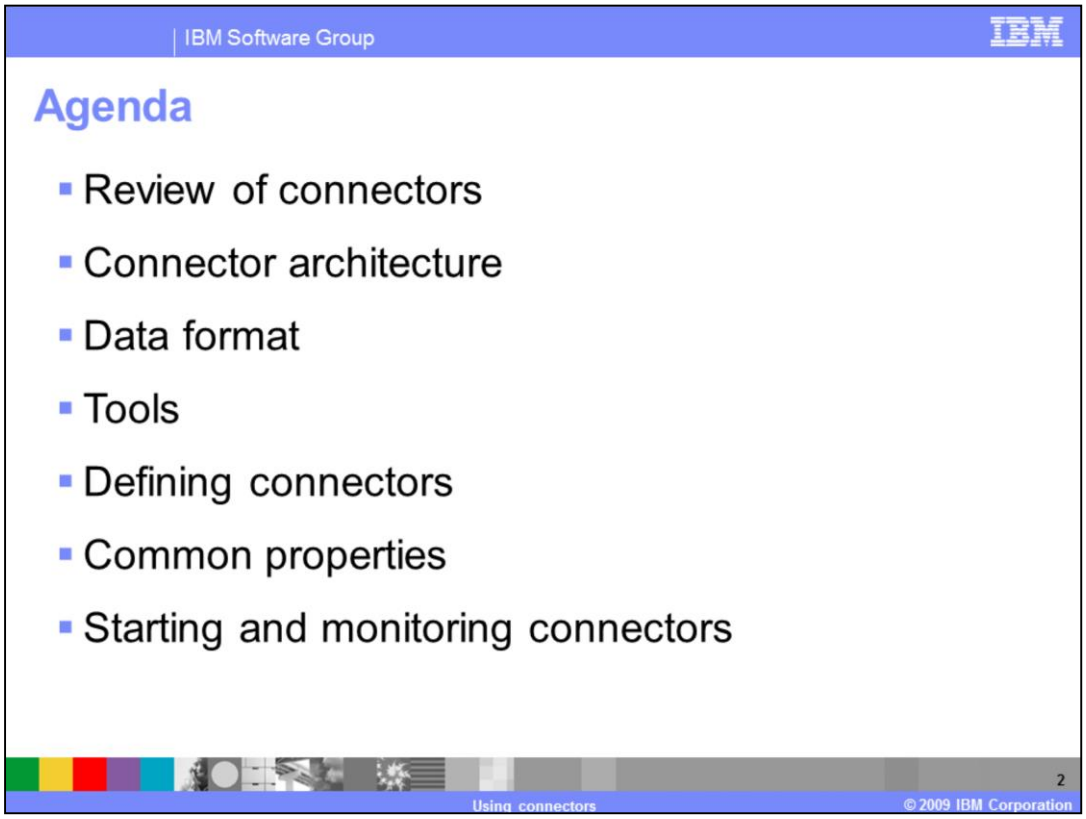
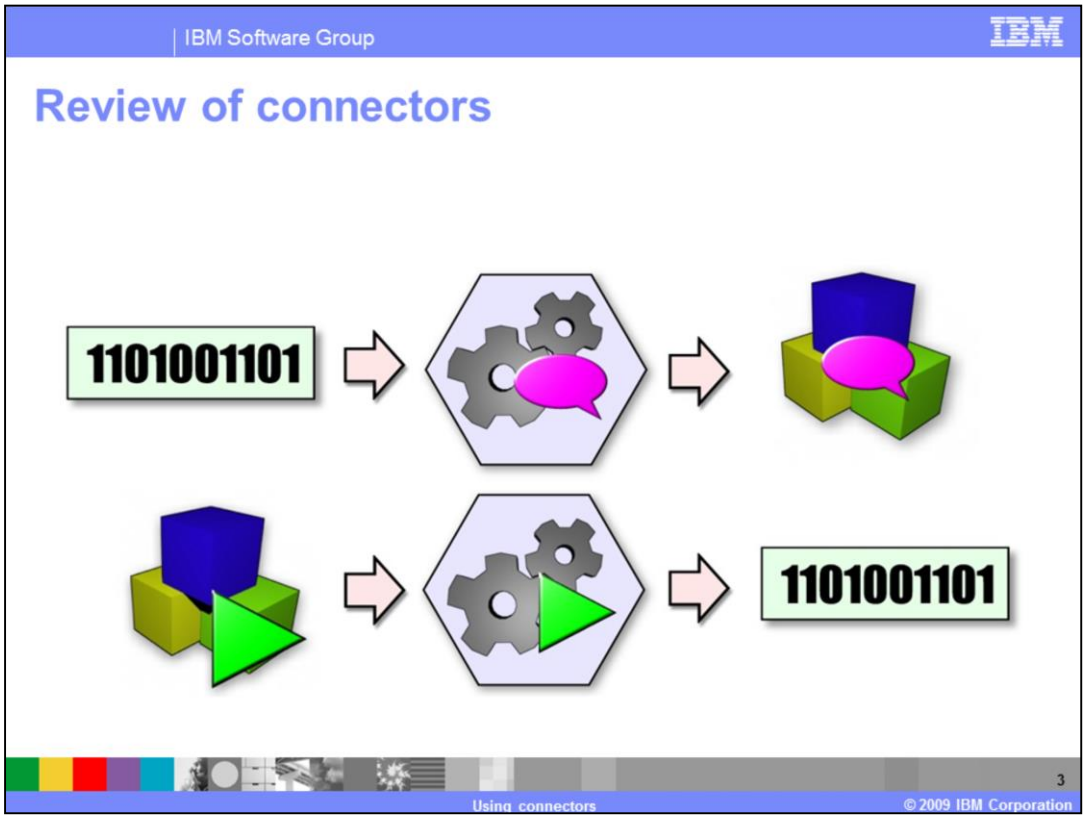
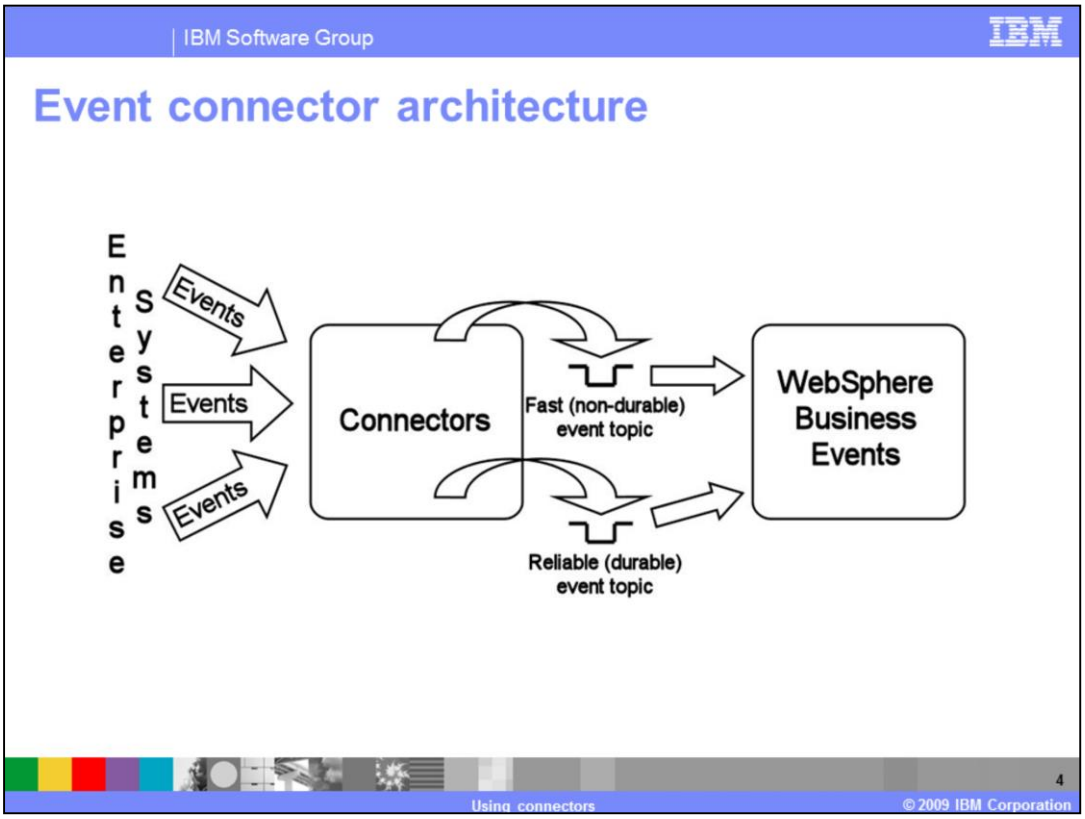
This presentation will give you a brief overview of what connectors are before going into detail on how the connectors fit into the WebSphere® Business Events solution architecture and how they can be configured and managed.



