



Discovering the Value of SOA WebSphere Process Integration

## WebSphere® Integration Developer V6.0.1.1

### *Best Practices*

*SOA on your terms and our expertise*

@business on demand.

© 2006 IBM Corporation

This presentation will cover best practices for WebSphere Integration Developer V6.0.1.1.

# Agenda

- **WebSphere Integration Developer Best Practice Categories**
  - ▶ **General**
  - ▶ **Development Environment**
  - ▶ **Team Support**
  - ▶ **Process Debuggers**
  - ▶ **.NET Interoperability**
  - ▶ **Business State Machine**
  - ▶ **Security**
  - ▶ **Integration Test Client**
  - ▶ **Selector**
  - ▶ **Business Rule**
  - ▶ **Business Objects**
  - ▶ **BPEL**
  - ▶ **Mediations**

2

SOA on your terms and our expertise

**ON DEMAND BUSINESS**  
© 2006 IBM Corporation

WebSphere Integration Developer best practices will be discussed as they apply to the topics shown here. It is assumed you have experience using WebSphere Integration Developer to develop applications to run on WebSphere Process Server.

**The best practices covered  
in this presentation are based on  
and were validated with  
WebSphere Integration Developer V6.0.1.1**

3

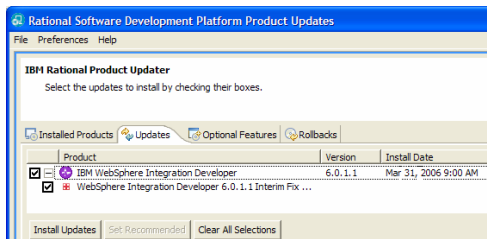
| SOA on your terms and our expertise

**ON DEMAND BUSINESS™**  
© 2006 IBM Corporation

The best practices discussed in this presentation apply specifically to WebSphere Integration Developer V6.0.1.1.

## General

- Upgrade WebSphere Integration Developer to the latest service level
  - ▶ Download fixes from WebSphere Integration Developer Support site:  
<http://www-306.ibm.com/software/integration/wid/support/>
  - ▶ ALWAYS! use Rational® Product Updater
    - If you shell-share Rational Application Developer and WebSphere Integration Developer you must use Rational Product Updater
- Upgrade WebSphere Process Server Universal Test Client to the latest service level
  - ▶ Download minor fixes from IBM WebSphere Process Server Support site:  
<http://www-306.ibm.com/software/integration/wps/support/>
  - ▶ Check for WebSphere Process Server approved WebSphere Application Server fixes from WebSphere Process Server Support site:  
<http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&uid=swg27006735>
  - ▶ Download WebSphere Application Server 6.0.2 fixes from WebSphere Application Server Support Site  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27006876>
  - ▶ Upgrade not needed when WebSphere Integration Developer update with a new third version digit is released
    - It includes new version of Process and Application servers
    - For example, WebSphere Integration Developer 6.0.2 will include a new “matching” version of WebSphere Process Server and WebSphere Application Server



4

SOA on your terms and our expertise

**ON DEMAND BUSINESS**  
© 2006 IBM Corporation

It is important that you check for available fixes frequently and keep WebSphere Integration Developer up to date by downloading the most current fixes from the support site. You should always use the Rational Product Updater and if you shell share with Rational Application Developer or Rational Software architect and WebSphere Integration Developer, you must use the Rational Product Updater. You should also update the WebSphere Process Server test server at the same time you update WebSphere Integration Developer. Keep in mind that WebSphere Process Server is based on WebSphere Application Server, so you should be aware of the fixes available and approved for that product that are approved for use with WebSphere Process Server as well. Whenever a new third version digit is released for WebSphere Integration Developer, the latest WebSphere Process Server fixes will be packaged with it.

## General

- Do not use the default path to install WebSphere Integration Developer, choose the shortest possible path (for example C:\W)
  - ▶ Also use the shortest possible path for server profile directory and workspace paths
- Minimize component name lengths
  - ▶ Having component with long names, results in deployment failure on Microsoft® Windows® operating system due to exceeding the path length limit
- Modules and Shared Libraries
  - ▶ After deployment, if shared resources change in the library, ALL modules using the resources have to be re-deployed

You should always use the shortest possible installation path for WebSphere Integration Developer, especially when installing on Windows® platforms. You should also use the shortest path possible for the server profile directory and workspace. Because there is a limit to the path length, you should keep component names short as well. Shared libraries are used to store static items such as WSDL and business objects. Shared libraries are included in each module, so if you change any one of the shared libraries, you must republish all the modules.

## General

- Do not put any user logic into generated EJB™ and Web modules
  - ▶ Project > Clean action will delete them!
- Perform a clean on the BI module project build before exporting as an EAR
  - ▶ It's always good to perform a clean build before exporting your application from WebSphere Integration Developer to ensure that all the projects are in sync
- When defining Interfaces, wrap primitives in business objects
  - ▶ Wrapping all arguments as business objects will result in the data types being handled correctly by SCA

You should not put any user logic in the EJB and Web modules that are generated by WebSphere Integration Developer unless it is absolutely necessary, as they will be deleted by the clean action. These generated modules are only visible after deployment to WebSphere Process Server. You should always perform a clean action just prior to publishing the EAR for the J2EE application generated by WebSphere Integration Developer in order to ensure a clean build that reflects the current state of the source. In addition, primitives should always be wrapped in business objects. For instance, if you are doing top-down development and you want to define your interfaces so you have a WSDL and WSDL has to have messages, in your WSDL you could define primitives such as INT and LONG and STRING. However, this practice is highly discouraged. Instead, you should always create a wrapped entity such as a business object to wrap these primitives. This is due to the fact that SCA is much better at handling business objects than it is at handling primitive types at the WSDL level.

