



IBM Software Group

# IBM WebSphere® Extended Deployment for z/OS® V6.0.1

## *Overview*



@business on demand.

© 2006 IBM Corporation  
Updated April 12, 2006

This presentation is an overview of WebSphere Extended Deployment Version 6.0.1.

## Agenda

- Introduction to WebSphere Extended Deployment
- Major features overview
  - ▶ J2EE (Java™ 2 Enterprise Edition) Batch
  - ▶ WebSphere Business Grid
  - ▶ On Demand Router
  - ▶ Advanced Management
  - ▶ Dynamic Operations Environment
  - ▶ High Performance Computing



The agenda is to first introduce WebSphere Extended Deployment, then give an overview of major features.

## Section

# *Introduction*

This section will introduce the WebSphere Extended Deployment product.

## Overview

- WebSphere Extended Deployment V6.0.1 is an ‘add-on’ product for WebSphere Application Server V6.0.2
  - ▶ Installation is a simple delta to an existing environment
  - ▶ Concepts and features can be “turned on” incrementally.
  - ▶ Integrates seamlessly into a network deployment environment
  - ▶ Integrates with standard z/OS features



WebSphere Extended Deployment is an add-on product for WebSphere Application Server version 6.0.2 that provides advanced features for your environment. The installation is a simple delta to your existing environment and does not require “migration” or restructuring of your current installation. WebSphere Extended Deployment has many new functions such as J2EE batch and dynamic horizontal scaling. These features can be selectively incorporated into your system on a schedule suitable to your organization. WebSphere Extended Deployment extends the administration console and scripting, simplifies application management, facilitates the creation of routing rules, and enhances the performance of WebSphere. WebSphere Extended Deployment integrates well into z/OS and compliments many of the standard features such as request routing, work flow management and high availability. Each of these topics will be discussed briefly in this presentation, and covered in more detail in follow on presentations.

## New concepts and features

- Platform consistent
- Integrated into z/OS platform
- WebSphere Business Grid
  - ▶ Goal-based On-Demand Autonomics
  - ▶ J2EE Batch Programming
- Enhanced Dynamic Operations Environment
  - ▶ Enhanced Multi-platform Workload Management and additional classification rules
  - ▶ Consistent WLM model
  - ▶ Improved multi-system workload distribution
  - ▶ Workload balancing for WebSphere platform messaging



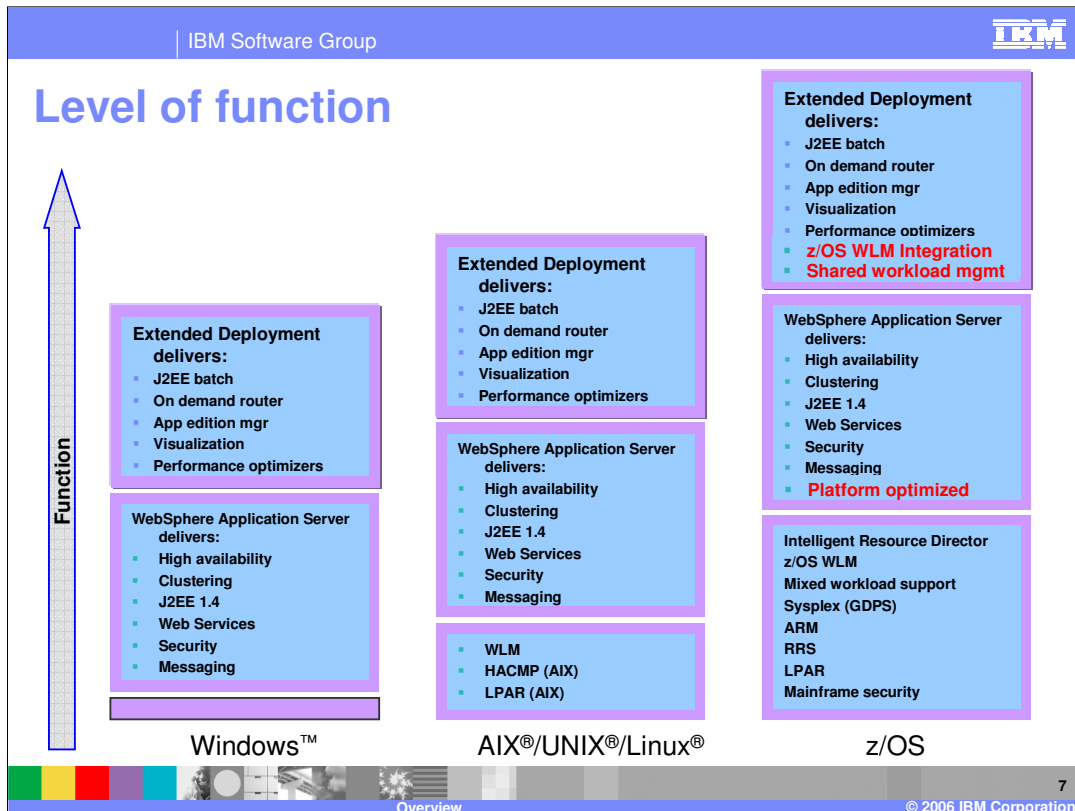
WebSphere Extended Deployment is built from a single shared common code base which assures platform portability and consistency between z/OS and other distributed platforms. This also ensures applications and skills are portable across platforms. At the same time WebSphere Extended Deployment for z/OS has been integrated into the z/OS platform. WebSphere Extended Deployment for z/OS leverages platform Quality of Service features such as Sysplex, WLM, RRS, RACF (SAF), SMF, and RMF. WebSphere Extended Deployment for z/OS optimizes local access to z/OS data resources. In addition, WebSphere Extended Deployment brings several interesting new features to WebSphere for z/OS. J2EE batch is part of WebSphere business grid and parallels MVS batch. The dynamic operations enhances Sysplex distributor and z/OS WLM in interesting ways.

## New concepts and features (cont.)

- **Enhanced Extended Manageability Capabilities**
  - ▶ Managed Rollout of New Applications
  - ▶ Continuous Application availability during application update in a server cluster
  - ▶ Improved application monitoring and visualization
    - Highly Available Deployment Manager (hot standby model)
    - Repository checkpoint and restore
    - Health monitoring
  - ▶ Scale Out Administration
    - Support for non-WebSphere server environments
- **High Performance Computing**
  - ▶ Object Grid
  - ▶ Global Cache - Transactional Replicated Distributed Caching
  - ▶ WebSphere Partitioning Facility



Several new features enhance application and system management and health monitoring. Finally support is offered to enhance high performance application creation and to increase WebSphere's performance.



WebSphere Extended Deployment enhances all the WebSphere platforms. These platforms each have their basic capabilities and offer different levels of quality of service.

WebSphere Application Server brings J2EE programming.

