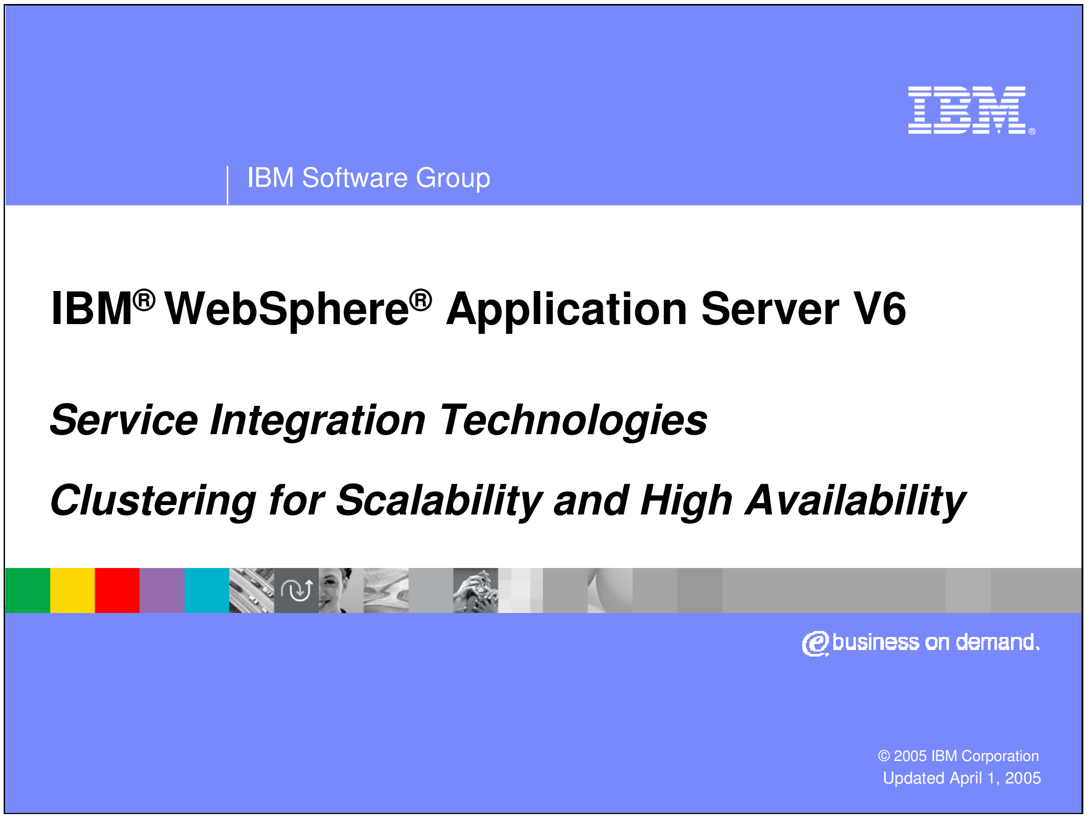


IBM Software Group

IBM® WebSphere® Application Server V6

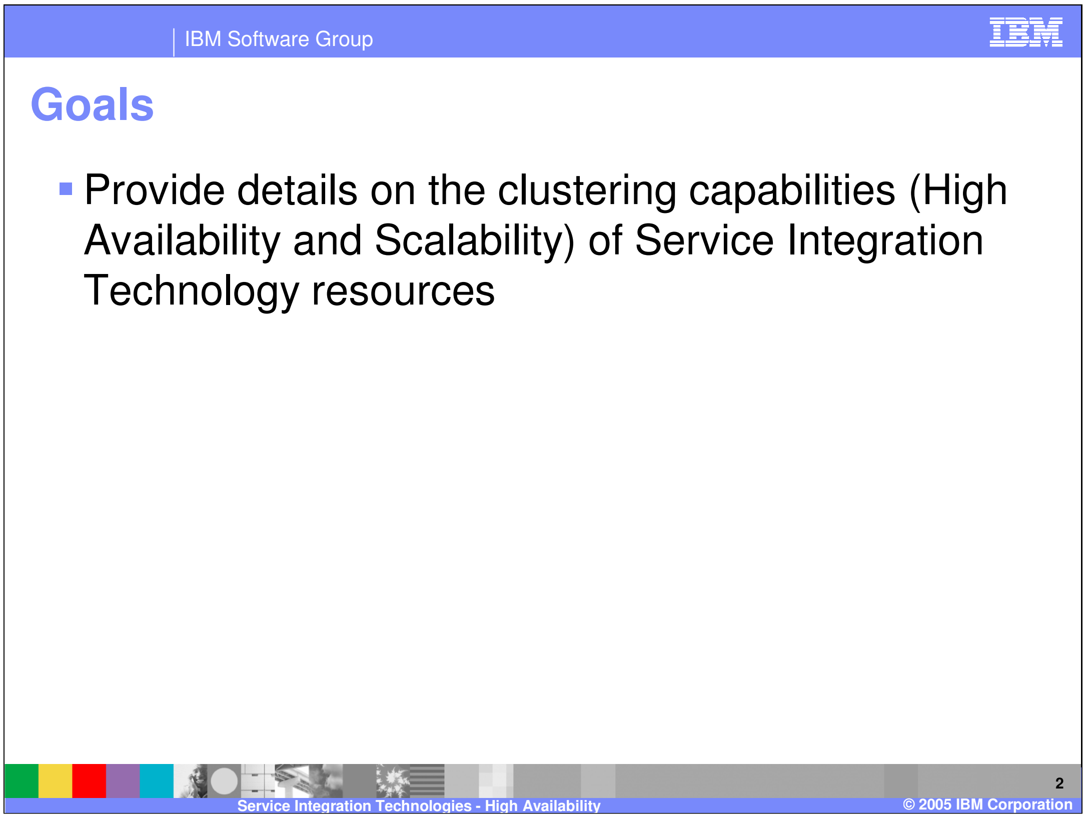
Service Integration Technologies

Clustering for Scalability and High Availability

@business on demand.