## Development Environment *Partition your modules*

- Recommended Project Environment:
  - Code ownership at **Module** level
  - Each module must have a **Shared Library** that describe the modules public interfaces
  - Automated project builds of all modules
    - This generates one **EAR** per **Module**

- Recommended WebSphere Integration Developer development environment:
  - Developers install only the **EARs** (generated by automated builds) and the associated **Shared Libraries** that their **Modules** depends on.
    - For example, Owner of Module **B** installs only **EAR EApp** and **DApp**
  - Developer checks out from SCM own source code and the dependent libraries.
    - For example, Owner of **Module B** checks out code for **Module B** and **Shared Libraries** for **Module E** and **D**

The image shows two screenshots. The left screenshot is from the 'Enterprise Applications' console, displaying a table of installed applications. A callout box points to the 'DApp' and 'EApp' rows, stating 'Owner of Module B installs EAR DApp and EApp'. The right screenshot is a 'References' window showing a module dependency graph. A callout box points to nodes B, E, and D, stating 'Module B's Module dependency graph'. The graph shows Module B depending on Modules E and D, while Modules C, G, and F have other dependencies.

7

SOA on your terms and our expertise

**ON DEMAND BUSINESS**  
© 2006 IBM Corporation

Suggestions for the development environment are presented in this slide.

Application development in WebSphere Integration Developer naturally results in a large number of EAR files, or modules. If you are working in a team development environment on a large application, you will no doubt need to give some thought to partitioning. This is similar to the days of C and C++ when include files, DLLs, XC, and the source. Similarly, you must make sure you establish ownership at the module level, for example, with a developer designated as the owner of one or more modules. Each module should have a shared library that describes how to call components in these modules. This is similar to a DLL and an HPT file. Then you conduct automated builds of the entire project on a build server, which generates one EAR for each module so that the administrative console here shows a BApp, DApp, and EApp as the EARs generated from modules. Once the build generates all the possible, runnable modules, a developer that is responsible for a single module (in this case module B) checks out from the source system his own source for B. He also checks out the libraries that define how to access modules in module E and D, but because he does not own modules E and D, he never checks out the source code for those modules. He must copy the EARs for DApp and EApp and install those into his own WebSphere Process Server. The design point is that a developer who works on an application with hundreds of modules does not need to compile all the modules whenever they are developing, otherwise the build times would be very long.

## Development Environment Large Projects

- What if even if you “partitioned your modules” the workspace is still very large?
- Consider using two machines, one for WebSphere Integration Developer and one for WebSphere Process Server:
  1. WebSphere Integration Developer box - 2.5 GB of RAM Minimum for
    - WebSphere Process Server UTE points to the remote WebSphere Process Server box
  2. WebSphere Process Server box - 1.0 GB of RAM



8

SOA on your terms and our expertise

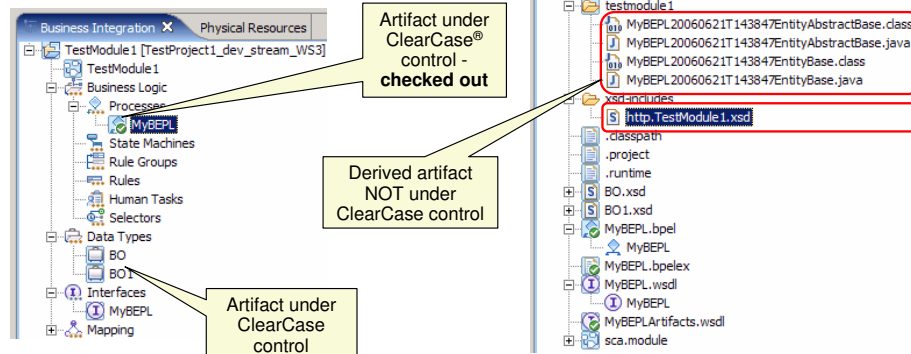
**ON DEMAND BUSINESS**  
© 2006 IBM Corporation

If your workspace is still very large, even after partitioning your modules, you might want to try using a single machine (with 2.5 GB RAM) for WebSphere Integration Developer and a remote WebSphere Process Server for the EARs that you need to invoke.



## Team Support

- Do not check in any generated artifacts into a team repository
- Check-in or Check-out only the files in the Business Integration view
  - ▶ Prior to V6.0.1.1 fix pack 5 some derived files were marked as not derived
  - ▶ In V6.0.1.2 all derived files are marked correctly



9

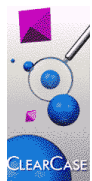
SOA on your terms and our expertise

ON DEMAND BUSINESS  
© 2006 IBM Corporation

Do not check any generated artifacts into a team repository. Only check in/out files in the Business Integration view. You should only check in and out files under ClearCase control, shown in the screen capture with a green checkbox. This will make things simpler, especially if you install V6.0.1.2, which does a better job of marking files correctly.

## Team Support

- Avoid concurrent modifications
  - Optimistic setting for Team Environment is not recommended unless you are very familiar with the artifact files at the text level (for example, XSD for BOs).
- Setup repositories to avoid concurrent modification of files
  - **ClearCase**
    - May be easily setup in exclusive checkout mode
    - For WebSphere Integration Developer development use these CC settings:
      - Use the **Base** model rather than UCM
        - To avoid the need for integration of views into main integration stream
      - When using the Snapshot view model always check-out first then disconnect
        - So that no one else can modify your files
  - **CVS**
    - Exclusive setup mode is against CVS design philosophy
      - You may achieve it using lock/unlock
        - "cvs admin -l" to lock it and "cvs admin -u" to unlock it
    - Set CVS watch/edit feature to notify users of file check-outs
    - More info of CVS with WebSphere Integration Developer:
      - [http://www-128.ibm.com/developerworks/websphere/library/techarticles/0604\\_beers/0604\\_beers.html](http://www-128.ibm.com/developerworks/websphere/library/techarticles/0604_beers/0604_beers.html)



Optimistic setting for team environments is not recommended unless you are interested in reconciling XSDs and BOs, which is digging into XML files. You can set up your repositories to avoid concurrent modification by following the recommendations shown here.

