IBM Software Group

WebSphere Enterprise Service Bus V6.2
WebSphere Process Server V6.2
WebSphere Integration Developer V6.2

Endpoint lookup mediation primitive

@business on demand.

This presentation provides a detailed look at the endpoint lookup mediation primitive.

Goals

- Understand the endpoint lookup mediation primitive

    Endpoint lookup

  ▸ Overview of function
  ▸ Use of terminals
  ▸ Definition of properties
  ▸ Service message object (SMO) usage
  ▸ Service registry cache
  ▸ Server administration of the registry
  ▸ Configuration of the registry
  ▸ Error handling
  ▸ Problem determination
  ▸ Example usage

2

© 2009 IBM Corporation

The goal of this presentation is to provide you with a full understanding of the endpoint lookup mediation primitive.

The presentation assumes that you are already familiar with the material presented in the presentations that cover common elements of all mediation primitives, such as properties, terminals, wiring and the use of promoted properties. The general knowledge of mediation primitives they provide is needed to understand the message filter primitive specific material in this presentation.

An overview of the function provided by the endpoint lookup primitive is presented, along with information about the primitive's use of terminals and its properties.

This primitive has a special relationship to the service message object (SMO). The presentation reviews those elements in the SMO context that are specifically used by endpoint lookup primitives.

In order to understand the endpoint lookup primitive, it is necessary to understand more than just the primitive's behavior and how it fits into a mediation flow. The endpoint lookup primitive interfaces with the WebSphere® Service Registry and Repository, which is referred to as the registry in this presentation. Use of the registry is enabled through capabilities provided by the WebSphere Enterprise Service Bus and the WebSphere Process Server, which are referred to as the server in this presentation. You are going to learn about the service registry cache and administration of registries, capabilities that are provided by the server. You are also provided with references to documentation describing how to configure the registry.

The presentation then returns to looking specifically at the endpoint lookup primitive, covering error handling, problem determination and an example of its usage.

Overview of function

- Uses a registry to find service endpoints
  - ▶ Performs the lookup based on selection criteria
  - ▶ Initializes SMO with results for downstream use by the mediation flow

- Numerous criteria can be used for selection
  - ▶ Which registry to use for the lookup
    - Available registries are administratively defined within a WebSphere cell
  - ▶ Specifics of the requested service port type
    - Name, Namespace, Version
  - ▶ Latest compatible version of a versioned SCA module
  - ▶ Associated classification, based on Web Ontology Language (OWL)
  - ▶ Associated properties and property values

3

The endpoint lookup primitive uses a registry to find service provider endpoints based on a set of selection criteria. The results of the lookup are reflected in the service message object which allows them to be used downstream in the mediation flow.

There are many different criteria that can be used for selection. First, multiple registries can be configured for use by the servers within a WebSphere cell, and the endpoint lookup can specify which of these registries should be used. The service port type of the requested service can be qualified based on name, namespace and version. Also, if you are using versioned SCA modules, you can request the latest compatible version to a specified version. The Web Ontology Language (OWL), provides a classification system which can be used as part of the selection criteria. Registered services can be associated with name value pairs which can also be used as part of the selection criteria. These selection criteria reflect the underlying capabilities of the WebSphere Service Registry and Repository.

Overview of function

- Context in SMO contains section for selected endpoints
  - ▶ List of endpoints selected, defining for each endpoint
    - Endpoint reference
    - Properties and property values
    - OWL Classifications
    - Relationships to other registry entities
- SMO header set for dynamic endpoint usage
  - ▶ Setting of the target and alternate target addresses
  - ▶ Controlled by the match policy property
  - ▶ Affects what can be done in subsequent flow

Endpoint lookup mediation primitive © 2009 IBM Corporation

4

The context of the SMO contains a section in which the selected endpoints are placed by the endpoint lookup primitive. For each endpoint there is the endpoint reference, a list of associated properties and property values, the OWL classifications and a list of relationships between the endpoint and other registry entries.

Service invoke primitives and callout nodes can use dynamic endpoints that are set in the SMO header. The setting of these fields, the target and alternate target address fields, is controlled by specification of the match policy property of the endpoint lookup primitive. Enabling the setting of these fields allows the primitive to initialize the SMO, preparing the SMO for dynamic endpoint usage by service invoke primitives and callout nodes.