Updated April 1, 2005

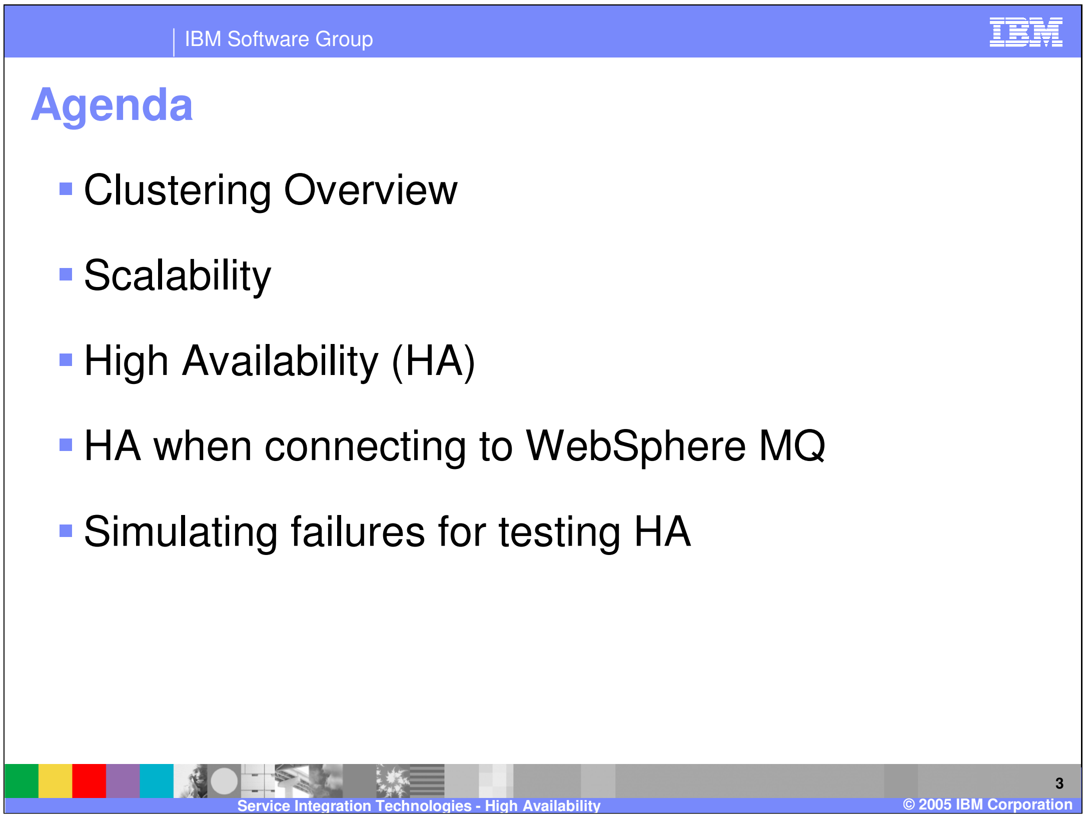
This presentation will focus on the Clustering capability of the new Service Integration Technology capability in WebSphere Application Server V6.



Goals

- Provide details on the clustering capabilities (High Availability and Scalability) of Service Integration Technology resources

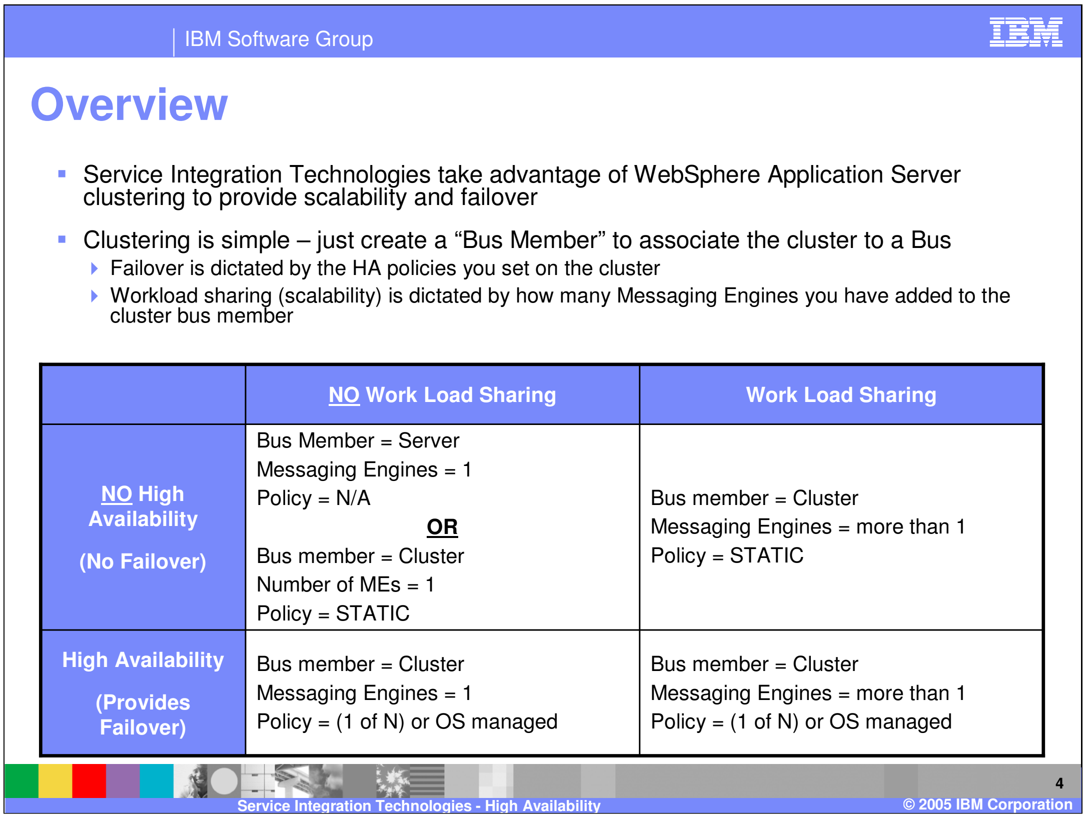
Understanding the architecture of the Service Integration Technology is needed to understand its clustering capabilities. These capabilities apply to a Network Deployment cell. A stand-alone Application Server environment does not have clustering capability.



Agenda

- Clustering Overview
- Scalability
- High Availability (HA)
- HA when connecting to WebSphere MQ
- Simulating failures for testing HA

This presentation will cover an Overview followed by some details on how to set up Scalability and High Availability, followed by some issues to consider when connecting to WebSphere MQ. Then it will show some commands to simulate failures for test.



