Proof of Technology
Discovering the value of IBM Integration Bus V9 for your ESB and SOA

Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
    - Assistance
  - Bar Files & Deployment
- What's New in V9
What do we mean by Integration?

- Enterprise systems consist of many logical endpoints
  - Off-the-shelf applications, services, web apps, devices, appliances, custom built software...

- Endpoints expose a set of inputs and outputs, which comprise
  - Protocols - e.g. MQ, TCP/IP, HTTP, File system, FTP, SMTP, POP3 etc.
  - Message Formats - e.g. Binary (C/COBOL), XML, Industry (SWIFT, EDI, HL7), User-defined

- Integration is about connecting these endpoints together in meaningful ways
  - Route, Transform, Enrich, Filter, Monitor, Distribute, Decompose, Correlate, Fire and Forget, Request/Reply, Publish/Subscribe, Aggregation, Fan-in, Complex Event Processing...
IBM Integration Bus has Everything You Need

- Transformation
- Multi-Protocols
- Content Based Routing
- Database Management System (DBMS) Integration
- Fully Transactional
- Effective, Powerful Tooling
- Simple
- Extensible
- Standards based
- Non-Invasive
Common Issues that IBM Integration Bus Solves

- Some common Integration Usage Patterns
  - Adding logging to existing service
  - Extend Reach of Existing Applications
  - Distribute Database information to where it's needed
  - File Hub to connect batch and online
  - Integrate Packaged Applications
  - Take advantage of .NET applications
  - Policy Enforcement Point for Secure Connectivity
  - Extend Enterprise to Devices and Mobile
  - Monitor business activity and act intelligently
  - Detect / Act Upon Business Events and Rules
  - Connectivity and Integration for Business Processes
  - Enable Policy based management

- New usage patterns continually emerging!

A Complete Solution

- Simply connect FROM anywhere TO anywhere
- Unparalleled range of protocols, routing, message formats and transformation options
- Easy to install, learn, develop, deploy and manage, including patterns to simplify solution creation
- High performing and scalable, with built-in failover support
- Connectivity Packs for Industry specific content (e.g. Healthcare)
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
    - Assistance
  - Bar Files & Deployment
- What’s New in V9

Components – What you get

| Toolkit – graphical development tool, Eclipse based; Windows, Linux (Intel) | X |
| Integration Server – runtime engine; various platforms | X | X |
| WebSphere MQ – interaction with MQ and administration; various platforms | X | X |
| WebSphere MQ/MB Explorer – administration; Windows, Linux (Intel) | X | X |
Component Configuration

- Developer Workstations
- Development Systems Testing Server
- QA Server
- Production Servers

Integration Server

Execution Groups Run Your Work

<table>
<thead>
<tr>
<th>EG 2</th>
<th>EG 3</th>
<th>EG 4</th>
<th>EG 5</th>
<th>EG n</th>
</tr>
</thead>
</table>

Development Artifacts

Go to 'View > Header and Footer' to change this to match the event title
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
    - Assistance
  - Bar Files & Deployment
- What’s New in V9
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
    - Message Models
    - Message Flows
      - Nodes
      - Assistance
    - Bar Files & Deployment
- What’s New in V9
Applications and Libraries Organize Resources

- Deployable containers of resources
  - Message flows, subflows, message definitions (DFDL, XSDs), ESLQ, JARs, XSL, Adapter files, etc

  » Applications promote encapsulation and isolation
    » “Main” message flows and required components
    » Deploy multiple applications to an execution group
    » Visibility of resource restricted to containing application

  » Libraries facilitate re-use
    » Resources shared across applications
    » Message definitions, common subflows, etc
    » Multiple libraries to group related resources
      » e.g. by type or function
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
    - Message Models
      - Message Flows
        - Nodes
        - Assistance
      - Bar Files & Deployment
- What’s New in V9

Message Models Define your Data

```java
class Address {
    public String street;
    public String city;
    public String state;
    public int zip;
}
```

```
struct Address {
    char street[40];
    char city[40];
    char state[20];
    int zip;
}
```
**Message Tree – Logical Representation of Message**

- **Root**
- **Properties**
- **Headers**
  - MQ
  - HTTP
  - SOAP
  - ...
- **Body**
  - Address
  - Name
  - Item
  - Qty
  - Price
  - Date
  - First
  - Last

