



IBM Software Group

WebSphere® Extended Deployment V6.0.1

Overview of New Features



@business on demand.

January 18, 2006

© 2006 IBM Corporation
Updated April 3, 2006

This presentation covers new and updated features in WebSphere Extended Deployment V6.0.1.

Agenda

- New Platform Support
- Enhancements
 - ▶ Dynamic Operations
 - ▶ Visualization
 - ▶ ObjectGrid
 - ▶ Application Edition Manager
 - ▶ Business Grid

Platforms supported by WebSphere Extended Deployment version 6.0.1 will be discussed first, followed by an overview of changes to the major functional areas.

New Platform Support

This section covers additional platforms supported by WebSphere XD version 6.0.1.

Now with z/OS Support

- Provides family consistency
 - ▶ Application portability
 - ▶ Skills portability
 - ▶ Commitment to z/OS® platform
- Delivers important enterprise-class features to the z/OS® platform
 - ▶ Batch/CI/Partitioned programming models
 - ▶ Application versioning
 - ▶ Repository checkpoint/restore
 - ▶ Dynamic visualization
- Provides improved edge-of-sysplex workload distribution
 - ▶ On-demand Router makes better-informed decision for distribution across multiple LPARs than existing mechanisms

WebSphere Extended Deployment V6.0.1 adds support for z/OS, providing enhanced portability for applications between z/OS and distributed platforms, and allowing you to maximize your existing WebSphere skills. WebSphere XD for z/OS adds advanced administration capabilities and programming functionality to that platform and complements and enhances existing z/OS features such as Sysplex Distributor and zWLM.

Stack Product Support

- XD 5.1 provided limited support for WebSphere Business Integration-Server Foundation V5.1.1
 - ▶ Did not provide any enhancements to WebSphere Business Integration-Server Foundation
 - ▶ Allow XD customers to use WebSphere Business Integration-Server Foundation Programming Model Extensions
- XD 6.0.0 does not support any of the WebSphere Stack Products
- XD Version 6.0.1 (distributed) adds support for the following WebSphere products:
 - ▶ WebSphere Process Server 6.0.1 or higher
 - ▶ WebSphere Portal Server 5.1.0.2 or higher running on ND 6.0.2.x
 - ▶ WebSphere Commerce Server 5.6.1.1 or higher
- Up to date prerequisite information:
- <http://www.ibm.com/software/webservers/appserv/extend/requirements>

WebSphere XD V5.1 provided limited support for WebSphere Business Integration Server Foundation programming model extensions, such as asynchronous beans. When WebSphere XD V6.0 was announced, WebSphere Business Integration – Server Foundation did not yet support WebSphere Application Server V6.0.2, which was required for that version of WebSphere XD. Version 6.0.1 of WebSphere XD adds full support for WebSphere Process Server, WebSphere Portal Server, and WebSphere Commerce Server. For up to date platform support and prerequisite information, including version numbers, consult the official prerequisites Web site at <http://www.ibm.com/software/webservers/appserv/extend/requirements>

Section

Dynamic Operations

This section describes changes to WebSphere XD Dynamic Operations.

Dynamic Operations Overview

- Virtualized, policy-based, dynamic workload management
- Dynamic application placement
 - ▶ Enables starting and stopping server instances based on application load and user-defined goals
- On-Demand Router
 - ▶ Enhanced version of the Proxy Server
 - ▶ Controls request prioritization, flow, and routing in an Extended Deployment (XD) environment

The dynamic operations features of WebSphere XD provide the capability to build a dynamic, virtualized, goal-oriented environment for workload management. The two major features that enable these capabilities are dynamic application placement and the On-Demand Router. Dynamic application placement enables starting and stopping of additional server instances to accommodate changes in load, balancing processing power among your applications to best meet your defined performance goals. The On-Demand Router is an intelligent HTTP proxy server that manages request prioritization, flow control and dynamic routing of requests to your application servers.

Section

Dynamic Application Placement

This section describes changes to Dynamic Application Placement.