Overview of function

- Match policy settings: resulting behavior and affect on flow
  - ▶ Return first matching endpoint and set routing target
    - Sets the target address
    - Clears the alternate target addresses
    - SMO ready for service invoke or callout node without alternate address retry
  - ▶ Return endpoint matching latest compatible service version
    - Sets the target address
    - Clears the alternate target addresses
    - SMO ready for service invoke or callout node without alternate address retry
  - ▶ Return all matching endpoints and set alternate routing targets
    - Sets the target address with the first endpoint
    - Sets the alternate target addresses with the remaining endpoints
    - SMO ready for service invoke or callout node including alternate address retry
  - ▶ Return all matching endpoints
    - Does not modify the target or alternate target addresses
    - Subsequent mediation logic must set the target addresses in SMO

The match policy setting determines how the target and alternate target address fields in the SMO header are affected by the endpoint lookup primitive. This has a resulting affect on what can be done downstream in the flow.

The first match policy setting is called return first matching endpoint and set routing target. This policy results in the target address field being set and causes the alternate target address field to be cleared of any addresses it might already contain. At this point, the SMO is ready to be used by a service invoke primitive or callout node configured for dynamic endpoints. However, it is not ready to be used for service call retry with alternate target addresses.
The next policy, called return endpoint matching latest compatible service version, has the identical affect on the SMO as the previous policy, and therefore results in the same downstream behavior. The difference between them is that this policy incorporates the SCA module version into the criteria when making the selection from the registry.

The next match policy setting is called return all matching endpoints and set alternate routing targets. This results in the target address field being set with the first endpoint returned and the alternate target addresses field being set with the other endpoints returned. At this point, the SMO is ready to be used by a service invoke primitive or callout node configured for dynamic endpoints and for service call retry with alternate target addresses.

Finally, there is the match policy setting called return all matching endpoints. In this case, the target address field and alternate target addresses field are not set. Therefore, when using this match policy, downstream processing in the mediation must perform some logic that selects an endpoint and places its address into the target address field. The logic might also set other endpoints in the alternate target addresses field. Specifically what logic is performed depends upon your application requirements.

Terminals

- Terminals:
  - ▸ Input terminal
  - ▸ Two Output terminals
  - ▸ Fail terminal

EndpointLookup

- ⊟ ▷] Input terminal
  - ▸ in
- ⊟ ▷ Output terminal
  - ▸ out
  - ▸ noMatch
- ⊟ ❗ Fail terminal
  - ▢ fail
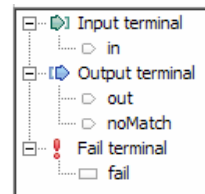
- Output terminals
  - out – if one or more matches is found the updated SMO is propagated
  - noMatch – if no matches are found, the unchanged SMO is propagated

- All terminals must be for the same message type

The endpoint lookup primitive has one input terminal, two output terminals and a fail terminal. There is one output terminal named out used when the endpoint lookup is successful and another output terminal named noMatch used when there was no service endpoint that satisfied the selection criteria. The output terminals must be for the same message type as the input terminal because the endpoint lookup primitive does not modify the message body structure. The slide shows an endpoint lookup primitive with its terminals and the terminals as seen in the properties view.

Properties

Endpoint Lookup : EndpointLookup

Build Activities | Properties | Problems | Server Logs | Servers | Console

Description	
Terminal	Name: Account
Details	Namespace: http://BankAppLibrary/Account
Advanced	Registry Name: BankProductionRegistry
Promotable Properties	Match Policy: Return first matching endpoint and set routing target
	Version: 2.1
	Module:
	Export:

- Registry Name
  - Identifies the registry against which to do the lookup
  - Registries are administratively defined in the server runtime at the cell level
  - Leaving this blank results in use of the registry designated as the default
- Match Policy
  - Return first matching endpoint and set routing target
  - Return endpoint matching latest compatible service version
  - Return all matching endpoints and set alternate routing targets
  - Return all matching endpoints

The properties of the endpoint lookup primitive, which are examined over the next few slides, are the basis for making the selection of the proposed endpoints.

The first property is the Registry Name, which identifies the registry against which lookup is to be done. The WebSphere Enterprise Service Bus and WebSphere Process Server manage registry references administratively, and this is the name the registry is known by in the server. The server designates one registry as the default. It is used if this property is left blank. More details about how the server manages registry references is provided later in this presentation.

The Match Policy property can have any of the four possible settings shown in the slide. A full description of these match policy values and the behavior associated with them was provided earlier in this presentation.