**Parsers Do the Work**

- **Raw data into logical view and vice versa**
- **Body Parsers:**
  - Handle Classes of Data
    - XML – Generic or Schema Based
    - DFDL – Binary data (Fixed length, Tagged / Delimited)
    - Standards - SOAP, MIME, JSON
    - Model – DFDL, MRM
    - DataObject - EIS systems (Adapters)
    - JMS – JMSMap, JMSStream
    - BLOB - No structure, just a sequence of bits
    - Other - WTX, User plugin parsers
  - Select based on Input/Response message parsing tabs
  - Override by header information such as MQRFH2
  - Or using SQL / Java plugin Interface calls.
- **Transport header parsers:**
  - Specific for transport nodes
    - WMQ - MQMD, MQRFH2, MQWH, MQCICS, MQRMRH ....
    - HTTP - HttpRequestHeader, HttpReplyHeader, HttpRequestHeader, ....
    - JMS - JMSTransport …
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
    - Assistance
  - Bar Files & Deployment
- What’s New in V9
Visual Programming is Faster and Easier

Visual Programming - How You Think
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
- Nodes
  - Assistance
  - BAR Files & Deployment
- What’s New in V9

Nodes are Reusable and Consistent

Action
Many Transformation Options

Scripting/Programming

- Compute:
  - No code to write.
  - Transformations are run in parallel.
  - Supports any Common Language Runtime (CLR) language.

- JavaCompute:
  - Uses Java programming language.
  - Ability to use XPath.

- PHPCompute:
  - Transform using PHP scripts.
  - PHP 5.2 compliant.

- .NETCompute:
  - Build transformations in C#, VB, F#.
  - C++/CLI, Jscript, etc., supports any Common Language Runtime (CLR) language.
  - Call .NET programs directly via CLR V4.

Graphical

- Mapping:
  - Graphical, easy to use.
  - Drag and Drop fields, apply functions.

- XSL Transform:
  - Convert XML to XML.
  - Uses standard XSL Style sheets.

- WTX Map:
  - Run a WebSphere Transformation Extender map.

The Environment: Your scratch pad

Root

Properties

Headers

Body

Environment

Variables

System Stuff

Your Stuff

Local Environment

Variables

System Stuff

Your Stuff

Passed through the nodes

Put $(Environment)/$(LocalEnvironment) in your Trace node to see it.

You can also see it in your debug session.

Setting it is easy:

- e.g., SET OutputLocalEnvironment.Variables.CheckAmount = 23.50;

Using it is easy:

- e.g., SET OutputRoot.MRM.CustomerMessage.CheckAmount = InputLocalEnvironment.Variables.CheckAmount;
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
  - Assistance
    - Bar Files & Deployment
- What’s New in V9

Patterns – Making Reuse Easy

- Many built-in patterns
- Organized into categories
  - Groups similar solutions together
    - Message-based integration
    - Service enablement
    - Service virtualization
    - …
- Or create your own patterns
  - Pattern Authoring
- Extensive help provided
  - Selection / implementation guidance
QuickStarts Simplify Common Tasks

- QuickStarts available for application / library creation
- Application or library can reference
  - Project created by QuickStart
  - Existing projects

Samples Show You How
Simplified Testing using the Integrated Test Client

- Toolkit includes Integrated Test Client
  - Also Interactive Debugger (explore in later lab)

- Test message flows containing following input nodes:
  - WebSphere® MQ, JMS, SOAP, HTTP and SCA

- Monitors output nodes
  - Execution path
  - Errors generated by flow

- Test scripts saved
  - Create repeatable test suites

Interactive Flow Debugger

- Convenient, easy to use graphical interface for flow testing

- Breakpoints between nodes or within node logic
  - Stop execution at any point
  - Current content of message trees examined and modified

- Step into node and debug at the source level
  - Compute (SQL)
  - JavaCompute (Java)
  - .NETCompute (requires Microsoft® Visual Studio Professional or higher)
  - PHP
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
    - Assistance
  - Bar Files & Deployment
- What's New in V9

Development – Bar Files & Deployment

Applications
Message Models
Message Flows
Libraries
Message Models
Message Flows
Bar File
Deployment Process
Development – Bar Files & Deployment

Integration Node

<table>
<thead>
<tr>
<th>Execution Group #1</th>
<th>EG 2</th>
<th>EG 3</th>
<th>EG n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar File #1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Flows</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bar File 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bar File 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bar File 4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Questions?