Dynamic Application Placement

- Autonomic placement of instances of applications on servers
- Starts/stops preconfigured application server instances to meet defined performance goals

Dynamic application placement uses dynamic clusters to increase or decrease the number of servers on which a particular application is running to help meet your performance goals. Each node within a dynamic cluster has an instance of an application server defined to run that cluster's applications. These server instances can be started and stopped dynamically as traffic for that application increases or decreases.

Dynamic Application Placement Enhancements

- Dynamic Workload Management integrated with Session Rebalancer to allow rebalancing based on response time differences in nodes
- Placement manager enhanced to include memory utilization information for non-WebSphere processes on a node

XD 6.0.1 includes the capability to monitor sessions in the cluster members and direct the XD Session Rebalancer in the cluster member to move certain Sessions to other cluster members within the same cluster with the objective of distributing the Session in the same proportion as the calculated weights for the cluster members. This ensures that requests will be balanced more proactively according to the calculated weights than they would be if Session Rebalancing were not done. You can turn off Session Rebalancing if that is deemed undesirable for your particular configurations.

It is important to note that Session Rebalancing will only take place if the cluster members (servers) are configured to use distributed sessions (with either database or in-memory replication) and sessions are tracked using cookies. If sessions are maintained using URL rewriting or Secure Sockets Layer (SSL) ids, no rebalancing is performed.

XD needs to know how much free memory is available on a node so it can keep all server instances in “real” memory. PMI data can indicate how much memory is being used by WebSphere, but not how much is being used by non-WebSphere processes. XD V6.0.1 includes several new custom properties that allow you to specify how much of the system

HTTP Session Rebalancing

- XD 6.0
 - ▶ Rebalancing occurs when servers are started or stopped
 - ▶ Reassign sessions so that each server has an equal proportion of all the existing sessions
 - ▶ The session rebalancing function runs in each application server
- XD 6.0.1 and later
 - ▶ Session moves can be initiated by any events that DWLM monitors, such as new servers starting, overloaded servers, and so on
 - ▶ Reassign sessions so that the number of sessions that are running in each server in the dynamic cluster is proportional to the assigned weight of the servers
 - ▶ The dynamic workload manager (DWLM) performs the session rebalancing function

WebSphere XD Version 6.0 HTTP session rebalancing can only occur when a new server instance is started or an existing instance is stopped. This could lead to a situation where a server instance remains overloaded due to session affinities. HTTP session rebalancing can occur on any event that the dynamic workload manager monitors, such as server weight changes. To facilitate this change, the session rebalancer component has moved from the individual application servers to the dynamic workload manager.

Sessions are spread equally across all servers, Live Sessions are monitored within cluster members and attempts are made to distribute sessions in the same proportion as the calculated weights for the cluster members. You can turn off Session Rebalancing if it is deemed undesirable for your particular configurations.

HTTP Session Rebalancing Scenario

1. In the dynamic cluster, a new server becomes available or an existing server becomes overloaded
2. The dynamic workload manager (DWLM) calculates the number of sessions to move
 - If a new server is available, DWLM can assign sessions to the new server from existing servers
 - If an existing server becomes overloaded, DWLM can move sessions from the overloaded server to other servers in the dynamic cluster
3. Meanwhile, the client generates a request for an existing session, sending the cookie that is established in the client browser
4. The cookie contains a session ID and a server ID, and is assigned to an existing application server that has an established session affinity
5. If DWLM determines that the established session needs to move a new or existing application server, the servlet filter sends a redirect back to the client that contains the new server ID
 - If the session has not moved, the servlet filters forward the request to the server that is identified by the server ID

If a new server becomes available in a dynamic cluster or an existing server becomes overloaded, the dynamic workload manager calculates new weights for all the servers in the cluster and determines how many sessions should move. If a client request is received while the move is taking place, the servlet filter sends a redirect back to the client with the new server ID.

Application Placement Custom Properties

- **cpuUtilizationThreshold**
 - ▶ define the percentage of CPU power that you can use on each node in a cell
- **GenerateUniquePorts**
 - ▶ generate unique ports for each dynamic cluster member
- **reservedMemoryFixed and reservedMemoryPercent**
 - ▶ define how much memory on each node to reserve for processes that are not related to WebSphere Application Server or WebSphere Extended Deployment

WebSphere XD Version 6.0.1 adds several new custom properties to control application placement.

