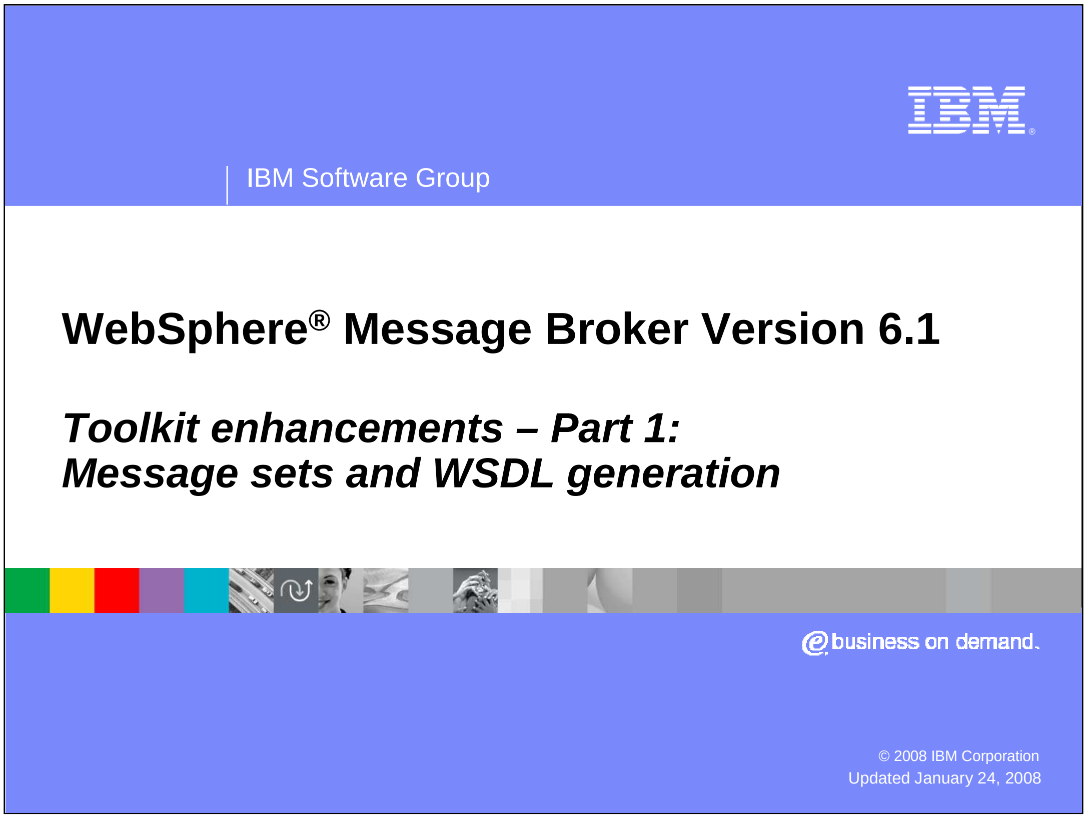


IBM Software Group

WebSphere® Message Broker Version 6.1

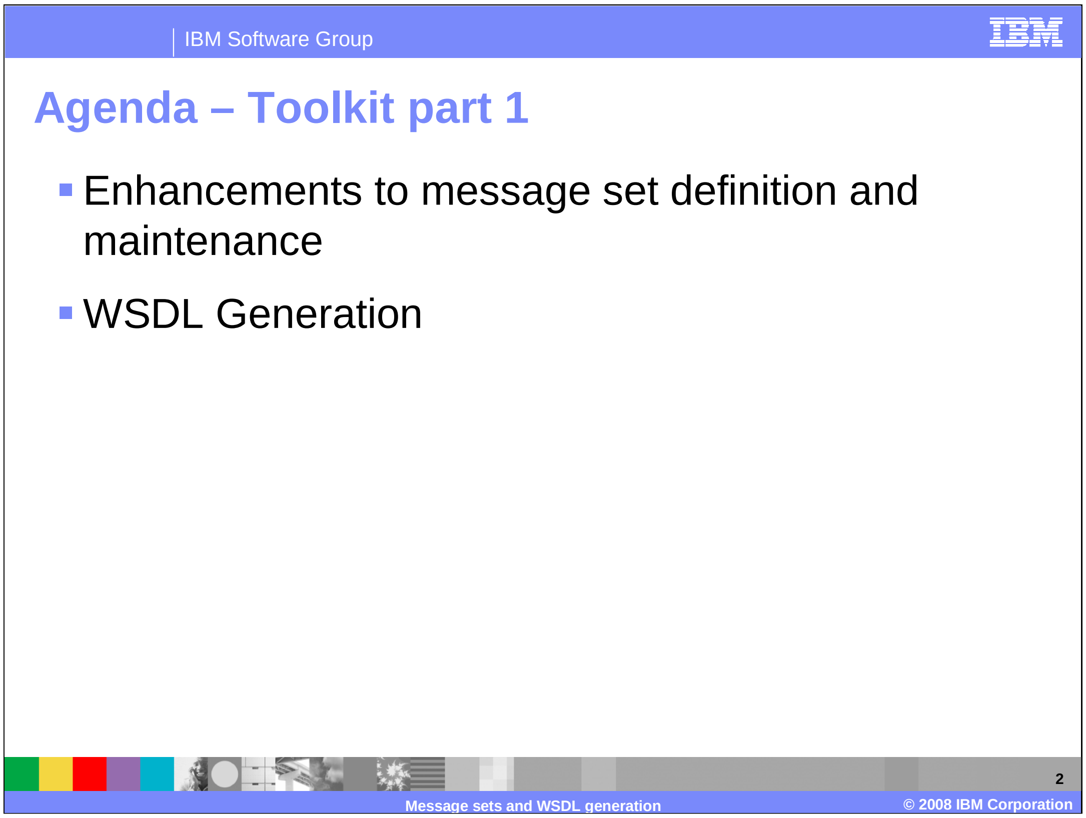
Toolkit enhancements – Part 1: Message sets and WSDL generation