Properties

IBM Software Group

Build Activities | Properties ⊠ | Problems | Server Logs | Servers | Console

Endpoint Lookup : EndpointLookup

Description	
Terminal	**Name:** Account Browse…
Details	**Namespace:** http://BankAppLibrary/Account
Advanced	Registry Name: BankProductionRegistry
Promotable Properties	Match Policy: Return first matching endpoint and set routing target ▼
	Version: 2.1
	Module:
	Export:

- Name and Namespace
  - ▶ Port type name and namespace of endpoint
  - ▶ Browse… button brings up the Interface Selection dialog
    - Selection sets both the name and namespace properties
  - ▶ Name and namespace match interface to be called
    - Same interface name and namespace configured on the reference used by the dynamic callout or service invoke

Endpoint lookup mediation primitive © 2009 IBM Corporation

8
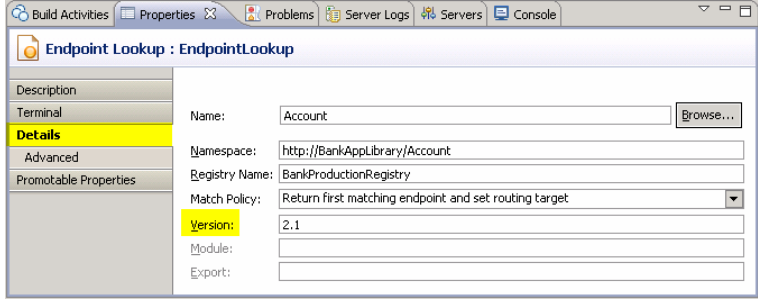
The properties, Name and Namespace, identify the port type of the endpoints to be selected. This defines the interface the service endpoint supports. There is a Browse… button that you can use to bring up the Interface Selection dialog which allows you to select the required interface. This results in the setting of both the name and namespace properties. The name and namespace are the same as those specified for the interface on the mediation flow component reference that is a associated with a dynamic callout node or a service invoke primitive in the flow.

Properties

Endpoint Lookup : EndpointLookup

Description
Terminal
Details
Advanced
Promotable Properties

Name: Account
Namespace: http://BankAppLibrary/Account
Registry Name: BankProductionRegistry
Match Policy: Return first matching endpoint and set routing target
Version: 2.1
Module:
Export:

- Version
    - Usage of depends upon the match policy used
    - For service selection of versioned SCA modules
        - Value is required
        - Usage explained on next slide
    - For all other match policies
        - Value is optional
        - Is a freeform string used to represent the version of the port type
        - Matched with version field in WebSphere Service Registry and Repository

9

Endpoint lookup mediation primitive © 2009 IBM Corporation

How the Version property is used depends upon the value of the match policy. When the match policy is set to the policy return endpoint matching latest compatible service version, the value for this property is required. How it is used in the lookup is explained on the next slide.

For all other match policies, specification of the version property is optional. When specified, it is a freeform string with no defined format or meaning. It is matched with a version specification in WebSphere Service Registry and Repository. Only endpoints with the identical freeform version string are selected.

Properties

Endpoint Lookup : EndpointLookup

Description	
Terminal	Name: Account — Browse...
Details	Namespace: http://BankAppLibrary/Account
Advanced	Registry Name: BankProductionRegistry
Promotable Properties	Match Policy: Return endpoint matching latest compatible service version
	Version: 2.*
	Module: BranchTransactions
	Export: Checking

- Module and Export
  - Specifies the name of the SCA module and export of endpoint
  - Only enabled if using service selection of versioned SCA modules
- Version
  - A dotted numeric version number
  - <version>.<release>.<modification>
    - version – must be an integer
    - release – can be omitted, the wildcard *, or an integer
    - modification – can be omitted, the wildcard *, or an integer

The Module and Export properties identify the name of the SCA module and the export within it that represents the endpoint. These properties apply when the match policy has been set to the policy return endpoint matching latest compatible service version. For all other match policies they are disabled.

The syntax rules for the Version property are dependent upon the match policy setting. When set to the policy return endpoint matching latest compatible service version, the rules are these.

The basic form that the version takes is a dotted numeric version number, in the form version, dot release, dot modification. The version digit must always be specified and must be an integer value. The release digit and the modification digit can be omitted, or they can be the wildcard character asterisk or they can be an integer.

How the specification actually affect the selection is explained later in the presentation.

Properties

Endpoint Lookup : EndpointLookup

Description
Terminal
Details
Advanced
Promotable Properties

Classifications:

http://www.ibm.com/wsrr/governance#Operational

Add...
Edit...
Remove

User Properties:

Name	Type	Value
status	string	productionReady
type	XPath	/headers/JMSHeader/JMSType
cellname	string	

Add...
Edit...
Remove

- Classifications
  - ‣ List of classifications associated with the selected endpoints
  - ‣ Each classification is specified as a URI as defined by the OWL classification system
- User Properties
  - ‣ List of user defined properties associated with the selected endpoints
  - ‣ String with value – selected endpoint must have specified property with specified value
  - ‣ String without value – selected endpoint must have specified property with any value
  - ‣ XPath – selected endpoint must have specified property with the value obtained from the element in the SMO defined by the XPath expression

Endpoint lookup mediation primitive © 2009 IBM Corporation

The endpoint lookup primitive has an Advanced details panel, as is shown here. It contains the Classifications property and the User Properties property.