Having understood what connectors are and how they fit within a solution you will look at how connectors interact with the enterprise systems. You will then see how you can define connectors and the configuration properties that are common to different connector types. Finally you will look at how connectors are started and how you can diagnose problems with the connectors you have configured.



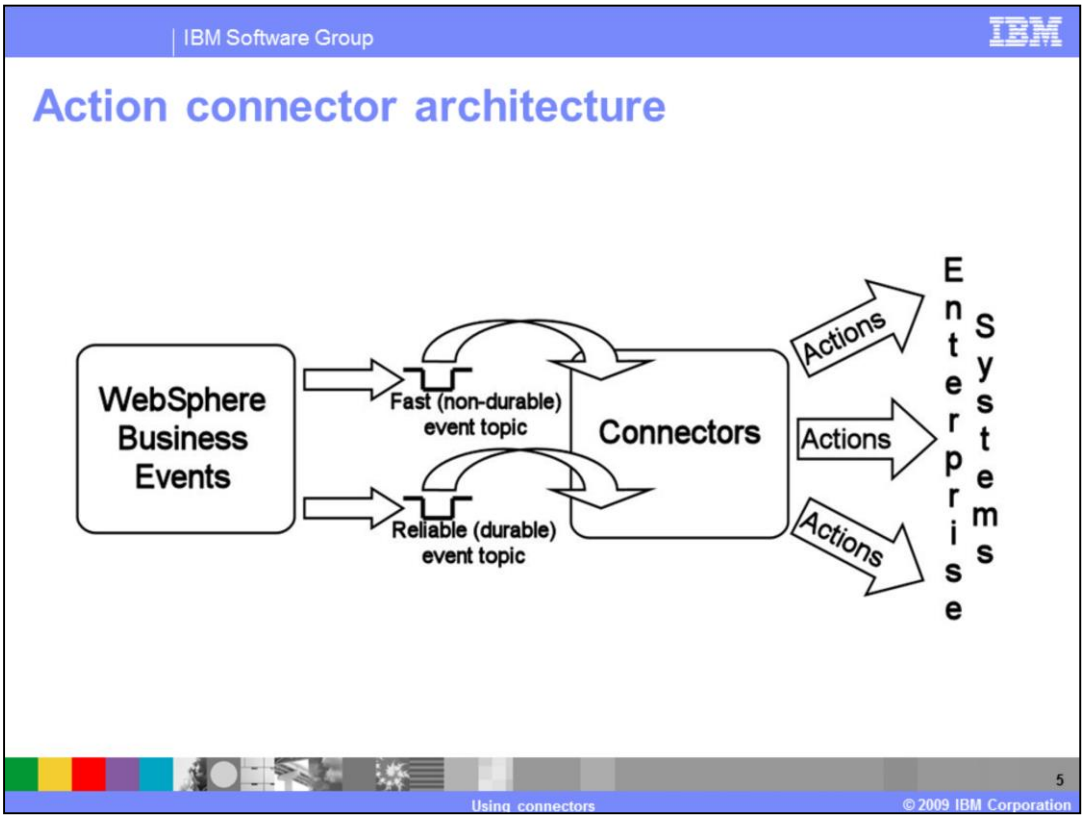
A connector is a bridge between the external world of bits and bytes and the internal world of WebSphere Business Events. Connectors are defined as part of an event or action and are designed to translate WebSphere Business Events XML packets to and from the appropriate protocols. A connector gets data from an external system which results in an incoming event. In addition, when WebSphere Business Events generates an action, a connector can be called which externalizes data to an external system. The connectors can interact with a wide variety of transport protocols and technologies.



There are two halves to the connector architecture. This diagram shows you the architecture of the inbound or event connectors, you will look at the action connector architecture in a moment.

Enterprise systems emit events which are received by the intended connector. The connector then parses the received event and puts it into the defined XML format which is then placed on the appropriate JMS topic. The WebSphere Business Events runtime engine monitors these topics and processes any events that it receives.

There are two topics available for incoming events to be placed on; the Fast (non-durable) topic and the Reliable (durable) topic. When placing messages on the durable topic they are persisted to a database. This means that in the event of a server failure they are either consumed by the server when it restarts or by another member of the WebSphere Business Events cluster. In the case of events which are placed on the non-durable event topic, when a server or cluster fails any events which have not been processed by the runtime engine are lost.



