenerateCorrelationToken	• Auto generated token
GetContextProperty	• One of the WCF context values
GetEndpointName	• The name of the WCF endpoint
GetOperationName	• The name of the current WCF operation
GetServiceContractCallPoint	• The current call “callpoint”
XPath	• An XPath statement against the WCF Message

WCF GetContextProperty

- **For the WCF interceptor there are two properties available**
  - SessionId
  - EventTime
- **SessionId will be available only if using a Channel Stack that has Reliable Sessions enabled**
  - Either WS-RM or WS-SecureConversation

Filter in WCF

- **GetServiceContractCallPoint must be part of all WCF filters**
  - Can't get a match without it

Value	Description
ClientReply	When a client channel is getting a reply
ClientRequest	When a client channel is sending its request
ClientFault	When a client channel faults
ServiceRequest	When a service channel is getting a request
ServiceReply	When a service channel is sending a reply
ServiceFault	When a service channel faults
CallbackRequest	When a request is made through a callback channel
CallbackReply	When a reply comes back to a callback channel
CallbackFault	When a callback channel faults

WCF Example filter

```
<bam:Filter>
        <bam:Expression>
                <bamwcf:Operation Name="GetServiceContractCallPoint“/>"
                        <bam:Operation Name="Constant">

        <bam:Argument>ServiceRequest</bam:Argument>
                        </bam:Operation>
                        <bam:Operation Name="Equals">
                        </bam:Operation>
                        <bamwcf:Operation Name="GetOperationName"/>
                        <bam:Operation Name="Constant">

        <bam:Argument>ProcessOrder</bam:Argument>
                        </bam:Operation>
                        <bam:Operation Name="Equals">
                        </bam:Operation>
                        <bam:Operation Name="And">
                        </bam:Operation>
                </bam:Expression>
</bam:Filter>
```

Configuration the WCF interceptor

- WCF interjects itself through an EndpointBehavior
- Add configuration for the BehaviorExtension
- Configure the behavior for the specific Endpoint

WCF BAM Configuration

```
<extensions>
        <behaviorExtensions>
                <add name="BamEndpointBehaviorExtension"
type="Microsoft.BizTalk.Bam.Interceptors.Wcf.BamEndpointBehavior,
Microsoft.BizTalk.Bam.Interceptors, Version=3.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
        </behaviorExtensions>
</extensions>
<!-- later in the same configuration file -->
<behaviors>
        <endpointBehaviors>
                <behavior name="bamEndpointBehavior">
                        <BamEndpointBehaviorExtension
primaryImportConnectionString="server=.;database=BAMPrimaryImport;trusted_con
nection=yes" pollingIntervalSec="5000" />
                </behavior>
        </endpointBehaviors>
</behaviors>
```

Configuring BAM tracing

- When debugging the interceptor it can sometimes be useful to get BAM API tracing

```
<system.diagnostics>
        <trace autoflush="true" />
        <sources>
                <source name="Microsoft BizTalk Bam Interceptors"
                   switchName="sourceSwitch"
                   switchType="System.Diagnostics.SourceSwitch">
                        <listeners>
                                <add name="console"
        type="System.Diagnostics.ConsoleTraceListener" />
                        </listeners>
                </source>
        </sources>
        <switches>
                <add name="sourceSwitch" value="Verbose"/>
        </switches>
</system.diagnostics>
```

Summary

- BAM provides a dynamic infrastructure for both business and operational in-flight data
- BizTalk Server provides interceptors for BizTalk Orchestration and messaging
- As of BizTalk Server 2006 R2 - interceptors for WCF and WF provide BAM reach end-to-end