business on demand.

© 2008 IBM Corporation

Updated January 24, 2008

This presentation is the first of three presentations looking at the toolkit in Message Broker Version 6.1. This presentation will look at the changes made to message set definitions, and to the generation of Web Services Definition Language.



IBM Software Group

Agenda – Toolkit part 1

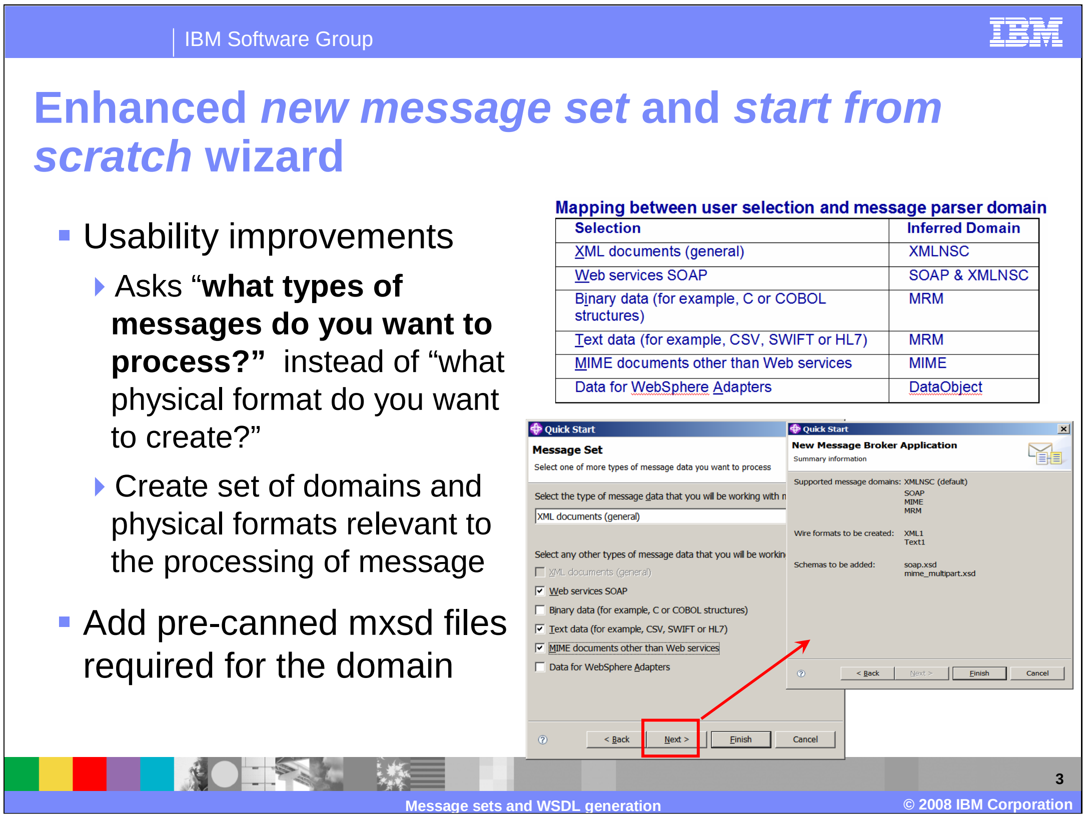
- Enhancements to message set definition and maintenance
- WSDL Generation

2

Message sets and WSDL generation

© 2008 IBM Corporation

The Message Broker toolkit has had some significant changes in version 6.1. One of the key themes for version 6.1 is the area of Web services, and the generation of message sets and the use of WSDL is fundamental to this area.