WebSphere Extended Deployment enhances WebSphere base.

WebSphere Extended Deployment for z/OS brings forward the basic z/OS features.

## Section

# ***WebSphere Business Grid***

This section will introduce the WebSphere Business Grid product.



## Business Grid overview

- WebSphere Application Server
  - ▶ Traditionally focused on transactional applications
  - ▶ Applications typically designed to handle large volumes of relatively small tasks
  - ▶ Not all applications fit this type of design



WebSphere Application Server has evolved some features that can be beneficial to long-running applications, such as the WebSphere work manager and message-driven beans for running asynchronous tasks in the EJB container. Still, applications designed to run on WebSphere Application Server tend to be transaction-oriented applications, made up of several relatively short-lived tasks.

## Business Grid

- Enhances WebSphere Application Server to support long-running applications
  - ▶ Provides capabilities to deploy long-running applications along with transactional applications
- Provides support for:
  - ▶ Submitting and running asynchronously
  - ▶ Submitting work from outside the WebSphere environment
  - ▶ Persisting work to highly available, non-volatile data stores
  - ▶ Administrative controls to monitor and manage units of work
  - ▶ Prioritizing and scheduling work



WebSphere Extended Deployment expands on these features by providing a full set of capabilities for running and managing long-running work. Business Grid is a new feature that expands the types of work that WebSphere Extended Deployment is designed to handle. These features include a special environment for long-running work, a separate job submission and scheduling environment, and the administrative tools to monitor and manage long-running work. Business Grid adds to WebSphere Extended Deployment the capability to run and manage long-running applications in the same cell as your transactional applications. WebSphere Extended Deployment coordinates with z/OS workload management (WLM) to dynamically manage both workload types.

## Types of Long Running Work

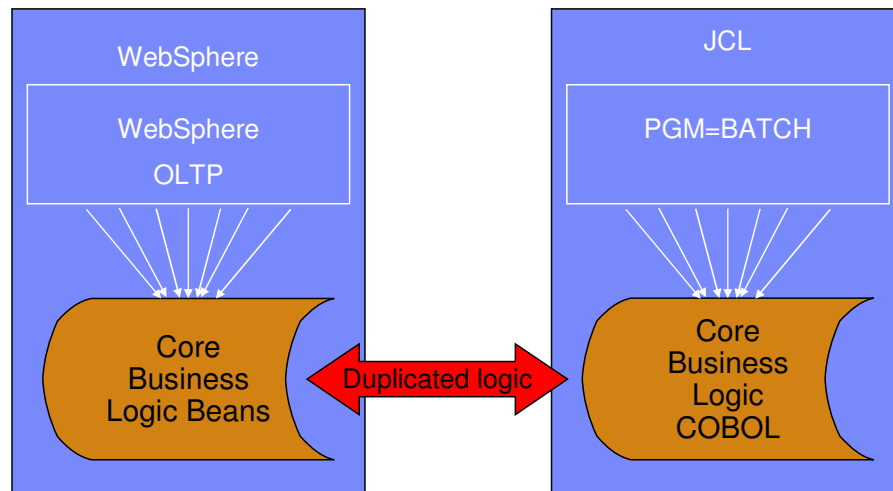
- WebSphere Extended Deployment V6.0 supports two styles of long-running work
- J2EE Batch applications focus on doing large amounts of work based on a specific task
  - ▶ Example: record processing
    - Application will provide logic for a single unit of work (process one record)
    - Container provides support for transactions and mechanisms checkpoint and restart work
- Compute-intensive applications focus on large amounts of processor-bound work
  - ▶ Container provides thread and has limited contact with the work after it is started
  - ▶ Application provides all other logic



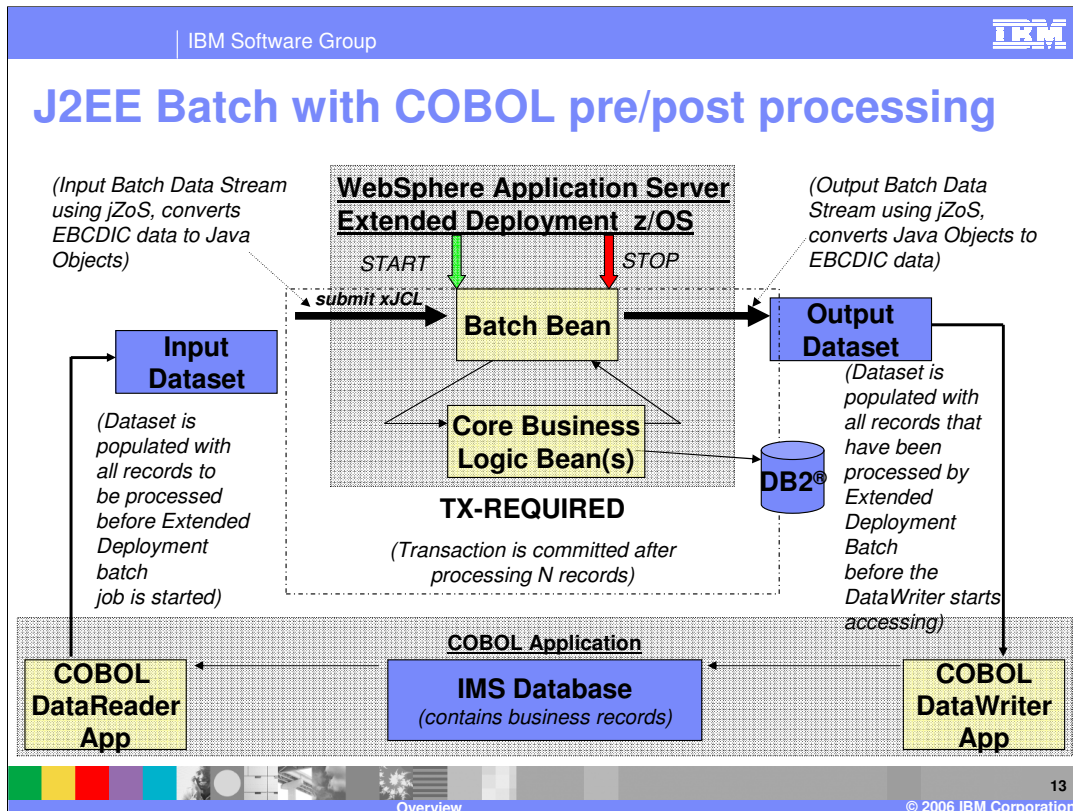
Long-running work can generally be divided into two types, both of which are supported by WebSphere Extended Deployment. J2EE batch applications are designed to repeatedly perform some specific unit of work, such as processing a database record, over a large set of data. Although J2EE batch applications parallel MVS batch in concept, J2EE batch applications are distinct from and extend MVS batch. If an application provides the logic for performing the work, WebSphere Extended Deployment can manage the transaction support and lifecycle aspects, like restarting interrupted work from the point it was stopped.