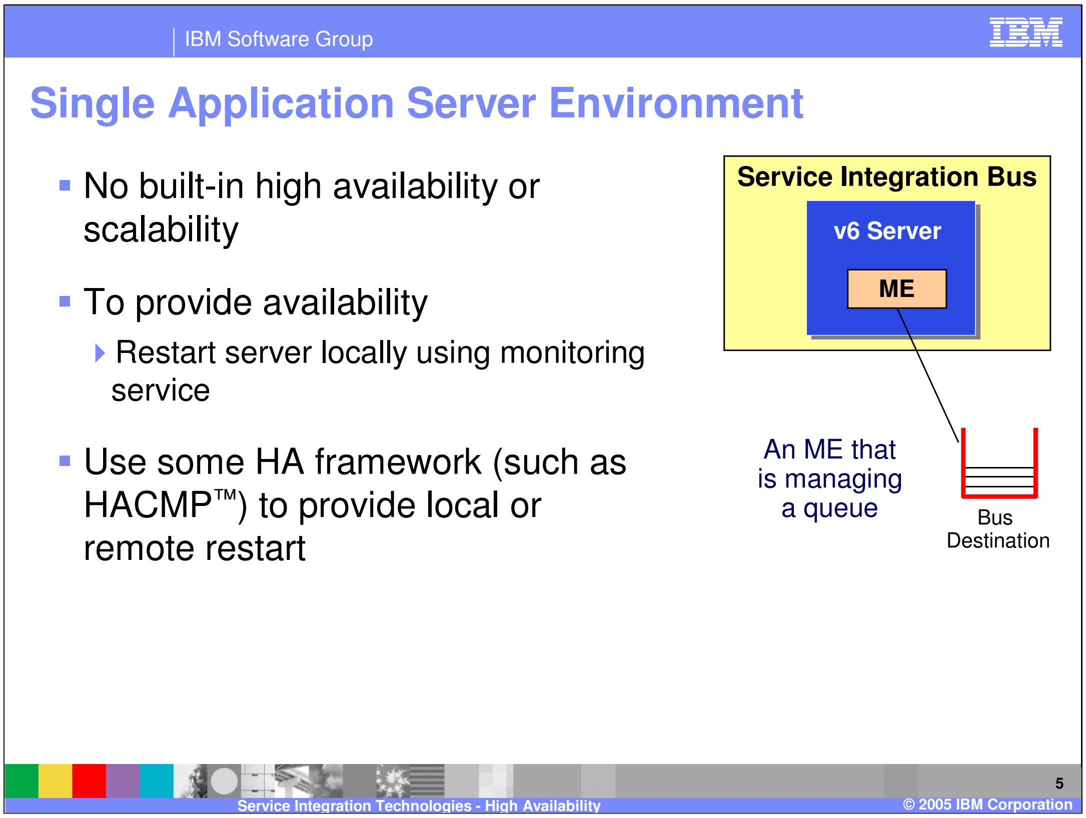
Overview

- Service Integration Technologies take advantage of WebSphere Application Server clustering to provide scalability and failover
- Clustering is simple – just create a “Bus Member” to associate the cluster to a Bus
  - Failover is dictated by the HA policies you set on the cluster
  - Workload sharing (scalability) is dictated by how many Messaging Engines you have added to the cluster bus member

	NO Work Load Sharing	**Work Load Sharing**
NO High Availability **(No Failover)**	Bus Member = Server Messaging Engines = 1 Policy = N/A **OR** Bus member = Cluster Number of MEs = 1 Policy = STATIC	Bus member = Cluster Messaging Engines = more than 1 Policy = STATIC
High Availability **(Provides Failover)**	Bus member = Cluster Messaging Engines = 1 Policy = (1 of N) or OS managed	Bus member = Cluster Messaging Engines = more than 1 Policy = (1 of N) or OS managed

Unlike WebSphere Application Server V5, the V6 implementation supports Clustering. You get the clustering capabilities by adding a Bus member of type Cluster to the Service Integration Bus. The High Availability is dictated by the HA policy on the Cluster. The scalability is dictated by the number of Messaging engines added in the cluster. By default, when you add a Cluster to a bus, one Messaging engine is created. To get scalability, all you need to do is add additional Messaging engines to the Cluster Bus member.

Looking at the table, you can see that based on the type of Bus member, number of Messaging engines and the HA policy, you can have scalability or High Availability or both. You have complete flexibility.



IBM Software Group

Single Application Server Environment

- No built-in high availability or scalability
- To provide availability
  - Restart server locally using monitoring service
- Use some HA framework (such as HACMP™) to provide local or remote restart

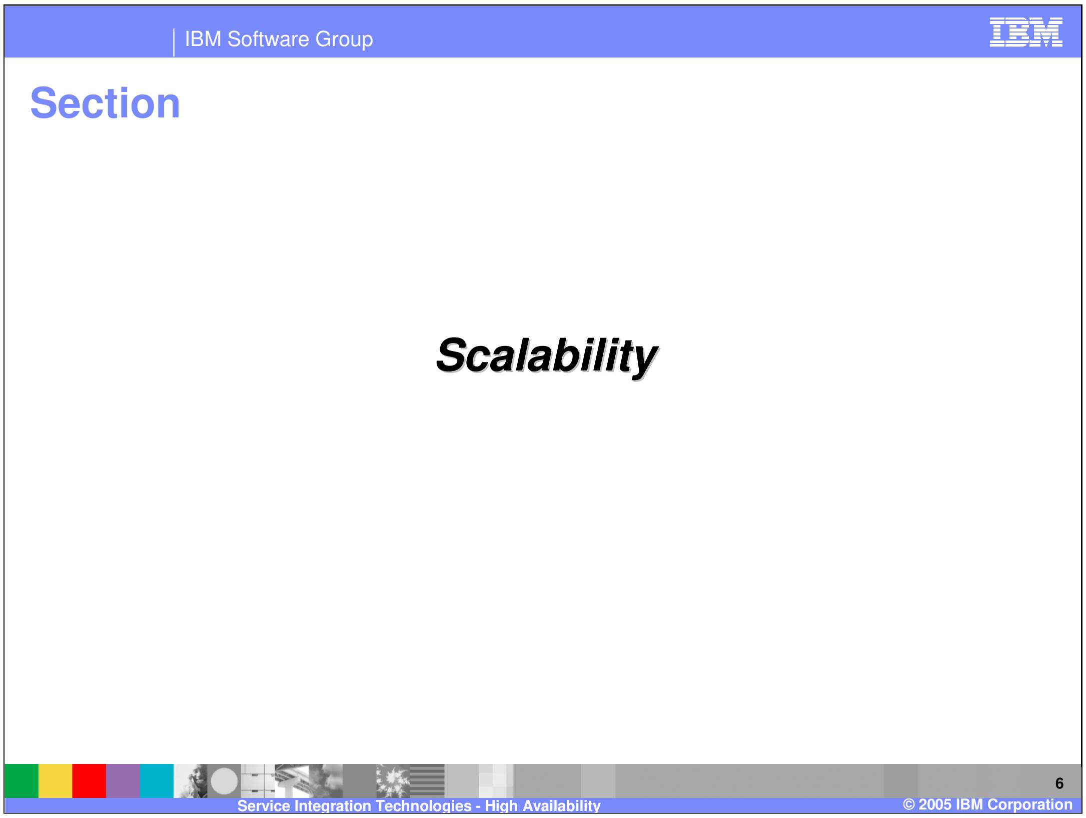
Service Integration Bus
v6 Server
ME
An ME that is managing a queue
Bus Destination