The Classifications property is a list of classifications that should be associated with the selected endpoint. A classification is specified as a URI which is defined by the OWL classification system.

The User Properties property is a table of name, type and value triplets identifying a name value pair that should be associated with the selected endpoint. The Name column contains the name of a user property. If the Type column contains XPath, then the Value column contains an XPath expression identifying an SMO element that contains the value for the user property. If the Type column contains string, then the Value column contains either the value for that user property or blank. When both Name and Value are present, the selected endpoint must have the specified user property with the specified value. When Value is left blank, then the selected endpoint must have the specified user property but it can have any value.

Promotable properties

IBM Software Group

Endpoint Lookup : EndpointLookup

Description
Terminal
Details
Advanced
Promotable Properties

Filter Property ▾ <Type in the filter string>

Property	Promoted	Group	Alias	Alias value	Description	
Name	☐					
Namespace	☐					
Registry Name	☐					
Match Policy	☐					
Version	☐					
Module	☐					Edit
Export	☐					
http://www.ibm.com/wsrr/governance#Operational [Classification]	☐					
status [Value]	☐					
type [Value]	☐					
cellname [Value]	☐					

- Promotable
  - Classification (individual rows)
  - User Properties (the Value column)
  - Registry Name
  - Match Policy
  - Name
  - Namespace
  - Version
  - Module
  - Export

Endpoint lookup mediation primitive   © 2009 IBM Corporation

12

All of the properties for the endpoint lookup are promotable.

The Classification property is a table with a single column. Individual rows in the table can be promoted.

The User Properties table designates the Value column as promotable, allowing values for individual rows to be promoted.

Promoting the Registry name allows administrators to dynamically manage which registry is being used.

The remaining properties are all promotable, but promoting them and changing their values at runtime has to be done with great care.

Changing the Match Policy typically implies that the mediation flow logic needs to be changed as well. Therefore, for most circumstances, this does not appear to be something that should be dynamically changed.

Changing the port type for the endpoint selected in most cases also implies a change to the mediation flow logic and possibly also to the assembly for the mediation module. Therefore, it is unlikely that Name or Namespace should be dynamically changed.

The changing of version, module or export has the potential to change the port type in a way that is not compatible with the flow. However, in some circumstances it can make sense to dynamically set these values in a way that does not modify the port type.

Versioned SCA module selection

- Version matching rules for SCA module selection
  - Specified as <version>.<release>.<modification>

	Specified as	Specification rules	Selected version
version	integer	required	exact match

release	integer
	omitted
	wildcard (*)
		all choices valid	if integer
			   exact match
			otherwise
			   greatest available

modification	integer
	omitted
	wildcard (*)
		if release is an integer
		   all choices are valid
		otherwise
		   must be omitted
			if integer
			   greatest available larger than integer
			otherwise
			   greatest available

- Examples

Registry Contains
1.0.0 1.1.0 1.3.0 2.0.0 1.0.1 1.1.3 1.3.1 1.0.2 1.1.4 1.3.4 1.0.3

Specified	Selected
1 or 1.*	1.3.4
1.1 or 1.1.* or 1.1.2	1.1.4
1.2 or 1.2.* or 1.2.1	no match
1.0.4	no match

The rules for version are looked at in detail on this slide. As previously stated, the version is specified as a dotted numeric, composed of a version, release and modification. The table at the top of the slide describes the rules for specifying this value. The column entitled, specified as, indicates the valid ways to specify each of the individual digits that make up the overall dotted numeric. The column entitled, specification rules, further qualifies this by describing the specification rules for the digits when related to the other digits of the dotted numeric string. Finally, the selected version column indicates how each digit of the version specification is applied to selection of a versioned module from the registry.

The basic rules can be summarized as stated here.

The specified version digit must be an integer and the version digit of the selected module must match exactly.

If the release digit is specified as an integer, it must match the release digit of the selected module exactly. However, if the release digit is omitted or is the wildcard character, then the module with the greatest available release digit is selected.

The modification digit can only be specified if the release digit is an integer, otherwise it must be omitted. If the modification is also an integer, the module with the greatest modification digit is selected provided that it is greater than or equal to the specified modification digit. When the modification digit is omitted or is the wildcard character, then the module with the greatest available modification digit is selected.