CPU utilization threshold defines the percentage of CPU resource that can be used on each node in a cell. The default value is 100, meaning that 100% of the CPU can be used on each node in the cell. The application placement controller attempts to compute placements so that the utilization of any node does not exceed the defined threshold. The application placement controller considers all processes, including processes that are not related to WebSphere Extended Deployment when making these calculations.

For configurations that include z/OS nodes, this property must be defined and must be set to a value less than 100.

The Generate Unique Ports custom property defines how port numbers are generated for each dynamic cluster member. By default, dynamic cluster members on distributed nodes

Section

On Demand Router

This section describes changes to the On Demand Router.

On Demand Router

- The On Demand Router (ODR) is an intelligent HTTP proxy server
 - ▶ Enhanced version of the Proxy Server from ND 6.0.2
 - ▶ Can replace or complement the HTTP server plug-in
 - ▶ Prioritizes requests and controls traffic flow according to operational policy
 - Ensures consistent quality of service
 - Enables more elegant degradation of performance when all resources are consumed
 - ▶ Integrates with application placement to route requests to dynamic cluster members

The On-Demand Router is an intelligent HTTP proxy server provided with WebSphere XD that serves as the point of entry into the WebSphere XD environment and is responsible for request prioritization, flow control, and distributing requests to application servers. These advanced features distinguish the ODR from both the HTTP server plug-in and the Proxy Server provided with WebSphere Application Server Network Deployment, and give the ODR the ability to ensure a more consistent quality of service for your enterprise applications. It can be used in place of, or in concert with the HTTP server plug-in, depending on your needs. While it is most often associated with WebSphere XD dynamic operations features, other features, such as interruption free application upgrades and Deployment Manager high availability also rely on the On-Demand Router.

ODR Enhancements

- Support for cross-cell routing and load balancing, including multiple data center routing
 - ▶ Backup and weighted cluster support (failure or load based routing to peer cluster)
- Support for weighted least outstanding requests (WLOR) load balancing
 - ▶ Provides much improved distribution of workload and error recovery
- Improve Debug support for XD dynamic configuration system (ODC)
 - ▶ TargetTreeMBean
 - ▶ RoutingDebugMBean

WebSphere XD Version 6.0.1 supports two types of multi-cluster routing: failover and load balancing.

Multi-cluster failover routing allows one or more backup clusters to be associated with a particular cell, application, or Web module. If no servers in the primary cluster are currently running the Web module, the request can be routed to one of the backup clusters. Failover clusters are typically generic server clusters. For example, suppose application A is running in a New York cell as well as in a Tokyo cell. The New York cell could be configured to send requests to Tokyo if application A is not currently running on any servers in New York.

Multi-cluster load balance routing is similar to failover routing, except that requests can be routed to multiple clusters more often, and not just when one cluster is down. For example, requests for application A are load balanced to both New York and Tokyo.

Section

Request Classification

This section describes new support for request classification.

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
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. XD Version 6.0.1 adds support for classification on Virtual Portal name.

JMS and IIOp Traffic Shaping

- WebSphere XD V6.0.1 adds request classification and flow control for JMS and IIOp requests
 - ▶ Gives WebSphere XD the ability to give higher priority to more important requests, similar to the treatment of HTTP requests in previous releases
- Traffic is not routed through an ODR
 - ▶ Traffic is routed to individual servers by traditional means
 - ▶ Traffic is classified and controlled by ARFM instances running on individual application server processes
- This feature is currently not available on z/OS

WebSphere XD Version 6.0.1 adds request classification and flow control for JMS and IIOp requests, which provides the capability to assign a higher priority to more important requests. These requests are not routed through the ODR, but are instead routed to individual servers just as they would be without WebSphere XD. An Autonomic Request Flow Manager (ARFM) running within the application server matches each request to a transaction class, and therefore work class, just like the ODR does with HTTP requests. The in-process ARFM limits the concurrency of message processing based on available processing power. Individual requests will flow through faster or slower than others based on priority.

