Introduction to IBM Integration Bus (IIB)
Agenda

- What is IIB (Message Broker)?
- Programming Concepts
  - Message Flows
  - Nodes
  - Message Model
- Product Overview
  - Components
  - Architecture
  - IIB on z/OS
- What’s new in Message Broker V8
- What’s new in IBM Integration Bus V9
Introducing IBM Integration Bus

- **IBM’s Strategic Integration Technology**
  - Single engineered product for .NET, Java and fully heterogeneous integration scenarios
  - DataPower continues to evolve as IBM’s integration gateway

- **A Natural Evolution for WebSphere Message Broker users**
  - Significant innovation and evolution of WMB technology base
  - New features for Policy-based WLM, BPM integration, Business rules and .NET

- **Designed to incorporate WebSphere Enterprise Service Bus use cases**
  - Capabilities of WESB are folded in to IBM Integration Bus over time
  - Conversion tools for initial use cases built in to IIB from day one
  - WESB technology remains in market, supported. Migrate to Integration Bus when ready
IBM’s plans, directions, and intent are subject to change or withdrawal

Product Roadmap

IBM Integration Bus V9 Q2 2013
- Policy-based Workload Management and Flow Management
- Web-based Visualization and Performance Analysis
- MQ and Database Service Discovery
- Business Rules and BPM Integration
- .NET Input, Dynamics, MSMQ

V9 FP1 Q4 2013

Healthcare Connectivity Pack
- DICOM Medical Imaging
- Web-based Operational Monitoring
- CDA Analytics and Reporting
- Message Broker V8 Exploitation

Healthcare Connectivity Pack V8 Q1 2013

Healthcare Connectivity Pack V8 FP1 Q3 2013

Message Broker V8 Q4 2011

Message Broker V8 FP1 Q2 2012

Message Broker V8 FP2 Q1 2013

Healthcare Connectivity Pack V7 FP2 Q1 2012

© 2013 IBM Corporation
What is an “Enterprise Service Bus (ESB)”?

"Gartner estimates that up to 30% of the cost of implementing an application is related to the development of the interfaces”

(Gartner, January 2000)

- The “Broker” (or “hub” or “gateway” or …) is central and the intermediary between the applications
- It typically has responsibility for routing and transforming data between the applications
- A Broker must be polyvalent with support for multiple data formats and protocols, extremely reliable and scalable
Business value of an ESB

- Traditional communications
  - Point to point
  - Each application must adapt the data to its particular need
  - The number of transformations grows exponentially n*(n-1)

- Communications with a ESB
  - Hub ‘n spoke logic
  - Maintenance can be centralised
  - Promotes reutilisation

(Note: although an ESB appears architecturally as a single middle point, scalability and high-availability requirements imply that multiple physical hubs are required)
What is IBM Integration Bus?

- IBM Integration Bus enables “universal connectivity” by integrating protocols, message formats and mediation patterns
  - Emphasis on application re-use

- Fits naturally with WebSphere MQ
  - Robust, scalable architecture
  - Optimized for high throughput
  - Flexible broker topologies

- Three programming constructs are used:
  - Message Flows
  - Nodes
  - Message Models
IBM Integration Bus (formerly WMB)…

- **Routes, Transforms, Augments** “messages”
- Supports **multi-format** (XML, SOAP, fixed, variable length, tagged, SWIFT, IDOC, etc.)
- Accepts **multi-protocol** (HTTP, JMS, MQ, SOAP, TCP/IP, local files, FTP, etc.)
- Offers full **database** support (DB2, Oracle, Informix, Microsoft SQL Server, Sybase, etc.)
- Supports common **ERP** and **EIS** interfaces (CICS, IMS, SAP, PeopleSoft, Siebel, etc.)
- Provides a drag ‘n drop **visual development** based upon Eclipse, supporting a variety of **development languages** (Java, eSQL, XSLT, PHP, .Net, etc.)
- Offers a **scalable, high-performance, resilient, low-latency** “execution container”
- Provides **transactional** (2PC) support (MQ, DB2)
- Supports Point-To-Point, Pub/Sub, Event, Synchronous and Asynchronous message processing styles
- Includes full life-cycle tooling (development, administration, runtime)
- Is extensible with open parser, node & administration interfaces
- Supports all major hardware and O/S platforms
Application Connectivity

- **Protocols**
  - e.g. MQ, TCP/IP, HTTP, File system, FTP, SMTP etc.

- **Message Formats**
  - e.g. Binary (C/COBOL), XML, Industry (SWIFT, EDI, HL7), User-defined

- **Mediation Patterns**
  - e.g. Route, Transform, Enrich, Filter, Monitor, Distribute, Decompose, Correlate, Fire and Forget, Request/Reply, Publish/Subscribe, Aggregation, Fan-in, Complex Event Processing
Mediation Patterns – Routing and Transformation

Mr. Smith,
Graphics Card, 32, 100, 11/07/2009

[Customer, Order, Quantity, Price, Date]

<order>
  <name>
    <first>John</first>
    <last>Smith</last>
  </name>
  <item>Graphics Card</item>
  <quantity>32</quantity>
  <price>200</price>
  <date>07/11/2009</date>
</order>

[Customer, Order, Quantity, Price, Date]
Application Connectivity with IIB

- IIB can act as an intermediary
- Flexible topologies
Message Flows

- Reusable
- Scalable
- Transactional
Message Flow Example
Nodes

- The building blocks of message flows

- Each node type performs a different (input, output or processing) action

- Many different node types
  - Grouped into logical categories in the message flow editor
Lots of Nodes are Built in [1]
Lots of Nodes are Built in [2]

- Many other nodes available through product extensions and supportpacs
  - For example, WebSphere TX, Tibco RV, VSAM, QSAM
- Write your own User-Defined Nodes in C or Java
Node Terminology
The Integration Toolkit

- The GUI used for all ESB development tasks
- Eclipse based, standard interface for Windows or Linux (http://www.eclipse.org/org/)
- Provides various “perspectives” for different tasks to be performed
  - Message Flow Developer Perspective
  - Debug Perspective
  - Java Perspective
  - etc...
- Let’s take a closer look at it…….
The Integration Toolkit tour

Editor pane

Navigator pane

Outline pane

Properties
Integration Toolkit - Flow definition

A "flow" defines the handling of a message.

