



IBM Software Group

# **WebSphere Process Server V6.0**

## **WebSphere Integration Developer V6.0**

### *Human Tasks Overview*



@business on demand.

© 2005 IBM Corporation  
Updated October 31, 2005

This presentation will cover the Human Tasks feature of WebSphere Process Server and WebSphere Integration Developer V6.0.

## Agenda

- **Overview and Architecture**
- Human Tasks Features
- Summary and References



This first section will provide an overview of support for Human Tasks to be used in integration applications.





## Human Interaction in Integration Applications

- Human interaction is key to many business integration applications
  - ▶ Human input or review required
    - Manager must approve/reject employee travel request
  - ▶ Amount of automation is limited
    - Document review or translation
  - ▶ Error or Exception situations must be handled
    - Flight unavailable, credit card check failed, loan amount too high
  
- Many challenges with adding human interaction
  - ▶ How are the correct people involved?
  - ▶ How to provide robust capabilities such as claim, transfer, complete, notify, expire and escalate?
  - ▶ How can human interaction be implemented in a flexible manner to allow for changes in the business?



When integrating different types of applications in an enterprise due to merging of departments, an acquisition of another company, or for any reason, some processes require human interaction. For example, sometimes a manager's approval is required for a travel request or purchase order. A document review is another example of a process where human interaction is required. Error situations can also require human interaction in order to resolve. Introducing human interaction can present some unique challenges in terms of application development and integration. Many times the user is not technically savvy, requiring more time spent on creating a more friendly user interface and not on how to fit the human interaction into the integration application. The task must also be assigned to the correct people and a means for people to claim work items must be provided. A notifications strategy and function are often necessary to alert users about a task or escalate the urgency of the task. Finally, in the anticipation of changes in the business, constructing the human interaction in such a way which allows for flexibility and an easy change can be difficult. The time spent on human interaction can become a major portion of the time spent on integration, shifting focus from the real business problems that must be solved.

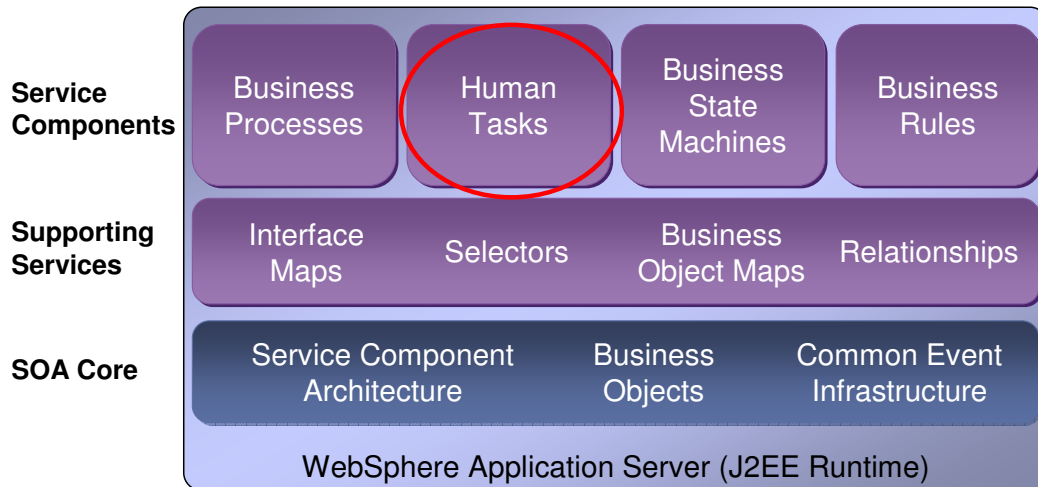
## Overview of Human Task Support

- Use Human Tasks within business integration applications following a service-oriented approach, either independently or within BPEL business processes
- Create and start tasks for different individuals registered in LDAP, Local OS, or Custom Registries
- Specify different levels of authority for the tasks
- Transfer and suspend tasks 
- Priority and due date attributes available on tasks 
- Enable expiration, escalation and notification 
- Build custom Web interfaces for tasks 