ClearCase is designed to be pessimistic and is easy to set up for exclusive checkout. For example, for WebSphere Integration Developer development, use the base rather than the UCM model to avoid multiple streams. When using the Snapshot view model always check-out first then disconnect.

CVS was designed to be optimistic, so if you choose to use it you should have exclusive ownership of files so that one user cannot check out another user's files and modify them. You should not use the lock/unlock feature to achieve exclusive checkout.

## Process Debuggers

- Start the server in the debug mode before you use the Integration Test Client
  - ▶ The debugger requires a couple of seconds to connect to the server and install the breakpoints
  - ▶ Since the Integration Test Client framework does not wait for this to occur, it is fairly easy to run a process before debug initialization has finished

- Ensure the debugger has properly connected to the server
  - ▶ Once all breakpoints have been installed, you will see this line in the Console:

```
[5/12/06 12:33:58:028 EDT] 00000076 SystemOut    O Server is ready for debugging
```

- ▶ If it does not appear the debugger may not have connected due to a Java Debug timeout.
- ▶ Fix this by increasing the values in 'Window>Preferences>Java>Debug' by a factor of ten.

Communication	
Debugger timeout (ms):	300000
Launch timeout (ms):	20000

- Debuggers do not work when security is enabled

If you are testing and debugging at the same time, ensure that the server has started properly and is fully operational before you start using the integration test client. Ensure that the debugger has connected and you see the “Server is ready for debugging” message. Otherwise, you can increase the debugger timeout and launch timeout. Debuggers currently do not work if security is enabled.

# .NET Interoperability with WebSphere Process Server



- When creating web services in WebSphere Integration Developer to be consumed by .NET, use supported data types
  - ▶ Information can be found at : <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/webservice/datatypes.asp>
- When .NET calls a WebSphere Process Server Component using a Web Service binding, and the Required attribute is not checked you will encounter data loss for that attribute
  - ▶ For example: Attribute *UnitInPackage*,
    - In your C# .NET proxy file, an extra variable called *UnitInPackageSpecified* will be created
    - Either delete this variable or set it to true
    - This will ensure that .NET will send the complete data to your WebSphere Process Server application

ASP.NET data type	XML data type	Supported
string	string	Yes
boolean	boolean	Yes
float(single)	float	Yes
double	double	Yes

Attribute in BO

Attribute - UnitInPackage

Name: UnitInPackage

Type: int

Required  Array

C# .NET proxy

```

/// <remarks/>
public int UnitInPackage;

/// <remarks/>
[System.Xml.Serialization.XmlIgnoreAttribute()]
public bool UnitInPackageSpecified;
    
```

When creating Web services in WebSphere Integration Developer to be consumed by .NET, use only supported data types.

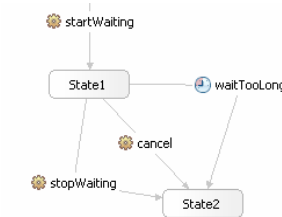
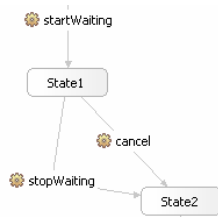
To call a WebSphere Process Server component from .NET using a Web service binding, you must create a WSDL for the WebSphere Process Server component and a proxy on the .NET side and consume that Web service. Always use the “required” check box in the BO editor when invoking a process from .NET.

## Business State Machine

- Conditions are only meant to return true or false
  - While computing the return value in a Java or visual snippet, do not update the value of a variable (declared in the Variables list) as a side effect

```
isCoinValid
Implementation:  Visual  Java  Invoke  Invert result
double coin = deposit_input_coin.getDouble("value");
boolean valid = (coin == 1.0d || coin == 2.0d || coin == 0.25d || coin == 0.1d || coin == 0.05d);
return valid;
```

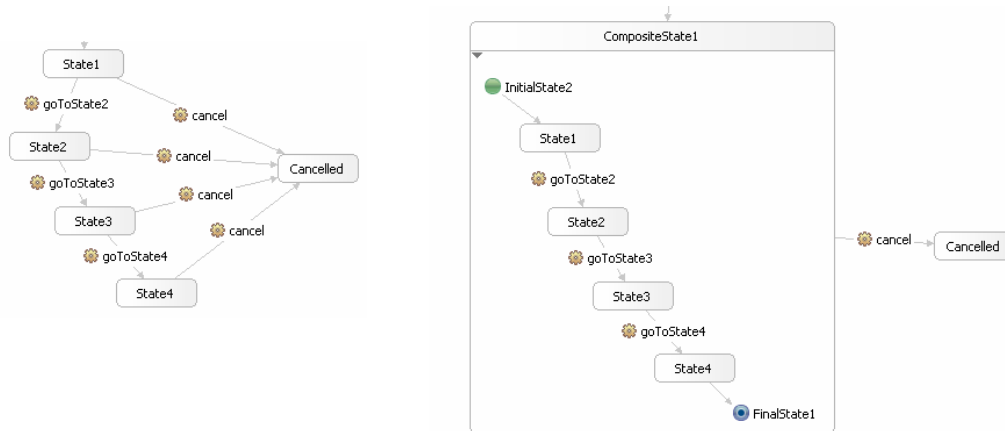
- Avoid a potential deadlock situation: use a timeout
  - In case events *cancel* and *stopWaiting* never fires
  - The time *waitTooLong* will fire and break the deadlock



When creating conditions, you should not use extraneous logic in that code. Instead, conditions should simply calculate the condition and return true or false. Deadlock situations are extremely difficult to debug because the symptom is that nothing happens. For this reason, you should avoid potential deadlocks by using a timeout whenever an event is fired. In the example shown here, *cancel* and *stopWaiting* does not fire the deadlock.