- A “flow” specifies the handling of a message.
- The IIB Toolkit is used to define a sequence of Nodes representing actions.
- Each Node allows for customization.
- "Drag 'n Drop" development for many standard operations.
- Some nodes allow complex operations (e.g., Compute, JavaCompute, XSLT).
Patterns

A *pattern* is a reusable solution that encapsulates a tested approach to solving a common architecture, design, or deployment task in a particular context.
Pattern example

1. Select a pattern
2. Validate it
3. Fill in basic info
4. Input Q info
5. Name artifacts

That's it! Working flow generated!
Message Modeling

```c
struct {
    int height;
    int age;
    char firstname[24];
    char lastname[24];
} Person;

172   32   Fred   Smith
```

```xml
<Person age='32' height='172'>
   <name>
      <first>Fred</first>
      <last>Smith</last>
   </name>
</Person>
```

- PERSON + 172 + 32 + Fred Smith
The Logical Message Model - Addressing

Examples:
Parsers

Input Message Bit-stream

Parser converts bit-stream to logical structure

Model

Parser converts logical structure to bit-stream

Output Message Bit-stream

Fred Smith, Graphics Card ...
Creating Message Models

- C Header
- COBOL Copybook
- XML Schema
- WSDL
- File Import

Enterprise Information System (SAP, Siebel, PeopleSoft)

- Pre-built
  - SOAP, MIME, CSV, IDOC, SWIFT, EDIFACT, X12, FIX, HL7, etc

- Define your own using the Eclipse-based Tooling

WebSphere Transformation Extender

- Type tree

Message Set

Parsers

IBM Integration Bus
Message Transformation
- The conversion of one message format into another

- Graphical, easy to use
- Drag and Drop fields, apply functions
- Describe powerful transformations quickly
- Uses SQL-based language (ESQL)
- Uses Java programming language
- Ability to use XPath
- Transform using PHP scripts
- PHP 5.2 compliant
- Convert XML to anything
- Uses standard XSL Style sheets
- Run a WebSphere Transformation Extender map
Examples of Message Addressing

```java
public class jcn extends MbJavaComputeNode {
    public void evaluate(MbMessageAssembly assembly) throws MbException {
        String lastName =
            (String)assembly.getMessage().evaluateXPath("/Body/Order/Name/Last");
    }
}
```

IF Body.Order.Date < '2008/01/01' THEN
    INSERT INTO Database.OldOrders (LastName, Item, Quantity)
    VALUES (Body.Order.Name.Last,
            Body.Order.Item,
            Body.Order.Quantity);
ENDIF;
When “drag ‘n drop” development with the standard nodes isn’t sufficient, programming can be done typically with eSQL (procedural language based upon the SQL99 standard), Java, PHP, .Net, XSLT or additional add-on extensions such as WTX.

- Custom nodes can also be developed either in Java or C++
- Toolkit supports tracing and interactive debugging
- Team development and administration is supported by scripting and/or standard market plugin extensions to the IIB Toolkit, eg. CVS, ClearCase, PVCS, TeamCode, etc.
### eSQL

**Data Insert**

```
IF Root.XML.Person.Taille > 183 THEN
  INSERT INTO Database.MesGrandsCopains
  (Name, Height, Age)
  VALUES (Body.Person.Nom,
          Body.Person.Taille,
          Body.Person.Age);
ENDIF;
```

**Compute**

```
IF (Body.Person.Name = 'Carl') THEN
  OutputRoot.Properties.MessageFormat = 'XML';
ELSE IF (Body.Person.Name = 'Rudi')
  OutputRoot.Properties.MessageFormat = 'CWF';
ELSE IF (Body.Person.Name = 'Saad')
  OutputRoot.Properties.MessageFormat = 'TDS';
ENDIF;
```

**Data Types**
- INTEGER
- FLOAT
- DECIMAL
- STRING
- BOOLEAN
- REFERENCE
- NULL

**Statements**
- Basic
- DECLARE
- SET
- IF ENDIF
- WHILE
- Tree
- MOVE
- CREATE
- DETACH
- ATTACH
- INSERT
- DELETE
- UPDATE
- PASSTHRU
- EVAL
- Node
- PROPAGATE
- RETURN
- THROW

**Operators**
- - + * /  ||
- AND OR NOT
- = <> > > = < <=
- IN BETWEEN
- LIKE

**Functions**
- String
  - LENGTH
  - TRIM
  - LTRIM
  - RTRIM
  - OVERLAY
  - POSITION
  - SUBSTRING
  - UCASE
- LCASE
- Numeric
  - ABS
  - BITAND
  - NOT
  - XOR
  - MOD
  - ROUND
  - SQRT
- Datetime
  - TRUNCATE
- EXTRACT
- CURRENTDATE
- CURRENTTIME
- Field
  - BITSTREAM
  - CARDINALITY
  - FIELDTYPE
  - SAMEFIELD
- Complex
  - CAST
  - SELECT
Java Compute Node

- Support for Java as IIB transformation language (Java Compute Node)
  - Complete support for Java JSE v7 integrated into IIB
  - Java classes deployed in the IIB Broker ARchive with the flow (.BAR)
  - "Wizards" used to simplify the development
  - Classes provided for XPATH message tree navigation
  - zAAP support for IIB on z/OS!

```java
public class CarlJNode extends MbJavaComputeNode {
    public void evaluate(MbMessageAssembly inAssembly, MbInputTerminal inTerm) throws MbException {
        MbMessage outMessage = new MbMessage(inAssembly.getMessage());

        // Add user code below
        MbElement cadet =
            outMessage.getRootElement().getLastChild().getFirstChild().getLastChild();
        rc = cadet.createElementAfter(MbElement.TYPE_NAME, "NewElm", "mon truc chouette");

        // End of user code
        MbMessageAssembly outAssembly =
            new MbMessageAssembly(inAssembly, outMessage);
        getOutputTerminal("out").propagate(outAssembly);
        outMessage.clearMessage();
    }
}
```
Other integrated transformation options…