Compute-intensive applications are also long-running tasks with little user interaction, but focus on performing a large amount of work that tends to be processor-bound, such as large-scale simulation work. In this case WebSphere Extended Deployment provides a runtime environment and a processing thread, and all other logic is implemented by the application.

## Example: Duplicated business logic



There is strong incentive to reuse core business logic wherever possible. The logic could be for calculation of sales commissions, account interests, stocking or pricing policies, stock market financials or any number of other functions. These functions are often used in both On Line Transaction Processing (OLTP) and batch processing. Often the business logic starts off coded in COBOL and duplicated into Java for the OLTP processing. Real problems can occur just ensuring the functional results are up to date and consistent. J2EE batch offers a solution to this problem. The same core business logic beans can be used for J2EE batch and OLTP. One solution would be to move the batch processing entirely over to J2EE batch. This has the draw back that the investment in the COBOL before and after the batch processing is lost. For example, the data that is used to create the batch input stream and interpret and save the batch output away might already be coded in COBOL. In this case, the supporting COBOL can be pretty much left alone and with all the batch processing performed by J2EE batch. In this scenario the COBOL program would generate the batch input, pass it off to a J2EE batch and then catch the output and write the results. This could be attempted with WebSphere without Extended Deployment by substituting OLTP calls for the batch piece, but all the batch characteristics such as check pointing would be lost.



Here is a graphical representation of the previous slide. The program flow starts in the bottom left corner. A COBOL application generates a native dataset containing all records to be batch processed. This is a normal batch processing step. Next a batch job is submitted to WebSphere Application Server Extended Deployment. xJCL specifies the input dataset, the output dataset and the Batch bean to be used for this record processing. The input stream is converted from EBCDIC to Java objects with jZoS, a very high performance tool. An alternative might be to convert the input to SOAP and pass it in that way, but this would offer poorer performance due to .xml translation and such. The input is then processed as transactions inside the batch bean using the core business logic beans. And of course this step can take full advantage of ZAAP processors. Checkpoints happen every N records where the value of N can be passed in the xJCL. This value can be tuned to optimize committing the current transaction and DB2 records. The final batch output can be converted by jZoS into a data set and finally the COBOL application persists the output dataset to its IMS database. A very important point to remember here is that the J2EE batch is asynchronous to classical processes. This means that the J2EE batch must wait until the input stream is ready, start processing and the output catcher, the COBOL data writer in this case, must wait until the J2EE batch finishes.

## Batch Bean internals

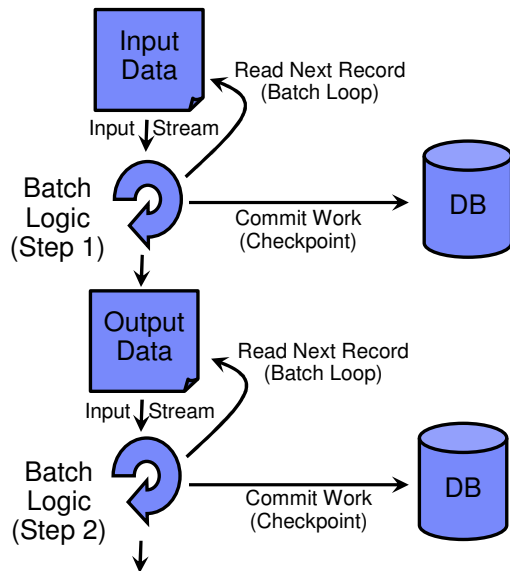
### WebSphere Extended Deployment Batch Framework

```
{  
  - Create a Batch Data Stream (BDS) using jZoS to read the records from the native dataset and a  
  BDS to write the record out to a native dataset.  
  While there are records to be processed in the input dataset  
  {  
    Start a global transaction  
    For every N records process  
    {  
      - For each record that is read in using this BDS, create a Java object that represents the record  
      - Pass the Java Object to the Kernel application so that financial calculations can be made to it  
      - Write the updated record out to the output BDS  
    }  
    commit the global transaction  
  }  
}
```



Here is the core J2EE batch processing. There is one bean that reads lines from an input data set, processes them and writes the results out to a data set. The lines in the dashed box is the record processing loop.

## Flow of a typical batch program

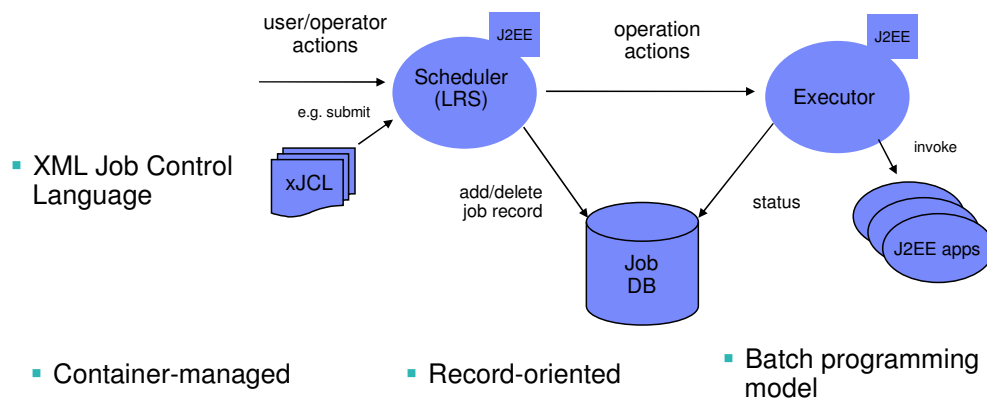


- Typical batch process
  - ▶ Read data (1 or more records) from an input stream
  - ▶ Perform batch process logic using data
  - ▶ Write any output data if needed
  - ▶ Apply checkpoint to work performed by batch process to a database
  - ▶ Loop to get next record
- Each batch job can be made up of multiple batch steps
- The output from one batch step can be an input to another batch step

To better understand what J2EE batch is, first take a high level look at a typical batch program. A batch job can be comprised of one or more batch steps. Dividing a batch application into steps allows for separation of distinct tasks in a batch application. In a typical batch process the application will read and process data, write output data if needed, commit the work to a database, and then move to the next record to repeat the process. The output from one batch step can be the input into another batch step in the process. Each batch job could be made up of any number of individual batch steps.

## J2EE batch

- Runs Asynchronously
- Implemented with J2EE components
- Transactional



J2EE batch is different from traditional MVS batch. J2EE batch jobs are composed of several J2EE applications that run inside a WebSphere process. J2EE batch runs asynchronously, and since it is container managed, it supports transactions.

The process starts when a J2EE batch job is submitted to the Long Running Scheduler (LRS). Part of the input is XML encapsulated JCL.