## Business State Machine

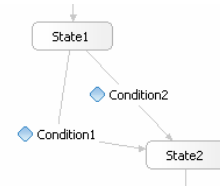
- Use a composite state to reduce the number of transitions needed
  - ▶ When the same event transitions to the same target state from several source states:
  - ▶ Use the *Composite State*



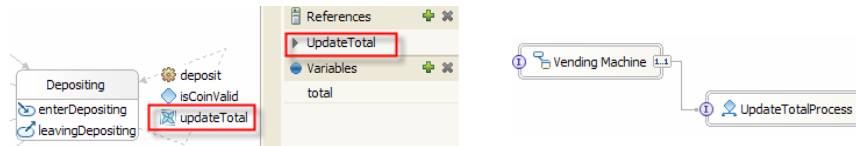
You should always use the composite state to reduce the number of transitions needed. Otherwise, you will do a lot of useless and extraneous coding. In the case shown here, all the states utilize the cancel event, so you should make a composite state and code the cancel event only once.

# Business State Machine

- Avoid a potential deadlock situation: make sure at least one condition will return true.
  - ▶ State2 may never be reached if Condition1 and Condition2 both evaluate to false



- Faults raised by invoking partner links in ( ) cannot be easily handled inside the BSM
  - ▶ If a partner link call raises faults create BPEL component to invoke the partner link and handle the faults
  - ▶ Invoke the BPEL component from BSM



15

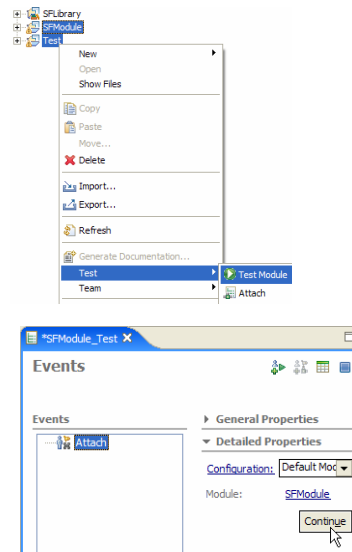
SOA on your terms and our expertise

**ON DEMAND BUSINESS**  
© 2006 IBM Corporation

Guards or conditions will prevent you from changing states and can result in deadlocks. For this reason, you should ensure that all your conditions cannot evaluate to true at the same time. By the same token, at least one of the conditions must return true. Because business state machines do not handle exceptions, you should not invoke one partnerlink from another, because if that partnerlink causes an exception, this will cause problems. One interesting pattern is to code that partnerlink in the BPEL process, which is capable of handling exceptions and acting on them, and then invoke the business process from the business state machine rather than from the partnerlink.

## Integration Test Client

- When testing more than one module, open multiple modules with a single Integration Test Client
  - ▶ Selecting all of the modules
  - ▶ Each module can be deployed on a different server if required
- When testing a module that is invoked by an external client or asynchronously, use Attach
  - ▶ Select the module and use the [Module] > Test > Attach
  - ▶ Click the Continue button to make the module ready for monitoring

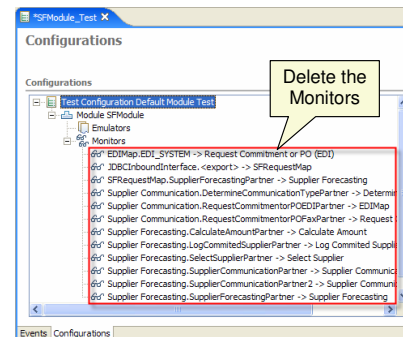
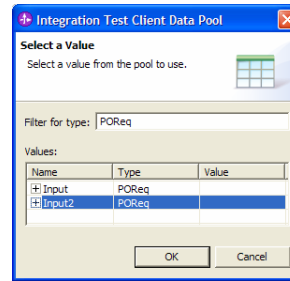


When using the Integration Test Client, you can select several modules to test at the same time. This is helpful in the case where one module calls another, because all modules will be loaded and ready for interaction with each other. Use “attach” when testing a module that is invoked by an external asynchronous event, such as a BPEL process waiting for something. “Attach” indicates that the module is ready for testing whenever something occurs that causes it to start.



# Integration Test Client

- Share the Data Pool
  - ▶ Data Pools save initialized input variables
  - ▶ Data Pools are local to each workspace
  - ▶ To “share” a Data Pool copy it between workspaces:
    - Directory:  
*./metadata/.plugins/com.ibm.wbit.comptest.ui*
    - File: *testclient.objectpool*.
- Using Integration Test Client for performance benchmarking of your component
  - ▶ Delete all Monitors
  - ▶ In one case a Web Service (with 8000 element XSD) performance was measured:
    - With Monitors: 45sec
    - After deleting Monitors: 230ms



Data pools allow you to save and share data when testing very complex objects, such as a large Business Object that serves as input for a BPEL process. You can copy data pools in order to share data across multiple workspaces. You should not use the Integration Test Client for performance benchmarking of components. If you do choose to do so, you should delete the monitors in order to eliminate the performance overhead created by them. Keep in mind however, that you will no longer be able to see the data collected by the monitor.

# Security

**Server**  
Enter settings for the server.

WebSphere profile name:

Update server status interval (in milliseconds):

Server connection type and admin port

RMI (Better performance)  
ORB bootstrap port: 2809

SOAP (More firewall compatible)  
SOAP connector port:

**Security**  
Enable and setup security.

Security is enabled on this server

Current active authentication settings:

User ID:

Password:

- WebSphere Process Server UTC considerations
  - ▶ Server connection type must be set to SOAP
  - ▶ Set security credentials for SOAP:
    - <INSTALL\_ROOT>pf\wps\properties\soap.client.props
  - ▶ Console view will not work; to re-enable Console view
    - wsadmin -profile redeployFileTransfer.jacl -lang jacl -c "fileTransferAuthenticationOn widCell widNode server1" -user <userid> -password <password>

18 | SOA on your terms and our expertise | **ON DEMAND BUSINESS**  
© 2006 IBM Corporation

In order to use WebSphere Integration Developer native security with the UTC, the server connection type must be set to SOAP and security credentials must be specified in a separate file. You will also need to run the wsadmin command shown here to get the administrative console to work after configuring and enabling security.