Enhanced *new message set and start from scratch* wizard

- Usability improvements
  - Asks “**what types of messages do you want to process?”** instead of “what physical format do you want to create?”
  - Create set of domains and physical formats relevant to the processing of message
- Add pre-canned mxsd files required for the domain

Mapping between user selection and message parser domain

Selection	Inferred Domain
XML documents (general)	XMLNSC
Web services SOAP	SOAP & XMLNSC
Binary data (for example, C or COBOL structures)	MRM
Text data (for example, CSV, SWIFT or HL7)	MRM
MIME documents other than Web services	MIME
Data for WebSphere Adapters	DataObject

Quick Start

Message Set

Select one of more types of message data you want to process

Select the type of message data that you will be working with m

XML documents (general)

Select any other types of message data that you will be workin

□ XML documents (general)
☑ Web services SOAP
□ Binary data (for example, C or COBOL structures)
☑ Text data (for example, CSV, SWIFT or HL7)
☑ MIME documents other than Web services
□ Data for WebSphere Adapters

Quick Start

New Message Broker Application

Summary information

Supported message domains: XMLNSC (default)
SOAP
MIME
MRM

Wire formats to be created: XML1
Text1

Schemas to be added: soap.xsd
mime_multipart.xsd

The first topic covers the enhancements made to message set definition.

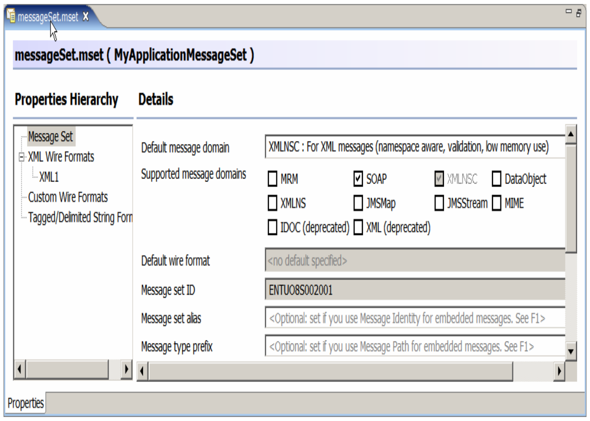
The “Start from Scratch” wizard allows you to create the basic resources that are required to develop a Message Broker application. The “Start from Scratch” Wizard creates a message flow project, a message set project, sets up the project dependency, creates a message set and, optionally, creates a message flow and working set. You have the option of selecting the type of message data that you will most often be working with and any other types of data you will likely be working with. The domains and physical formats are then created that are relevant for processing the message.

Message set – Associating multiple domains

Associate multiple domains with message set to ***better support composite messages***

Primary motivation comes from:

- ***MIME domain*** - where message itself is parsed by MIME domain but contents of the embedded message are parsed by another domain such as XMLNSC
- ***SOAP domain*** - New SOAP nodes use SOAP domain to build SOAP message tree but a follow-on SOAPExtract node puts the extracted body message in XMLNSC domain



messageSet.mset (MyApplicationMessageSet)

Default message domain: XMLNSC : For XML messages (namespace aware, validation, low memory use)

Supported message domains: □ MRM ☑ SOAP ☑ XMLNSC □ DataObject □ XMLNS □ JMSMap □ JMSStream □ MIME □ IDOC (deprecated) □ XML (deprecated)

Default wire format: <no default specified>

Message set ID: ENTUO8S002001

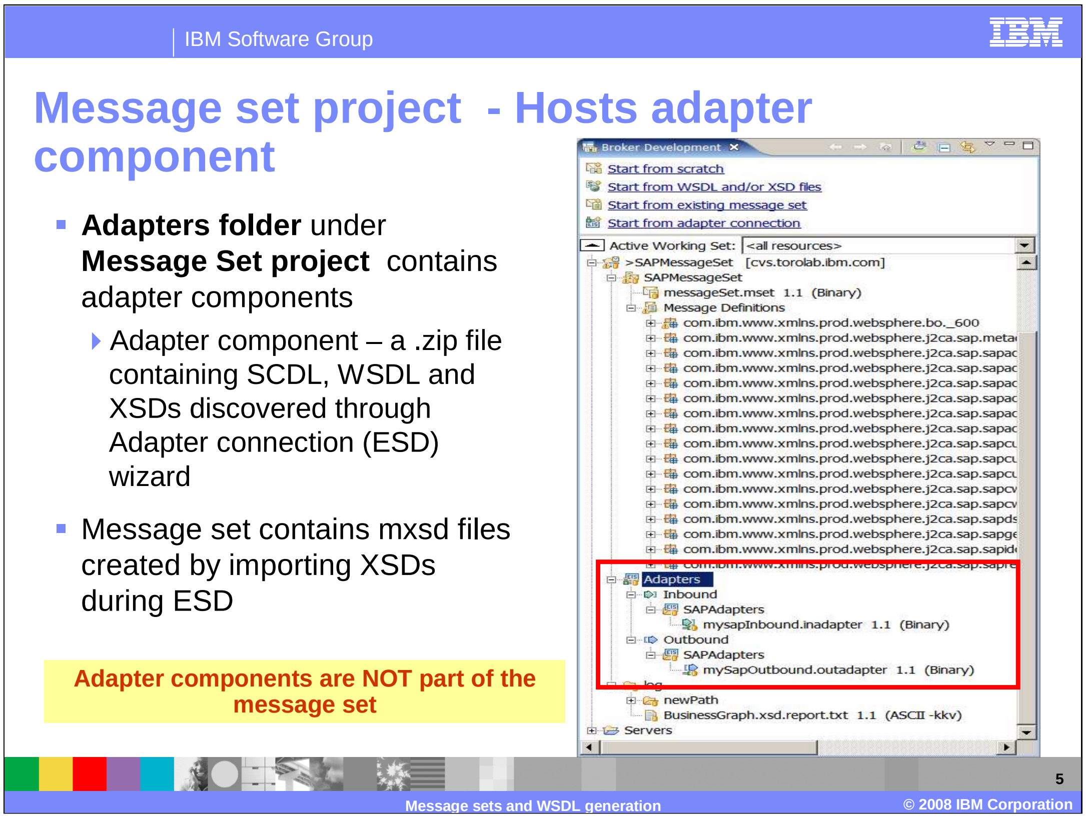
Message set alias: <Optional: set if you use Message Identity for embedded messages. See F1>