QUESTIONS ANSWERED HERE
EVEN THE SILLY ONES
Time to Explore

- **Lab 1** - Create a new Application
  - Construct a simple Message Flow in the Application
    - MQInput node gets messages from a queue
    - MQOutput node puts messages to a queue
    - Trace node renders message structure into a readable format
  - Test your Application
    - Use the Integrated Test Client to deploy and test
      - XML Message In/Out, No Transformation
    - Use the Message Broker Explorer to create MQ queues

- **Lab 2** - Extend your flow to do message parsing and validation
  - Use the Interactive debugger to test your flow
    - Set breakpoints within the flow
    - Examine message tree
    - Modify message tree

- **Lab 3** - Modify your message flow to perform content-based routing
  - Use the Interactive debugger to test your flow
    - Set breakpoints within the flow
    - Examine message tree
    - Modify message tree

It’s Time For Labs 1-2-3!

- **Introductory Labs** (approx. 45 – 60 mins)
  - Building and Executing a Simple Message Flow
  - Extending the Message Flow for Parsing and Validation
  - Extending the Message Flow for Content-Based Routing

- **Tools You Will Use**:
  - MB Explorer
  - Integrated Test Client
  - XPath Expression Builder
  - Integrated Flow Debugger
Introduction to IBM Integration Bus

- Objectives of an ESB
- Components and Configuration
- Development
  - Eclipse design
  - Components
  - Message Models
  - Message Flows
    - Nodes
    - Assistance
  - Bar Files & Deployment
- What's New in V9
Worklight and Mobile

- Simple to "mobile enable" enterprise services
- Enable mobile application access to back-end systems

Global Cache - Storing state for integrations

- With a global cache, each node can handle replies – even when the request was processed by another node.
Global Cache - Caching infrequently changing data

- Number of clients can increase
  - Maintain predictable response time

Data Format Description Language (DFDL)

- New open standard
  - Open Grid Forum (OGF)
  - http://www.ogf.org/
  - Version 1.0
  - ‘Proposed Recommendation’ status

- Way of describing data…
  - NOT a data format itself!

- Describe any data format …
  - Textual and binary
  - Commercial record-oriented
  - Scientific and numeric
  - Modern and legacy
  - Industry standards

- High performance …
  - Right data format for the job

- Leverages XML technology and concepts
  - W3C XML Schema subset & type system
  - Annotations within the XSD
    - Physical representation of data
    - XPath to reference fields within data

- Round-tripping
  - Read / write data in described format
  - From same description

- Keep simple cases simple
  - Simple descriptions human readable

- Generality
  - Think “Type Tree + MRM” & more
An IBM Proof of Technology

**Record & Replay**

- Enable Record and Replay of In-flight Data
  - Comprehensive audit of messages, web, ERP, file & other data
  - Flexible topology: single or multiple nodes for recording, capture & replay

- Data Recording, Capture & Store
  - Graphically configure binary, text, XML payload capture, including whole, partial & multi-field data
  - Source data is currently limited to MB flows, including MB6.1, MB7 & MB8

- Monitor tab or monitoring profiles identify captured events
  - Capture events on *any node*, local or remote
    - Any node EG can be configured as capture agent
    - Configurable service identifies topic, target database
  - Agent stores data in a database

- Web Tooling to View, Query data
  - Friendly editors to view and query payloads
    - Key data fields, including application data
  - Independent web admin & capture for scalability
    - Configure multiple EG listeners for web

- Replay for redelivery or flow reprocessing
  - Replay selected data to flows or applications
  - MB admin configures logical destinations
    - Maps to physical protocol, e.g. MQ: {Qmgr, Q}
  - User selects destinations from auto-populated drop-down list

**Integrating .NET with IBM Integration Bus**

- Extremely tight language agnostic integration
  - Integrates any CLR language at a very low level

- Create your own .NET Compute Nodes using Visual Studio
  - Integrate new or existing .NET applications directly with your Message Flow
    - Write nodes in C#, VB, F#, C++/CLI, and many more

- Tightly integrated with Visual Studio
  - Integration toolkit can launch Visual Studio
  - Visual studio plugin to simplify node development

