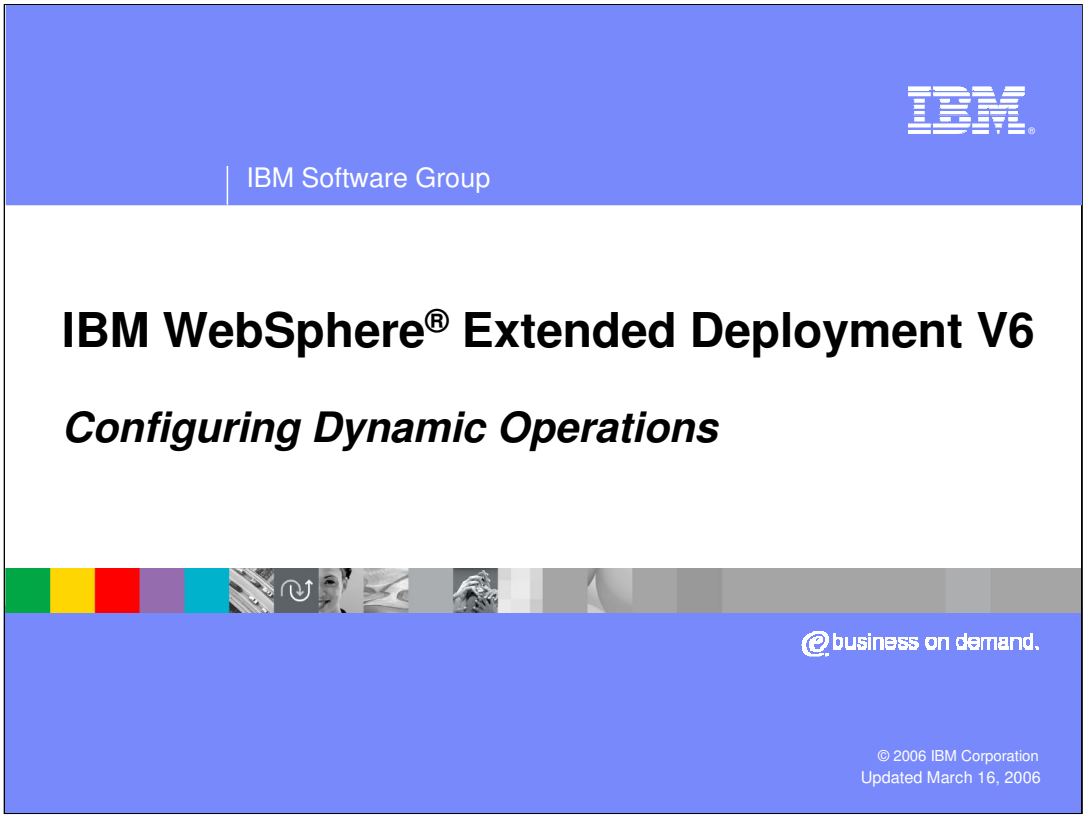


This presentation will cover configuring the dynamic operations features of WebSphere Extended Deployment.

Agenda

- Creating runtime resources
- Configuring operational policies

This presentation will first cover creating the runtime resources necessary for dynamic operations. It will then cover configuring operational policies, which define work types and goal levels that are used to drive a dynamic operations environment.

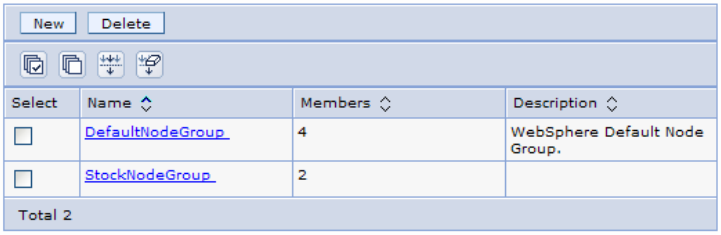
Section

Creating Runtime Resources

This section will cover creating runtime resources.