The Human task support in WebSphere Process Server and WebSphere Integration Developer solves many of these difficult problems with introducing human interaction into integration applications. Human tasks can be created as independent, separate service components outside of a BPEL process. This allows the human task to become a reusable task without the requirement of BPEL when the features of a business process are not needed. The tight integration of human interaction within a BPEL process is still supported for implementing more complicated business processes. Different user registries can be used as a source of information about who should receive tasks and different levels of authority can be specified, limiting access to tasks. Tasks can be transferred from one person to another. V6.0 includes new support that allows you to suspend work items and resume them at a later time. You will also find support for priority and due date attributes available for tracking work items. Other new capabilities include expiration, configuration of escalation chains based on expirations, and notifications. You can also now create custom Web interfaces for working with tasks.

## Fitting into WebSphere Process Server V6



Support for human tasks builds upon support for Service Oriented Architecture or SOA core components and is a separate component from Process Choreography support, allowing it to be used outside of a BPEL process. Because it is built upon the core capabilities, it is an SCA component and will use Business Objects in terms of the information a person will work with in the task. Human tasks can also be monitored using the Common Event Infrastructure that is also part of that SOA core set of components.

## Different Type of Human Tasks

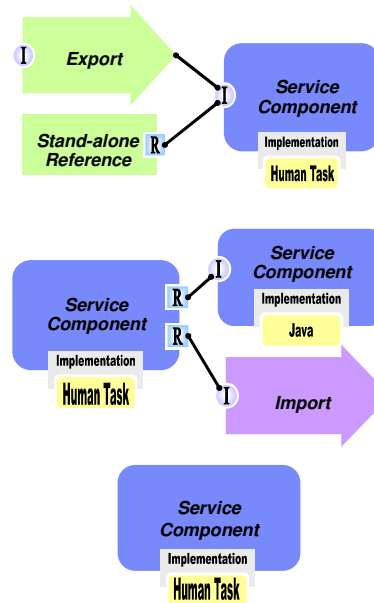
- Different type of Human Tasks available for different integration situations
- Participating Task
  - ▶ Service creates a work item for Human interaction (WSBPEL)
- Originating Task
  - ▶ Human interaction invokes a Service (Business State Machine for example)
- Pure Human Task
  - ▶ Human interaction invokes a Service which creates a work item for another Human



There are several types of human tasks which can be used in WebSphere Process Server V6.0. The type of human task that you use depends on your integration situation and how you want human interaction to be introduced or used within your application. Participating Task can be used when an application or system is going to assign work to an individual. This is very similar to the Staff Activity supported in business processes in WebSphere Business Integration Server Foundation V5.1 where you have an automated work flow process and at some point you assign work to an individual. After the person completes the task, control returns to the work flow and processing continues. Originating Task can be used when a person is going to assign work to some type of application or system. When the task is complete, a response is returned providing a record of the task and the completion status. This could be used when you are calling a business state machine and updating the state. It might also be used to start a BPEL process where you are starting a business process. A human initiates the process and the machine or service called is a BPEL process. The service could also be any SCA component. The final human task is a pure human task where there is no SCA invocation going out or coming in to this task, but you can still take advantage of the human task manager and other escalation and notification capabilities. Using this type of task, an application could be created where a manager could generate "to dos" for his/her department. After tasks are completed, the manager can view the results for each task.

## Fitting into SCA programming model

- Participating Task
  - ▶ Interface invoked through SCA, JMS, Web Service, or Java
- Originating Task
  - ▶ Reference wired to either a service component or Import (not both)
- Pure Human Task
  - ▶ No interfaces or references