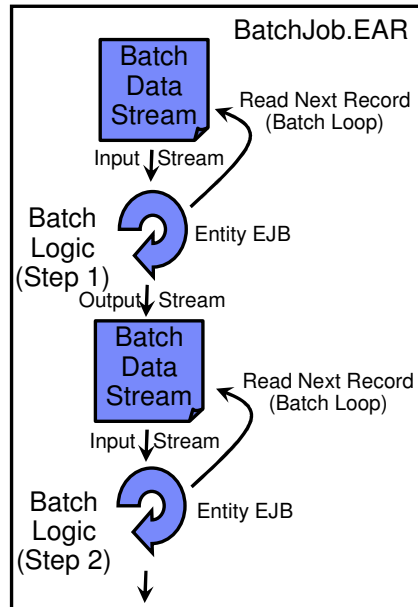
The LRS sends operations to the process that invokes and supports J2EE applications.

The J2EE applications are user created code that follows the batch programming model which is discussed on the next slide.

In contrast, MVS batch consists of conventional programs written in any of a variety of languages such as COBOL or PL/I. More importantly, MVS batch runs in a JES Initiator address space – not a WebSphere process. This means that there is no access from MVS batch programs to WebSphere container facilities and management such as communications, security, and transactions.



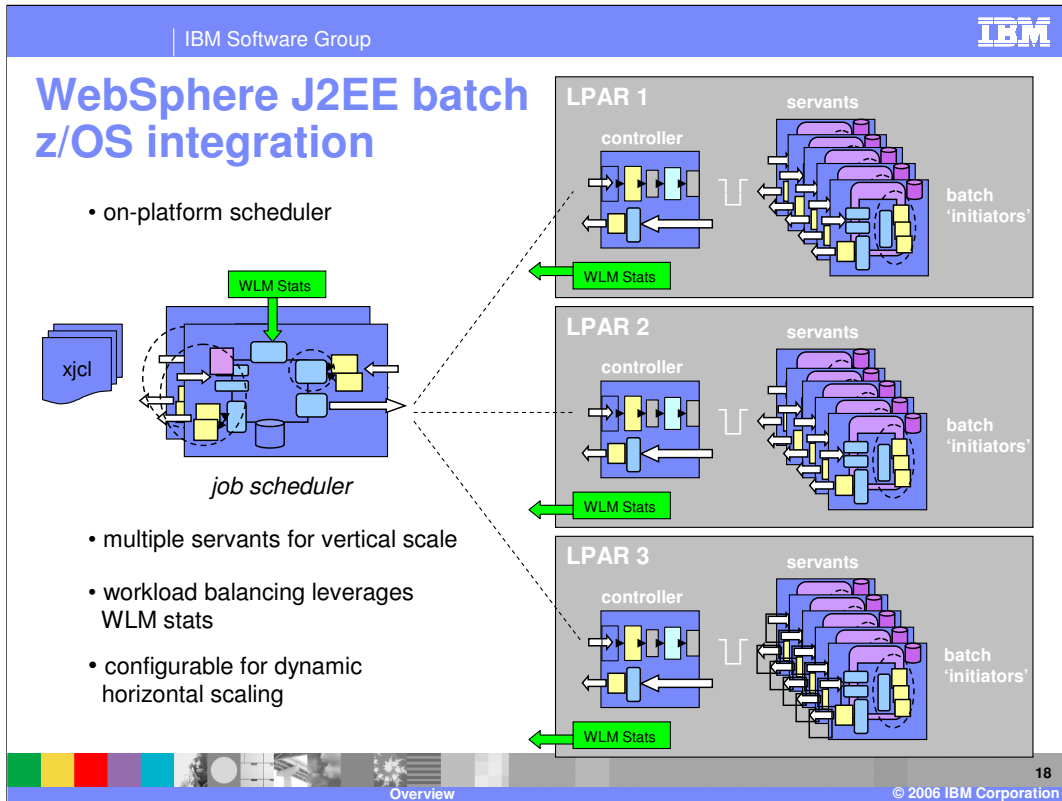
## J2EE batch programming model



- Input and output streams needed by a batch job step are wrapped by a Batch Data Stream object
- The business logic for a batch step is provided by a CMP Entity EJB which implements a Batch Step Interface

A J2EE batch step is implemented as a local container-managed persistence entity bean that uses the WebSphere provided home, business and key interfaces. The business interface of a batch step EJB, `com.ibm.websphere.batch.BatchJobStepLocalInterface`, provides methods that the batch processing environment invokes to control a batch application. A batch step can have zero or more batch data streams associated with it. A batch data stream (BDS) is a java class that implements the `com.ibm.websphere.batch.BatchDataStream` interface. A BDS is a Java object that reads the input stream that contains the data to be processed by a batch step. A BDS can also be an output stream that writes data instead of reading it.

Methods on the batch data stream interface allow the batch processing environment to manage the data stream being used by a batch step. For example, one of the methods retrieves current cursor information from the stream to keep track of how much data has been processed by the batch step.



Extended Deployment for z/OS extends WebSphere Application Server z/OS basic features to provide both horizontal and vertical scaling to support even the most demanding workloads. z/OS WLM integration enables Extended Deployment schedulers to make cooperative WLM decisions and facilitate administration through leverage of the Extended Deployment classification engine.

For J2EE batch, the scheduler is a WebSphere application, the Long Running Scheduler, running in a cluster.

z/OS WLM provides vertical scaling by starting multiple servant regions.

At the same time, z/OS WLM statistics are fed back into the Long Running Scheduler.

As required, the Long Running Scheduler can start servers to provide horizontal scaling

## Section

# *On Demand Router*



This section will introduce the WebSphere On Demand Router.

## On Demand Router

- On Demand Router 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 that is provided with WebSphere Extended Deployment. It is the point of entry into a WebSphere Extended Deployment environment, and is responsible for request prioritization, flow control, and distribution to application servers. It can momentarily queue requests for less important applications in order to allow requests for more important applications to be handled more quickly. It is aware of the current location of a dynamic cluster's server instances, so that requests can be routed to the correct endpoint. The On Demand Router can also dynamically adjust the amount of traffic sent to each individual server instance based on processor utilization and response times. The On Demand Router can detect and react to a "storm drain condition". These and other advanced features distinguish the On Demand Router from the HTTP server plug-in, and give the On Demand Router 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.