5

Service Integration Technologies - High Availability

© 2005 IBM Corporation

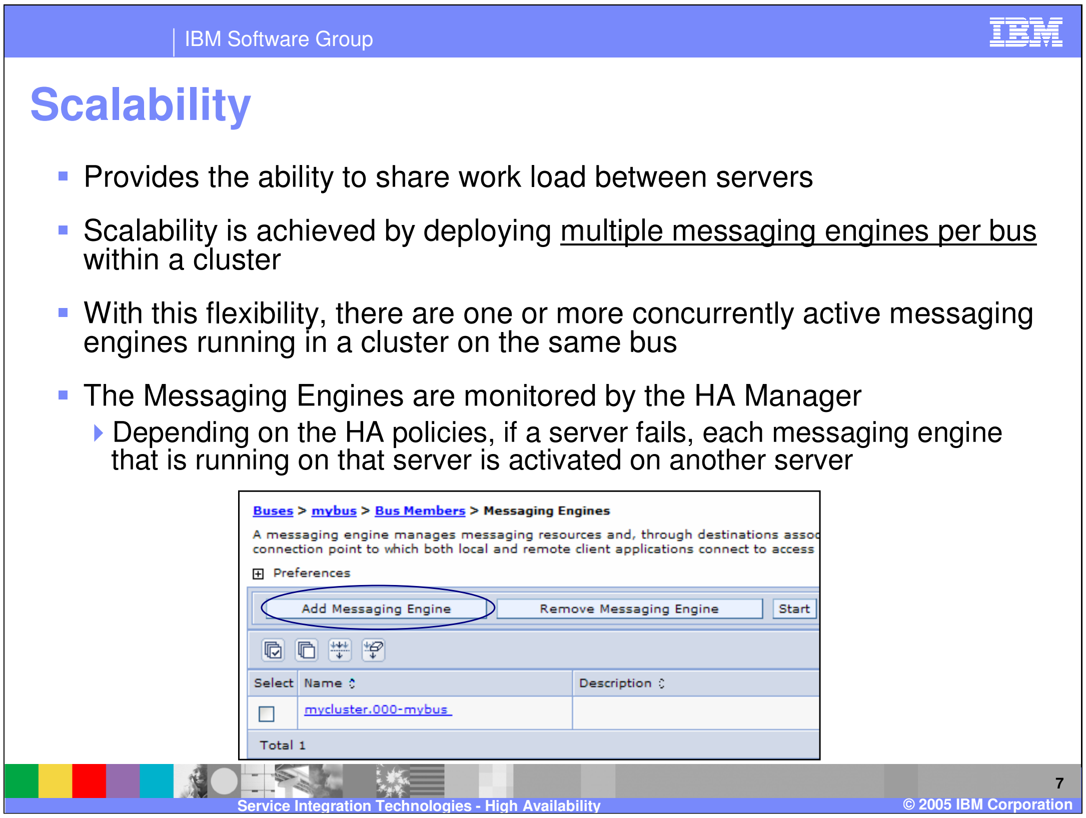
For a Single Application Server (Express or Network Deployment package), there is no clustering capability, since there is only one Application Server in that environment. If you need any HA capability, you will need external software like HACMP.



Section

Scalability

Next few foils will discuss the Scalability of how to achieve higher throughput



Scalability

- Provides the ability to share work load between servers
- Scalability is achieved by deploying multiple messaging engines per bus within a cluster
- With this flexibility, there are one or more concurrently active messaging engines running in a cluster on the same bus
- The Messaging Engines are monitored by the HA Manager
  - Depending on the HA policies, if a server fails, each messaging engine that is running on that server is activated on another server

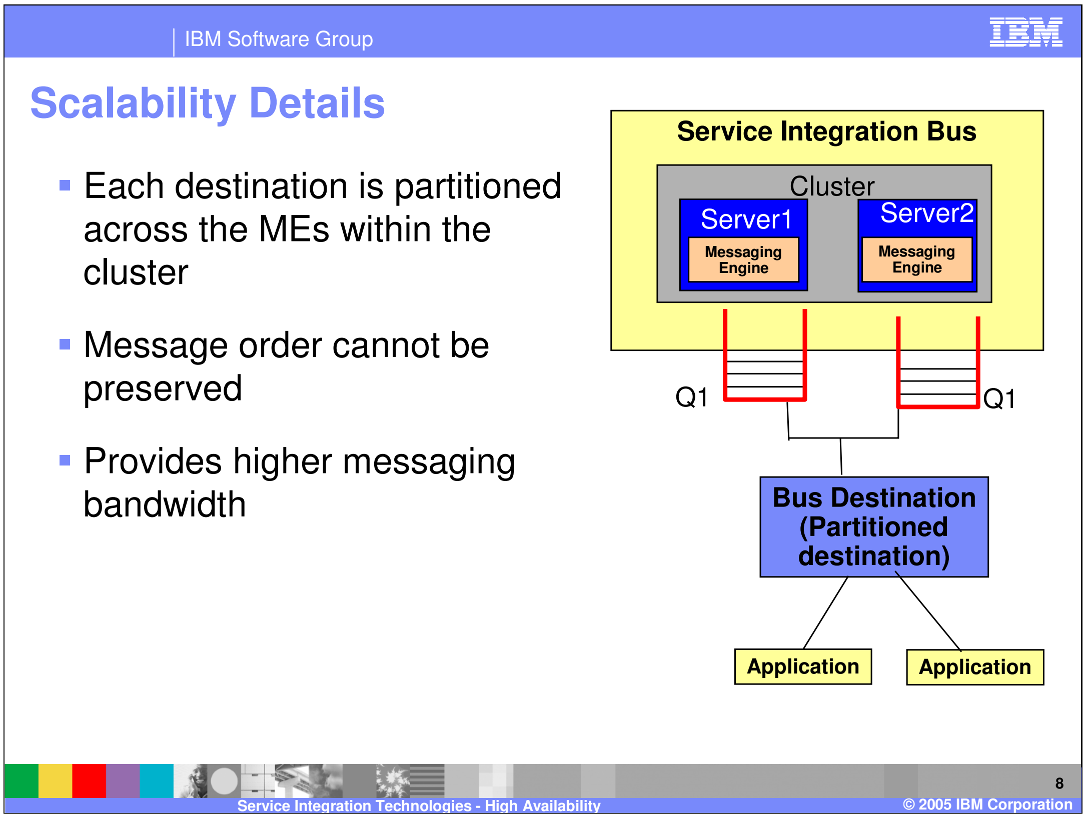
Buses > mybus > Bus Members > **Messaging Engines**

A messaging engine manages messaging resources and, through destinations asso
connection point to which both local and remote client applications connect to access

⊞ Preferences

Select	Name ◊	Description ◊
□	mycluster.000-mybus	
Total 1		

Adding additional Messaging Engines in the Cluster will provide improved scalability. However, the destination will be partitioned, meaning a message can be handled by any Messaging Engine in the cluster. In addition, based on the HA policy, the Messaging Engine could be floating between the cluster members of the bus. The HA policy defines the fail over of the messaging engines within the cluster.