The supported version of WebSphere for z/OS, Version 6.02, does not support the necessary ORB SPIs to enable this support. This release of WebSphere XD for z/OS provides a comparable level of IIOp workload management through z/WLM service classes and balancing through the location service agent, which is a daemon process, and provides a comparable level of JMS workload management through z/WLM service

JMS and IIOP Request Classification

- Incoming JMS messages are classified based on destination (queue or topic) name
 - ▶ No other classification operands are supported
- IIOP requests can be classified based on several attributes, including EJB™ name and method name

JMS messages can be classified based on queue name or topic name.
IIOP type work classes use EJB and EJB method names to map to the goal.

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 available to classify IIOp requests.

Section

Visualization Features

This section describes changes to the WebSphere XD Visualization Features.

Runtime Topology Enhancements

- Visualization and Monitoring of Database nodes in runtime topology and charting, including JDBC service time and throughput aggregated at different levels
- Added Unmanaged/Non-WebSphere nodes to runtime topology
- Native SVG support (no plug-in) for Firefox 1.5



The WebSphere XD visualization components can now be configured to monitor database nodes to help detect performance bottlenecks in the database tier. In addition, WebSphere XD can now monitor unmanaged nodes and non-WebSphere nodes with the WebSphere XD Mixed Server Environment installed.

WebSphere XD 6.0.1 also adds support for the Firefox 1.5 browser with native support for Adobe SVG.

Charting Enhancements

- Improve charting resolution
- Chart size preferences (small, medium, large)
- Ability to show/hide chart goal lines
- Charting of data for non-WebSphere servers/tiers
- Charting support for
 - ▶ work factors
 - ▶ node capacity
 - ▶ background CPU
 - ▶ CPU consumption
 - ▶ usable node CPU capacity
 - ▶ total node memory
 - ▶ external node stats
 - ▶ tier stats
 - ▶ memory high-water marks

The WebSphere XD Runtime Topology charting capability has been enhanced in several areas, including improved resolution, control over chart size, and the ability to show or hide service and transaction class goal lines. Charting data is now available for non-WebSphere servers and database nodes; and many new metrics are available for charting, including those listed here.

Section

Task Management

This section describes changes to WebSphere XD Task Management.

Runtime Task Persistence

- V6.0
 - ▶ Runtime tasks are stored in memory only
 - ▶ Tasks are lost if the Deployment Manager is restarted
- V6.0.1
 - ▶ Tasks are serialized and stored on disk
 - Files are stored in <WORKSPACE_ROOT>/tmsStorage
 - ▶ Tasks are recovered after Deployment Manager restart

In WebSphere XD Version 6.0, runtime tasks were stored in memory on the Deployment Manager. If the Deployment Manager stops or fails, all tasks are lost. In version 6.0.1, tasks are serialized to disk, so if the deployment manager stops or fails the task list can be recovered.

Health Monitoring

- Added Service Policy Violation Health Policy to enable administrator notification when service policy is consistently violated above some threshold
- Health Management support for Grid nodes and Business Grid Applications

Service Policy definitions for Average Response Time and Percentile Response Time goals can now optionally be configured to monitor for persistent service policy violations.

When a policy is consistently violated above a configured goal delta value for a specified period of time, a notification is submitted to the Task Management Service. The violation notification includes a delta value and time period parameters. For Average Response Time service policies, the delta value is the amount above the configured goal. For Percentile Response Time service policies, the delta value is the amount that the actual percentage of requests served falls short by.

The WebSphere XD Health monitor has been enhanced to interact with the business grid controller to provide better monitoring for Business Grid nodes and applications.

Section

ObjectGrid

This section describes changes to the WebSphere XD ObjectGrid.