## On Demand Router (cont.)

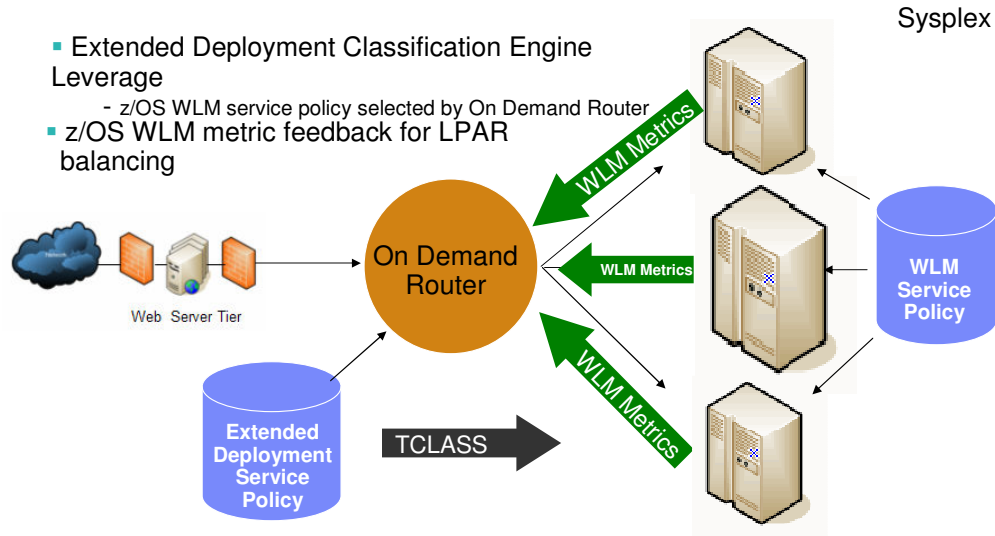
- Important participant in many Extended Deployment functions
  - ▶ Dynamic Operations
  - ▶ Edition Management
  - ▶ Scale Out Management
  - ▶ Object Grid (partitioning)
  - ▶ Not directly part of Business Grid
- World Class edge of Sysplex distribution
  - ▶ Compliments Sysplex Distributor
  - ▶ Complements z/OS WLM

The On Demand Router is one of two primary WebSphere Extended Deployment workload schedulers. The other scheduler is the Long Running Scheduler discussed earlier. As such the On Demand Router is a critical piece of WebSphere Extended Deployment and is required for most of the new functionality, except for WebSphere business grid applications. For example, the On Demand Router maintains both an application view of offered load and a two-way coordination with z/OS WLM. The On Demand Router is key to application roll out for edition management by routing request to the correct cluster or temporarily caching request as necessary.

The On Demand Router also provides world class edge of Sysplex distribution. When placed as end points for Sysplex distributor, the On Demand Router uses WLM information to enhance the Sysplex Distributor. The On Demand Router is integrated with z/OS WLM and the On Demand Router rules builder actually enhances service policy selection. The On Demand Router can be placed on a non z/OS based platform and still maintain integration with z/OS WLM.

## z/OS WLM integration

- Extended Deployment Classification Engine Leverage
  - z/OS WLM service policy selected by On Demand Router
- z/OS WLM metric feedback for LPAR balancing



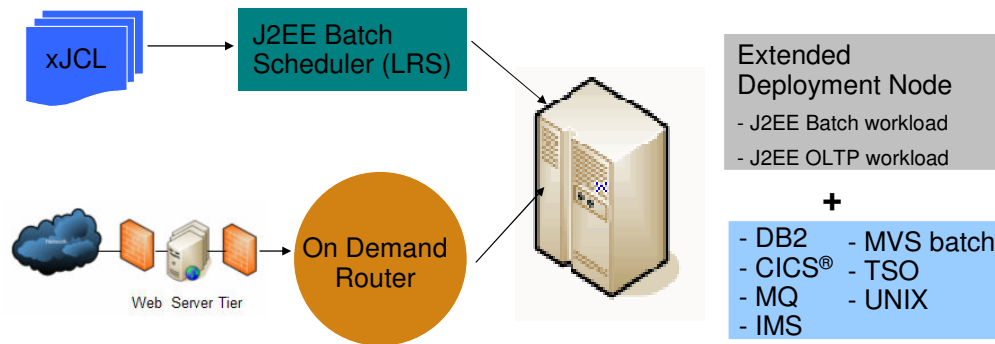
z/OS WLM integration is a unique value-add that Extended Deployment brings to the z/OS platform.

It provides two-way integration:

- 1) Service unit metric data is retrieved from the platform to enable the On Demand Router to make informed decisions on workload placement.
- 2) The classification engine in Extended Deployment is leveraged to assign and propagate the transaction class (TCLASS). This class will be used downstream by a WebSphere cluster to register the work request with z/OS and to select the z/OS service policy that governs how it runs.

## Shared workload management

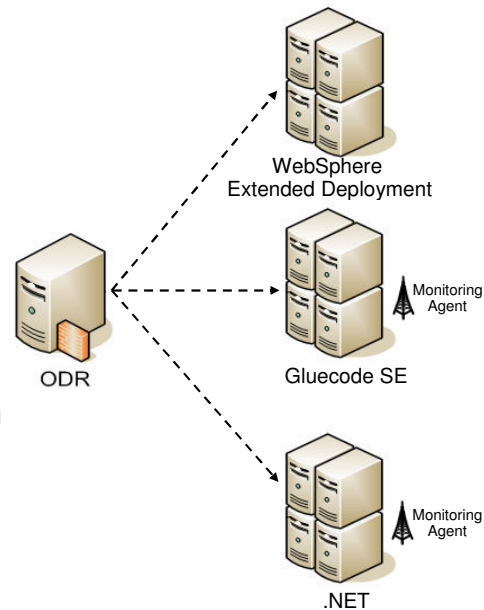
- Prioritization & Queuing delegated to z/OS WLM
- Enables z/OS inherent strength to manage diverse workload



Shared workload management is an additional z/OS specific capability. Extended Deployment manages workload on distributed platforms by dedicating each node to a particular task, such as processing J2EE batch or OLTP. In that environment, each node is typically a separate system and the resource management operates with the default presumption that the entire system is dedicated to executing WebSphere applications for a single cell. In contrast, the z/OS system is inherently designed and used to manage large, diverse, mixed workloads and manage those workloads with its built-in Sysplex-aware workload manager. Extended Deployment on z/OS changes its resource management style to cooperate with the z/OS platform workload manager. This allows Extended Deployment nodes to host mixed J2EE batch and OLTP along with whatever other work the platform may be hosting. This way, Extended Deployment conforms to and supports the normal operational and management approach used to host workloads on z/OS.

## Integration with other servers

- The On Demand Router can route traffic to non WebSphere Extended Deployment servers
  - ▶ Any HTTP(S) endpoint, including Gluecode SE, Tomcat, or .NET
- Optional monitoring agent provides better integration
  - ▶ Communicates processor capacity and utilization information to the On Demand Router



Non WebSphere Extended Deployment servers can also be integrated into a WebSphere Extended Deployment environment. The On-Demand Router can route traffic to any HTTP endpoint, meaning that you can use it as a front end to applications running on third-party servers, including Gluecode SE, Tomcat, or .NET. While dynamic application placement cannot be performed on these non WebSphere Extended Deployment servers, you can benefit from the On-Demand Router's request prioritization and flow control features. For enhanced integration, you can also install a lightweight monitoring agent on each of the non WebSphere Extended Deployment servers that will communicate processor capacity and utilization information back to the On Demand Router. This enables the On Demand Router to make better decisions as to how much traffic should be routed to each server.



