



IBM Software Group

IBM WebSphere Application Server V6

Programming Model Extensions Overview



@business on demand.

© 2006 IBM Corporation
Updated May 13, 2015

This presentation will discuss a high-level overview of the WebSphere® Application Server V6 Programming Model Extensions (PMEs).

Goals

- Provide an introduction of the Programming Model Extensions
- Understand which extensions are available in IBM WebSphere Application Server V6

The goals for this presentation are to provide a high-level overview of the Programming Model Extensions and to understand what functionality is available in the IBM WebSphere Application Server V6.

Agenda

- Overview
 - ▶ What Are Programming Model Extensions?
 - ▶ Most functionality migrated from WebSphere Business Integration Server Foundation V5.1
 - ▶ High-level Functional description of each extension
- Summary

The agenda for this presentation is to define Programming Model Extensions, describe how you would use them, review which extensions are now available in WebSphere Application Server V6, and provide a brief description of each extension.

Section

Overview



This section will provide an overview of Programming Model Extensions.

What are the Programming Model Extensions?

- Set of functions that extend or complement the standard Java™ 2 Enterprise Edition (J2EE) programming model
- Enable the creation of sophisticated applications
 - ▶ J2EE compliance is preserved whenever possible
 - Defined with extended deployment descriptors unique to the WebSphere Application Server platform
- Build upon and extend the industry and open standards supported in WebSphere Application Server V6



Programming Model Extensions consist of a set of function that builds upon J2EE standards, in addition to other industry and open standards. While J2EE provides the building blocks for developing typical applications, there comes the time when developers need additional APIs to help them build more robust applications. They need APIs that can provide them with increased power to help solve large-scale business problems. These additional APIs are the extensions you will be previewing in this presentation.

Programming Model Extensions

Core extensions included in all WebSphere versions

▶ Formerly only in WebSphere Business Integration Server Foundation

- Last Participant Support
- Internationalization Service
- WorkArea Service
- ActivitySession Service
- Extended JTA Support
- Startup Beans
- Asynchronous Beans
- Scheduler Service
- Object Pools
- Calendar
- Dynamic Query
- DistributedMap
- Application Profiling

▶ Newly added

- CommonJ Work Manager
- CommonJ Timer Manager

Many of the extensions that were previously available only in the IBM WebSphere Business Integration Server Foundation product, are now available in all versions of WebSphere Application Server. Here you see a listing of those extensions, and the newly added Work Manager and Timer Manager extensions.

High-Level Functional Description

- Last Participant Support
 - ▶ Mix resources in a transaction
- Internationalization Service
 - ▶ support for international user sets
- Work Area Service
 - ▶ Implicitly propagate application context on thread
- ActivitySession Service
 - ▶ Alternate unit-of-work model to EJB transactions
- Extended JTA Support
 - ▶ Added APIs



Last Participant Support enables the use of a single, one-phase commit capable resource with any number of two-phase commit capable resources in the same global transaction.

Internationalization Service enables applications to support international user sets by creating a J2EE Locale context with appropriate locale and time zone information.

Work Area Service allows application developers the capability to implicitly propagate application context on the running thread.

ActivitySession Service provides an alternate unit-of-work model to standard EJB transactions.

Extended JTA Support provides APIs in addition to the JTA User Transaction interface.

High-Level Functional Description (cont.)

- **Startup Beans**
 - ▶ Run business logic at application start or end
- **Asynchronous Beans**
 - ▶ Initiate and control parallel processing
- **Scheduler Service**
 - ▶ Schedule time-dependent tasks
- **Object Pools**
 - ▶ Reuse application-defined objects
- **Calendar**
 - ▶ pluggable calendar capabilities



Startup Beans provide the capability to run business logic automatically whenever an application starts or stops in a normal fashion.

Asynchronous Beans enable applications to initiate and control parallel processing using application-spawned threads, passing along context, within the J2EE framework, and provide many other additional asynchronous processing capabilities

Scheduler Service provides a persistent scheduling service which allows for the scheduling of time-dependent tasks

Object Pools provide support for the pooling of application-defined objects for the purpose of reuse

Calendar provides default or custom-defined pluggable calendar capability

High-Level Functional Description (cont.)

- **Dynamic Query Service**
 - ▶ Runtime EJBQL enhancements
- **Distributed Map API**
 - ▶ Cache and share Java object references
- **Application Profiling**
 - ▶ EJB tuning capabilities
- **CommonJ TimerManager**
 - ▶ Schedule and receive timer notifications
- **CommonJ WorkManager**
 - ▶ API for concurrent work items



Dynamic Query Service allows you to formulate and run EJB Query Language queries at runtime, and provides enhancements to EJB Query Language, substantially increasing its usefulness.

Distributed Map API provides J2EE applications with an API to cache and share Java objects by storing a reference to the object in the cache.

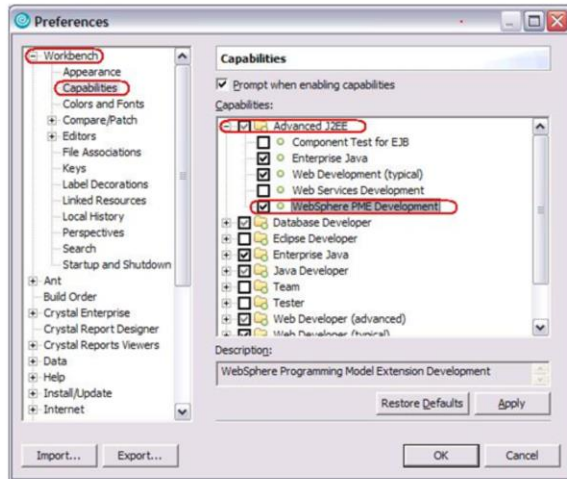
Application Profiling provides fine-grained, application-focused EJB tuning capabilities, and is particularly useful in environments with EJB reuse.

CommonJ TimerManager enables J2EE applications to schedule subsequent timer notifications and receive timer notifications.

CommonJ WorkManager provides an API for application-server supported concurrent running work items. The WorkManager programming model is a simplified subset of the Asynchronous Beans programming model

Enable Development Capabilities

- To enable WebSphere PME Development capabilities in IBM Rational® Application Developer V6
 - ▶ Open the Workbench
 - ▶ Select Window > Preferences > Workbench > Capabilities > Advanced J2EE and check WebSphere PME Development



IBM Rational Application Developer V6.0 or the WebSphere Application Server Toolkit V6.0 may be used to configure PME settings. It is worth noting that you will need to enable the WebSphere PME Development capabilities when using Application Developer, as this is not enabled by default. Open the Workbench, select Window > Preferences > Workbench > Capabilities, check and expand the Advanced J2EE box, and check the WebSphere PME Development box. Click OK.

Section

Summary

Summary

- Numerous extensions to the J2EE programming model are available with Programming Model Extensions
- Robust APIs and resources provide a head-start to developers for building enterprise applications
- Solve such issues as integration with enterprise systems, EJB performance, asynchronous programming, and scheduled programming
- Further details provided in WebSphere V6 Information Center and additional presentations



In this presentation you learned about the Programming Model Extensions available in WebSphere Application Server V6 and how they may be used to develop robust, enterprise-level applications that require additional function over and above that found in the J2EE programming model. Further details regarding individual extensions are provided in the WebSphere V6 Information Center, and in additional IBM Education Assistant presentations.

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	Series	OS/400	xSeries
AX	DB2 UniversalDatabase	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 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.