- Call .NET code directly from ESQL
  - Jump straight from ESQL into .NET code

- Integrate with existing COM applications
Understanding Runtime Behaviour

- Activity Logging: Allows users to understand what a message flow is doing
  - Complements current extensive product trace by providing end-user oriented trace
  - Can be used by developers, but target is operators and administrators
  - Doesn't require detailed product knowledge to understand behaviour
  - Provides qualitative measure of behaviour

- End user oriented with external resource lifecycle
  - Focus on easily understood actions & resources
  - "GET message queue X", "Update DB table Z"...
  - Complements current extensive product trace by providing end-user oriented trace

- Flow & resource logging
  - User can observe all events for a given flow
    - e.g. "GET MQ message", "Send IDOC to SAP"
  - Users can focus on individual resource manager if required
    - e.g. SAP connectivity lost, SAP IDOC processed
  - Use event filters to create custom activity log
    - e.g. capture all activity on JMS queue REQ1 and C:D node CDN1
  - Progressive implementation as with resource statistics, starting with JMS, C:D and SAP resources

- Comprehensive Reporting Options
  - Reporting via MB Explorer, log files and programmable management (CMP API)
  - Extensive filtering & search options, also includes save data to CSV file for later analysis

- Log Rotation facilities
  - Rotate resource log file when reaches using size or time interval

Graphical Performance Monitor

- Detailed Statistics and Reporting
  - Understand behaviour of deployed solutions
  - Comprehensive resource usage

- Reporting Mechanisms
  - Graphically reported through MB Explorer
    - Sort, filter and chart performance characteristics
    - View CPU, IO and other metrics
    - Log data to file in CSV/Excel readable format for post processing
  - User Configurable Reporting Interval
    - XML report messages consumed by any end user application

| CKI | CDRB | ISCtag | FTP | File | JDBCConnectionPool | VM | MDB | Parities | GigAppInput | GigInput | Security | TCPFConnection | TCPFServerNodes | numSamples | avgTime | stdTime | avgLive | avgIdle | avgRt   | avgTrans | numMessage | avgMessageSize | avgTotalSize | avgMaxSize | avgQueueSize | avgConnects | avgMessage'
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Discovering the value of IBM Integration Bus V9 for your ESB and SOA.

IBM Software

© 2013 IBM Corporation

Go to 'View > Header and Footer' to change this to match the event title
An IBM Proof of Technology

Browser Administration for Universal Access

- Web Administration Console
  - Objective is to provide comprehensive web management interface
  - Focus on non-administrators to understand nodes & resources
  - Supports all major browsers Firefox, IE, Opera, Safari, Chrome
  - Designed as a complement to MB Explorer
    - MB Administrators can continue to use MB Explorer

- Easy to configure
  - No extra moving parts - uses internal HTTP server to serve data
  - Just start a port for web admin, and go!
  - Can reconfigure to listen on user port or disable
  - SSL connector configured via mqsichangeproperties
  - View resources only for V8 GA
  - Design allows for future role based access to modify resources

- Using Web Admin
  - Intuitive tree view shows hierarchy of MB resources
  - View resource details with click or button
  - Includes full suite of resources
    - Apps, Libs, Flows, Configurable services etc

- Web Admin & MB Explorer
  - MBX & web admin designed for concurrent use

Hypervisor Edition

- Easy and Faster to Instantiate
  - Simplify provisioning MB on x/Linux & AIX
    - Initial system deploy resulting in quicker time to solution value
  - Private Clouds
  - Test and/or Production

- Hyper Visor Edition Packages
  - Pre-built installed VM image for OS+HW combination
    - RHEL 5.5 for VMWare ESX x86 64 & IWD images
    - AIX 6.1 TLT mksysb & IWD images
    - Package includes all MQ & MB components
  - Updated when new fix pack levels released
    - Fix pack deploy reduces recurring maintenance cost
    - Download to customer site from IBM web
    - Uploaded to IWD using CLI scripts or Image Loader tool

- Configuration Patterns and Scripts
  - HVE Scenario Configuration Information
    - Base Pattern with configuration script packages
    - Base Pattern for most popular MQ and MB topology configurations
      - Basic Configuration, Advanced Configuration, <User Pattern>
    - Script Packages configure base pattern
      - Create Exec Group, Deploy BAR, Run MQSC, <User Script>
    - Emergency Fix also possible: iFix binary + script package to drive installer

- Full Function Deployed Configuration
  - Interaction with deployed MQ, IIB components as per regular deployment
Making File Processing Simple