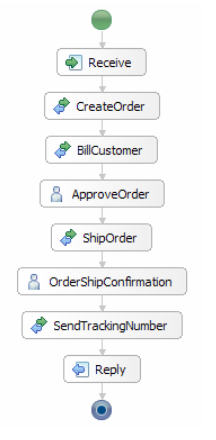


Human Task support is built on the SOA core components of SCA and Business Objects. With this architecture, human tasks follow a consistent programming and more specifically packaging model along with other service components. For the different human tasks, the bindings for the interfaces must be generated and any references resolved. For a Participating Task, which is invoked by another service or application, the bindings for the interface of the task must be generated. The different choices are SCA, JMS, or Web Service which allow different clients to utilize the human task support. For pure Java clients, a stand-alone reference can be created and invoked through a WSDL or Java port-type. A Participating Task has no references to resolve as the implementation is a “person”. For Originating tasks which are started by a person, there is no interface which needs a binding generated. Instead a reference must be resolved to the service or application which will perform the “task”. The reference could be to another component in the module or to an Import component which is bind to another service component through an SCA binding, a Web Service, or a message queue using JMS. For a Pure Human Task, because you have humans calling or starting the task and a human completing the task, there is no interface binding to generate or references to be resolved. Because a pure human task is running on WebSphere Process Server, it still must follow the packaging model and be packaged as a service component.

## Human Tasks and Business Processes

- Business Process may have tight integration with human tasks
  - ▶ Human task activity “in-lined” into business process
- Business Process provide a context which can be used between different human tasks and activities to support more robust situations
  - ▶ Multiple levels of approval (first line and second line managers)
  - ▶ Separation of Roles (“4-eyes”)
- Architectural decision on how tightly coupled human interaction is with a business process
  - ▶ In-lined tasks provide more capabilities, but are less flexible.

### Business Process Instance Context

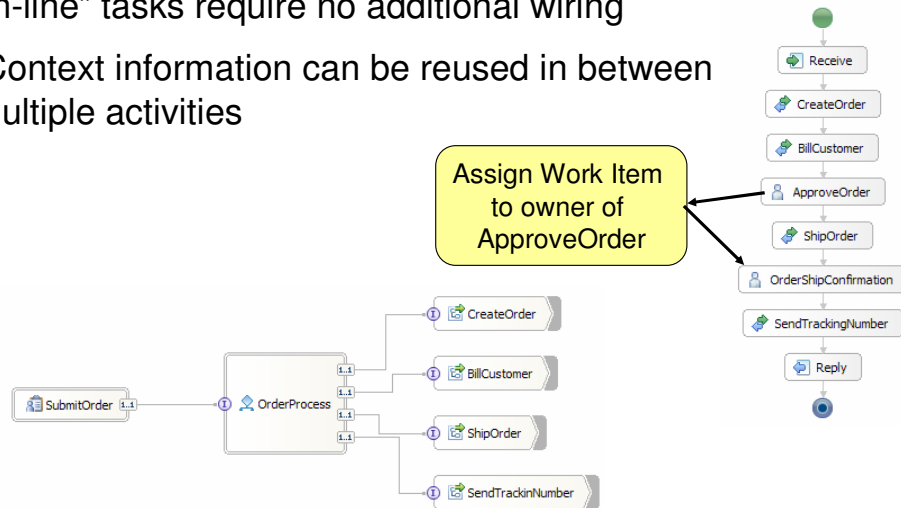


Human tasks can also be used directly inside a business process, which is identical to the support provided in WebSphere Business Integration Server Foundation V5.1. Human tasks in business processes are not separate service components and therefore do not need to be resolved through a reference. Instead, a human task in a business process is implicitly wired into the business process and known as an “inline” human task. This tight integration provides the support for more robust solutions involving human tasks. Business processes have a context that is created for each instance. This context contains the state of the business process, such as which activities have completed and who may have completed an activity. This information can be used by different human tasks in the business process in order to create situations where you want the same person to perform multiple operations, multiple levels of approval or multiple approvals in a group otherwise known as separation of Roles or “4-eyes”. For example, a business process might be used to track the completion of filling and shipping a customer order. In this business process there are two human tasks, one for approving the order contents and another task for confirming the order is shipped. In order for the same person to be assigned the task of confirming the order, the owner of the first task must be used. The business process context contains this information and can be used. The question on whether to use a human task inline in a business process or to use a human task that is a separate service component comes down to a decision between functionality or flexibility. Inline human tasks offer more robust features. However they are difficult to change as they are part of the business process. Separate service component Human Tasks are not as robust as they do not have access to the business process context. However they can be updated easily as they are independent.