# Security

- WebSphere Process Server setup
  - ▶ Configuring the J2C Authentication data

[Global security](#) > [J2EE Connector Architecture \(J2C\) authentication data entries](#)

Specifies a list of user IDs and passwords for Java 2 connector security to use.

Preferences

Select	Alias	User ID
<input type="checkbox"/>	SCA Auth Alias	admin
<input type="checkbox"/>	widCell/BPEAuthDataAlias/MS_widNode_server1	admin
<input type="checkbox"/>	widCell/widNode/server1/EventAuthDataAlias/CloudScope	none
<input type="checkbox"/>	widNode/CommonEventInfrastructure/MSAuthAlias	admin

- ▶ Configuring the Task and Process Container Security Information

[Enterprise Applications](#) > [BPEContainer\\_widNode\\_server1](#) > [Map security roles to users/groups](#)

Map security roles to users/groups

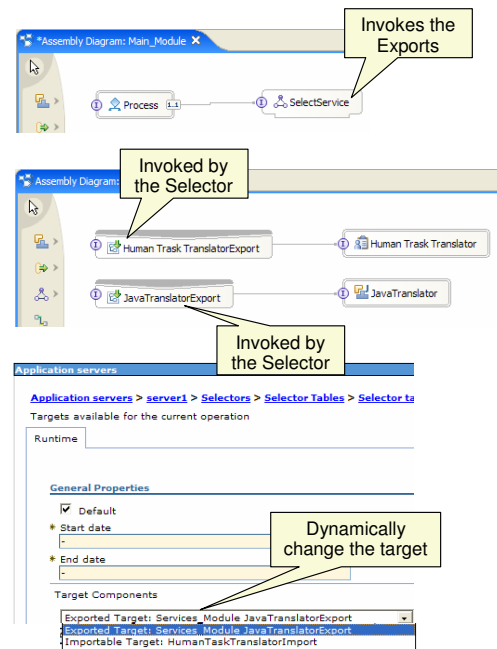
Each role that is defined in the application or module must map to a user or group from the domain user registry.

Select	Role	Everyone?	All authenticated?	Mapped users	Mapped groups
<input type="checkbox"/>	BPESystemAdministrator	<input type="checkbox"/>	<input type="checkbox"/>	cn=admin,ou=admins,o=IBM,c=US	
<input type="checkbox"/>	BPESystemMonitor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	cn=admin,ou=admins,o=IBM,c=US	
<input type="checkbox"/>	WebClientUser	<input type="checkbox"/>	<input checked="" type="checkbox"/>	cn=admin,ou=admins,o=IBM,c=US	
<input type="checkbox"/>	JMSAPIUser	<input type="checkbox"/>	<input checked="" type="checkbox"/>	cn=admin,ou=admins,o=IBM,c=US	

You must configure the J2C authentication alias to utilize the appropriate user and password and the task and process container to utilize that alias.

# Selector

- Key value of Selector is that they invoke the target Components dynamically using the SCA late-binding mechanism
- Use selectors if...
  - ▶ You know or anticipate that the service being invoked will need to change in the future
  - ▶ The target Service must be changed dynamically without redeploying the client component
- Package your selector client component in one module
- Define the selection targets in another module
- Use Administrative Console to dynamically select the target



20

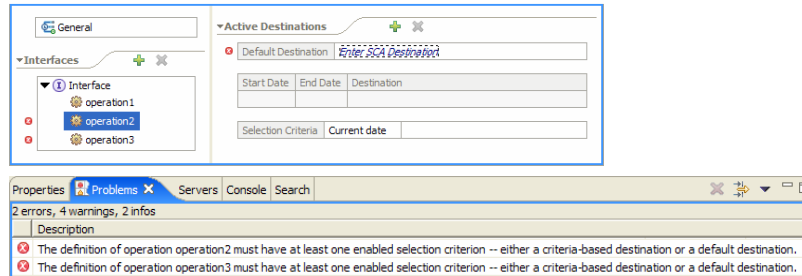
SOA on your terms and our expertise

ON DEMAND BUSINESS  
© 2006 IBM Corporation

The key value of Selector is that they invoke target components dynamically using the SCA late-binding mechanism. Selectors can also invoke based on date or, through skillful programming, based on content, but in reality the SCA late-binding mechanism is the key value of selectors. You should use selectors if you expect that the service will need to change in the future. Selectors are more expensive to invoke, so they should be used for dynamicity, allowing the client to remain unchanged if the target invocation component changes. Package the selector client component in a separate module and define the selection targets in another. If you place everything in a single module, you defeat the point of using a selector, so you should ensure the selector exists in a separate module for easy redeployment. You can use the administrative console to dynamically select the target when you undeploy the old module with the target and deploy a new one.

# Selector

- Selectors and Interfaces with multiple operations
  - ▶ Selectors require that all operations in an interface have at least a default implementation
  - ▶ Example, below is an interface with three operations of which only one is used in the selector



- Top-down development – design it right for selector
  - ▶ Do not group operations in the same interface unless you mean to provide selector support for all of them)
- Bottom-up development - you have no choice
  - ▶ Provide a dummy implementation for operations you do not care about

Selectors require that all operations in an interface have at least a default implementation and each operation must have a target. There are two solutions to meet this requirement. One is to define a “dummy” implementation for operations that you don’t care about, or if you do top-down development, then ensure that you design your interfaces for selectors using only operations that you intend to use.