IBM Software Group

Scalability Details

- Each destination is partitioned across the MEs within the cluster
- Message order cannot be preserved
- Provides higher messaging bandwidth

8

Service Integration Technologies - High Availability

© 2005 IBM Corporation

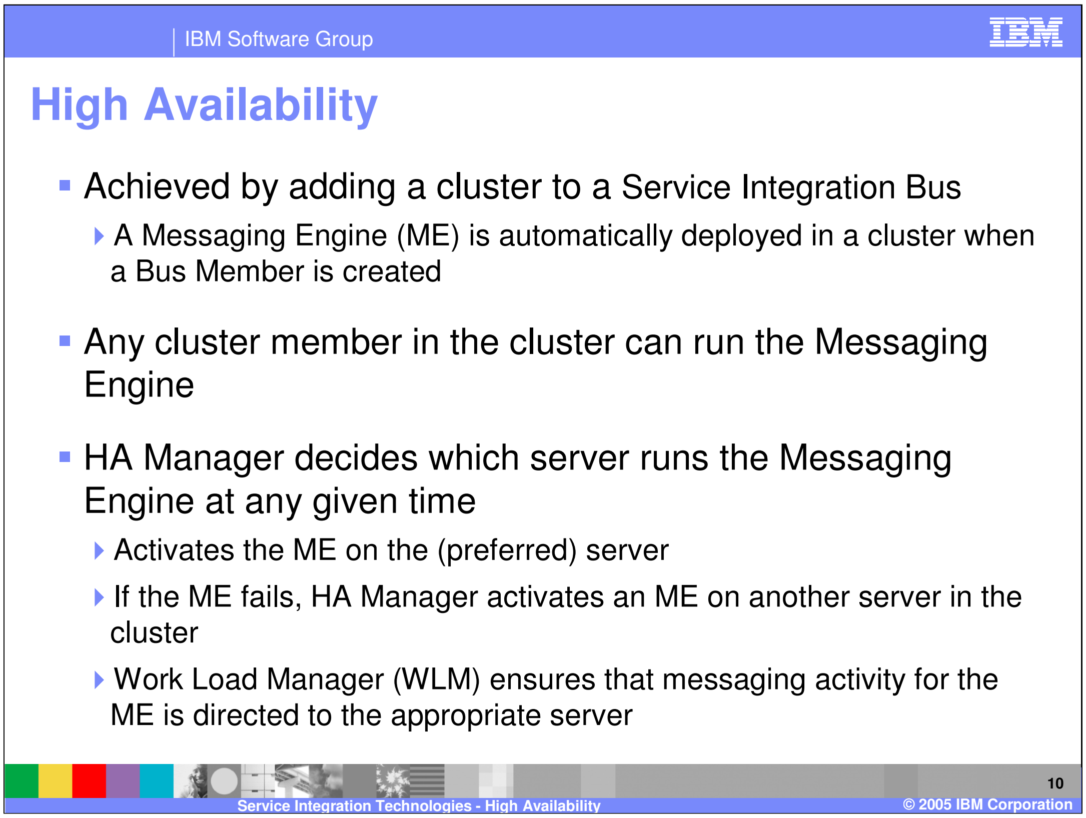
This case provides scalability. The destination has been localized to a Cluster Messaging Engine. It is therefore partitioned across the Messaging Engines within the cluster. With a partitioned Destination, the recoverable objects associated with the destination are split between separate Data Stores. This configuration has the disadvantage that message order cannot be preserved, but has advantages. One is that multiple consumers (or producers) can be deployed across the same Cluster to provide high messaging bandwidth. Messaging operations would always be locally fulfilled.

Scalability can be increased by adding cluster members to run additional messaging engines.

Section

High Availability

This section will discuss *failover and availability for* Messaging Engines in a cluster.



IBM Software Group

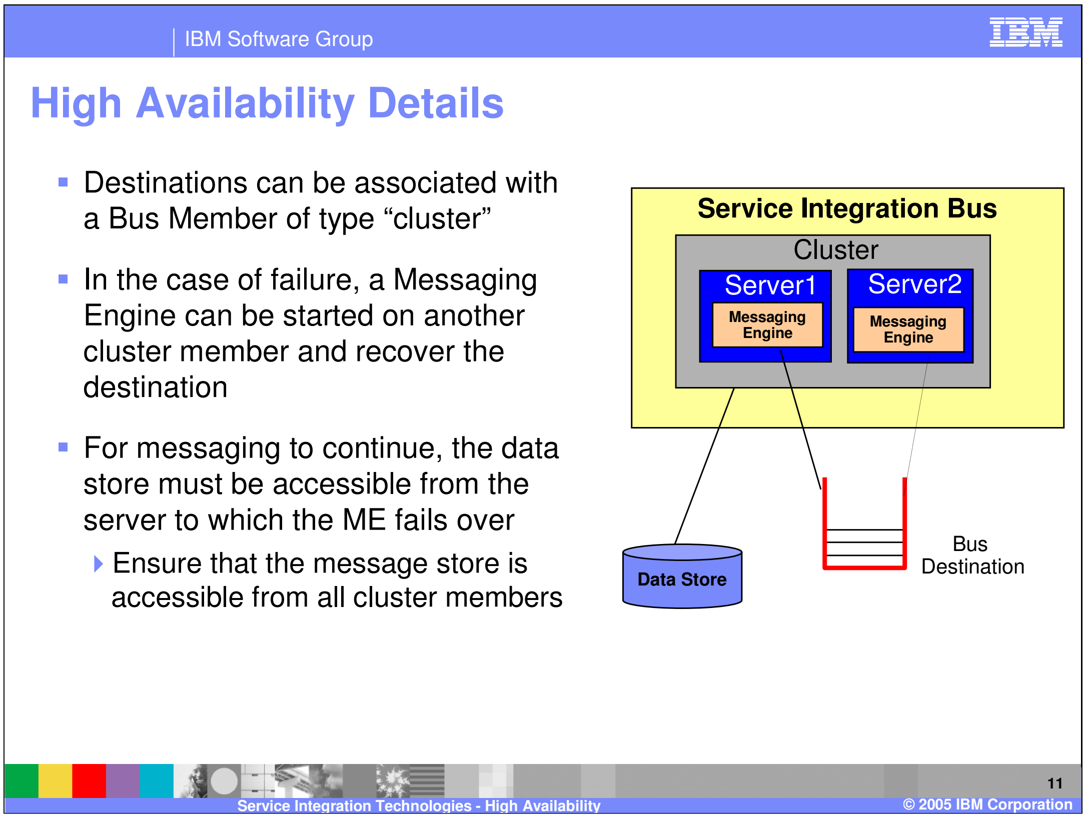
High Availability

- Achieved by adding a cluster to a Service Integration Bus
  - A Messaging Engine (ME) is automatically deployed in a cluster when a Bus Member is created
- Any cluster member in the cluster can run the Messaging Engine
- HA Manager decides which server runs the Messaging Engine at any given time
  - Activates the ME on the (preferred) server
  - If the ME fails, HA Manager activates an ME on another server in the cluster
  - Work Load Manager (WLM) ensures that messaging activity for the ME is directed to the appropriate server

10

Service Integration Technologies - High Availability

By default, a messaging engine within a cluster is managed by the High Availability manager, meaning that it can be restarted on any server in the cluster, should the server that is currently running the messaging engine fail. The configurable HA policy gives you the ability to control the failover behavior, or specify which server should run the messaging engine.