There are some examples provided at the bottom of the slide that help clarify the application of the rules. On the left is a box representing the version specifications for a group of modules in the registry. Next to it is a table, where the column entitled specified shows possible values configured in the endpoint lookup, and the column entitled selected indicates the version of the selected module. The first row matches the version digit exactly with the highest available release and modification digits. The second row matches the version and release digits exactly and the highest available modification digit. The third row results in no match because the release, specified as an integer, cannot be matched exactly. The fourth row results in no match because the modification digit is greater than any found in the registry for the specified version and release.

The next few slides look at the SMO content that is specific to the endpoint lookup primitive.

Looking at the screen capture, you can see that within the context section of the SMO there is a primitiveContext section. Its purpose is to allow mediation primitive types to define a specific usage of the SMO which is unique to primitives of that type.

Looking again at the screen capture, you can see that the primitiveContext contains an EndpointLookupContext, which is an array of endpoint information. This array is populated by the endpoint lookup primitive with the results of a registry lookup. Each endpoint in the array has endpointReference data defining the endpoint and registyAnnotations data defining additional information about the endpoint from the registry. These are both covered in more detail on the upcoming slides.

IBM Software Group

SMO usage

EndpointLookupContext : EndpointLookupContextType [0..n]
  endpointReference : EndpointReferenceType
    anyAttribute
    Address : AttributedURI
    ReferenceProperties : ReferencePropertiesType
    ReferenceParameters : ReferenceParametersType
    PortType : AttributedQName
    ServiceName : ServiceNameType
    any [0..n]
  registryAnnotations : RegistryAnnotationsType

- endpointReference
  - Schema defined by the WS-Addressing specification
    - Schema → http://schemas.xmlsoap.org/ws/2004/08/addressing
    - Specification → http://www.w3.org/Submission/ws-addressing/
    - Use of this schema is consistent with:
      - Service Oriented Architecture (SOA) Core
      - Business Process Choreographer
      - WebSphere Application Server SPIs
  - Address
    - The URI needed to contact the endpoint
    - This value is used when setting the SMO header target address and alternate target address fields
  - Other fields – see ws-addressing specification for full details

15

Endpoint lookup mediation primitive   © 2009 IBM Corporation

The schema for the endpointReference is defined by the WS-Addressing specification as defined by the World Wide Web Consortium (W3C). Looking at the slide, you see the URLs identifying where to find this schema definition and specification. Using this schema provides consistency with the Service Oriented Architecture Core, Business Process Choreographer and WebSphere Application Server SPIs.

This presentation does not attempt to describe this schema. The key element from the schema that you need to understand is the Address element. This contains the URI that is needed to contact the service endpoint. It is this value that is placed into the target address field or alternate target addresses field of the SMO header for use as a dynamic endpoint for a service invoke primitive or callout node.

This slide examines the registryAnnotations, which is composed of three arrays.

The first is the property array that contains the name value pairs for the user properties associated with the endpoint.

The next is the classification array, containing the URIs of the OWL classifications associated with the endpoint.

Finally, there is the relationship array. It contains information about the relationship between this endpoint and other entities in the registry. Each relationship is defined by a relationship name and the name, namespace and version of the target entity.

Service registry cache

- Server runtime provides a registry cache
  - ▸ Intended to boost performance for registry lookups
    - Caching is an effective strategy because registries are not normally subject to frequent updates
  - ▸ The cache is not exposed by the Mediation Flow editor
    - It is a hidden implementation detail in WebSphere Integration Developer
  - ▸ The cache is exposed to the administrator
    - Registries are administratively configured in the server
    - Registry definitions contain cache management information
    - Registry definitions are maintained at the cell level
    - Each server in the cell has one cache per configured registry

17

Performing frequent lookups to the WebSphere Service Registry and Repository can be expensive in terms of performance. Registries do not normally have frequent updates, so caching is a viable way to address this. Therefore, WebSphere Enterprise Service Bus and WebSphere Process Server provide a registry cache which is intended to boost the overall performance of registry lookups by reducing the number of calls made to the registry.

The endpoint lookup primitive and the mediation flow editor do not expose the registry cache. Therefore, from an integration developers perspective, the cache is hidden.

Registries are defined administratively and therefore are exposed to the administrator. The configuration of a registry includes configuration information for the management of the cache. The registry administration information is maintained at the cell level, and therefore the registry cache configuration information applies to all servers in the cell. There is one cache per registered registry in each server in the cell.