Configuring Node Groups

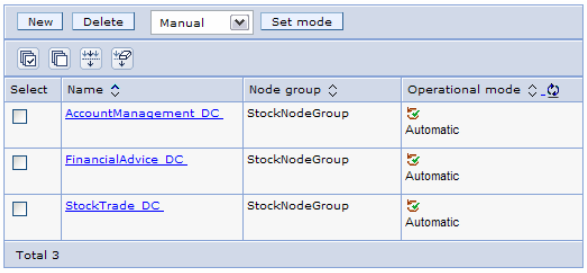
- DefaultNodeGroup is created by default during ND installation
  - New nodes are members of this group by default
  - No Dynamic Clusters can be created in this node group
- Node groups are created and modified under *System administration > Node groups*



Node Groups, the boundary within which Dynamic Clusters grow and shrink, are created under the 'system administration' menu item in the Administrative Console. The only property of a Node Group is a list of member nodes, which can be added after the group has been created. These nodes can overlap with other Node Groups. By default there is a single Node Group, called 'DefaultNodeGroup', that contains all nodes in your cell unless you specify otherwise. You cannot create Dynamic Clusters that are bound by this Node Group, so you must create a second Node Group before you can create any Dynamic Clusters.

Configuring Dynamic Clusters

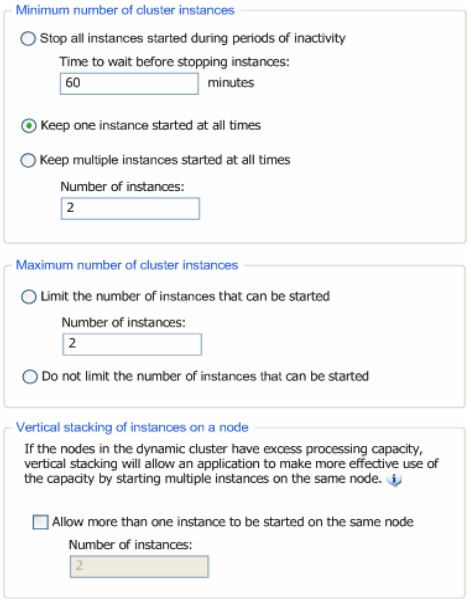
- Dynamic Clusters are created and modified under *Servers > Dynamic Clusters*
- The main panel enables creating or deleting clusters and setting operational modes
  - Manual, supervised, or automatic modes



Similar to static clusters, Dynamic Clusters are created based on an existing template. Unlike static clusters, however, you do not specify the nodes on which to create cluster members. Instead, you choose a bounding Node Group, and cluster members can be dynamically started and stopped on any node in that Node Group. To enable this dynamic behavior, put the Dynamic Cluster into automatic mode by selecting the check box for the cluster. Then select ‘Automatic’ from the pop-up menu, and click ‘Set mode’ on the screen shown here.

Modifying a Dynamic Cluster

- General options
  - Min/max running servers
  - Vertical stacking
- Server template link enables modifying the template for all cluster members
  - Looks like configuration page for a single server

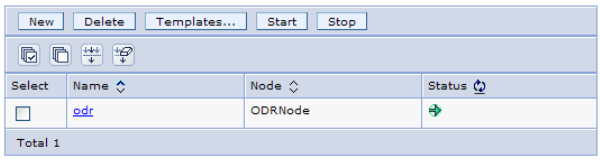


The main configuration panel for a Dynamic Cluster gives you control over the minimum and maximum number of instances of the Dynamic Cluster that can be simultaneously active. New in this release, you can specify that no instances of a given Dynamic Cluster should be running after a defined period of inactivity. This can be useful for rarely used applications for which users are willing to wait for application startup. 'Vertical stacking' is also a new option, allowing multiple instances of the same Dynamic Cluster to run on the same node, to take full advantage of nodes with extra processing capacity.

Clicking on the 'server template' link for a Dynamic Cluster displays a page that looks very similar to the configuration page for a single application server. This page, however, controls the properties of every member of the Dynamic Cluster. This differs from static cluster members, which must be configured individually after they have been created.