Message type prefix: <Optional: set if you use Message Path for embedded messages. See F1>

You no longer have to create multiple message sets to overcome the limitation of a single parser domain on the message set that existed in 6.0 releases. The primary parser domain now becomes default domain.

A new feature in Version 6.1 is the ability to associate multiple message domains with a single message set. Previously, users had to create multiple message sets if multiple domains are required. In the message set editor you specify the supported domains, with the primary parser coming from the default message domain.

This is particularly useful for messages in the MIME domain, where the embedded part of the message is parsed by a domain other than the MIME domain. It is also useful in the SOAP domain, where the SOAP body will be parsed in the XMLNSC domain after the headers have been parsed in the SOAP domain.



Message set project - Hosts adapter component

- **Adapters folder** under **Message Set project** contains adapter components
  - Adapter component – a .zip file containing SCDL, WSDL and XSDs discovered through Adapter connection (ESD) wizard
- Message set contains mxsd files created by importing XSDs during ESD

Adapter components are NOT part of the message set

Broker Development
Start from scratch
Start from WSDL and/or XSD files
Start from existing message set
Start from adapter connection
Active Working Set:
<all resources>

```
>SAPMessageSet   [cvs.torolab.ibm.com]
  SAPMessageSet
    messageSet.mset  1.1  (Binary)
    Message Definitions
      com.ibm.www.xmlns.prod.websphere.bo._600
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.meta
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapac
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapac
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapac
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapac
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapac
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapac
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapcu
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapcu
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapcu
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapcv
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapcv
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapds
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapge
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapid
      com.ibm.www.xmlns.prod.websphere.j2ca.sap.sapre
  Adapters
    Inbound
      SAPAdapters
        mysapInbound.inadapter  1.1  (Binary)
    Outbound
      SAPAdapters
        mySapOutbound.outadapter  1.1  (Binary)
  log
    newPath
    BusinessGraph.xsd.report.txt  1.1  (ASCII -kkv)
Servers
```

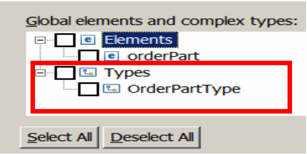
© 2008 IBM Corporation

For projects using adapters, the adapters folder in the Message Set project will contain the adapter component that was discovered through Enterprise Service Discovery. This uses the Adapter Connection Wizard.

This contains WSDL, SCDL and schema definitions. The adapter components are not part of the message set. The message set contains the message definition files that are created by importing the discovered schemas during the enterprise service discovery process.

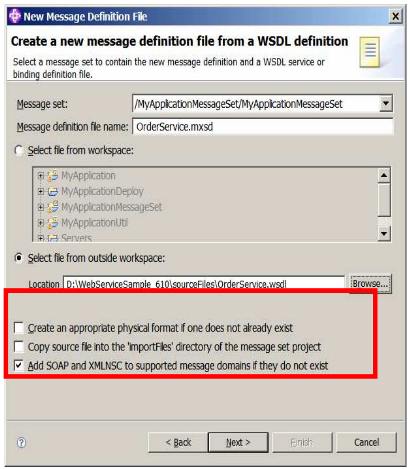
Importer enhancements

- Create appropriate physical wire format required for the import resource
- Copy input source in **importFiles** folder
- XSD importer and quick start wizard
  - start from WSDL or XSD or both
  - Allow creation of message definition based on global complex types in addition to global elements



Global elements and complex types:

- □ Elements
  - □ orderPart
- □ Types
  - □ OrderPartType



Create a new message definition file from a WSDL definition

Select a message set to contain the new message definition and a WSDL service or binding definition file.