Management of the service registry cache

- Management of cache contents
  - ▸ The cache is populated "lazily" as lookups occur
  - ▸ A timeout period is associated with the cache
    - Individual cache entries are invalidated based on the timeout period
    - There is administrative capability to set the timeout value
    - Timeout value can be set to "0" to indicate not to cache entries
  - ▸ Administratively clearing the cache
    - From the administrative console
      – Caches for designated registry are cleared in all servers in the cell
    - Using the wsadmin commands
      – Cache for designated registry is cleared in a single server
      – Must loop through servers to clear cache in all servers in the cell

Endpoint lookup mediation primitive © 2009 IBM Corporation

Population of the cache occurs in a lazy fashion, with entries being added as lookups occur. The administrator can assign a timeout period that is applied to cache entries, causing entries to be invalidated once they are older than the timeout period. If a timeout period is set to zero, the effective result is that caching does not take place. This is a reasonable thing to do in a development environment, but in a stable production environment the benefits of caching suggests using a large timeout value.

In the event that the registry is updated so that the existing cache entries are out of date, there is a capability for the administrator to issue a command to clear the cache. When this is done from the administrative console, the caches for that registry are cleared in all of the servers in the cell. However, the wsadmin command only clears the cache in a single server. Therefore, if using the command line, clearing the cache in all servers in the cell requires looping through each server to clear its cache.

Server administration of WebSphere Service Registry and Repository

- Service registries need to be configured
  - Endpoint lookup primitives must use configured registries
  - Registry definitions found in the administrative console:
    - **Service integration → WSRR definitions**
  - Several actions can be performed using the buttons at the top of the list

Endpoint lookup mediation primitive

© 2009 IBM Corporation

19
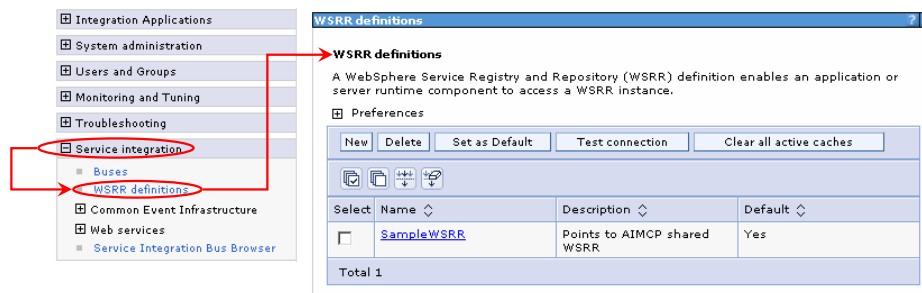
The next couple of slides show how the WebSphere Service Registry and Repository registries are administered in WebSphere Enterprise Service Bus and WebSphere Process Server.

Note that in order for an endpoint lookup primitive to use a registry, the registry must be administratively defined as described here.

This slide shows the navigation in the administrative console. On the left panel, open Service integration and then select WSRR definitions. This opens a panel containing a list of the configured registries.

Notice the buttons at the top of the list of registry definitions. There is a New button and Delete button for managing which definitions are in the list. There is also a Default button, allowing you to designate one of the registries to be the default registry. The Test connection button enables you to make sure the configuration information is sufficient to enable the server to connect with the registry. Finally, the Clear all active caches button clears the caches for this registry that exist in every server in the cell.

By clicking on one of the registries in the list, the registry properties panel opens, as shown on the next slide.

Server administration

- Registry definitions maintained at cell scope

- WebSphere Service Registry and Repository definition name
    - ▸ Used to identify this registry instance
    - ▸ Endpoint lookup configuration uses this value to identify the registry

- Default WebSphere Service Registry and Repository definition
    - ▸ Designates this registry as the default
    - ▸ Leaving registry name blank in endpoint lookup causes this registry to be used

- Timeout of cache
    - ▸ How long cache entries remain valid
    - ▸ Value of 0 prevents caching

Test connection

General Properties

WSRR definition name
SampleWSRR

Description
Points to AIMCP shared WSRR

Default WSRR definition
Yes

Timeout of cache
300 seconds

Connection type
Web service

Apply OK Reset Cancel

Additional Properties
- Connection properties

This slide shows the properties that are used to configure a WebSphere Service Registry and Repository. As previously mentioned, within a WebSphere cell, these definitions are maintained only at a cell scope and cannot be specified at the node or server scope.

The first field is the 'WSRR definition name' field. This property is a name that is used to identify this registry instance. This is the value that you specify in the endpoint lookup primitive that identifies the registry to use for the lookup.

The Description property provides a text comment describing this registry.

The 'Default WSRR definition' field, if set to yes, designates this registry instance as the default registry. The default registry is the one that is used when the registry name property in the endpoint lookup is left blank.