- Standard XML Transformations via XSLT
- PHP (Hypertext pre-processor) support
  ```php
  $message->a->b->c = $input_body->Message;
  for ($index = 0; $index < $output_root->Menu->Food->count; $index++) {
    $item = $output_root->Menu->Food[$index];
  }
  ```
- WebSphere Transformation Extender (WTX) integration - for bulk transformations
  - Full Development-time integration into toolkit
  - Full Runtime integration
    - Invoke WTX parser
    - WTX mapping
- .Net integration with IIB v8 (Windows platforms only)
Architected for High Performance and Scalability

- **IBM Integration Bus Toolkit**
  - Development and Test Environment
  - Built on Rational Application Developer

- **IBM Integration Bus Explorer**
  - Advanced Administration Tool
  - Built on MQ Explorer

- **Broker (Integration Server)**
  - Standalone runtime environment that runs message flows
  - Execution groups for isolation and scalability
  - Many different platforms
  - Builds on an MQ queue manager
Components – Broker Runtime

- Runs message flow processing logic
- Made up of one or more ‘integration server’ processes that can run multiple message flows each
  - Provides isolation and scalability
- Available on Windows, z/OS, AIX, HP, Solaris, Linux (Intel, zSeries)
- Requires a local MQ Queue Manager
IIB Administration
IBM Integration Bus Administration

- IBM Integration Bus offers a wide variety of flexible administration tools
  - This reflects its underlying nature as a production strength tool
  - 3rd party administration tools are also available (BMC, CA, IBM Tivoli…) for integrating into a corporate administration

- IBM Integration Explorer is the visual (GUI) interface for casual administration
  - Eclipse based plugin for WMQ Explorer
  - Available on Linux and Windows

- Lightweight Web Administration for zero-footprint visual administration
  - Supports IE, Firefox, Safari, Chrome….

- A full Command Line interface is also available
  - Consistent interface on all platforms
  - Useful for administration from scripting environments
  - SDSF commands also provided for IIB on z/OS

- The IIB Java API is the underlying administration API
  - Available to any administrator to develop routines
  - A solid base for corporate, repeatable, controlled administration

- REST-based administration interface supporting HTTP clients
  - Compatible with Java API
The Integration Explorer
Industrial strength administration – OMEGAMON XE
IBM Integration Bus business monitoring

- A message flow can be configured to emit events through Monitor Properties on each node.
- Events are available for transaction start, transaction end, transaction rollback for Input nodes, and for a message passing into or out of any terminal on any node.
- Each event can be manually added – then they can be enabled or disabled.
  - Events can also be defined by the administrator using a monitoring profile.
- These Events are for functional & business monitoring, e.g., KPIs.
  - The entire message or selected fields can be included in the event.
- Events are published to a WebSphere MQ topic.
- Any subscribed application will receive the events.
  - Any 3rd party subscriber can make use of the events, for example WebSphere Business Monitor, via an IBM-supplied program that converts them to industry-standard CBE format.
Statistics – technical monitoring

Possible outputs
- MQ(XML) by Publish/Subscribe
- UserTrace
  BIP2380/2381/2382/2383 Messages
- z/OS SMF
  Type 117
    subtype 1: Flow and threads
    subtype 2: Nodes, optional terminals

Reporting Scope
- Flow
- Execution Group
- Broker
- Domain

Brother
Subscribers
Published
SMF
archive
snapshot
DATA
Pub/Sub Writer
UserTrace Writer
SMF Writer
Statistics –MQ(XML) format

```xml
<psc>
<Command>Publish</Command>
<PubOpt>RetainPub</PubOpt>
<Topic>$SYS/Broker/MQ02BRK/StatisticsAccounting/Archive/default/XMLflow</Topic>
</psc>

<WMQIStatisticsAccounting RecordType="Archive" RecordCode="StatsSettingsModified">
<MessageFlow BrokerLabel="MQ02BRK" BrokerUUID="7d951e31-f200-0000-0080-efelb9d849dc"
MessageFlowName="XMLflow" StartDate="2003-01-17" StartTime="14:44:14.550824"
TotalInputMessages="1" TotalSizeOfInputMessages="367" TotalNumberOfBackouts="0" />
<Threads Number="1">
<ThreadStatistics Number="0" TotalNumberOfInputMessages="0" TotalElapsedTime="0" ...
MinimumSizeOfInputMessages="0" />
</Threads>

<Nodes Number="3">
<NodeStatistics Label="FAILQueue" Type="MQOutput" TotalElapsedTime="0"
MaximumElapsedTime="0" NumberOfInputTerminals="1" NumberOfOutputTerminals="2">
<TerminalStatistics Label="failure" Type="Output" CountOfInvocations="0" />
<TerminalStatistics Label="in" Type="Input" CountOfInvocations="0" />
<TerminalStatistics Label="out" Type="Output" CountOfInvocations="0" />
</NodeStatistics>...
</Nodes>
</WMQIStatisticsAccounting>
```
Accounting and statistics: Example output – User Trace

```plaintext
.../wmqi/<broker>/log/<broker>/agent.userTrace.bin.0

BIP2380I: WMQI message flow statistics. ProcessID='196767', Key='3',
Type='SnapShot', Reason='Snapshot', BrokerLabel='MQ01BRK',
BrokerUUID='a0a1a981-f000-0000-0080-9f945b3d6b5b',
ExecutionGroupName='PubSubGrp', MaximumElapsedTime='20457211',
MinimumElapsedTime='20457211', TotalNumberOfBackouts='0'.

Statistical information for message flow 'PubSubTest' in broker 'MQ01BRK'.
This is an information message produced by WMQI statistics.

BIP2381I: WMQI thread statistics. ProcessID...
Key='3', Number='0', TotalNumberOfInputMessages='1',
TotalElapsedTime='20457211', TotalCPUTime='395405',
CPUTimeWaitingForInputMessage='10425',
ElapsedTimeWaitingForInputMessage='3302147', TotalSizeOfInputMessages='690',
MaximumSizeOfInputMessages='690',

BIP2382I: WMQI node statistics. ProcessID=..., Key='3', Label='', Type=' ',
TotalElapsedTime='0', MaximumElapsedTime='0', MinimumElapsedTime='0',
TotalCPUTime='0', MaximumCPUTime='0', MinimumCPUTime='0',
NumberOfOutputTerminals='1'.
```
Accounting and statistics: Example output – z/OS SMF Record