Message set: /MyApplicationMessageSet/MyApplicationMessageSet

Message definition file name: OrderService.mxsd

○ Select file from workspace:

- MyApplication
- MyApplicationDeploy
- MyApplicationMessageSet
- MyApplicationUtil
- Servers

● Select file from outside workspace:

Location D:\WebServiceSample_610\sourceFiles\OrderService.wsdl

□ Create an appropriate physical format if one does not already exist

□ Copy source file into the 'importFiles' directory of the message set project

☑ Add SOAP and XMLNSC to supported message domains if they do not exist

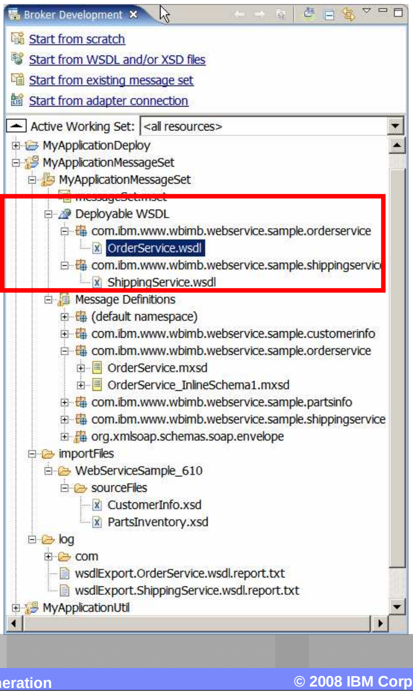
© 2008 IBM Corporation

Several enhancements have been made to the Importers. New options include the ability to create the appropriate wire format required for the imported resource, such as XML for a schema import. The source of an import can also be copied in to an import-Files folder.

When using the schema importer or the Quick Start Wizard for WSDL or schemas, you can create the message definition based on global complex types that are in the validated schema, in addition to global elements.

Message Set - WSDL key artifact

- WSDL files imported into message set are
  - Stored in subfolder derived from nsURI of WSDL
  - Shown under new logical category “Deployable WSDL”
  - Schema type definitions referenced externally or defined in-lined are changed to reference mxsd files in the message set
  - WSDL files are annotated to
    - Keep track of mxsd files created by WSDL importer.
    - Keep WSDL files in synch with MXSD files



Start from scratch
Start from WSDL and/or XSD files
Start from existing message set
Start from adapter connection

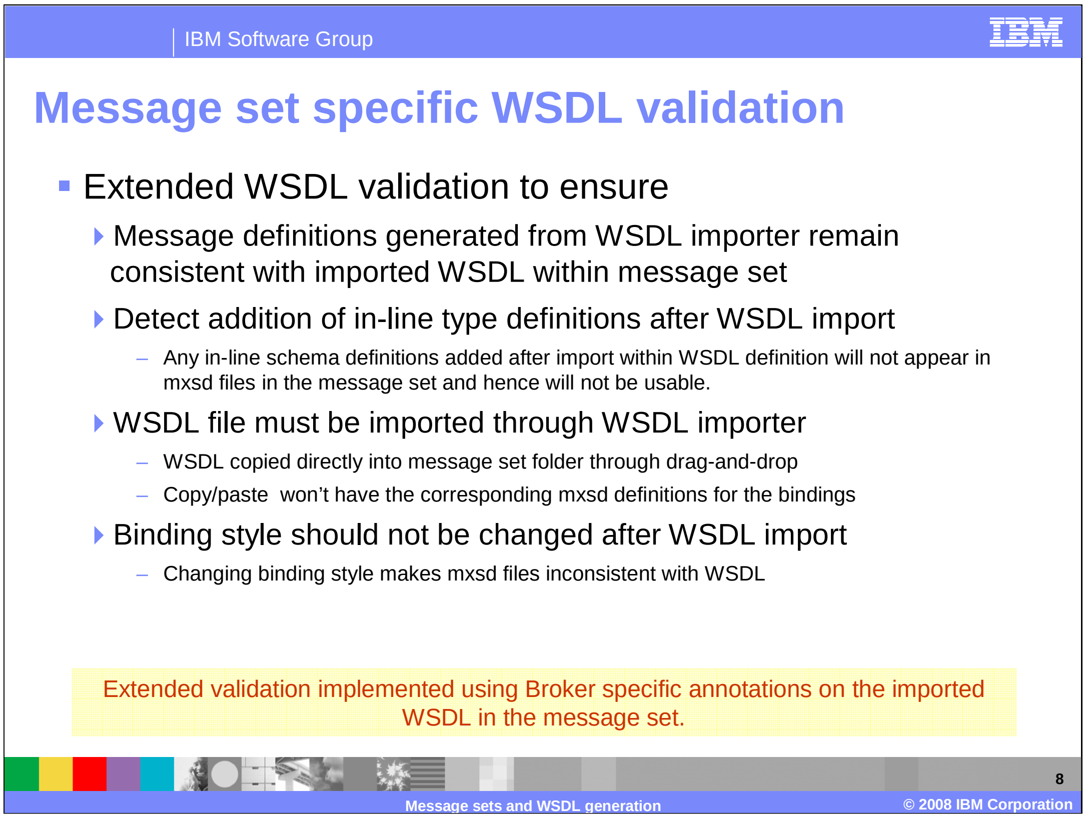
Active Working Set: <all resources>

```
MyApplicationDeploy
MyApplicationMessageSet
  MyApplicationMessageSet
    messageSet.mset
    Deployable WSDL
      com.ibm.www.wbimb.webservice.sample.orderservice
        OrderService.wsdl
      com.ibm.www.wbimb.webservice.sample.shippingservic
        ShippingService.wsdl
    Message Definitions
      (default namespace)
      com.ibm.www.wbimb.webservice.sample.customerinfo
      com.ibm.www.wbimb.webservice.sample.orderservice
        OrderService.mxsd
        OrderService_InlineSchema1.mxsd
      com.ibm.www.wbimb.webservice.sample.partsinfo
      com.ibm.www.wbimb.webservice.sample.shippingservice
      org.xmlsoap.schemas.soap.envelope
  importFiles
    WebServiceSample_610
      sourceFiles
        CustomerInfo.xsd
        PartsInventory.xsd
  log
    com
    wsdlExport.OrderService.wsdl.report.txt
    wsdlExport.ShippingService.wsdl.report.txt
MyApplicationUtil
```