The Timeout of cache property specifies how long cached entries remain valid. A value of zero prevents any caching of entries for this registry instance.

Server administration

- Connection type
  - Currently Web service is the only allowed choice
- Registry URL
  - URL to connect to registry instance
  - http://<host:port>/WSRRCoreSDO/services/WSRRCoreSDOPort
- Authentication alias
  - Defines user ID and password needed to authenticate with the registry
- SSL Configuration
  - Defines the SSL configuration in the server needed to communicate with the registry
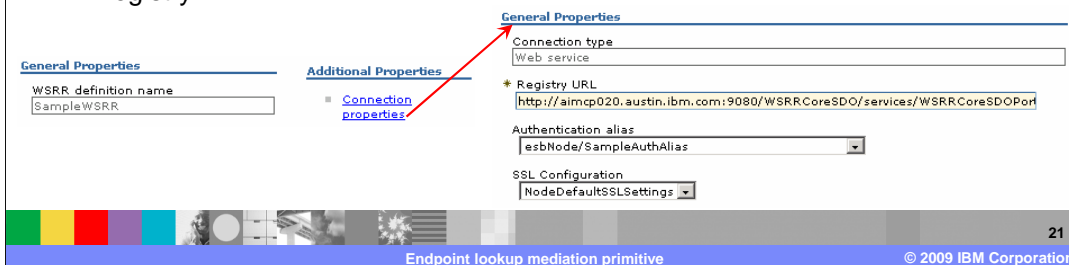
From the panel shown on the previous slide, clicking on the Connection properties link allows you to specify properties for the connection to the WebSphere Service Registry and Repository.

The Connection type defines what type of protocol is used to connect with the registry. Currently, Web service is the only connection type allowed.

The Registry URL property provides the specific URL needed to connect to this registry instance. Your URL should be the same as the one on the slide, with only the host and port needing to be unique to your environment.

The Authentication alias property identifies an authentication alias containing the user ID and password needed to authenticate with the registry.

The SSL Configuration property identifies the SSL configuration definition needed to connect with the registry.

Configuration of the WebSphere Service Registry and Repository

- Services must be correctly defined in the registry
  - ▸ This presentation does not address how this is done

- See the WebSphere Service Registry and Repository Information Center for complete information
  - ▸ http://publib.boulder.ibm.com/infocenter/sr/v6r2/index.jsp
  - ▸ See topic: **Loading SCA module definitions**

Describing how to configure and administer the WebSphere Service Registry and Repository is beyond the scope of this presentation. However, it is important that the services be correctly defined in the registry if the endpoint lookup requests are to result in matches.

Information on configuration of the WebSphere Service Registry and Repository is provided in the information center at the URL shown on this slide. For initializing the registry with service endpoints defined by SCA modules, see the topic entitled Loading SCA module definitions.

Error processing

- MediationBusinessException (fail terminal flow)
  - ▸ Registry not currently available
  - ▸ Configured URL for registry is incorrect
    - Correct syntax but wrong, such as an incorrect port specified
    - Incorrect syntax (malformed URL)
  - ▸ Registry name not found
    - Incorrect name specified in mediation primitive
    - No administrative entry in the server for the for specified registry
  - ▸ User Property XPath expression problem
    - Valid XPath but element does not exist in the SMO
    - Invalidly formed XPath expression

23

© 2009 IBM Corporation

There are several conditions that cause a MediationBusinessException to occur. When any of these happen, the mediation flow continues through the fail terminal if it is wired, otherwise the exception is re-thrown and the mediation flow is ended.

One issue is that the registry specified is not currently available and therefore cannot be contacted with the lookup request.

Another possible reason is if the URL for the registry has been incorrectly specified in the administrative definition of the registry. This can be a URL with a good syntax but has a mistake such as an incorrect host or port specification. It can also be for a malformed URL.

The MediationBusinessException can also occur if the administrative definition for the registry specified in the endpoint lookup properties cannot be found. This can occur if the administrative name for the registry is misspelled, or if the registry was never administratively defined.

Another reason for this exception is if the User Properties table contained an XPath expression which can not be resolved. This can occur if the XPath expression is incorrect or if this instance of the SMO does not happen to have the element defined by the XPath.