ObjectGrid Overview

- ObjectGrid is a high-performance, transactional cache framework for storing Java™ objects
 - ▶ Scalable from a local JVM cache to a distributed, partitioned cluster of 100 JVMs
- Can be backed by a hardened data store
- Customizable cache lifecycle features
 - ▶ Declaration, configuration, invalidation, size management, cache loading

ObjectGrid provides a highly scalable, high-performance, transactional cache framework for storing Java objects. An ObjectGrid can be used as a generic object cache, and can be optionally persisted to a hardened store. It can also be used as a local cache for objects stored in a database. ObjectGrid is a highly customizable feature, with interfaces provided for custom data loaders, invalidation and size management schemes, and more.

ObjectGrid Overview

- XD 6.0
 - ▶ supports a simple push-based model for change propagation
 - ▶ changes can be propagated to peer ObjectGrids using a message transport
- XD 6.0.1
 - ▶ Distributed ObjectGrid data is replicated from a primary server to one or more secondary servers to ensure availability
 - ▶ This cluster of replication servers is referred to as a replication group
 - ▶ Can include one or more standby servers
 - ▶ Peer-to-peer model still available

In WebSphere XD Version 6.0, ObjectGrid supported a simple peer-to-peer model for change propagation between ObjectGrid instances. In version 6.0.1, the ObjectGrid has been improved to be a fully distributed, highly available, secure caching solution, including remote client support to an ObjectGrid cluster of servers offering caching services. Additionally, ObjectGrid distributed capabilities can be used within WebSphere Application Server, or as a complete stand-alone approach outside the WebSphere environment.

Runtime Environment Support

- ObjectGrid caches are supported in WebSphere XD 6.0 (or greater) or WebSphere Application Server 6.0.2 (or greater) runtime environments
 - ▶ ObjectGrid libraries are provided only with WebSphere XD, not WebSphere Application Server
 - ▶ ObjectGrid can be installed without WebSphere XD using the “Mixed Server Environment” installation option on distributed platforms
- ObjectGrid can also be run ‘stand-alone’ in any J2SE 1.4.2 or higher JVM

While ObjectGrid technology is provided only with WebSphere Extended Deployment, its use is also supported in a WebSphere Application Server Network Deployment V6.0.2 server by including the ObjectGrid libraries (contained in wsubjectgrid.jar) in your application classpath. An ObjectGrid can also be run in a stand-alone JVM or other application server product by using the ObjectGrid.jar file.

Runtime Environment Support

- ObjectGrid Client
 - ▶ Where customer applications interact with ObjectGrid APIs
 - ▶ Has the routing logic to connect to ObjectGrid servers
 - ▶ Application can also create local objectmaps using the client jars
 - ▶ Supported on both z/OS and distributed platforms
- ObjectGrid Server
 - ▶ Server runtime supports full clustering, replication, and partitioning of distributed object caches
 - ▶ Applications can connect to a server directly, or declaratively through configuration files
 - ▶ Client logic is included in the server jar so a co-located client can be used in the same JVM as the server
 - ▶ The XD ObjectGrid server is not currently supported on z/OS