© 2008 IBM Corporation

When WSDL files are imported into a message set project, they are stored in a new folder named Deployable-WSDL. The subfolder names are derived from the nsURI of the WSDL.

Inline schema or external schema type definitions are change to reference message definition file in the message. The WSDL file in the Deployable WSDL folder is annotated to keep track of message definition files created by the importer, and keep WSDL files synchronized with Message Definition files.



IBM Software Group

IBM

Message set specific WSDL validation

- Extended WSDL validation to ensure
  - Message definitions generated from WSDL importer remain consistent with imported WSDL within message set
  - Detect addition of in-line type definitions after WSDL import
    - Any in-line schema definitions added after import within WSDL definition will not appear in mxsd files in the message set and hence will not be usable.
  - WSDL file must be imported through WSDL importer
    - WSDL copied directly into message set folder through drag-and-drop
    - Copy/paste won't have the corresponding mxsd definitions for the bindings
  - Binding style should not be changed after WSDL import
    - Changing binding style makes mxsd files inconsistent with WSDL

Extended validation implemented using Broker specific annotations on the imported WSDL in the message set.

8

Message sets and WSDL generation

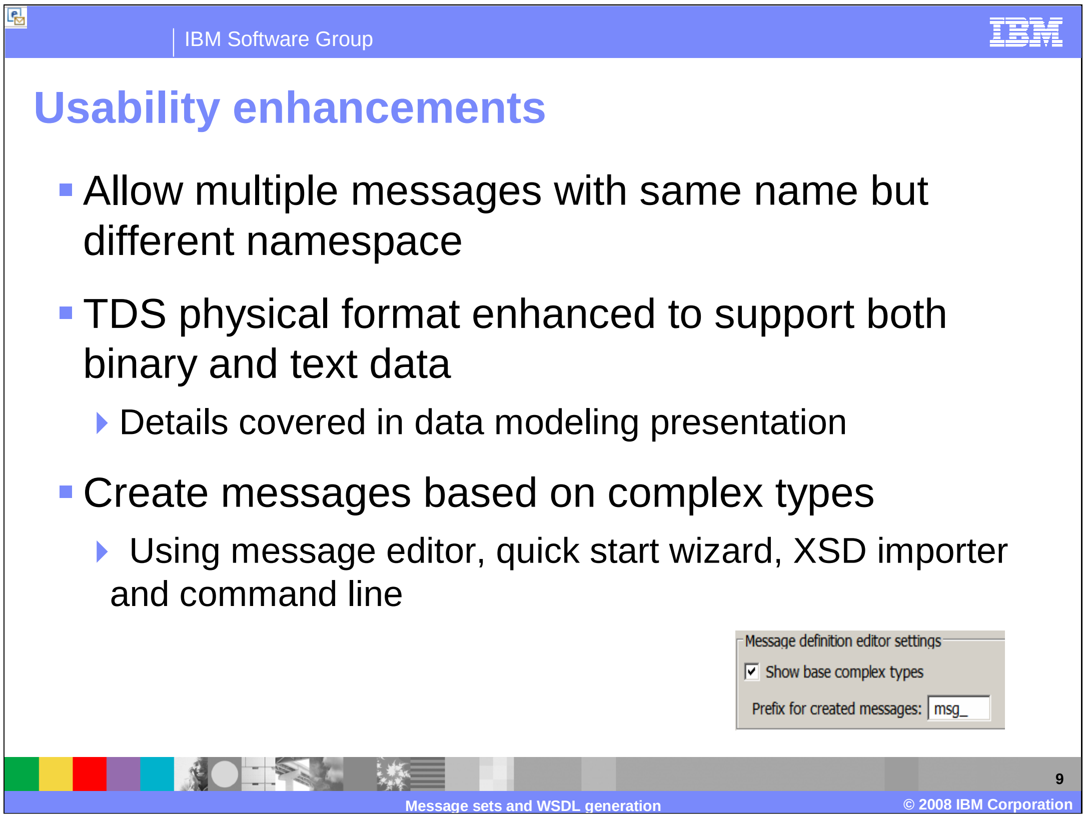
© 2008 IBM Corporation

The validation of WSDL for message set generation has been extended to ensure that message definitions that are generated using the WSDL importer remain consistent with the imported WSDL within the message set. Validation includes the detection of the addition of any in-line schema definitions after import. Any in-line schema definitions that are added to the WSDL after the import will not appear in the message definition file in the message set, and consequently cannot be used.