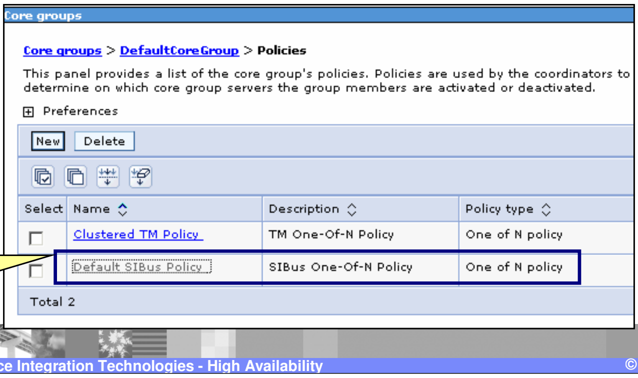
High Availability Details

- Destinations can be associated with a Bus Member of type “cluster”
- In the case of failure, a Messaging Engine can be started on another cluster member and recover the destination
- For messaging to continue, the data store must be accessible from the server to which the ME fails over
  - Ensure that the message store is accessible from all cluster members

The requirement for the fail over to work is that the data store needs to be accessible by all the cluster members. So, if a Messaging Engine over on one cluster member, another cluster member can host the Messaging Engine using the same Data Store.

HA settings for Messaging

- The HA policies determine the high availability characteristics of a Messaging Engine
  - Can be set using Administrative Console or wsadmin
- By default, all Messaging Engines have the default Service Integration Bus policy (1 of N)
  - Set the HA policy for messaging on the Core Group
  - All clusters belong to the “DefaultCoreGroup” by default



Core groups

Core groups > DefaultCoreGroup > Policies

This panel provides a list of the core group's policies. Policies are used by the coordinators to determine on which core group servers the group members are activated or deactivated.

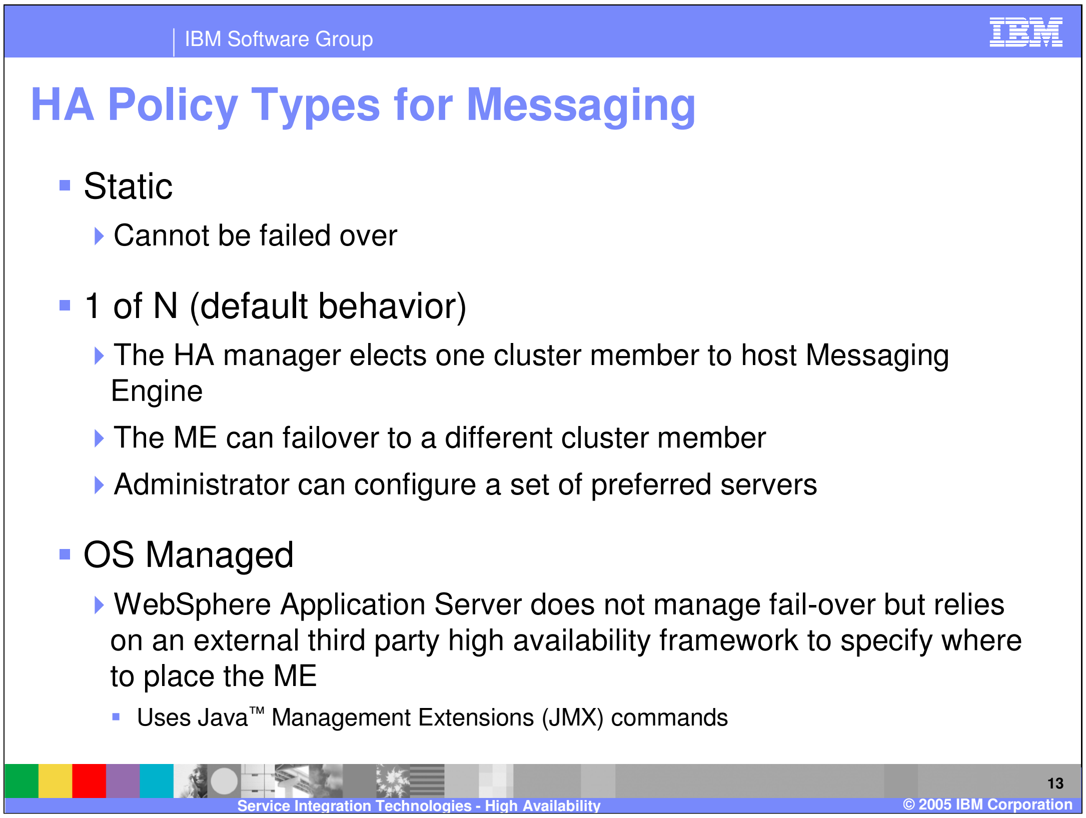
⊞ Preferences

Select	Name	Description	Policy type
□	Clustered TM Policy	TM One-Of-N Policy	One of N policy
□	Default SIBus Policy	SIBus One-Of-N Policy	One of N policy