## Using “in-line” tasks

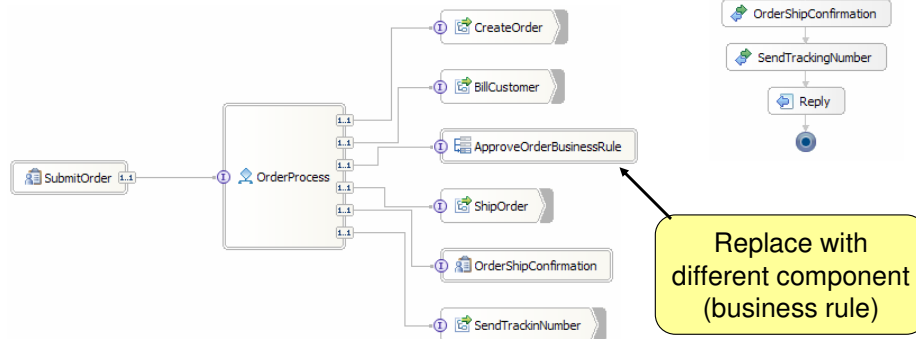
- “in-line” tasks require no additional wiring
- Context information can be reused in between multiple activities



To further emphasize the differences in this important concept, consider the assembly of a business process which has an in-line task. At assembly time there is no reference to be resolved for the inline human task. Unlike the invoke activities of the business process which have references that must be resolved, the human tasks, named approve-order and order-ship-confirmation, do not. The human tasks are already implicitly wired into the business process. With the human tasks inside the business process, they can take advantage of the business process context and the same individual could be assigned to both tasks. While this does provide robust functionality, it is more difficult to update the tasks as they are part of the business process.

## Using Human Tasks as Components

- Human tasks described as components can be wired into business processes
- Allows for components to be changed more easily



If the human tasks in the order process are replaced with invoke activities, the assembly of the business process changes. The references for the invoke activities must now be resolved. The references can be wired to Human Tasks components in the same module or they can be wired to import components which are bound to human components in another module. With the human tasks outside the business process, the reference could be easily updated to point to another service component such as a business rule rather than a human task. This implementation provides a great deal of flexibility. However, the two tasks would not be able to share information to easily perform things such as assigning the task to the same person.

## Agenda

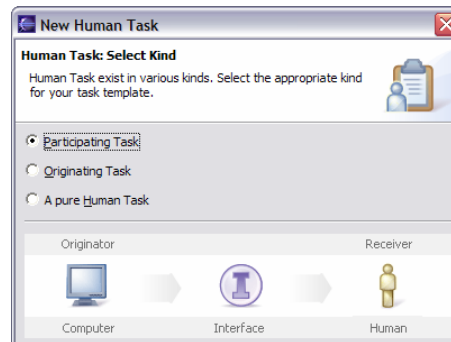
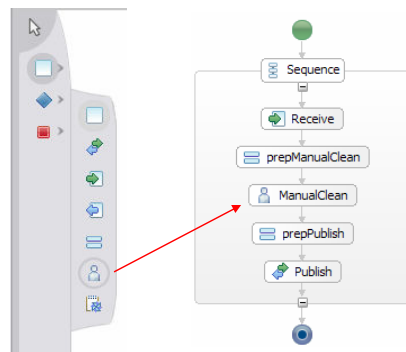
- Overview and Architecture
- **Developing and Administrating Human Tasks**
- Summary and References



This section will briefly cover the development and administration of human tasks.

## Creating Human Tasks

- Specific wizard for creating Participating, Originating, and Pure Human tasks
  - ▶ Directly uses SCA and Business Objects



- Special activity for adding a Human Task to a WSBPEL business process (inline task)
  - ▶ More capabilities over calling a Participating task

There is now a specialized wizard for creating human tasks outside the Business Process Editor. All three types of human tasks can be created using this wizard. When you create these human tasks, you are creating a service component that is separate from a business process. If you want to use a human task directly in a business process, there is a specific activity that can be used. The icon is available along with the other activities and can simply be dragged from the palette onto the editor canvas. These are known as inline human tasks. In reality, the definition of an inline human task is a participating task that is implicitly wired into the business process with no SCA assembly necessary.