# Business Rules

- Know the difference between when to use a ruleset and when to use a decision table.
  - ▶ Rulesets provide a sequential set of rules which can be run with multiple results

▼ Rules

Name	Rule1
Template	Template 2
Presentation	The default value for translation is that human performs the translation (t/f): <b>False</b>
Name	Rule2
Template	Template 1
Presentation	If the language is <b>German</b> then translation is performed by a human (t/f): <b>True</b>
Name	Rule3
Template	Template 1
Presentation	If the language is <b>Italian</b> then translation is performed by a human (t/f): <b>True</b>

- ▶ A decision table has multiple conditions which can be evaluated, but only the first one evaluated to true will fire the action.

Conditions		
in.AccountBalance	in.Gold	out
< 3000000	true	true
	false	false
>= 3000000	true	true
	false	true

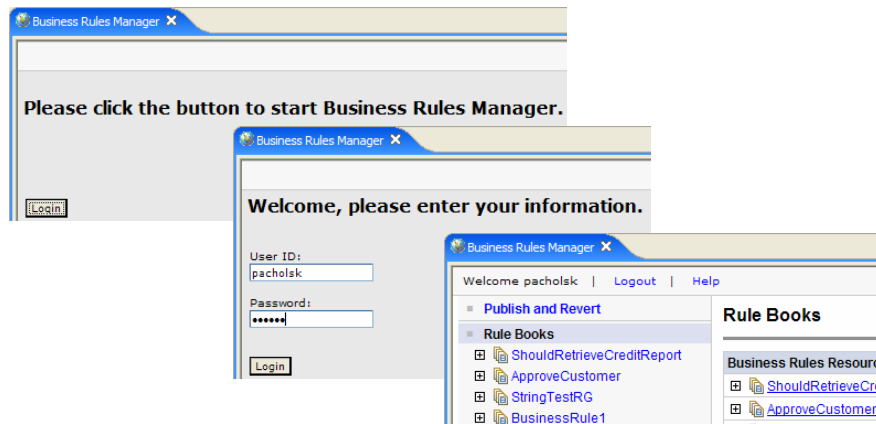
Actions

- ▶ Trying to write a ruleset which will have one outcome may become very complex and difficult to maintain.

The most important thing to understand about business rules is when to use a ruleset and when to use a decision table. Very simply, rulesets provide a sequential set of rules implemented as if statements, which are run sequentially. A decision table consists of multiple conditions, which are evaluated sequentially. However, only the first condition that evaluates to true will fire the action. Decision tables can be thought of as having only one outcome, while rulesets are something more complex. Trying to construct a ruleset with only one outcome could become difficult to maintain, because it would require complex programming logic.

## Business Rules

- If business rules are installed on the system, make sure security is enabled prior to installing the Business Rule Manager
  - Currently the level of security for the Business Rule Manager is tied to J2EE role settings on the Business Rule Manager and with global security not enabled, anyone can modify any business rule



23

SOA on your terms and our expertise

**ON DEMAND BUSINESS**  
© 2006 IBM Corporation

If you want to enable security, and you probably do in your production environment, you must configure J2EE security on the Web project, which represents the business rule manager. Otherwise, everybody will be able to change your rules.

## Business Objects

- After importing XSDs ensure that namespaces are not null and are unique
- Example
  - ▶ BO **M** exists in a module with namespace **http://Same**, and
  - ▶ BO **L** exists in a library with the same namespace
  - ▶ Then BO **L** becomes inaccessible

The screenshot displays the IBM Business Architect interface. On the left, a project tree shows two namespaces: 'http://NameSpaceLibrary' and 'http://Same'. Under 'http://Same', there are two business objects, 'L' and 'M'. A callout box points to these objects with the text: "BOs with the same namespace in different XSD files".

In the center, the 'Interface Editor' shows a table for 'operation1' with input and output parameters. A callout box points to the editor with the text: "Interface Editor shows error; L cannot be selected".

On the right, a dropdown menu is open, showing a list of data types. The business object 'L' is selected, but it is dimmed. A callout box points to it with the text: "Attempt to select L for BO".

At the bottom, a red error message reads: "The namespace http://Same exists in multiple projects; as a result, the selected artifact is not visible from project NameSpaceModule."

24

SOA on your terms and our expertise

ON DEMAND BUSINESS  
© 2006 IBM Corporation

When using business objects, namespaces must be used and must be kept unique. If for instance, two business objects exist in the same namespace, then one will become inaccessible.



## Business Objects

- Beware of *Only permit certain values* (restrictions) checkbox in the BO Editor
  - Using restrictions in data types will lead to build time performance issues

**Attribute - LoanApproved**

Name:

Type:

Required  Array

Minimum length   Collapse whitespace

Maximum length

Only permit certain values

Enumerations

Patterns

Be aware that using the “Only permit certain values” checkbox in the BO editor can result in performance degradation and should only be used when absolutely necessary.

# Business Objects

- Support for "variable" XSD types in the Business Object Editor
  - Supported: xsd:anyType, xsd:choice
    - Possible to access through BO Maps, Java Snippets, etc...
  - Tolerated: xsd:any, xsd:anyAttribute
- Runtime (SCA and SDO) support may be in-doubt

The diagram illustrates the mapping between XSD code and the Business Object Editor. On the left, the XSD code for a complex type named 'BookType' is shown. A yellow box labeled 'Tolerated' points to two specific XSD elements: `<xsd:any maxOccurs="4" />` and `<xsd:anyAttribute/>`. On the right, the BO editor view for 'BookType' is shown, listing various elements and their types. Red boxes highlight 'fiction string', 'bigraphy string', 'history string', and 'Publisher anyType' in the editor, with red dashed arrows pointing from the corresponding XSD code to these elements. The 'Publisher anyType' element in the editor is also highlighted with a red box, corresponding to the 'Tolerated' label in the code.

```
<xsd:complexType name="BookType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="fiction" type="xsd:string"/>
      <xsd:element name="bigraphy" type="xsd:string"/>
      <xsd:element name="history" type="xsd:string"/>
    </xsd:choice>
    <xsd:element name="Title" type="xsd:string"/>
    <xsd:element name="Author" type="xsd:string"/>
    <xsd:element name="Date" type="xsd:string"/>
    <xsd:element name="Publisher" type="xsd:anyType"/>
    <xsd:element name="ISBN" type="xsd:string"/>
    <xsd:any maxOccurs="4" />
  </xsd:sequence>
  <xsd:attribute name="Topic" type="xsd:string"/>
  <xsd:anyAttribute/>
</xsd:complexType>
<xsd:element name="Book" type="bo:BookType"/>
```

Variable XSD types allow you to defer the decision of what the types are until the end. This slide shows which XSD types are supported, and which are tolerated. Choice and anyType are supported XSD types because you can edit them and therefore you can use them in Java snippets. Any and anyAttribute are tolerated XSD types, and the BO editor will not even display them so therefore the question arises of how do you access them programmatically.