- **Native File Processing Support**
  - Local Files
    - Including FTP/S
- **WebSphere MQ File Transfer Edition (FTE)**
  - End-to-end transactional file processing
  - Receive and send FTE transfers
  - Typical scenarios
    - Reliable file to queue
    - Database to file
    - File to file
- **IBM Sterling Connect:Direct (C:D)**
  - Greatly simplifies using Connect:Direct
    - No need to understand process scripts
  - Use the CD network without
    - Creating intermediate files
    - Using CD Server interface

More than just File Nodes

- Process large files without large memories
  - Process record at a time
  - Parsing on demand
    - Build tree as needed
    - Delete input and output trees
- **Advanced record detection**
- No artificial constraints
  - 64-bit runtime
File Support - Overview

- File processing examples
  - Split batch of records to set of messages
  - Combine messages to produce batch of records in file
  - Transforming data formats
  - Routing based on file content
  - Connecting and sending to different protocols:
    - MQ, JMS, HTTP, Raw TCPIP, SAP, Siebel, WebServices, IMS, CICS, Databases …
- Local files
  - FileInput and FileRead nodes read files from local file system
  - FileOutput node writes files to local file system
- Options for remote files:
  - FTP/SFTP
    - FileInput and FileOutput nodes
  - Managed File Transfer Options:
    - WebSphere MQ Managed File Transfer nodes
    - IBM Sterling Connect:Direct
  - Additional Z/OS specific nodes (VSAM, QSAM)

Record detection – One record at a time

- Split file into separate records
  - Single record
    - Whole file treated as single record
  - Fixed-length records
  - Delimited Records
    - Specify delimiter
  - Parsed Record Sequence
    - Records recognized by specified parser
- One record in memory at a time
  - Very large files streamed efficiently
    - DFDL, MRM (CWF and TDS), and XMLNSC parsers
Managed File Transfer (MFT) nodes – Move Data Reliably

- **End-to-End Transactional File Processing**
  - Receive / send FTE transfers
  - Consistent with FileInput/FileOutput nodes but make full use of the power of FTE
  - Sophisticated file processing includes timely inbound and outbound transfers & intelligent metadata
  - Typical scenarios are reliable file-to-queue, database-to-file, file-to-file

- **Input node**
  - Message flow starts processing file when FTE agent notifies of complete file transfer
  - FTE Metadata provided in LocalEnvironment
  - Allows intelligent processing of transfers & scripting
  - Whole file or record-at-a-time (same as file nodes)

- **Output node**
  - Creates file output records and requests FTE transfer with appropriate metadata
  - LocalEnvironment allows transfer overrides and customizable metadata

- **Agents automatically Installed, Configured and Managed**
  - Installed seamlessly as part of regular install, auto configured client name, transfer directories etc.
  - Agents started when message flows are started or stopped
  - Configurable service allows overrides

Connect:Direct (C:D) Nodes - Common standard for file transfer

- **Extend File Processing Capabilities**
  - Receive / send Connect:Direct transfers
  - Uses standard CD client API
  - Consistent with FileInput/FileOutput nodes but make full use of the power of Connect:Direct
  - Simplifies using Connect:Direct (no need to understand process scripts)

- **Input node**
  - Monitors C:D servers stats for completed transfers
  - Processes files immediately.
  - Can leave file unchanged after processing (just delete notification message)
  - C:D Metadata available to flow

- **Output node**
  - Destination C:D server, directory etc are defined on node
  - Wild card file names
  - Sequential and partitioned datasets
  - Properties overridden using local environment
    - Including any C:D options
  - C:D Metadata in local environment
IBM Integration Bus –Summary

• Powerful, flexible, extensible, production strength product
• Key Concepts
  – Message flows represent application connectivities
  – Message nodes modularize integration operations
  – Message Tree and Logical Message Model provides focus on business data
  – Patterns enable rapid, top-down development of Connectivity solutions
• Rich, simple message and database processing using Graphical Mapping
  – Java, ESQL, .NET, PHP can also be used
• Support for Web Services, Enterprise Messaging, and a variety of transports
• Scalable architecture for high capacity
• IBM Integration Bus is a key IBM connectivity technology
  – Unparalleled range of connectivity options and capabilities
  – Supports users’ range of experience and needs
  – Industry leading performance in a broad range of scenarios

Questions?