This diagram shows you the architecture of the outbound or action connectors.

When WebSphere Business Events processes an interaction set which results in an action being triggered it places this action, using the defined XML format, onto an action topic. The intended action connector detects the arrival of this action object and sends the action to the configured enterprise system.

As with inbound events, there are two topics available for actions to be placed on; the Fast (non-durable) topic and the Reliable (durable) topic. When placing messages on the durable topic they are persisted to a database. In the event of a server failure, since the messages are persisted they can be consumed by the connectors when the server restarts. In the case of actions which are placed on the non-durable topic, when a server or cluster fails any events which have not been processed by the runtime engine and are waiting on the non-durable topic are lost.

In terms of how connectors are installed in an actual WebSphere Business Events deployment, connectors run in a separate JVM™ to the WebSphere Application Server instance that is running the WebSphere Business Events runtime. It is up to the solution architect as to whether this JVM is on the same physical machine as the WebSphere Application Server. Only one connector instance should be installed and configured per WebSphere Business Events runtime engine. In some solution scenarios it can be architecturally more sound to have the enterprise systems emit and consume events directly to and from the event and action topics themselves. When doing this the enterprise systems will of course have to emit and consume events and actions in the correct WebSphere Business Events XML packet format.

Format of event and action data

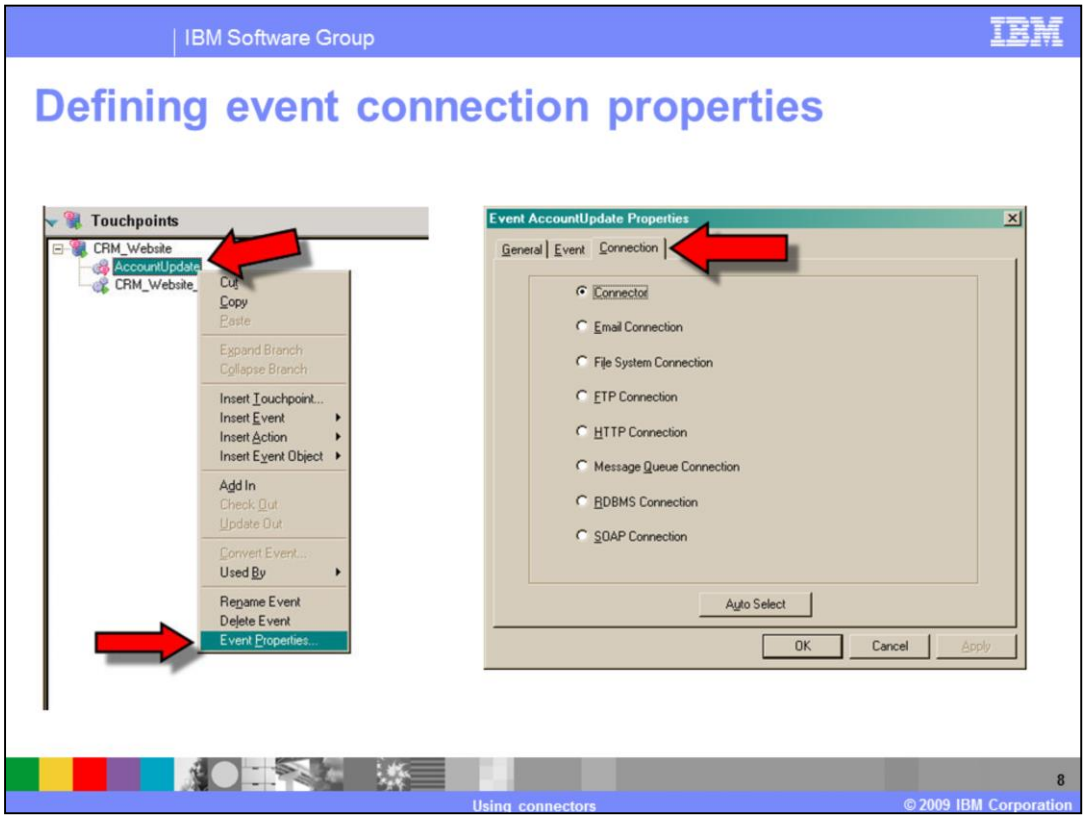
- Format of event and action data is XML
- XML data well defined and XML schema definition provided or generated
- Non compliant data can be transformed via:
  - XSLT
  - WebSphere Enterprise Service Bus
  - WebSphere Process Server
  - WebSphere Transformation Extender
  - Others