Problem determination

- Call to the registry fails
  - Ensure the registry service is running
  - Ensure registry is configured correctly in the server
  - Ensure mediation primitive has specified the right registry entry
- Not getting expected services returned from registry
  - Ensure selection properties configured on primitive are correct
  - Ensure correct WSDLs, XSDs and SCA modules have been loaded into the registry
  - Ensure your primitive is configured for the right registry
  - Consider possibility that a cached entry is being returned
    - Result does not reflect recent updates to the registry
    - When doing development or test, it is be best to run with cache timeout = 0
    - Administrative clear the cache to ensure you are not getting old entries
- Target address not set in SMO
  - Check logic for setting the target address

Endpoint lookup mediation primitive   © 2009 IBM Corporation

Several things can go wrong at runtime that are inherent in the complexity of the interactions involved. There is interaction between the endpoint lookup primitive, the server administration of registries, and the call to the registry. This slide gives you some things to look for when your endpoint lookup primitive fails at runtime.

The first set of issues revolve around a failing call to the registry. If this is the case, ensure that the registry is running and that the configuration of the registry in the server is correct. Also check that the mediation primitive has specified the correct administrative name for the registry to be used.

The next set of issues revolve around the results from a lookup apparently not returning the expected services. The first thing to check is the configuration of the endpoint lookup primitive to ensure that the various search criteria properties are correctly specified. If so, then make sure the correct XSD and WSDL files and SCA modules have been loaded into the registry. And finally, make sure that the endpoint lookup is configured to use the right registry.

If there have been any registry updates, consider the possibility that your endpoint lookup is being resolved through the registry cache rather than a call to the registry. This is not likely to be an issue in a production environment, but is definitely an issue during development and test. As a best practice, use a cache timeout of zero to prevent caching during the development phase and possibly also during the test phase. If you are using caching, you can administratively clear the cache to ensure that you are not picking up old entries.

If the dynamic callout is failing because the target address is not set in the SMO, there are a couple of things you can check. If the endpoint lookup is configured to have a Match Policy of one, check to see why no endpoint was returned from the lookup. If the endpoint lookup is configured to have a Match Policy of all, check if any endpoints were returned. If they were, check the mediation flow logic following the endpoint lookup to determine why the target address was not set.

Example

- Select endpoint by domain
  - ▶ Endpoint lookup finds all endpoints for a service
  - ▶ Custom code then:
    - Examines the endpoints returned
    - Selects an endpoint based on domain
    - Selection favors providers in the same domain

getBalance : Account   LookupServiceProviders   NoProvidersFound   SelectServiceProvider   getBalance : Account

Custom code sets target address

Endpoint Lookup : LookupServiceProviders

Description		
Terminal	Name:	Account	Browse...
Details		
Advanced	Namespace:	http://BankAppLibrary/Account
Promotable Properties	Registry Name:	<Use default registry>
	Match Policy:	Return all matching endpoints
	Version:	
	Module:	
	Export:	

Service Requested

Use the default registry

*Return all endpoints
for requested service*

25

Endpoint lookup mediation primitive   © 2009 IBM Corporation

In this example, the selection of the endpoint is optimized to use a service provider in the same domain as the mediation. In the screen capture, you can see the mediation flow and the properties for the endpoint lookup. Notice that the lookup uses the default registry configured in the server and the match policy results in all endpoints for this service being returned. Following the endpoint lookup is a custom mediation primitive. It examines the address URIs of the returned endpoints looking for one that is in the same domain. If found, it places that into the target address element of the SMOHeader. If none are in the same domain, one endpoint is arbitrarily picked.

IBM

Summary

- Examined the endpoint lookup mediation primitive

  Endpoint lookup

  ▸ Overview of function
  ▸ Use of terminals
  ▸ Definition of properties
  ▸ Service message object (SMO) usage
  ▸ Service registry cache
  ▸ Server administration of the registry
  ▸ Configuration of the registry
  ▸ Error handling
  ▸ Problem determination
  ▸ Example usage

In summary, this presentation introduced you to an overview of the function provided by the endpoint lookup primitive, along with information about the primitive's use of terminals and its properties.

Since this primitive has a special relationship to the service message object elements in the SMO context, the schema for this was examined.

In order to understand the endpoint lookup primitive, it is necessary to understand more than the behavior of the primitive itself and how it fits into a mediation flow. The endpoint lookup primitive interfaces with the WebSphere Service Registry and Repository. Use of the registry is enabled through capabilities provided by the WebSphere Enterprise Service Bus and the WebSphere Process Server. You learned about the service registry cache and the administration of registries that is provided by the server. You were also provided with references to documentation describing how to configure the registry.

The presentation then returned to looking specifically at the primitive, covering error handling, problem determination and an example of its usage.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv62_EndpointLookupPrimitive.ppt

This module is also available in PDF format at: ../WBPMv62_EndpointLookupPrimitive.pdf

27

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

28

Endpoint lookup mediation primitive