IBM Software Group IBM

## Human Task Editor

New V6

- **General Properties**
  - ▶ Staff Plug-in Provider
  - ▶ Restrict transfer abilities
  - ▶ Calendar
- **Permission settings**
  - ▶ Specify Verb and Parameters for Roles
- **Client Interface**
- **Escalations and Notifications**

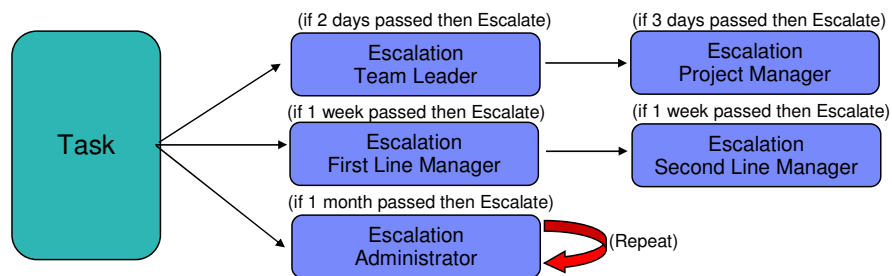
13  
© 2005 IBM Corporation

When you create a task, whether it is a participating, originating, human, or inline task in your BPEL process, you use the Human Task Editor. Clicking the button at the top of the editor will display general settings in the Properties view. Details such as the duration for the task, starting priority, and JNDI name of the staff plug-in provider can be specified here as well. There is also a calendar setting that can be specified, which is used to track the durations for Escalations. In the second section, you can specify permissions for individuals on the work items for the task. There is a variety of permissions to select from and they will vary depending on the task. Most tasks have administrator, potential owner, editor, and reader as available permissions. For each permission, a verb and group of parameters is specified to indicate the individuals to be assigned the appropriate permission. The client interface section allows you to specify custom JSP or JSF files that you have created for working with these items and displaying them in the BPC explorer. The Escalations settings section allows for different escalation chains to be specified based on the state of the task.

## Escalation Support

New  
V6

- Escalation may be set on tasks (participating, originating, pure human and in-line) for different states of the task
- Single or multiple chains of escalations can be defined
- Time and state of task are checked before escalation
- Used to create notifications for overdue tasks
  - Work items, e-mails, custom programmatic events

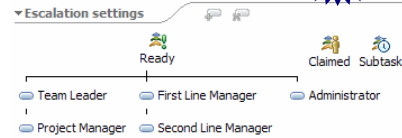


Escalation support provides for a human task to pass through various states as it progresses towards completion. At each of these states, you can specify an escalation chain to begin. When the specified duration expires for an escalation, a check is performed to determine if the appropriate state has been met. If the appropriate state has not been met, a notification is performed and the escalation chain can continue on for another duration. If the appropriate state has still not been met after that duration, another notification will occur. The example shown here consists of a human task with three escalation chains. The first chain checks to see if the task is completed after two days. If it has not, it will be escalated to the team leader and notification will be performed. The next check is performed three days after the first duration expires, or five days after task creation. If the task has not completed, the task is escalated to the project manager. The second escalation chain starts at the same time as the first escalation chain and performs a check after one week. If the task has not completed, it is escalated to the First Line Manager. Another check is performed one more week after the first duration expires, or two full weeks, and if the task has not completed it is escalated to the Second Line Manager. The third chain can also run in parallel and check for completion after one month. If the task has not completed, the administrator is notified. This escalation can also be set up to repeat multiple months.