Type 117

SubType 1
- Message flow
- Threads?
- Message flow data
- Thread Data
- Thread Data

SubType 2
- Message flow
- Nodes
- Terminals==0
- Message flow data
- Node data
- Node data
- Node data

SubType 2
- Message flow
- Nodes
- Terminals>0
- Message flow data
- Node data
- Node data
- Terminal data
- Terminal data
- Terminal data

1 Flows/Threads
1 Including Nodes only
N Including Nodes+Terminals

BipSmf.h

Including Nodes only
BipSmf.h

Including Nodes+Terminals
IIB Architecture
IBM Integration Bus System Architecture

Runtime Platforms

- IBM AIX®
- Sun Solaris SPARC
- Sun Solaris x86-64
- HP-UX
- IBM z/OS®
- Windows Server 2012
- Windows 2008
- Windows 7 & 8
- Linux® (Intel) (Suse & RHEL)
- Linux on POWER®
- Linux on System z®
- Ubuntu
Broker on z/OS
The Broker address spaces
What’s different on z/OS?

● The Broker is essentially identical on all platforms as far as the facilities and any APIs.
  ▸ Almost all flows run “as-is” regardless of the platform
● IIB z/OS does benefit, however, from a few z/OS-specific nodes for better exploiting z/OS:
  ▸ VSAM nodes for direct access to VSAM datasets
  ▸ QSAM nodes for direct access to QSAM datasets
● IIB z/OS also leverages several other z/OS-exclusive features…
  ▸ z/OS is the only platform that can offer the highest levels of **scalability and high-availability** by taking full advantage of the z/OS Parallel Sysplex and WebSphere MQ **Shared Queue** technology
  ▸ IIB uses z/OS **ARM** feature used to auto (re-)start in case of failure
  ▸ z/OS **RRS** is used to ensure IIB transactionality
  ▸ IIB z/OS takes advantage of **WLM** and corporate business goals can be assigned to Execution Groups
  ▸ IIB z/OS makes full use of **SMF** for performance monitoring

Customers choose to run the Broker on z/OS typically when interfacing with host data and/or when they require the best QOS only found on z/OS
IIB z/OS high-availability

- Any Broker in the QSG can access messages
  - “Natural” load-balancing based upon availability
- Any Broker can recover messages in case of an outage
- z/OS ARM can restart any stopped component
1. Install Path

2. Authorization
   - extattr +a bipimain
   - extattr +l *.lib *.lil
   - per install

3. Create component PDS(E)

4. Copy procedures from SBIPPROC
   - BIPCRBK, (QMZABRK)...

5. Copy sample profiles SBIPSAMP

6. Tailor BIPEDIT
   - per broker

7. Create component

8. Start component
   - /S QMZABRK
IIB SDSF console commands

<table>
<thead>
<tr>
<th>Short</th>
<th>Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>Start component</td>
</tr>
<tr>
<td>PC</td>
<td>Stop component</td>
</tr>
<tr>
<td>CT</td>
<td>Change trace</td>
</tr>
<tr>
<td>RT</td>
<td>Report trace</td>
</tr>
<tr>
<td>L</td>
<td>List</td>
</tr>
<tr>
<td>RE</td>
<td>Reload</td>
</tr>
<tr>
<td>CB</td>
<td>Change broker</td>
</tr>
<tr>
<td>CS</td>
<td>Change flow stats</td>
</tr>
<tr>
<td>RS</td>
<td>Report flow stats</td>
</tr>
<tr>
<td>DP</td>
<td>Deploy</td>
</tr>
<tr>
<td>CX</td>
<td>Change flow user exits</td>
</tr>
<tr>
<td>RX</td>
<td>Report flow user exits</td>
</tr>
<tr>
<td>RC</td>
<td>Reload security</td>
</tr>
<tr>
<td>CM</td>
<td>Change flow monitoring</td>
</tr>
<tr>
<td>RM</td>
<td>Report flow monitoring</td>
</tr>
<tr>
<td>CR</td>
<td>Change resource stats</td>
</tr>
<tr>
<td>RR</td>
<td>Report resource stats</td>
</tr>
</tbody>
</table>

Start a broker  /S <Broker>
Stop a broker   /P <Broker>
Modify a broker /F <Broker>,cmd

/F MQRBBRK,PC
/F MQRBBRK,cb l=/usr/lpp/wmqi/lil:/u/carlf/lil
/F MQRBBRK,SC
/F MQRBBRK,list e='default'
/F MQRBBRK,changeflowstats -a -e 'default' -o smf
Key Usage Scenarios

1. Extending the reach of existing applications
   - Using IIB to integrate with other systems

2. Moving Batch Into Online
   - Migrating batch processing to real-time systems

3. Business Monitoring
   - Monitoring business performance and operations

4. Making an Application Inventory and Governing Processing
   - Inventory management with WSRR

5. Making the Most of Packaged Applications
   - Integrating with SAP and Siebel

6. Participating in a Secure Infrastructure
   - Using IIB and LDAP for secure access

**IIB**

Extending the reach of existing applications

Moving Batch Into Online

Business Monitoring

Making an Application Inventory and Governing Processing

Making the Most of Packaged Applications

Participating in a Secure Infrastructure
Additional updates in Version 8
Web Administration for Universal Access

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

- **Easy to configure**
  - No extra moving parts - uses internal HTTP listener to serve data
    - Web admin started by default on port 7050
  - Can reconfigure to listen on user port or disable
    - SSL connector configured via `mqsichangeproperties`
  - Role based access provides custom class user control
    - Default is read-only access to MB resources
    - More authority required to create, change or delete 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
  - Web admin requires MB8 broker
  - Explorer can manage both MB8 & MB7 brokers
Record & Replay

- **Enable Record, Edit and Replay of In-flight Data**
  - Comprehensive audit of messages, web, ERP, file & other data
  - Flexible topology: single or multiple brokers 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 broker*, local or remote
    - Any broker EG can be configured as capture agent
    - Configurable service identifies topic, target database
  - Agent stores data in any supported broker database
    - Oracle, DB2, SQL Server, Sybase, Informix...