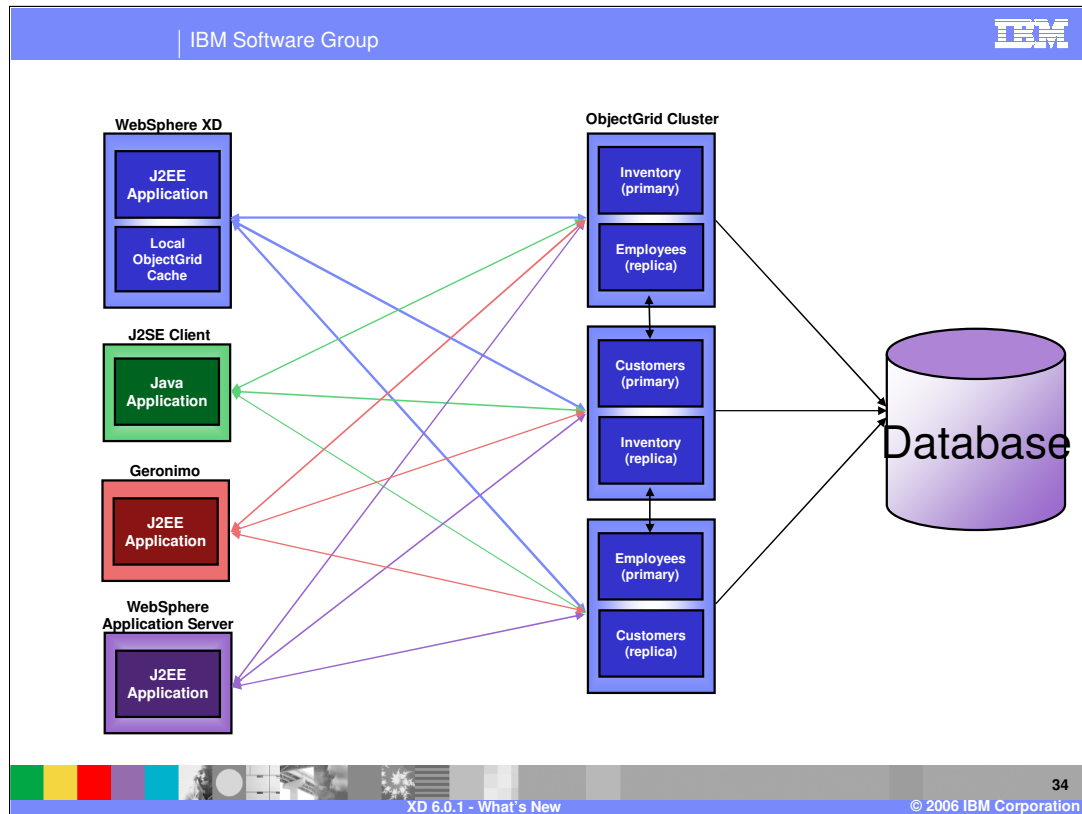
The client/server object grid model employs a dedicated-purpose object grid server. This server does not run in a WebSphere server; but is a scaled-down server in its own right. Both client- and server-side caches can be defined programmatically, or declaratively using property files.

The local ObjectGrid cache and the ObjectGrid client are supported on both distributed and z/OS platforms. However, the stand-alone object grid server will not be available for z/OS in this release. An application deployed to WebSphere XD V6.01 for z/OS that requires an ObjectGrid server is expected to use an ObjectGrid server running on another platform, such as Linux®.

ObjectGrid Client Support

- ObjectGrid data can be accessed by code running on:
 - ▶ WebSphere Application Server 5.0.2 or higher
 - ▶ J2SE 1.3.1
 - ▶ J2SE 1.4
 - ▶ J2SE 5

The ObjectGrid client runtime is supported on J2SE and J2EE at JDK level 1.3.1 and later, including WebSphere Application Server Version 5.0.2 and later.



This example illustrates a topology in which several different client types have access to a set of data hosted by an ObjectGrid cluster. The three maps – Inventory, Customers, and Employees – each have a primary instance and a replica. Data changes are periodically sent from the primaries to the replicas to ensure that the replica remains up to date, and can serve client requests in case the primary fails.

Section

Managing Application Editions

This section describes changes to Application Edition Management in WebSphere XD.

Edition Rollout Enhancements

Configure the edition rollout

Configure the edition rollout

Rollout Strategy

Atomic

Grouped

Group size

1

Reset Strategy

Soft reset

Hard reset

Drainage Interval

Drainage Interval

30 seconds

OK Cancel

The edition rollout wizard has been enhanced to provide more control over the behavior of the application rollout.

Rollout Strategy specifies the rollout method, either groups of nodes updated serially or the divide and swap atomic strategy.

Reset Strategy determines whether the entire server or only the application is restarted.

The Drainage Interval is the amount of time that will be allowed to pass after incoming requests have stopped being forwarded to the application server before stopping the application.