## Section

# ***Advanced administration features***

This section will introduce the Advanced Administration Features

## Advanced administration features

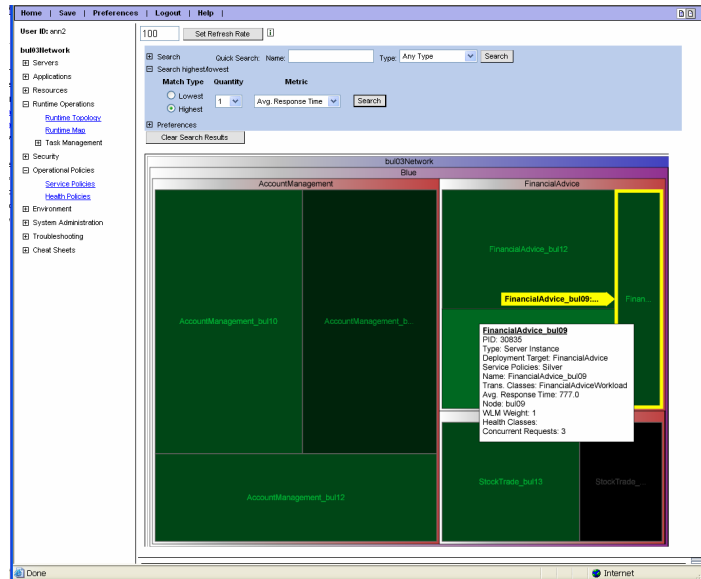
- Visualization tools simplify health and performance assessments
- Edition Control Center supports multiple application editions and seamless rollout
- Health monitoring tools provide means for remediation of software health problems
- Repository checkpoints enable returning to a previously saved 'snapshot' of the configuration repository
- Standby Deployment Managers can be created for administrative high availability



WebSphere Extended Deployment includes many features that make life easier for the administrator of a WebSphere environment. The visualization components enhance the Administrative Console to provide live data on the performance and health characteristics of the entire cell. The Administrative Console also features the Edition Control Center, which enables management of multiple editions of an application, including interruption-free rollout. WebSphere Extended Deployment also has health monitoring capability, and can take remediation steps when software health problems are detected. The extended repository service is a new feature that lets you create saved checkpoints of your configuration repository. These checkpoints allow you to later return to a saved configuration, or they can be used as an 'undo' feature for configuration changes. WebSphere Extended Deployment also lets you create multiple Deployment Managers in a single cell. One of these standby Deployment Managers can take over if the active Deployment Manager fails, so that a Deployment Manager is always available for managing your cell.

## Runtime visualization

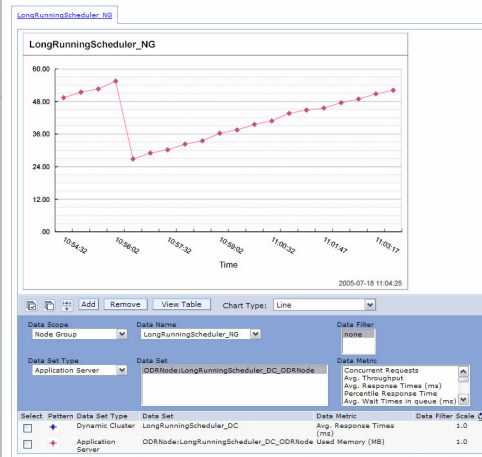
- Runtime Map
  - ▶ Summary view of entire cell
  - ▶ Quickly assess overall health



The visualization features can be accessed from the Administrative Console. The Runtime Map displays an overview of the entire cell for quick assessment of the overall performance relative to defined goals. Color is used to indicate goal-relative performance, and size is used to indicate the amount of work performed, relative to the other servers. You can also drill down or search within this view to get information about a specific server.

## Runtime visualization

- Runtime topology
  - ▶ Detailed view of what is running and where
  - ▶ Custom charting of performance and goal data
  - ▶ Data can be logged for reference or chargeback



Overview

28

© 2006 IBM Corporation

The runtime topology view displays a simple overview of your entire topology, and shows where individual components (such as dynamic cluster instances or highly available runtime components) are currently running. You can also display custom charts that can represent numerous statistics about components of your cell, from an application's average response time to the processor utilization of an individual server.

## Application edition management

- WebSphere Extended Deployment supports managing multiple editions of an application in a WebSphere cell
  - ▶ Interruption-free rollout of application updates
  - ▶ Ability to “roll back” to a previous application version
  - ▶ “Validation mode” to verify functionality using a subset of users
  - ▶ Routing policies can be used to define which particular requests are handled by which application edition
- An *edition* is a distinct instance of a J2EE application (similar to *version*)
  - ▶ May be a distinct build version, or the same build with different deployment bindings



WebSphere Extended Deployment allows you to deploy and manage multiple editions of the same application within a cell. You can use the Edition Control Center to manage the rollout of new editions to some or all of your servers without service interruption. You also have the ability to roll back to previous editions if you find a problem with the new edition after deployment. ‘Validation’ of an edition rolls it out to a new cluster that is a clone of the cluster to which a previous edition is deployed. This way you can verify the functionality of the edition in your production environment. If you have multiple editions of an application deployed concurrently, you can create routing policies that dictate which edition should serve each incoming request. For example, all requests from users in a certain group, or from a certain IP address range, might be served by a different edition than all other users for testing purposes.

## The Edition Control Center

### [Edition Control Center](#) > Manage Editions

Manage editions of an application. The deployment targets for each edition were specified during the application install process. After install, an edition is initially in the inactive state. Inactive editions cannot be started. Activating an edition makes it eligible to be started. Validating an edition puts it into a special "validation mode" that configures the edition to run on a clone of its original deployment target. Validation mode requires assignment of a routing policy to the edition to control who may access it. Rolling out an edition performs an interruption-free upgrade of one edition to another on the same deployment target. Rolling out an edition that is in validation mode performs an interruption free upgrade of the edition on the deployment target from which the validation mode target was cloned. After the rollout, the clone is deleted. Deactivation makes an edition ineligible to be started. An edition must be stopped before deactivation.