Configuring On Demand Routers

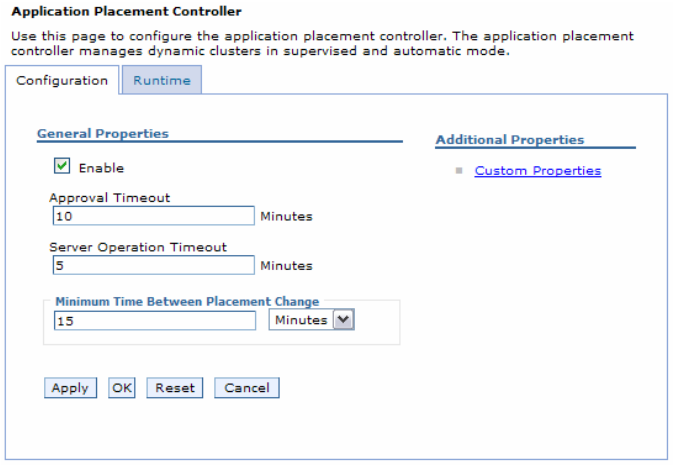
- Create, modify, and manage ODRs under *Servers > On Demand Routers*
- Click on the name of the ODR to configure it just like a server
  - Manage server attributes like ports, thread pools, cache rules here
  - Service policies and routing policies are not configured here



On Demand Routers can be created and configured in the Administrative Console under the ‘Servers’ menu item. ODRs can be placed on any managed node; no special installation is required. While properties of the ODR itself, such as thread pool sizes or cache rules, can be configured in this panel, service policies, work classes, and routing policies are not configured here. Service policies are created under the ‘operational policies’ menu item, while work classes and routing policies are attributes of individual applications. A wsadmin script, ‘createODR.jacl’ is also provided for creating an On-Demand Router from the command line.

Configure Application Placement Controller

- *Operational Policies > Autonomic Managers > Application Placement Controller*



The Application Placement Controller has its own configuration page under the 'operational policies' menu, where you can modify its behavior.

The 'approval timeout' defines how long notifications live before expiration while running in supervised mode.

The 'server operation timeout' is the amount of time that the placement controller should wait for a server to start or stop before considering the action a failure.

'Minimum time between placement change' is the value that defines how long to wait after performing a set of placement actions until performing the next set of changes.

Section

Configuring operational policies

This section will cover configuring operational policies.

Service policies

- Service policies define a level of importance and a response time goal
- Each service policy contains one or more transaction classes
  - Transaction classes enable work classes to be mapped to the service policy
  - Also enable finer-grained monitoring than just by application, server, or work class
  - A default transaction class is created for each service policy
    - This is sufficient unless you need finer-grained monitoring or reporting

A service policy defines a level of importance and a response time goal, which allow you to describe how requests into your environment should be treated. In addition to defining goals, each service policy contains one or more transaction classes. As you will see later, a work class is a set of rules used for mapping incoming requests to transaction classes. Therefore incoming requests that match a particular rule are associated with a transaction class. This in turn is contained by a service policy that defines the goals and importance for that request. Each service policy contains a single transaction class by default. However, you can create multiple transaction classes within a service policy. This enables you to differentiate (for monitoring or reporting purposes) between requests that you want to be handled with the same class of service.

Creating a service policy

- Create service policies in the Administrative Console under *Operational Policies > Service Policies*

Service policies are created in the Administrative Console under the ‘operational policies’ menu. Pause this presentation and click the ‘show me’ icon to view a demonstration of creating a service policy.

Work classes

- Work classes define a certain type of work and associate it with a transaction class
- Each work class contains a default policy and an optional list of rules
  - Requests matching the work class rules will be classified into the service policy that contains that transaction class
  - If no rules match, work will be classified to a default transaction class

A work class is a set of rules that allow you to differentiate between incoming types of work. Each rule created within a work class has an associated transaction class. Work that matches a rule will be treated according to the service policy that contains that transaction class. Each work class also has a default transaction class, which defines how work that does not match any rules will be treated.