The WSDL file must be imported using the WSDL importer. WSDL that is copied directly into the message set folder will not have the corresponding message definitions for the binding, and consequently will not be useable.

Also, the binding style should not be changed after the WSDL import. Changing the binding style will cause the message definition files to be inconsistent with the WSDL.

Finally, extended validation is implemented using Message Broker specific annotations on the imported WSDL in the message set.



IBM Software Group

Usability enhancements

- Allow multiple messages with same name but different namespace
- TDS physical format enhanced to support both binary and text data
  - Details covered in data modeling presentation
- Create messages based on complex types
  - Using message editor, quick start wizard, XSD importer and command line

Message definition editor settings

☑ Show base complex types

Prefix for created messages: msg_

9

Message sets and WSDL generation

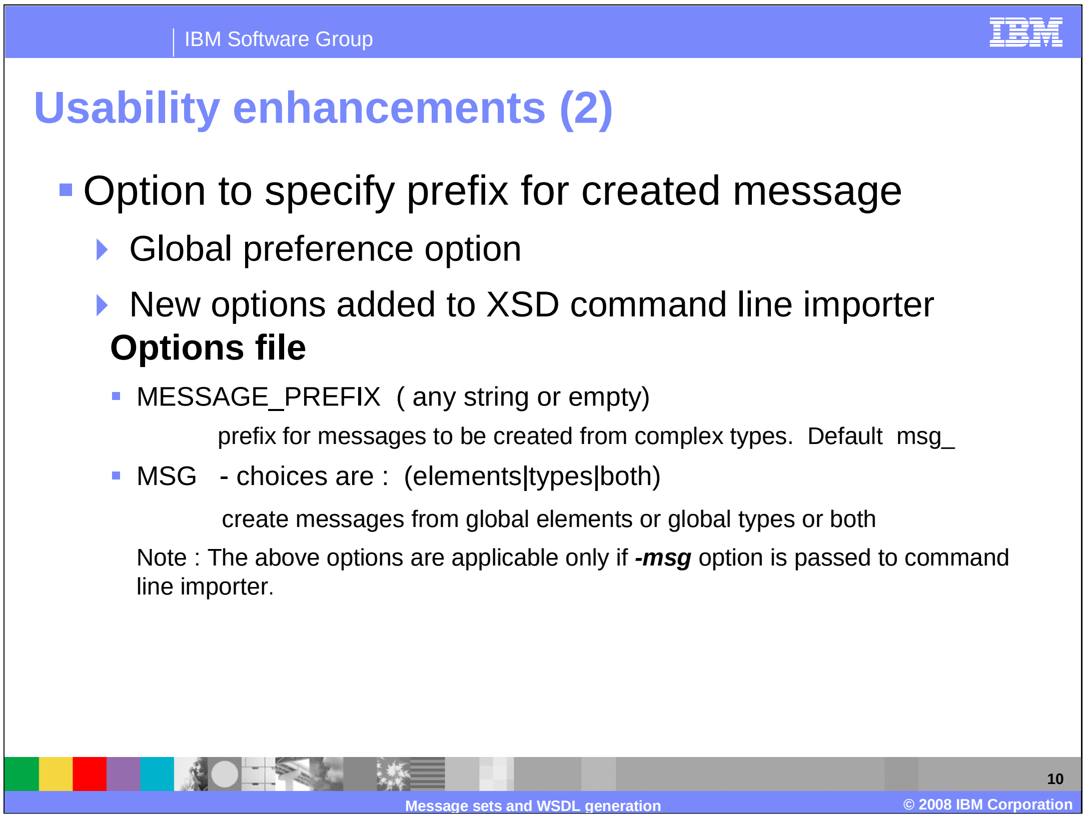
© 2008 IBM Corporation

The next two slides discuss some additional improvements in the area of usability.

A message set can contain multiple messages of the same name as long as each message is identified by a unique, and different namespace.

The “Tagged Delimited String” wire format has been enhanced to support both binary and test data type. Tagged Delimited String is sometimes shortened to TDS.

Messages can be created based on complex types using the message editor, the quick start wizards, and command line schema importers. Preferences can be set in the message definition editor settings.



Usability enhancements (2)

- Option to specify prefix for created message
  - Global preference option
  - New options added to XSD command line importer

Options file

- MESSAGE_PREFIX (any string or empty)