Application=**XDStockTradeEdition**

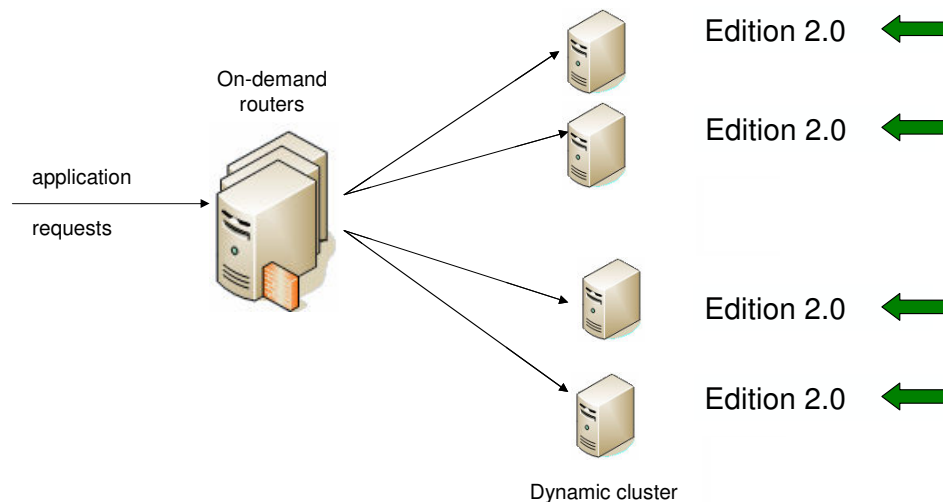
Preferences

<input type="button" value="Activate"/> <input type="button" value="Validate"/> <input type="button" value="Rollout"/> <input type="button" value="Deactivate"/>				
<input checked="" type="checkbox"/> <input type="checkbox"/>				
Select	Editions	Description	Target	State
<input type="checkbox"/>	1.0		WebSphere:cell=wsbeta156Cell01,cluster=StockTrade_DC	INACTIVE
<input type="checkbox"/>	1.0.1		WebSphere:cell=wsbeta156Cell01,cluster=StockTrade_DC	ACTIVE
Total 2				



The Edition Control Center is a new section of the Administrative Console that enables you to manage the various editions of your installed applications. It allows you to see which editions are currently active in your environment, and manage how those editions are rolled out. From here you can choose to roll a new edition out to your cluster one server at a time, or use a 'divide and switch' approach. You can also put an application edition into the previously mentioned 'validation mode'.

## Operational scenarios: Atomic rollout



31

Overview

© 2006 IBM Corporation

This example illustrates one of the available rollout methods, called 'atomic rollout'. It uses a 'divide-and-switch' technique to rollout a new edition across a cluster while ensuring that only one edition is actively serving traffic at any given time.

- 1) The On Demand Router will first quiesce traffic to half of the servers in the cluster, and when existing requests complete,
- 2) the new edition is installed to those servers,
- 3) and they are restarted.
- 4) Traffic will then be quiesced on the other half of the servers.
- 5) While waiting for those requests to finish, there may be a brief build-up of requests in the On Demand Router's dispatch queue, since no servers are receiving traffic.
- 6) After the existing requests have been served,
- 7) requests will be routed to the first set of servers, where the new edition of the application is available.
- 8) The new edition will then be installed on the second set of servers.
- 9) Once they have been restarted, the entire cluster will be running the new application edition.

Note that since your cluster will effectively be running at half capacity during this process, it is wise to perform this type of an edition upgrade during a low-traffic time period.

## Health monitoring

- A WebSphere Extended Deployment environment can be monitored for various software health conditions
  - ▶ Age, work performed, memory usage, response time
  - ▶ Stuck requests, storm drain detection (new in Extended Deployment 6.0)
- Health conditions and reactions are defined by Health Policies
  - ▶ Servers can be restarted as a corrective or preventive action
  - ▶ Java heap can be dumped for memory problems



Health monitoring is another useful administration feature. WebSphere Extended Deployment allows you to configure 'health policies', which define software conditions that may indicate a problem. You can look for excessive memory usage, stuck requests, or simply uptime, and take a user-defined action if those conditions are matched. For example, if your application has a known bug that causes degraded performance after a week of uptime, you could choose to automatically restart the server weekly. Similarly, you could also choose to generate a dump of the Java heap every time memory consumption exceeded a certain level, and restart the server as well. Note that heap dumps can only be generated automatically on platforms that utilize the IBM Java Runtime Environment.



## Repository checkpoints

- Repository checkpoints give you advanced control over the configuration repository
- A full checkpoint is a snapshot of the repository
  - ▶ Restoring to a full checkpoint returns to the configuration state from the time the snapshot was taken
- A delta checkpoint records the most recent change to your configuration
  - ▶ Delta checkpoints can be rolled back individually, in order
  - ▶ Can be created automatically when changes are made