- **Web Tooling to View, Query, Edit data**
  - Friendly editors to view, query & edit 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
Open Management with REST

- **REST based management API**
  - MB now supports HTTP/REST management API
    - Complements & compatible with existing CMP interface
  - HTTP client can manage MB independent of CMP
    - Includes new interface for message record & replay

- **URI for all MB Resources**
  - New ATOM data format for payload describes MB resources & related entities
  - ATOM service documents & feeds map intuitively mapped to MB artefacts
  - Provides very natural navigation of MB resources
    - e.g. Execution group document contains EG properties & per-message flow ATOM feed

```
GET /admin/eg/MYEGNAME HTTP/1.1
From: machine@ibm.com
User-Agent: MyApp/1.0
```

```
HTTP/1.1 200 OK
Date: Sun, 1 Oct 2011 21:46:59 GMT
Content-Type: text/html
Content-Length: 426

<?xml version="1.0" encoding='utf-8'?>
<service xmlns=http://www.w3.org/2007/app xmlns:atom="http://www.w3.org/2005/Atom">
    <workspace>
        <atom:title>Execution group feeds</atom:title>
        <collection href="http://my.broker.com/admin/eg/MYEGNAME/resources" />
    </collection>
    <workspace>
        <executiongroup description.long="" description.short="" ... />
    </executiongroup>
</service>
```

- **Fully open interface can be exploited by 3rd party tools**
  - HTTP REST/ATOM formats published & maintained for use by external users
  - Enables widgets, mash-ups & other situational applications
Easy to Develop, Deploy & Manage

- **Streamlined AD, Deploy & Management**
  - New & migrated resources grouped into Apps & Libs
    - Encourages designing for reuse
    - Simplifies deployment & management

- **Apps & Libs contain all solution resources**
  - Apps contain solution specific resources
    - e.g. main processing flow, specific transformations
  - Libs contain common resources
    - e.g. data definitions, error routines...
    - Can reference other libraries
  - Apps & Libs created from migrated workspaces
    - Automatic migration from MB6.1 & MB7

- **Easy deployment**
  - Drag & drop apps to run them immediately
  - Simple to package with 1-click for each app
  - Override deployment properties for promotion

- **Consistent Operations**
  - AD artefacts are visible in runtime with same structure
    - MBTK, MBX, Web admin all reflect same structure
  - Can manage using apps e.g. start, stop, delete
  - Commands updated to refer to Apps & Libs e.g. **mqsilist**
  - Full lineage available, e.g. version, deploy date…
Data Modeling – Why DFDL?

• Much of the data in the world resides in files, is not XML, is a mixture of textual and binary with custom syntax and encodings, and does not have a shareable description

• But there has been no universal standard for modeling this data!
  – XML -> use XML Schema
  – RDBMS -> use database schema
  – Text/binary -> ??

• Existing standards are too prescriptive: “Put your data in this format!”

• Organizations including IBM evolved their own way of modeling text and binary data based on customer need.

• IBM examples…
  – IBM® WebSphere® Message Broker: MRM message set
  – IBM WebSphere ESB: Data Handlers
  – IBM WebSphere Transformation Extender: Type Trees
  – IBM DataPower : FFD
  – IBM Cast Iron: Flat File Schema
  – Sterling B2B Integrator: DDF and IDF files

✓ DFDL: a universal, shareable, non-prescriptive description for general text & binary data formats
Data Format Description Language (DFDL)

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

- A way of describing data…
  - It is NOT a data format itself!

- A powerful modeling language …
  - Text, binary and bit
  - Commercial record-oriented
  - Scientific and numeric
  - Modern and legacy
  - Industry standards

- While allowing high performance …
  - You choose the right data format for the job

- Leverage XML Schema technology
  - Uses W3C XML Schema 1.0 subset & type system to describe the logical structure of the data
  - Uses XSDL annotations to describe the physical representation of the data
  - The result is a DFDL schema

- Both read and write
  - Parse and serialize data in described format from same DFDL schema

- Keep simple cases simple

- Annotations are human readable

- Intelligent parsing
  - Automatically resolve choice and optionality

- Validation of data when parsing and serializing
Easy Data Modelling with DFDL

- **Simple & powerful standard for data modelling**
  - New standard for binary, text & industry data formats
    - Logical structure with physical annotations
    - e.g. endian, ASCII/EBCDIC, padding, justify…
  - Data Format Description Language (DFDL)
    - For use in IBM and non-IBM products
    - [forge.gridforum.org/projects/dfdl-wg](http://forge.gridforum.org/projects/dfdl-wg)

- **Built-in facilities to model data easily**
  - Quick wizards for (e.g.) CSV, record oriented data
  - Auto-model importers (e.g.) COBOL copybooks
  - DFDL editor for power users
    - Create logical model & physical refinements

- **Test parsing and test data generation**
  - Test whether sample data fits with DFDL definition
    - Parse trace provide success & error case explanation
  - Auto-generate test data for test & debug scenarios

- **All broker nodes can exploit new DFDL parser**
  - Configure as existing XML, JSON, MRM, MIME… parsers
    - Interacts with message tree in usual manner
  - Excellent performance characteristics
    - (e.g.) element type, size, structural complexity etc
    - Supports streaming, partial parsing etc…
Easy Data Modelling with DFDL

- Simple & powerful standard for data modelling
  - New standard for binary, text & industry data formats
    - Logical structure with physical annotations
    - e.g. endian, ASCII/EBCDIC, padding, justify...
  - Data Format Description Language (DFDL)
    - For use in IBM and non-IBM products
    - http://www.ogf.org/dfdl/

- Support more features of the DFDL specification
  - Fields with length prefixes (eg, PL/1, ISO8583)
  - Default values for missing structures when serializing