Total 2

Default Policy created for Service Integration Bus

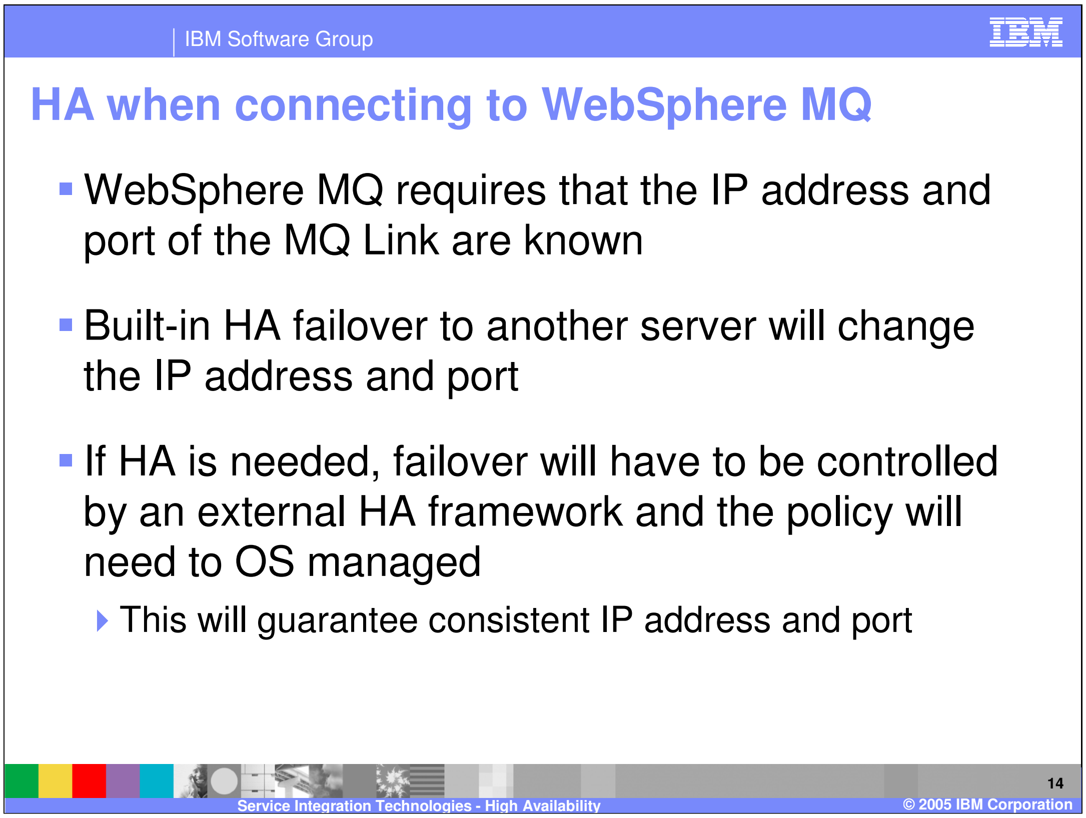
The HA policies for Service Integration Bus is automatically created for you, as “Default SIBus Policy”, with a default policy type of “1 of N”.



HA Policy Types for Messaging

- Static
  - Cannot be failed over
- 1 of N (default behavior)
  - The HA manager elects one cluster member to host Messaging Engine
  - The ME can failover to a different cluster member
  - Administrator can configure a set of preferred servers
- OS Managed
  - WebSphere Application Server does not manage fail-over but relies on an external third party high availability framework to specify where to place the ME
    - Uses Java™ Management Extensions (JMX) commands

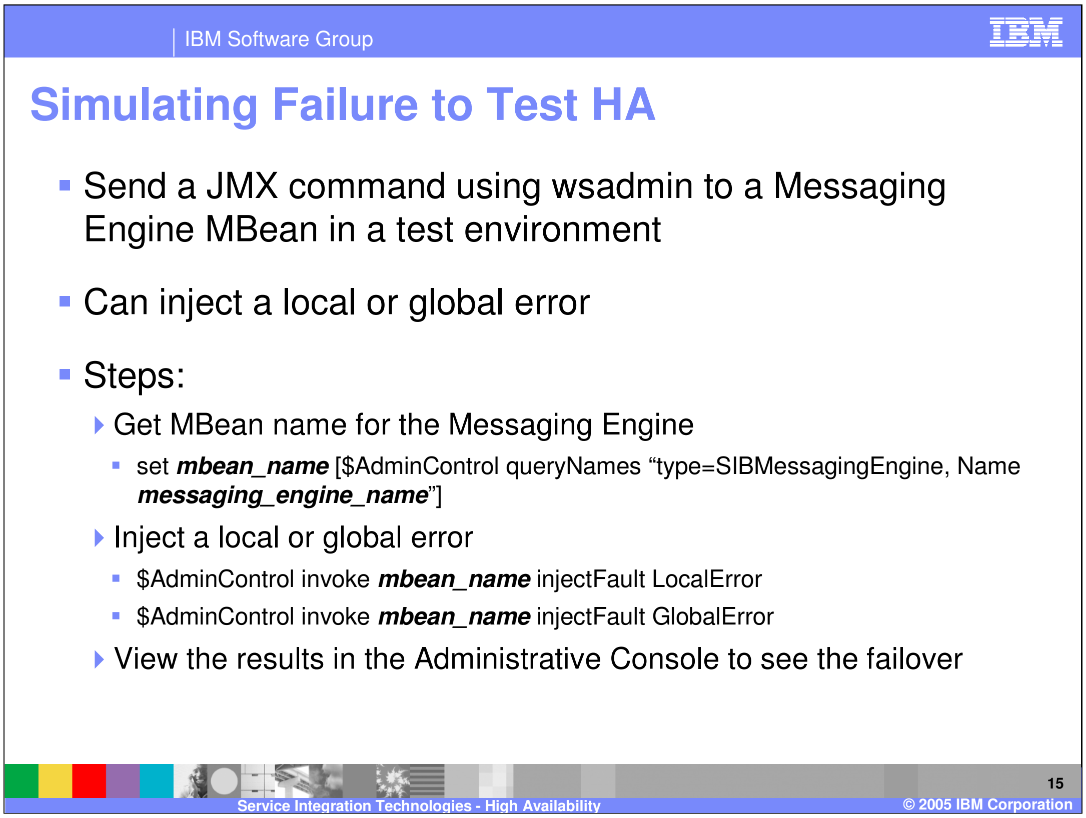
Different HA policy types are defined here. Static policy indicates that you do not need any failover. The default “1 of N” indicates that 1 elected member becomes the host for the messaging engine. The High Availability Manager determines the elected cluster member based on policies you have defined. Finally, OS-Managed is where some 3rd party software decides where to failover the messaging engine. They would use normal WebSphere JMX (wsadmin) commands to tell WebSphere to fail over the ME. These are the same policies that are available for any other highly available service in WebSphere Application Server.