Edition Rollout Enhancements

Rollout Strategy

- Specifies the rollout method, either groups of nodes updated serially or the divide and swap atomic strategy

Atomic

- Specifies that only one edition of the application can serve requests during the rollout period
- This results in half of the application server cluster being taken offline and updated, and then the other
- Application requests that arrive while both halves of the cluster are offline are queued by the ODR

Grouped

- Specifies that both old and new editions of the application can serve requests during the rollout period
- Specify the group size with a sub-option
- Group size gives the number of nodes to process at a time
- Users might be served by either edition until the new edition completely replaces the old

Configure the edition rollout

Configure the edition rollout

Rollout Strategy

Atomic

Grouped

Group size
1

Reset Strategy

Soft reset

Hard reset

Drainage Interval

Drainage Interval
30 seconds

OK Cancel

The Atomic rollout strategy specifies that only one edition of the application can serve requests during the rollout period. This results in half of the application server cluster being taken offline and updated, and then the other. Application requests that arrive while both halves of the cluster are offline are queued by the ODR.

The Group strategy specifies that both old and new editions of the application can serve requests during the rollout period. You can specify the group size with a sub-option. The group size gives the number of nodes to process at a time. All servers in a group are restarted before the next group is processed.

Edition Rollout Enhancements

Reset Strategy

- ▶ Specifies whether to recycle, for example, stop and restart, the application or the entire application server
- ▶ **Soft Reset** (Application)
 - Reset the application by stopping or restarting the application in each server of the cluster as the next edition replaces the old edition in that server
 - Native libraries are not unloaded from memory
 - Generally safe for applications that use no native libraries
 - In a production environment, need to monitor the application server process to ensure that there is sufficient virtual memory
- ▶ **Hard Reset** (Server)
 - Recycles the entire application server, refreshing both process memory and any native libraries used by the application
 - This prevents virtual storage exhaustion
 - Allows new versions of native libraries to be loaded
 - When rolling out an application edition that is accompanied by new versions of native libraries on which it depends, you must select hard reset as your reset strategy

Configure the edition rollout

Configure the edition rollout

Rollout Strategy

Atomic

Grouped

Group size
1

Reset Strategy

Soft reset

Hard reset

Drainage Interval

Drainage Interval
30 seconds

OK Cancel

When a soft reset strategy is chosen, the old edition will be stopped, and the new edition started within a running application server. Choosing hard reset dictates that the application server should be stopped and restarted with the new edition. This is generally only necessary if the application makes use of resources that are loaded by the host operating system, such as native libraries.

Reset strategy specifies whether to recycle the application or the entire application server.

Use a Soft reset strategy to reset the application by stopping or restarting the application in each server of the cluster members as the new edition replaces the old edition in that server. With soft reset, native libraries are not unloaded from memory. Soft reset is generally safe for applications that use no native libraries.

A hard reset recycles the entire application server, refreshing both process memory and any native libraries used by the application. This prevents virtual storage exhaustion and allows new versions of native libraries to be loaded. When rolling out an application edition that is accompanied by new versions of native libraries on which it depends, you must

Edition Rollout Enhancements

▪ Drainage Interval

- ▶ The amount of time that an application server serves clients with affinity to that server after the rollout process has begun and before the reset strategy is started
- ▶ Affinities, such as transaction, activity, and compensation-scope, and activities unknown to WebSphere Extended Deployment lengthen the effective drainage interval, because the server does not stop until these units of work complete
- Applications with activities unknown to Extended Deployment can use the AppEditionManager MBean quiesce initiated notification as a trigger to begin shutdown processing and exploit the drainage interval as a time period during which to complete the shutdown

Configure the edition rollout

Configure the edition rollout

Rollout Strategy

Atomic

Grouped

Group size
1

Reset Strategy

Soft reset

Hard reset

Drainage Interval

Drainage Interval
30 seconds

OK Cancel

The drainage interval is the amount of time that will be allowed to pass after incoming requests have stopped being forwarded to the application server before stopping the application. This might be necessary due to affinities, such as transaction, activity, and compensation-scope; and activities unknown to WebSphere Extended Deployment; which lengthen the effective drainage interval because the server does not stop until these units of work complete. This interval should be set to give those activities adequate time to complete. Applications with activities unknown to WebSphere Extended Deployment can use the AppEditionManager MBean quiesce initiated notification as a trigger to begin shutdown processing and exploit the drainage interval as a time period during which to complete the shutdown.

Section

Business Grid

This section describes changes to the business grid component of WebSphere XD.