The data that arrives at the WebSphere Business Events runtime as an event and the data that is externalized by WebSphere Business Events as an action is physically formatted and transmitted as an XML document. The structure of this XML is described by an XML schema that is supplied with the product or generated for you in the case of custom data. If the incoming data you want to work with is either not XML or not appropriately formatted XML for WebSphere Business Events usage, then it needs to be modified into a conforming format. The data can be modified inside WebSphere Business Events through XSLT transforms or before arriving at WebSphere Business Events through technologies such as WebSphere Enterprise Service Bus, WebSphere Message Broker, WebSphere Transformation Extender or others. The same story is also true for Action data that is externalized for consumption by external applications. The data is externalized as an instance of an XML document. You will see later that the source and destination of the XML can be configured to be transmitted over a variety of protocols.

The development tools

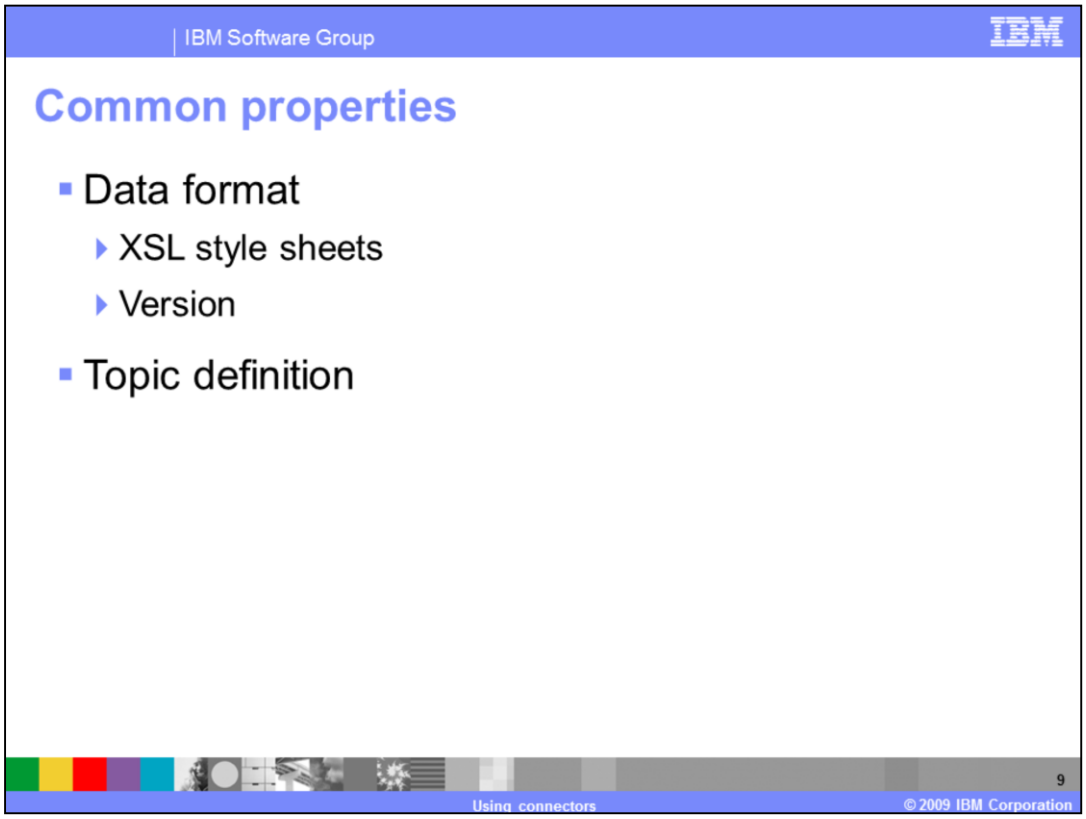
- **WebSphere Business Events: Design Data** used to define
  - Touchpoints
  - Events
  - Actions
  - Intermediate objects
  - Connectors

WebSphere Business Events provides two primary tools for development. These are called WebSphere Business Events:Design Data and WebSphere Business Events: Design. WebSphere Business Events:Design Data is used by the IT developer to define events, actions, touchpoints, intermediate objects and connectors. Therefore, it is the WebSphere Business Events: Design Data tool that is used for configuring connectors. The WebSphere Business Events:Design tool is used by the business user to define the rules, in the form of interaction sets and filters, that are evaluated when the events arrive.