Request classification

- Requests can be classified by a number of variables:
  - Virtual host or Uniform Resource Identifier (URI)
  - HTTP headers, query parameters, and cookies
  - Web service and operation name
  - Client or server IP address, port, and host names
  - Time
  - User or group ID (forwarded by WebSEAL)

Work class rules can match requests by a combination of several variables, including virtual host or URI, client or server IP address, time of day, HTTP headers, query parameters, and cookies. You can also classify requests based on the user that made the request if WebSEAL is used in your environment, because WebSEAL will forward user and group data to the On Demand Router in the HTTP header.

Classification operands

Operand	Protocols	Description
clientHost, serverHost, clientIPv4, serverIPv4, clientIPv6, serverIPv6	HTTP,SOAP	Client and server host names and IP addresses
Port	HTTP,SOAP	Server listening port
UID, GID	HTTP,SOAP	WebSEAL authentication identities
protocol	HTTP,SOAP	The request protocol
header$*name*, *queryParm$name*, cookie$*name*	HTTP,SOAP	HTTP header, query parameters, and cookies
HTTPMethod	HTTP,SOAP	HTTP method
MIMEType	HTTP,SOAP	MIME type
service	SOAP	Web service name
operation	SOAP	Web service operation
virtualportal	**HTTP**	**Virtual Portal name**

The table shown here lists all of the operands that are available to classify HTTP and SOAP requests. These operands are used in the strings that define classification rules.

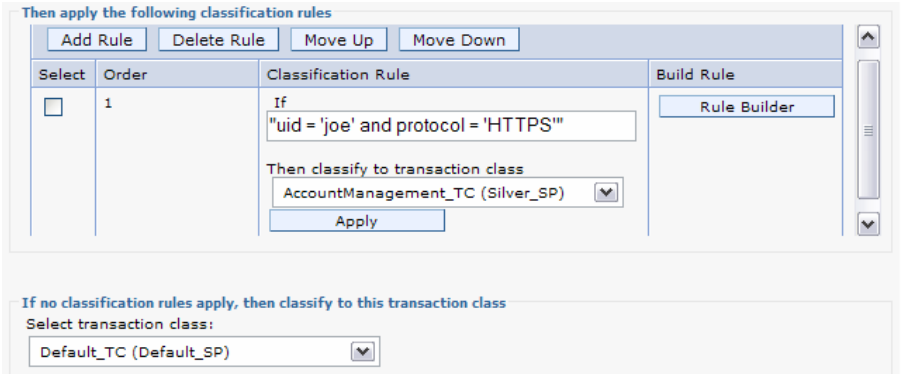
Classification operands

Operand	Protocols	Description
application	IIOP	Application name
ejbmodule	IIOP	EJB Module name
ejbname	IIOP	EJB name
ejbmethod	IIOP	EJB method name
clienthost, serverhost, clientport, port	IIOP	Client and server hostnames and ports

The table shown here lists all of the operands that are available to classify IIOP requests.

Creating work classes

- Work classes are defined under the *Service Policies* tab of each application



Work classes are created as attributes of individual applications. To create a work class, choose an installed application from the 'enterprise applications' menu item, and then click the 'service policies' tab from the application's configuration page. Here you can create rules like the one shown here. Rules can define Boolean combinations of classification operands, and each rule specifies a transaction class to which the request will be classified if it matches. If you don't know the exact syntax of the rule you want to create, the 'rule builder' button will let you create rules using a menu-driven wizard.

Using the Rule Builder

- The Rule Builder provides a set of menus to help you create Routing Policies and learn the correct syntax

To see a demonstration of the rule builder in use, pause this presentation, and click the ‘show me’ icon.

Summary

- Configuring dynamic operations requires both creating runtime resources and configuring operational policies
  - Runtime resources include Node Groups, Dynamic Clusters, and On-Demand Routers
  - Operational policies require service policies, transaction classes, and work classes
    - Work matching a rule in a work class is classified to a transaction class, which associates it with a service policy

In summary, configuring dynamic operations requires creating runtime resources, such as Node Groups, Dynamic Clusters, and On-Demand Routers, and creating operational policies. Service policies define classes of importance and response time goals, and work is mapped to those policies by work class rules that associate the work with transaction classes.

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