## Business Objects

- Use xs:dateTime instead of xs:date
  - ▶ Using a xs:date type may result in unpredictable or undesirable results due to conversion of local to GMT of the date value through a deserialization and serialization process
  - ▶ Differences in the source and server time zones could result in the loss or gain of a day during normalization

```
<?xml version="1.0" encoding="UTF-8"?>
<bo:customerDoc xsi:type="bo:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:bo="http://BOTest">
  <name>Shane</name>
  <dateOfBirth>1970-03-03+02:00</dateOfBirth>
</bo:customerDoc>
```

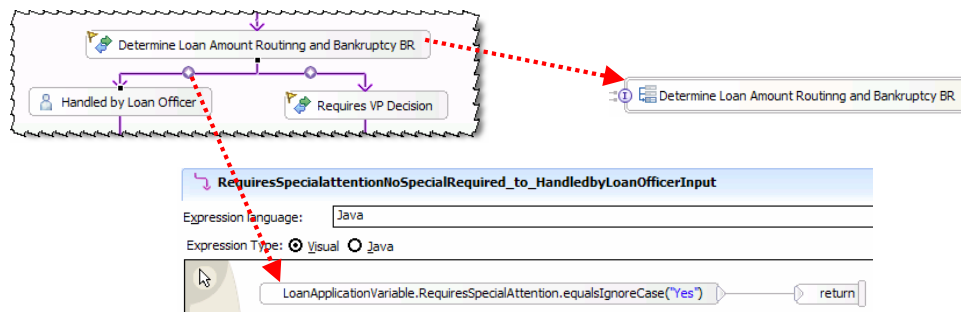
```
<?xml version="1.0" encoding="UTF-8"?>
<bo:customerDoc xsi:type="bo:Customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:bo="http://BOTest">
  <name>Shane</name>
  <dateOfBirth>1970-03-02Z</dateOfBirth>
</bo:customerDoc>
```

- There is a technote and “fix” available:  
[http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&dc=D600&uid=swg21240393&loc=en\\_US&cs=utf-8&lang=en](http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&dc=D600&uid=swg21240393&loc=en_US&cs=utf-8&lang=en)
  - ▶ Read the technote before applying the “fix”; it is not a complete fix and may not apply to your circumstances
  - ▶ The technote describes alternatives and best practices for using date in XSDs

Using xs:date might result in a loss or gain of a day during normalization. There is a fix available for this issue, which can be obtained at the URL shown on this slide. You should, of course, read the technote before applying the fix because it is not a complete fix and might not apply to your circumstances. If at all possible, you should use xs:dateTime instead of xs:date.

## BPEL

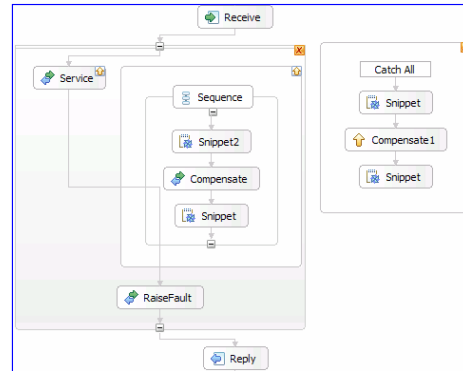
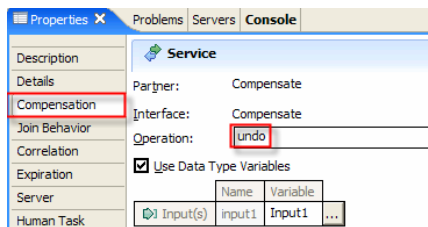
- Use an invoke activity prior to the flow or switch activity.
- Wire the invoke activity to a business rule component.
- In the business rule, define the appropriate logic and for the result return a value
- Use the value in the links or switch to determine navigation.
- This allows a certain amount of logic to be easily modified without having to redeploy the business process.



Whenever you have a decision to make, you should use a business rule to create an activity, which is a business rule that returns very simple things like true or false, or a numerical value, and contains the complex business logic to determine the outcome of the condition. The BPEL process simply acts on these conditions. You should not put business logic in a BPEL process. Specifying business rule business logic externally simplifies the design and allows you to change the values and thresholds dynamically. The screen capture shown here illustrates that point.

# BPEL

- There two choice of Compensation support:
  - ▶ (1) Classic “do-undo-pairs”; (2) Compensation Handlers
- **Classic**
  - ▶ May be used in short and long-running processes
  - ▶ Not standard compliant
  - ▶ Simple to implement
  - ▶ Inflexible - essentially they are a compensation handler with a single Compensate activity
- **Compensation Handlers**
  - ▶ May only be used in long-running processes
  - ▶ BPEL standard compliant
  - ▶ Allow to implement complex compensation algorithms
  - ▶ Allow for partial flow compensating without the need to terminate the process



29

SOA on your terms and our expertise

ON DEMAND BUSINESS  
© 2006 IBM Corporation

The previous release of WebSphere Process Server, called WebSphere Business Integration Server Foundation, provided a form of compensation support referred to as “classic” compensation. In the current release, compensation support is provided in the form of compensation handlers. The classic support existed before the specification, so it is non-compliant and is not standard. It can be used in both short and long-running processes and is very simple to implement, but it is very inflexible, meaning you have to either compensate the entire process or none of it and compensation occurs at the end of the process. Compensation Handlers, on the other hand, are specification compliant and can be used in long running processes and allows you to implement very sophisticated compensation algorithms that can be invoked not when the process is about to terminate, but anywhere in the process while it is running. You can compensate certain activities and then continue on to run the process.