We'll now take a look at how a user goes about creating an event connector.

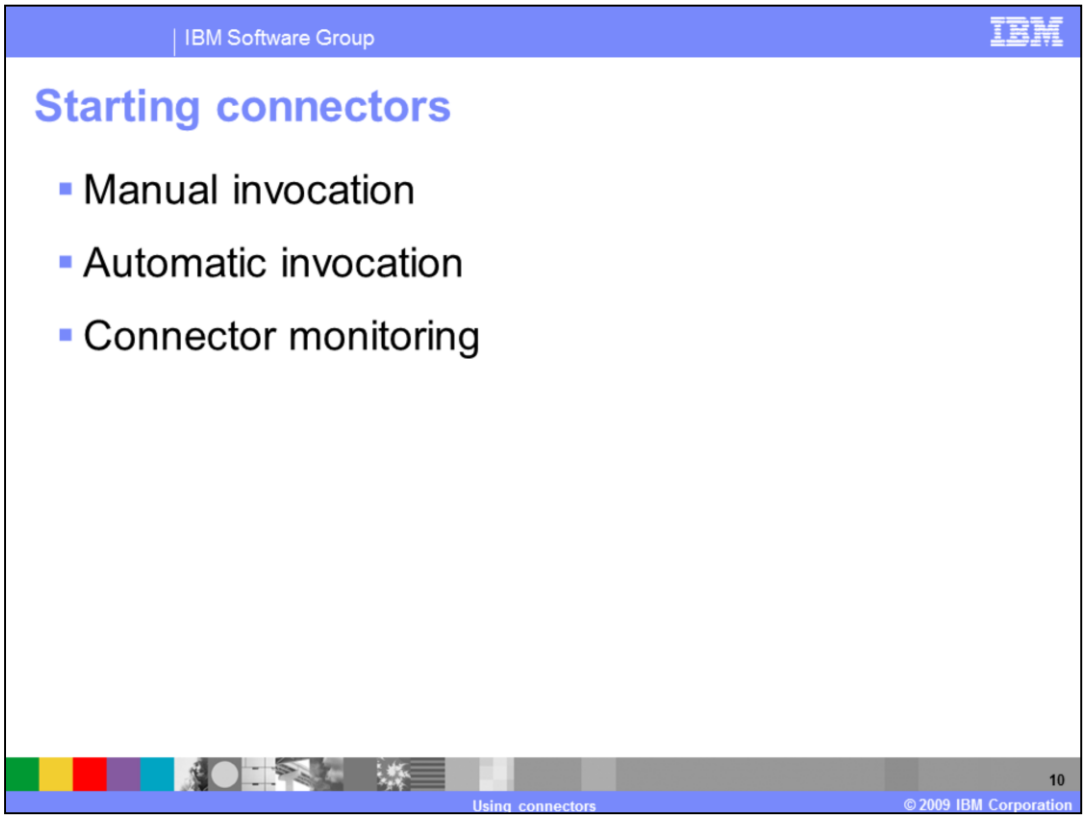
Having defined an event or action for a specific Touchpoint you then define the connector for the given event by right clicking on the event, selecting Event Properties and then clicking on the Connection tab. Here you are presented with the various connector types which can be configured for this event. Selecting the radio button for the required connection type will result in the connection properties window for the specific connector type being displayed. The user must then provide the required configuration information to complete the setup of the connector. The details of this configuration are described in detail in later presentations in the series and in the product information center.



There are two areas of commonality across the range of connectors which WebSphere Business Events provides, the data format and the topic definition. All of the connector configurations allow you to define the format of the incoming and outgoing data. This might be to define the input file as being a comma separated file or an e-mail as being plain text.

With the exception of the Relational Database Management System (or RDBMS) connector, all of the connectors allow you to specify that this data is in a custom XML format. If this option is selected then you must configure the XSL style sheet that should be used to convert the data to and from the standard format. Additionally, action connectors allow you to define which version of the WebSphere Business Events packet format should be used for the emitted actions. This can be one of either version 2, version 2.2 or version 6.2. The recommended format is version 6.2.

The second area of commonality is with regards to the configuration of the topic that the connectors use to pass messages between the connector instance and the WebSphere Business Events runtime engine. When configuring an event connector, the Advanced panel for each connector provides the option to define whether a fast (non-durable) or reliable (durable) topic is used. When configuring an action, the Action tab of the action definition provides the option to define whether a fast or reliable topic is used.