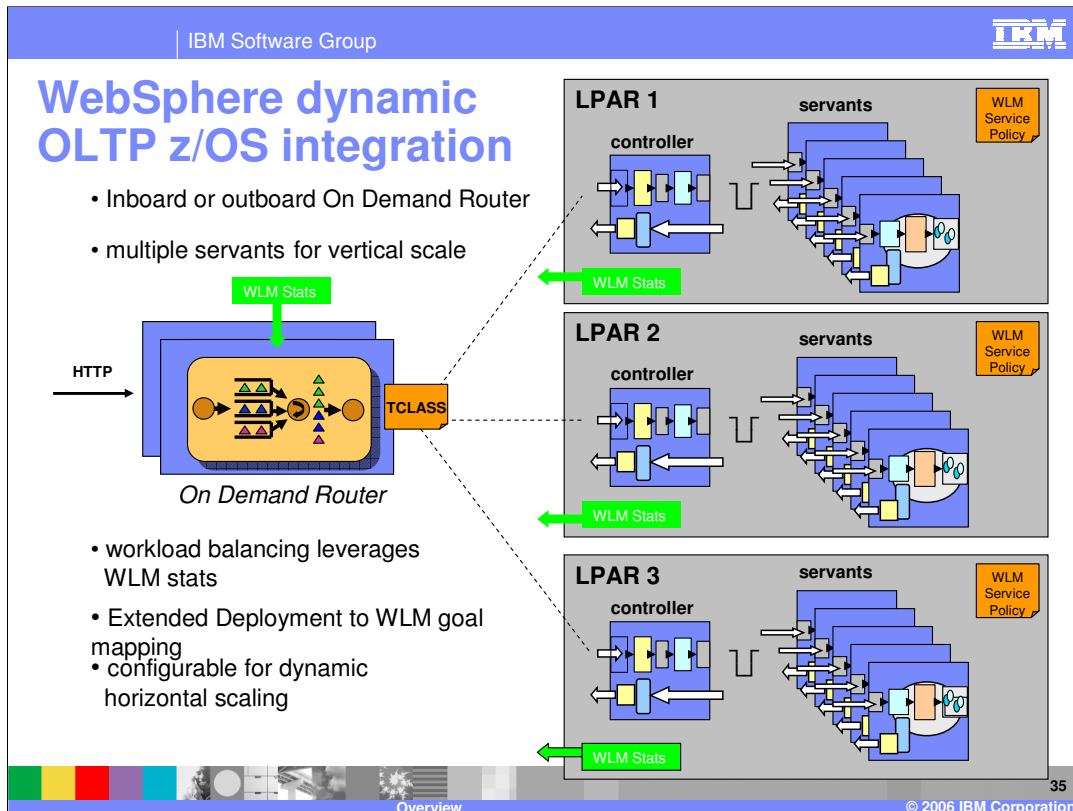


Repository checkpoints are a new feature that make it easier to manage your WebSphere Extended Deployment configuration. A 'full checkpoint' is a snapshot of your entire configuration repository that you can create at any time, and use in the future to restore your configuration back to that state. You can also configure WebSphere Extended Deployment to create 'delta checkpoints' each time you save changes to your configuration repository. Delta checkpoints contain a record of configuration objects that were changed during the last save action, and you can roll back one or more delta checkpoints to undo a series of changes in reverse order.

## Section

# *Dynamic Operations Environment*

This section will introduce the Dynamic Operations Environment.



On Line Transaction Processing (OLTP) uses the same principles as on the J2EE batch topology chart earlier. Here OLTP requests come in from the left and follow a path to servant regions that satisfy the request. Again, Extended Deployment for z/OS extends WebSphere Application Server for z/OS basic features to provide both horizontal and vertical scaling to support even the most demanding workloads. z/OS WLM integration enables Extended Deployment schedulers to make cooperative WLM decisions and facilitate administration through leverage of the Extended Deployment classification engine. For OLTP, the schedulers are WebSphere applications running in clusters and On Demand Router.

- 1) A unique feature of the On Demand Router is that it may be deployed on or off a z/OS platform (inboard or outboard) and still deliver the z/OS WLM integration features.
- 2) z/OS WLM provides vertical scaling by starting multiple servant regions.
- 3) At the same time, z/OS WLM statistics are fed back into the LRS.
- 4) A powerful rules engine assigns a Transaction Class (TCLASS) to each request. The TCLASS maps directly to a z/OS WLM service policy. The TCLASS is passed along with the request to the server (controller region) which by default will enter the subsequent request into WLM using the TCLASS as the service policy descriptor.
- 5) As required, the LRS can start servers to provide horizontal scaling

## Section

# *Object Grid*

This section will introduce the WebSphere Object Grid product.

## 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 is a new WebSphere Extended Deployment technology that provides a high-performance, transactional cache framework for storing Java objects. An ObjectGrid can be used as a generic object cache, and optionally persisted to a hardened store. It can in the same fashion 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.

## Partitioning programming model

- Caching of resource expensive data
- *Reduces lock contention*
- *Extensive caching*
- *Application portability*
- *Partitions are highly available*
- *Requires user design/coding*
- *Limited to JMS connections on z/OS*

The partitioning facility is a programming framework and runtime environment that makes it possible for high transaction-volume applications to scale linearly and increasing resource utilization. To accomplish this, an application is partitioned across multiple servers in a cluster. Each partition is a uniquely addressable endpoint within the cluster, to which requests for certain data sets are always routed. Partitioning solves some of the traditional challenges of very large clustering, because it can reduce data contention and reduce the overhead of replicating shared data, like caches or state information.

While partitioning reduces the number of locations within a cluster where a particular piece of logic may be running, your entire application remains highly available, because partitions are managed by WebSphere Application Server's High Availability Manager. The High Availability Manager ensures that every partition is running at all times, even in the case of a server failure.

While the standard J2EE programming model is available to partitioned applications, you must also create a Partitioned Stateless Session Bean. This bean tells the partitioning facility how your application should be partitioned, and how to associate requests with the correct partition, so they can be routed properly. For V6.0.1 on z/OS the implementation is limited to JMS traffic.

## Summary

- WebSphere Extended Deployment 6.0 is an 'add-on' product for WebSphere Application Server that provides many advanced features
- Advanced administrative capabilities
  - ▶ Administrative visualization provides new ways to look at performance data about your environment
  - ▶ Application edition management features enable seamless rollout of multiple editions of an application
  - ▶ Health monitoring can trigger alerts and take action when software health problems are detected
- Object Grid is a high-performance, transactional, and extensible cache framework



In summary, WebSphere Extended Deployment is an add-on product that provides several advanced features for WebSphere Application Server. The dynamic operations features enable a virtualized, dynamic environment for better utilizing shareable hardware resources based on user-defined performance goals. The partitioning facility provides a programming model and runtime environment for implementing highly scalable solutions for transaction processing. ObjectGrid is a new high-performance, transactional, and extensible object cache for Java applications.

## Value summary: Extended Deployment V6.0.1

- Do more with less
  - ▶ Autonomic resource management
  - ▶ Qualities of Service for non-WebSphere Application Server environments
- Create high performance applications
  - ▶ partitioned programming model
  - ▶ Object Grid cache technology
- Build more flexible solutions
  - ▶ long running programming model (batch, compute-intensive)
- Achieve operational efficiency
  - ▶ at-a-glance performance viewer
    - health policies and much more...
- Ensure application continuous availability
  - ▶ application versioning



WebSphere Extended Deployment gives you the capability to create and manage high performance applications on WebSphere, manage non WebSphere applications, monitor system health and preserve investments in both skills and application code.



## Value summary: Extended Deployment for z/OS

- Important enterprise-class features
  - ▶ J2EE Batch
  - ▶ Application versioning
  - ▶ Storm Drain
  - ▶ Application Lazy Start
- Family Consistency
  - ▶ Application portability
  - ▶ Skills portability



Extended Deployment z/OS offers value to WebSphere Application Server z/OS deployments because it offers multiple features important to enterprise class customers. These features include J2EE batch, application versioning, storm drain health policies, application lazy start, and repository checkpoint and restore.

## Value summary: Extended Deployment for z/OS

- Superior workload management
  - ▶ On Demand Router for intelligent LPAR routing
  - ▶ z/OS WLM service policy selection through Extended Deployment classification engine
  - ▶ On Demand Router cooperation with J2EE Batch Scheduler (LRS) for managing mixed workload
- J2EE Batch
  - ▶ Asynchronous access to core business logic
  - ▶ Shared work load management



The On Demand Router offers many functions that go beyond what the other HTTP routing alternatives offer. J2EE batch provides secure, asynchronous access to the same core business logic used by OLTP. The access to the shared business logic is secure. Shared workload management is an additional z/OS specific capability. Extended Deployment manages workload on distributed platforms by dedicating each node to a particular task, such as processing J2EE batch or OLTP.

## 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.

© Copyright International Business Machines Corporation 2005,2006. 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.