# BPEL

- Use XPath rather than Java for condition logic if:
  - ▶ Really complex, data-driven expressions
  - ▶ For standards-compliance, as the Java extensions are IBM specific
- Use XPath in Assign Statement when and Expression is required
  - ▶ For example, need to create a Business Object

The screenshot displays three panels from a BPEL editor:

- Link2:** Shows an XPath 1.0 expression: `bpws:getVariableData("library_asset", "count(book_instances[status='onShelf' and book_id=bpws:getVariableData('lend_book_request', 'book_id')]) > 0")`. A callout box labeled "XPath Condition" points to the expression.
- Link3:** Shows an equivalent Java condition: 

```
java.util.List list = library_asset.getList("book_instances");
java.util.Iterator listIterator = list.iterator();
int book_id = lend_book_request.getInt("book_id");
int cnt = 0;
while(listIterator.hasNext()){
    DataObject item = (DataObject)listIterator.next();
    int id = item.getInt("book_id");
    String status = item.getString("status");
    if ( book_id == id && status.equalsIgnoreCase("onShelf")) cnt++;
}
return cnt != 0;
```

 A callout box labeled "Equivalent Java Condition" points to the code.
- ReadLibraryAsset:** Shows an assign statement with an XPath 1.0 expression: `bp:create('http://com.ibm/library','library_asset')`. A callout box labeled "XPath Assign Expression" points to the expression.

30

SOA on your terms and our expertise

ON DEMAND BUSINESS  
© 2006 IBM Corporation

BPEL includes support for both XPath and Java for things like assign statements and conditions. You should use XPath rather than Java for condition logic if things are fairly complex and you want to simplify. The screen capture here shows how you can dramatically simplify the logic by using XPath instead of Java. A solid understanding of XPath expressions is required as there is currently no integrated XPath editor. You can use an external XPath editor to create the expression visually, then copy over the data. Use XPath in assign statements when expressions are required that need to create things like Business Objects.

## Mediation Module

- When using the XML transform to map BO's that contain arrays, map only the first entry of the array, and leave the other ones unmapped.

The screenshot displays the configuration of a mediation module. It includes two tables for Business Objects (BOs) and an XSLT Transform configuration window.

Name	Type	Default Value	Min Occurs	Max Occurs
attribute1	string		0	1
attribute2	string		0	1
attribute3	string		0	1
attribute4	double		10	10

Source BO

Name	Type	Default Value	Min Occurs	Max Occurs
attribute1	string		0	1
attribute2	string		0	1
attribute3	string		0	1
attribute4	double		0	-1

Destination BO

The XSLT Transform configuration window shows the mapping between the Source and Target BOs. The Source BO has an array of attribute4 with a fixed size of 10. The Target BO has an array of attribute4 with a variable size of 0..\*. The XSLT Transform maps the first element of the Source array to the first element of the Target array.

31

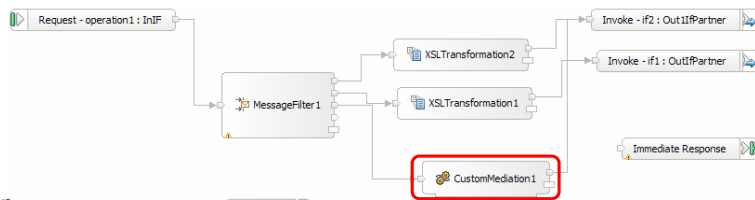
SOA on your terms and our expertise

ON DEMAND BUSINESS  
© 2006 IBM Corporation

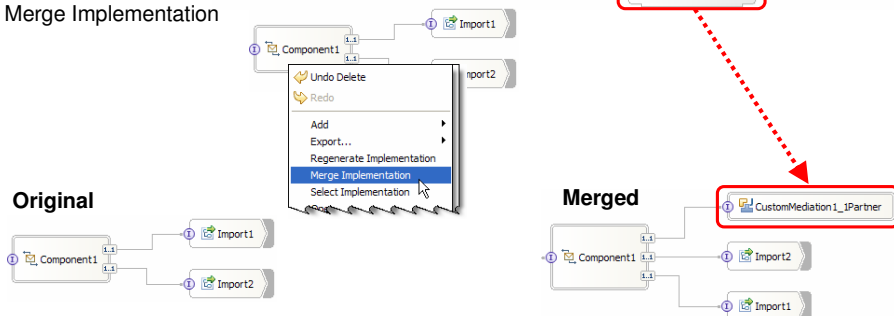
If you are writing a mediation module, you will notice that you don't have BO maps. Instead, you have XSL transports and one particular case that occurs quite often is when you have two arrays that you need to map and one of these arrays contains a min/max specifying the number of elements and the other contains a variable number of elements. In order to map these arrays, as you can see on the right hand side of this slide, the one with unlimited number of elements only has one entry. The workaround is to map the first array with the defined number of entries into the array with the variable elements. This issue should be resolved in the latest fix pack.

## Mediation Module

- After adding a custom mediation you need to do a merge implementation on the Mediation Component
- Added Custom Mediation



- Merge Implementation



If you use a mediation component, such as the mediation between two import statements shown here, you will find that in the middle of your work you created a custom mediation node that is essentially a piece of Java code. In order to add that node back in the assembly editor, you can use the merge implementation choice to add the reference to the custom mediation to the assembly editor.



# 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.