## Setting Escalations and Notifications

New  
V6

- Assignment of escalation notification uses Verb and Parameter format
- Notification occurs if condition has not been met
- If duration has exceeded and state not reach then escalate



Properties Problems Servers

Escalation

Description

Details

Verb: Users by user ID

Assigns users, given their user ID.  
Supported by sample XSLT files for:  
- LDAP  
- User Registry  
- System

Parameters:

*	Name	Value
*	UserID	manager
	AlternativeID1	
	AlternativeID2	

Expected task state:  Claimed  Subtasks finished  Finished

Duration until escalated: 2hours

Notification type:  Work item  E-mail  Event

Duration until repeated: \_\_\_\_\_

Increase priority:  No  Increase this time only  Increase per repetition

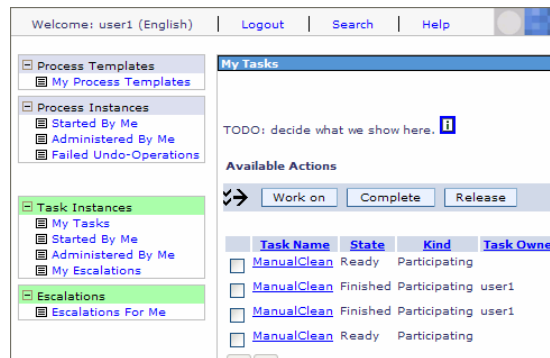
Human Task Overview © 2005 IBM Corporation 15

Shown here is the lower part of the editor, where you can set up escalations. You can set up multiple chains to run in parallel. You can see the various states, indicating when the chain of escalations should begin. As soon as the human task is created, it enters the ready state and timers begin for the escalation chains specified for the ready state. When the human task is claimed, timers will start for any escalation chains that begin at the time the task is claimed. The third state is if any sub tasks are created, escalation chains can be created there as well. You can perform various types of escalations, including assign a work item to a specific person, such as project manager or team leader, send e-mail to a person or group, or invoke a specific event handler to send an SMS or text message. You can also increase the priority when that escalation occurs. If a work item or e-mail notification is used, the person or group of people who should receive notification is set using the same verb and parameter style used to establish the original permissions of the task.

## Administrating Tasks with BPC Explorer



- BPC Explorer sample included for basic task and process administration.
  - ▶ View Tasks and Escalations
  - ▶ Claim, Complete, Transfer, Suspend or Resume Tasks
  - ▶ Located at <http://localhost:9080/bpc>
- Built with reusable JavaServer Faces (JSF) components
  - Starting point for customized clients



Task Name	State	Kind	Task Owner
<input type="checkbox"/> ManualClean	Ready	Participating	
<input type="checkbox"/> ManualClean	Finished	Participating	user1
<input type="checkbox"/> ManualClean	Finished	Participating	user1
<input type="checkbox"/> ManualClean	Ready	Participating	

16

Human Task Overview

© 2005 IBM Corporation

Included with WebSphere Process Server V6.0 is the BPC Explorer. The BPC Explorer is a basic client application that is meant as a sample to demonstrate the types of administrative clients that could be created from managing and administrating human tasks and business processes. In this sample you can perform basic tasks such as claiming, completing, transferring, suspending and resuming tasks. If the definitions of the human task specifies a different JSP to be used on the display of the human task in the client section, they will be used at the appropriate times when you use the BPC explorer. In order to modify the interface of the BPC explorer, you will have to work with your own copy of the application. The BPC Explorer is built using JSF components with a set of tag libraries that you can use to build your own custom clients.



## Agenda

- Overview and Architecture
- Developing and Administrating Human Tasks
- **Summary and References**

Summary and References.

## Summary

- Human Tasks use service-oriented (SCA) design to easily integrate into business applications
- Human Tasks allow for the involvement of people in Web service based applications
- Tight integration with WSBPEL business process provides support for more robust scenario
- Enhanced capabilities for Escalations, Notifications, priorities and due dates



In summary, the human task support fits tightly within the service oriented architecture, is built upon SCA, and utilizes business objects. With the Human Task support you can now easily involve people in Web service applications. You can make human tasks available as separate reusable service components and very quickly and easily wire those into your integration applications in a flexible manner. There is tight integration within the BPEL business processes, allowing you to do more robust things. There are new capabilities related to escalation, notifications, priorities and due dates.

## 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. 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.