HA when connecting to WebSphere MQ

- WebSphere MQ requires that the IP address and port of the MQ Link are known
- Built-in HA failover to another server will change the IP address and port
- If HA is needed, failover will have to be controlled by an external HA framework and the policy will need to OS managed
  - This will guarantee consistent IP address and port

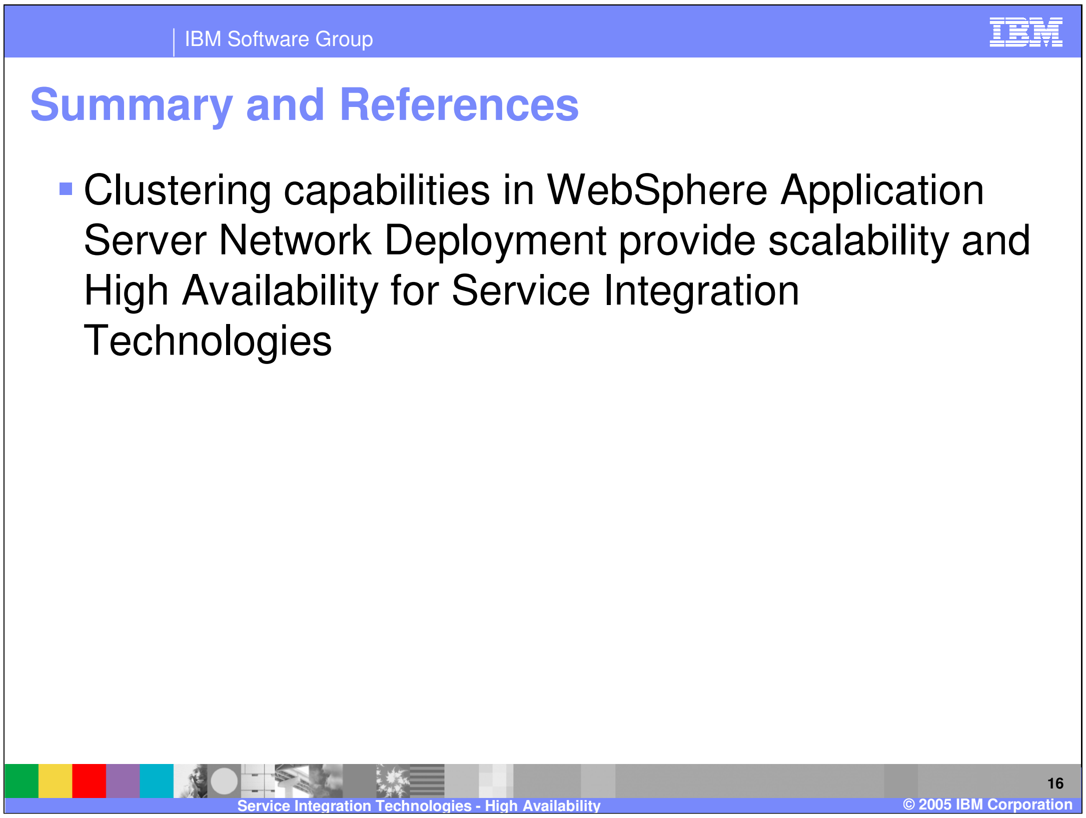
If you are connecting to an external WebSphere MQ queue manager, there are some considerations that limit HA functions. The limitation is due to the fact that WebSphere MQ requires that the IP address and port of the MQ Link be known. Moving a messaging engine from one cluster member to another will change the IP address. Hence the built-in HA will not work here. You will need external solutions like HACMP.



Simulating Failure to Test HA

- Send a JMX command using wsadmin to a Messaging Engine MBean in a test environment
- Can inject a local or global error
- Steps:
  - Get MBean name for the Messaging Engine
    - set ***mbean_name*** [$AdminControl queryNames “type=SIBMessagingEngine, Name ***messaging_engine_name***”]
  - Inject a local or global error
    - $AdminControl invoke ***mbean_name*** injectFault LocalError
    - $AdminControl invoke ***mbean_name*** injectFault GlobalError
  - View the results in the Administrative Console to see the failover

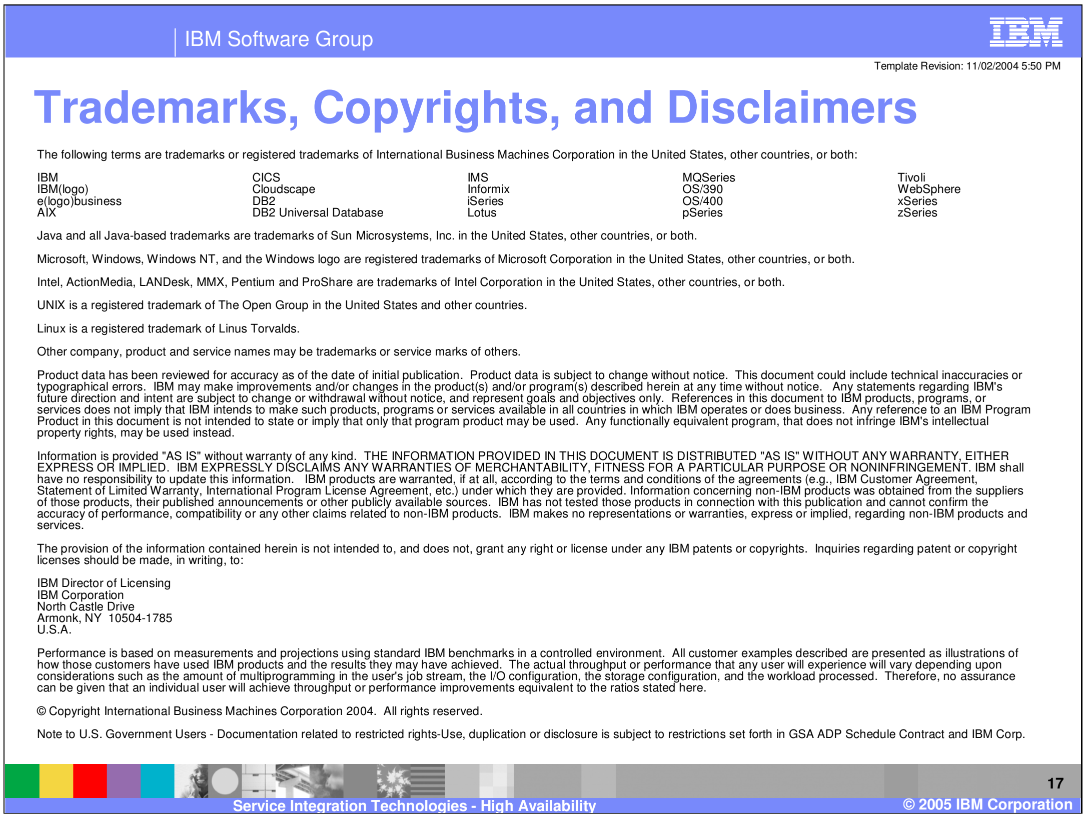
Wsadmin commands exist to inject an error on the messaging engine to simulate failure to test HA capabilities. The commands are shown on this page. Once you inject the failure, you can see the failover in the Administrative Console or in the System Log file for the Application server.



Summary and References

- Clustering capabilities in WebSphere Application Server Network Deployment provide scalability and High Availability for Service Integration Technologies

In summary, this presentation has focused on showcasing the new clustering capabilities of the Service Integration Bus in a V6 Cell environment.



Template Revision: 11/02/2004 5:50 PM

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
IBM(logo)
e(logo)business
AIX
CICS
Cloudscape
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
MQSeries
OS/390
OS/400
pSeries
Tivoli
WebSphere
xSeries
zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel 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.

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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.

Information is provided "AS IS" without warranty of any kind. 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 (e.g., 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.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.