Connectors can either be started manually or they can be configured to be started automatically by the operating system.

To start the connectors manually you should run the connectors.bat or connectors.sh script which is provided as part of the installation. In order for the connectors to start successfully the WebSphere Business Events runtime engine must have been previously started.

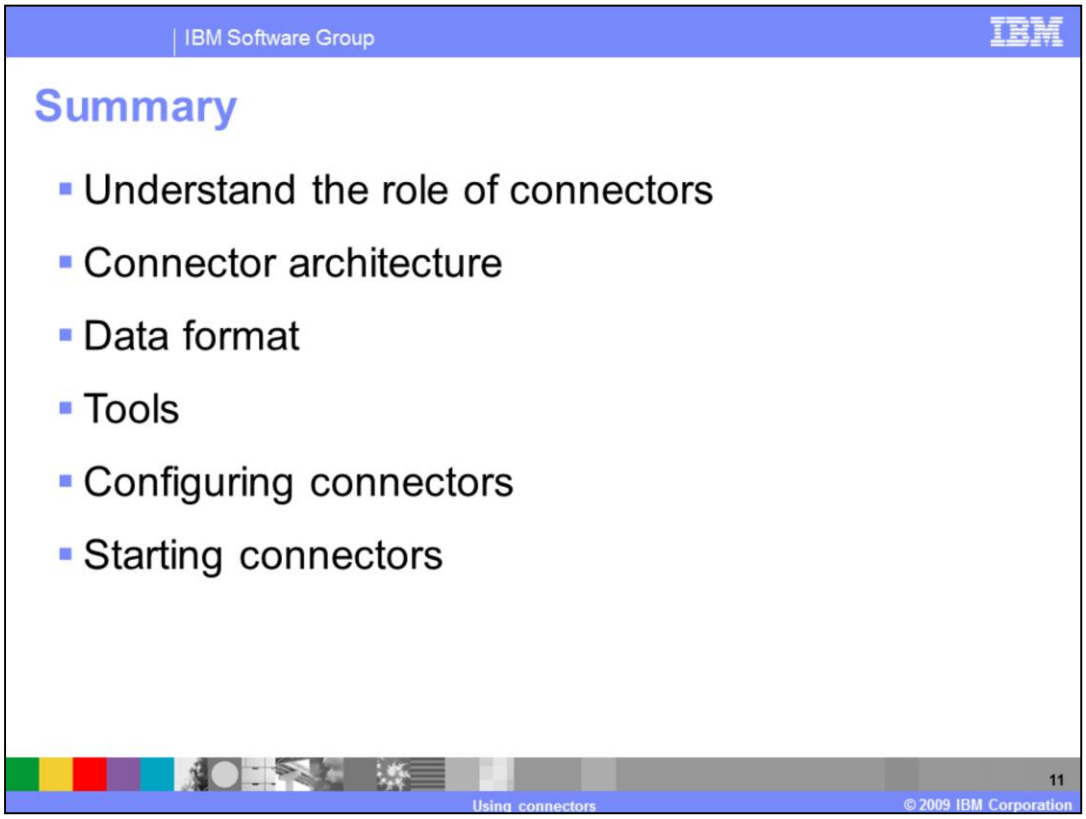
Having run the script to start the connectors, a user can determine which connectors are active by typing ‘modules’ at the connector console prompt.

In order to stop the connectors you can type ‘stop’ at the connector console prompt.

In order to configure the connectors to be started automatically you can configure the connectors to be run as a Windows® service or to be run as a background process in Linux® and UNIX®. On Linux and UNIX this background process can be started automatically during system start.

There are several configuration parameters available when registering the service as a Windows service to allow for finer grained configuration. These allow you to define whether the service is started automatically, any dependencies on other services and any required start-up delays. Details of these configuration parameters can be found in the product information center or by running the command ‘WBEConnectors.exe’.

If you have problems with your connectors not behaving as expected then there are two possible sources of information to assist in finding the cause. The connectors.log file (located in the logs directory) provides information on the connector status during start-up, execution and shutdown. Additionally, if there is a serious failure that causes a Java™ dump then the details of the stop can be found in the directory from which the connectors were started.



In this presentation you have seen how connectors fit into the solution architecture, how they can be defined for specific events and actions and how they can be configured, started and monitored.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java, JVM, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.