- More ways to create DFDL models
  - Import from C header files

- Usability enhancements to DFDL editor
  - More copy/paste & keyboard shortcuts
  - Multiple object selection

- Improved performance
  - Continued improvement when parsing & writing
  - Big gains for text numbers and packed decimals
Graphical Transformations

- **IBM Graphical Data Mapper (GDM)**
  - Visually map and transform source to target data
    - Code-free, high performing & scalable
  - GDM designed for whole IBM product set
    - e.g. Full map exchange with MDM Server V9
    - Mapping Script Language (MSL) format

- **Simple & powerful graphical mapping experience**
  - Apply transformations to single and multiple elements
    - Conditionals (*if then else*), Loops (*for each*), Functions (*target = f(source)*) & more!
  - Database mapping sources and targets for routing and enrichment scenarios
    - Broad database support (Oracle, DB2, SQL Server, Sybase, Informix…)
  - Complements and supports existing transformation languages
    - Call user defined transformation in Java, SQL, XPath, .NET…

- **Simple deployment, high performing & scalable**
  - Maps can be deployed with solution BAR file or stand-alone
  - JIT compile means technology advances improves existing, deployed solutions
    - Source deploy + runtime compilation = enhanced performance

- **Migration from pre-version 8 maps**
  - Existing maps developed before version 8 continue to work as-is
    - Existing maps opened in read-only mode for visualization & comprehension, cannot be modified
    - Automatic conversion of `.m fmap` format to MSL not currently built-in
JMS Receive Node and other JMS Enhancements

- **New JMSReceive Node supports all JMS 1.1 Providers**
  - Process JMS messages in the middle of a message flow, c.f. MQGET node
  - Typical scenarios include request response, routing & augmentation
  - Works with any JMS 1.1 provider, MQ is default provider

- **JMS Receive node**
  - Works on JMS queues: receive paradigm is not applicable to topics!
  - Can be configured for destructive read or browse

- **Comprehensive & Flexible options**
  - Retrieve particular JMS with message properties
  - Per message customization
    - Many **LocalEnvironment** overrides!
  - Flexible data locations
    - Incoming & received message can be kept

- **Activity Logging**
  - All JMS nodes updated to provide activity logging
  - Allows operators to understand JMS operations without understanding detailed flow design
    - e.g. failed to open or start JMS session, message sent to destination

- **Other JMS Enhancements**
  - Allows generic session object to be overridden as queue or topic
  - `JMSDestinationList.DestinationList.Queue = topic|queue;`
Making it Easier to Understand Broker Behaviour

- **New 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 quantitative resource statistics

- **Flow & resource logging**
  - User can observe all events for a given flow
    - e.g. “GET MQ message”, “Send IDOC to SAP”, “Commit transaction”…
  - 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 and C:D 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
Dynamic Deployment of AD Artefacts

- Allow sub-flows to be deployed independently of main flow
  - Additional to existing build-time sub-flow; no performance impact
  - New “Route to sub-flow” allows dynamic addition of new/changed logic
  - Intuitive Drag and drop deploy & simple BAR file packaging
    - Sub-flow is fully visible as development artefact c.f. message flow

- Independently deployable ESQL
  - Particularly useful for dynamic transformation scenarios
  - Allows new/changed transformation without whole-flow redeploy
  - Intuitive Drag and drop deploy & simple BAR file packaging

- Deploy Flow Stopped provides fine grained initialization control
  - Important in “order-of-initialization” type scenarios
    - Allows operator to declare initial state for deployed flow resources
      - Manual: always needs to be started by user
      - Automatic: always started by broker
      - Maintained: remember
      - Persists over expected or unexpected restarts

- Deployable Maps & Schemas
  - Graphical maps & XSDs (XML and DFDL) can now be deployed independent of flow
  - Simplifies change management for incremental solutions
    - Just deploy changed artefacts rather than whole flow!
Web-based Patterns for Easy Solution Creation

- **Patterns Based Development**
  - Quickly create best practice solutions from pre-built templates
    - e.g. WS façades, message processing, file to queue…
  - IBM pre-supplied & User Defined Patterns
    - Create & share user patterns
    - Including community downloads

- **Web-based quick & simple pattern generation**
  - Allows end users to configure repeatable solutions
  - New tool aimed at web user
  - Configure and deploy patterns directly to broker
  - Role-based access & security for appropriate authorization

- **Complements existing tools**
  - Pre-built, user-defined and imported patterns
  - Design allows for future inclusion of user patterns
    1. Build .patternzip in MB Toolkit
    2. Import for web tool
    3. Configure and deploy
  - Exploits source deployment (see later)
  - Move from test to QA to production

- **Operational Management**
  - Manage patterns using standard MB tools
    - e.g. MBX, web UI, CMP…
IBM Integration Bus

What’s New in Version 9
Migration from WebSphere Message Broker V6.1, V7 and V8

- **Migration from WMB V6.1, V7 and V8**
  - All development assets (e.g. message flows, ESQL, DFDL, Java, Maps and XSLT) import directly
    - Right-click convert action for pre-V8 maps; some manual tasks may be required
  - Migrate brokers using a single command, or create new brokers for phased migration
    - No broker redeployment necessary when using built-in migrate command
    - All existing BAR files can be deployed to IB V9 brokers without change

- **Migration commands for in-place migration**
  - Includes migration of configuration data including broker databases, queues and registry
  - Forwards and backwards migration of existing components, in situ
    - `mqsimigratecomponents` command (includes `–t` option for rollback to V7 and V8)

- **Flexible co-existence options remove the need for additional hardware when migrating**
  - IB V9 co-exists on the same OS with all previous MB versions
  - MQ V7.5.0.1 required for all IB V9 brokers
    - MQ V7.5.01 supported with V7 and V8 brokers for the purposes of V9 migration
    - For V6.1 migration, upgrade MQ and MB simultaneously

---

1. Install IB V9
2. Stop broker
3. Run migrate command
4. Start broker

or

1. Install IB V9
2. Create new broker
3. Deploy existing assets
4. Stop old broker
Web Visualisation and Analytics

- **A comprehensive tool for web management**
  - Manage all integration resources from zero-footprint client
  - Analyze integration performance in real-time
  - Supported on a variety of browsers: IE10, Firefox, Safari...
  - Complements MQ Explorer and WAS Admin consoles