Business Grid Enhancements

- Health Management support for Grid nodes and Business Grid Applications
- Support for overlapping node groups that contain grid nodes
- Improved status and monitoring information for the Job Scheduler
- Informix support for Batch runtime and scheduler
- Enable grid dynamic cluster to support min/max DC size

- Several situations exist where the typical health management behavior is different for long-running applications. For example, because a long-running application can be running for hours or days, restarting servers is delayed until the pending business grid jobs on the dynamic cluster member complete. Certain health policy conditions that use data from the on demand router (ODR) or Web container do not apply because the business grid scheduler submits jobs directly to the execution environment.
- WebSphere XD V6.0.1 now supports deploying long running applications on node groups containing nodes which are also contained in other node groups. That is, overlapping node groups.
- It is now possible to chart many job statistics in the Runtime Topology chart. Some examples are Jobs requested, completed, queued, dispatched, and job queue and dispatch time.
- Informix is now fully supported for both the Scheduler and Batch Runtime databases.
- The Balancer component now supports minimum and maximum cluster sizes for grid dynamic clusters.

Business Grid

- Support for time-based scheduling of jobs (run job at time x)
- New submit parameters:
 - ▶ -startDate :
 - the date in which the job should be submitted for execution where the required startDate format is yyyy-MM-dd. This parameter requires the -startTime parameter to be defined as well.
 - ▶ -startTime :
 - the time in which the job should be submitted for execution where the required startTime format is HH:mm:ss. This parameter requires the startDate parameter to be defined as well.
 - ▶ Ex: ...-startDate=2005-11-25 -startTime=23:59:00

WebSphere XD 6.0.1 adds the ability to submit a long running job to execute at a requested future start time.

Business Grid

- Support for recurring jobs
- Ircmd -cmd=submitRecurringRequest ...
 - ▶ -xJCL=<XML file name>
 - ▶ -request=<name of request>
 - ▶ -startDate=<date where first job gets submitted>
 - date in which the first job gets submitted for execution where the required startDate format is yyyy-MM-dd. Note that this parameter requires the -startTime parameter to be defined as well
 - ▶ -startTime=<time where job gets submission>
 - time at which the first job and all subsequent recurring jobs get submitted for execution where the required startTime format is HH:mm:ss. Note that this parameter requires the -startDate parameter to be defined as well
 - ▶ -interval=<time period between job submissions>
 - The time period between 2 job submissions for this recurring job request, where the supported time periods are daily, weekly and monthly

WebSphere XD 6.0.1 also adds the ability to submit a long running job to execute at a specified interval. The interval can be daily, weekly, or monthly, beginning at the requested initial start time.

Business Grid

- Modify an existing recurring job request
 - ▶ `-cmd=modifyRecurringRequest`
- Cancel an existing recurring job request
 - ▶ `lrcmd -cmd=cancelRecurringRequest ...`
- Show details of an existing recurring job request
 - ▶ `lrcmd -cmd=getRecurringRequestDetails ...`
- List all existing recurring job requests
 - ▶ `lrcmd -cmd=showAllRecurringRequests`
- Show all recurring jobs of a request
 - ▶ `lrcmd -cmd=showRecurringJobs -request=<name of request>`

WebSphere XD version 6.0.1 also provides the lrcmd commands shown here to support time-based and recurring requests.

Summary

- Platform Support
- Stack Product Support
- Installation
- Enhancements
 - ▶ Dynamic Operations
 - ▶ Visualization
 - ▶ ObjectGrid
 - ▶ Application Edition Manager
 - ▶ Business Grid

In summary, WebSphere Extended Deployment version 6.0.1 adds support for the z/OS platform as well as the WebSphere stack products, WebSphere Process Server, WebSphere Portal Server, and WebSphere Commerce Server. Version 6.0.1 also provides numerous fixes and enhancements to Dynamic Operations, Visualization, ObjectGrid, Application Edition Manager, and Business Grid.

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
eI (logo)/business	DB2	Series	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.

© Copyright International Business Machines Corporation 2005. All rights reserved.

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.