- **Managing Integration Resources**
  - View top-level integration node properties
  - Add/remove/change integration servers
  - Start/Stop integration data flows
  - Role based access to control usage
  - Advanced options include data replay, policy & monitoring
  - Exploits underlying public REST/JSON API

- **Integration Performance Analysis**
  - Operational experience; no developer intervention required
    - New and existing flows can exploit without change
  - Many metrics of integration flow available in real-time
    - CPU & I/O time shown by default in integration analyzer
    - Other metrics include thread, data sizes, errors...
  - Flexible display includes data tables and flow profile
    - Drill down to understand detailed behaviour
  - Exploits underlying MQTT web sockets technology
    - Asynchronous notification at low CPU cost
IBM DFDL in IBM Integration Bus

- IIB v9 uses IBM DFDL v1.1 component
- DFDL domain and parser
  - Available in nodes, ESQL, Java, …
  - Use instead of MRM CWF/TDS
  - More capable and higher performing
  - Adopts XMLNSC tree shape
- DFDL models
  - Schema files reside in IIB libraries
  - Not in Message Sets
- Tooling for creating DFDL models in IIBTK
- DFDL model debugger
  - Debug parsing & writing of data in IIBTK
  - No deploy to runtime necessary!
- DFDL schema deployed in BAR file
  - No dictionary file
- Migration from v8
DFDL Adoption

- IBM DFDL reusable component ships with:
  - IBM WebSphere Message Broker 8.0
  - IBM Integration Bus 9.0
  - IBM Rational® Performance Test Server 8.0.1
  - IBM Rational Test Virtualization Server 8.0.1
  - IBM Rational Test Workbench 8.0.1
  - IBM Rational Developer for System z 8.5
  - IBM InfoSphere ® Master Data Management 11

- Further IBM products and appliances investigating adoption

- Open-source DFDL implementation in progress ‘Daffodil’
  - Available as an alpha release (parser only)
  - Partly funded by a major US government agency who are adopting DFDL as their standard way to parse text and binary data

- DFDL web community on GitHub for collaborative authoring of DFDL schemas for commercial and scientific data formats
DFDL Schemas Web Community

- Free public repository for DFDL models
- Hosted on the popular GitHub community website
- Unlimited read-only access
- Collaboration encouraged
- Evolving content
Controlling Integrations with Policy

- **Integration Workload Management**
  - Provide intelligent mechanisms to control processing speed
  - Most common scenario is to reduce back-end server load
  - Design allows more policy-based processing over time
  - Can be applied to new or existing integration data flows

- **Policy defines threshold limits and relevant actions**
  - Set thresholds for integration data flow throughput
  - Specify actions at threshold, for example:
    - NOTIFY: Higher (or lower) than threshold generates publication
    - DELAY: Excessive workload will have latency added to shape throughput

- **Web Console used to manage WLM policy**
  - Sophisticated behaviour controllable by broker WLM policy
  - Workload can be managed across classes of message flows (e.g. batch vs. online)
  - Policies stored in local registry, and dynamically configurable
  - Developer can also specify limits as integration data flow properties
Managing Unresponsive Integration Flows

- **Target unresponsive flows through policy to improve overall system reliability**
  - Additional WLM option aimed at unresponsive integration flows
  - An integration flow can become unresponsive for multiple reasons
    - e.g. Waiting for external system, infinite loop, deadlock, malformed XML

- **Flexible configuration, actions and reporting options**
  - Specify threshold at which flows are considered unresponsive, e.g. 30 seconds for processing
    - Configured via WLM policy, or directly on the flow in the BAR file
  - Define action to trigger when flow considered unresponsive
    - Administrative notification through a new “timeout exceeded” event message
      - If flow eventually continues through to completion, a second event is published
    - Restart the integration server (execution group) on which the unresponsive flow is running
  - New command option to forcibly stop integrations manually: `mqsistopmsgflow -f`
Understand and Act on In-flight data

- **Provide business insight during integration data flows**
  - e.g. intelligent decision making; score then action in-flight request based on a business rule
  - User creates (e.g.) if-then-else rules using tool of choice (Excel, Word, Eclipse…)
  - The bus acts on these rules in flow, e.g. for business level routing

- **New Decision Service node**
  - Identifies inputs to business rules from in-flight data
    - e.g. **details of book order** from request
    - e.g. **the item price** from key fields…
  - Invokes built-in rule engine to perform business logic
    - Open interfaces for 3rd party and user engines
  - Captures rules output for downstream processing
    - Business objects mapped back to in-flight data

- **Create rules directly inside Integration Bus toolkit**
  - Significant rules authoring facility built-in
  - Automatic package & deploy with integration assets
  - Dynamically reconfigure business rule using configurable service policy
  - Optionally refer to business rules on external ODM decision server
  - Exploit separate full ODM Decision Center for BRMS scenarios

- **Embedded rules engine for high performance**
  - Rule is executed in the same OS process as integration data flow
    - Succeeds IAM9 Support Pac
  - Rule update notification ensures consistent rule execution
  - Optional governance of rules through remote ODM Decision Center
Graphical Transformations

- **IBM Graphical Data Mapper (GDM)**
  - Visually map and transform source to target data
  - GDM designed for whole IBM product set, e.g.
    - Integration Bus V9, WebSphere Message Broker v8, DataPower
    - InfoSphere Master Data Management v10, Integration Designer v7.5/v8
    - Rational Application Developer for WebSphere Software v8.5
    - Rational Software Architect v8.5, RSA for WebSphere Software v8.5
    - Other products yet to announce
  - Rich feature set and simplicity make this a good default transformation choice

- **Directly access stored procedures from within a map**
  - Complements existing database select, insert, update, delete
  - Incorporate user-defined database functions into your graphical transforms

- **Maps available to user patterns**
  - Graphical creation of flows which require transformation logic
    - e.g. new input or output messages
  - Invocation of mapper when pattern instances are generated
  - User guidance through HTML pattern help and task list
  - Patterns to demonstrate include CRM account mapping

- **Migration of pre-V8 maps to IBM GDM**
  - Most sophisticated maps can now be converted in a single step
  - Editor provides enhanced feedback about conversion to assist user understanding
Scenario 1 - Storing state for integrations

- With a global cache, each broker can handle replies – even when the request was processed by another broker.
Scenario 2 - Caching infrequently changing data

- With a global cache, the number of clients can increase while maintaining a predictable response time for each client.
WebSphere eXtremeScale Overview

- **Elastic “In-Memory” Data Grid**

- **Virtualizes free memory within a grid of JVMs into a single logical space**
  - Accessible as partitioned, key addressable map by applications and subsystems

- **Provides fault tolerance through replication**
  - e.g. Primary/secondary stores with failover, voting etc…

- **Easy to Scale**
  - Add more JVMs dynamically while it’s running without restart

- **Available as component or standalone software and hardware appliance**
  - Foundational technology used “under the covers” in Message Broker
WMB Global Cache in A Nutshell

- **WMB contains an embedded WebSphere eXtreme Scale grid**
  - WXS components are hosted within execution group processes

- **It works out of the box, with default settings, with no configuration**
  - You just have to switch it on!

- **The default scope of one cache is across one Broker**
  - Starts with multiple execution groups but easy to extend to multiple brokers

- **Advanced configuration available**
  - Execution group properties and Policy Profiles for more sophisticated topologies

- **MB developer has simple artefacts for working with the global cache**
  - Unaware of the underlying technology (WXS) or topology
Global Cache Enhancements

- **IB contains a built-in facility to share data between multiple brokers**
  - Improve mediation response times and dramatically reduce application load
  - Typical scenarios include multi-broker request-reply and multi-broker aggregation
  - Uses WebSphere Extreme Scale coherent cache technology

- **Support for external software and hardware caches**
  - Access separate eXtreme Scale and DataPower XC10 appliances from within the broker
  - Allows broker to interact with enterprise caching solution without embedding additional libraries
  - Cache access, activity log, resource statistics etc. just like embedded cache
  - Operationally configured using dynamic configurable service
  - New EG options to specify SSL connections to external WXS grids
    - Uses existing MB SSL infrastructure to configure certificates

- **Cache Expiry options**
  - New getGlobalMap() variant to set the time to live for data in the embedded global cache.
    ```java
    MbGlobalMap evictMap = MbGlobalMap.getGlobalMap("...", new MbGlobalMapSessionPolicy(30));
    evictMap.put("key", "val");
    ```
  - Specify a value in seconds. The default value is 0, which means data never gets automatically removed.

- **Programming and operational enhancements**
  - Insert and lookup map data using a wider range of Java object types for simplified programming logic
  - Support for highly available multi-instance configurations
Improvements for our z/OS Users

- **IBM Integration Bus is a compelling choice for z/OS users**
  - Broad connectivity options to support processing of z/OS subsystems
    - WAS, CICS, IMS, DB2, File...
  - Makes use of z/OS specific features such as Sysplex, security, automatic restart and WLM
  - New IIB features demonstrate commitment to the z/OS platform

- **Standard Edition Pricing on z/OS**
  - New entry-level edition offers flexibility to fulfil either broad-capability or high-performance scenarios

- **Different users per execution group**
  - The userid associated with each execution group address space is now configurable on z/OS
  - The execution group exhibits that userid for all resource manager interactions (e.g. MQ, DB2)
  - Configurable via execution group profile; takes effect when an execution group is started

- **Co-ordinated transactions for CICS requests**
  - The CICSRequest node now supports broker coordinated transactions (one-phase commit)
  - Allows multiple requests to a CICS server to be handled as part of the same transaction

- **Activity log for CICS transactions**
  - Provides a high-level overview of the recent interactions between IBM Integration Bus and CICS
  - Includes CICS invocation successes, failures, abends, security, timeouts and transactional state
Other Features Our Users Requested

- **Developer Edition**
  - Free edition of IB with all nodes available and no time limitations
  - Throughput rate limited to 1TPS per integration flow
  - Assistance through user community (e.g. mqseries.net)
    - No formal IBM support
  - Simple to download, install and use
    - Single installation package contains ALL required software:
      - MQ 7.5, Integration Bus (Runtime, Toolkit, Explorer)
    - Available on Windows and Linux platforms

- **DFDL may be used in standalone applications**
  - Strategic modelling technology now available as separable components
  - Simple to configure: Install Integration Bus, copy DFDL libraries to appropriate location

- **Flexible statistics output**
  - Performance statistics can now be directed to multiple destinations (publication, user trace, SMF)

- **Sub-second timeout on Aggregation nodes**
  - More granular timeout values (ms) can now be specified on the aggregation nodes
  - Allows for quicker timeouts when aggregating data from usually fast responding systems

- **ODBC Database verification (Linux/UNIX)**
  - Broker environment, and ODBC connections defined both to the broker and in odbc.ini are verified
  - Run at broker startup (or with the mqsicvp command) ensures early capture of potential problems
IBM Integration Bus V9 - Summary

- **IBM Integration Bus is IBM’s Strategic Integration Technology**
  - Single engineered product for .NET, Java and fully heterogeneous integration scenarios
  - Unparalleled range of connectivity options and capabilities
  - Supports users’ range of experience & needs
  - Industry leading performance in a broad range of scenarios

- **A strong feature set for V9 and beyond**
  - We are working on a significant number of features for the next evolution of Message Broker tech
    - More to come - this is not a definitive list!
  - Continuous delivery throughout 2013 and beyond; features rolled back as available
  - Builds on the continued success of V7 and V8 major engineering releases
  - Content heavily influenced by user requirements, participation and feedback

- **Diverse connectivity requirements**
  - Simple & Productive to make connectivity easy and powerful
  - Universal & Independent to connect everything you need in the way you want to manage it
  - Industry Specific & Relevant to help solve business problems
  - Dynamic & Intelligent to create flexible solutions that provide business insight
  - High Performing & Scalable to maximize hardware and grow with you
Demo