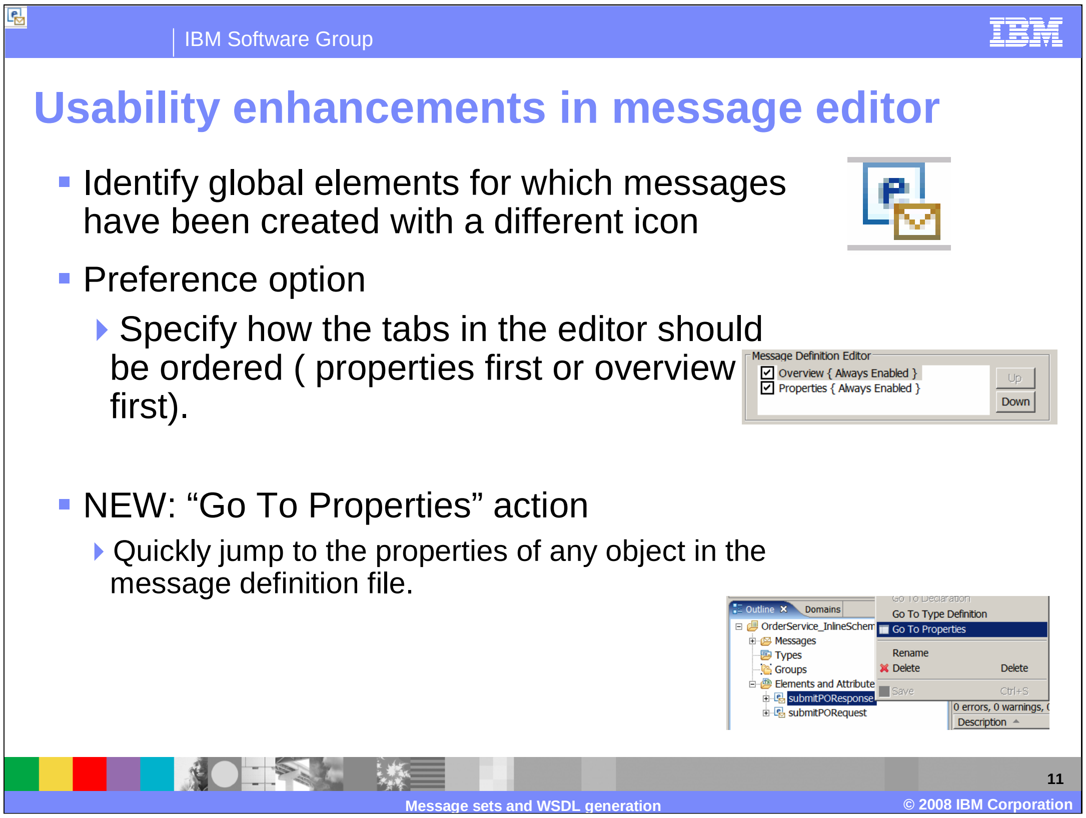
  prefix for messages to be created from complex types. Default msg_
- MSG - choices are : (elements|types|both)

  create messages from global elements or global types or both

Note : The above options are applicable only if ***-msg*** option is passed to command line importer.

As seen on the previous slide, also in the message set editor preferences is an option to set a global option for the prefix for newly created messages.

Added to the schema command line importer are additional parameters to nominate an options file which can be used when the –msg option is used. This options file can be used to set preferences for the import.



IBM Software Group

Usability enhancements in message editor

- Identify global elements for which messages have been created with a different icon
- Preference option
  - Specify how the tabs in the editor should be ordered (properties first or overview first).

Message Definition Editor

☑ Overview { Always Enabled }
☑ Properties { Always Enabled }

- NEW: "Go To Properties" action
  - Quickly jump to the properties of any object in the message definition file.

11

Message sets and WSDL generation

© 2008 IBM Corporation

The message editor has been enhanced in version 6.1.

First, a new icon identifies global elements for which messages have been defined.

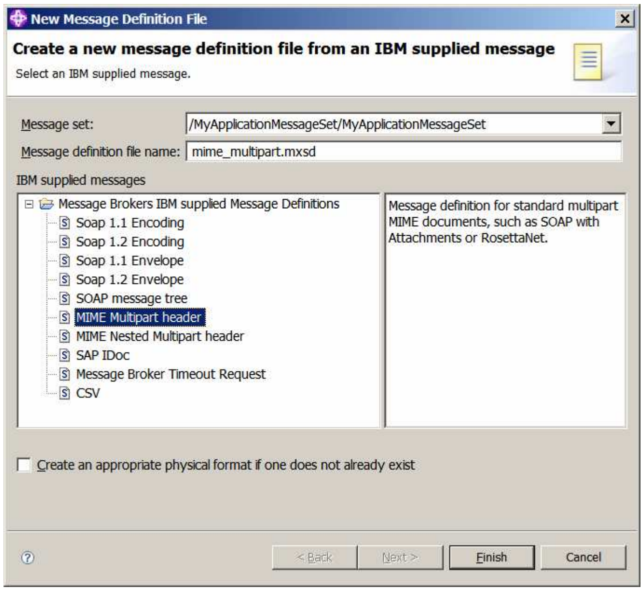
Second, you can set preferences for the message definition editor to specify how the tabs in the editor should be ordered.

Finally, the "Go To Properties" action allows you to quickly jump to the properties of any object in the message definition file.

IBM Software Group

Improved create message from message wizard

Shows short and long description for each of the pre-supplied schemas rather than just file names.



New Message Definition File

Create a new message definition file from an IBM supplied message

Select an IBM supplied message.

Message set: /MyApplicationMessageSet/MyApplicationMessageSet

Message definition file name: mime_multipart.mxsd

IBM supplied messages

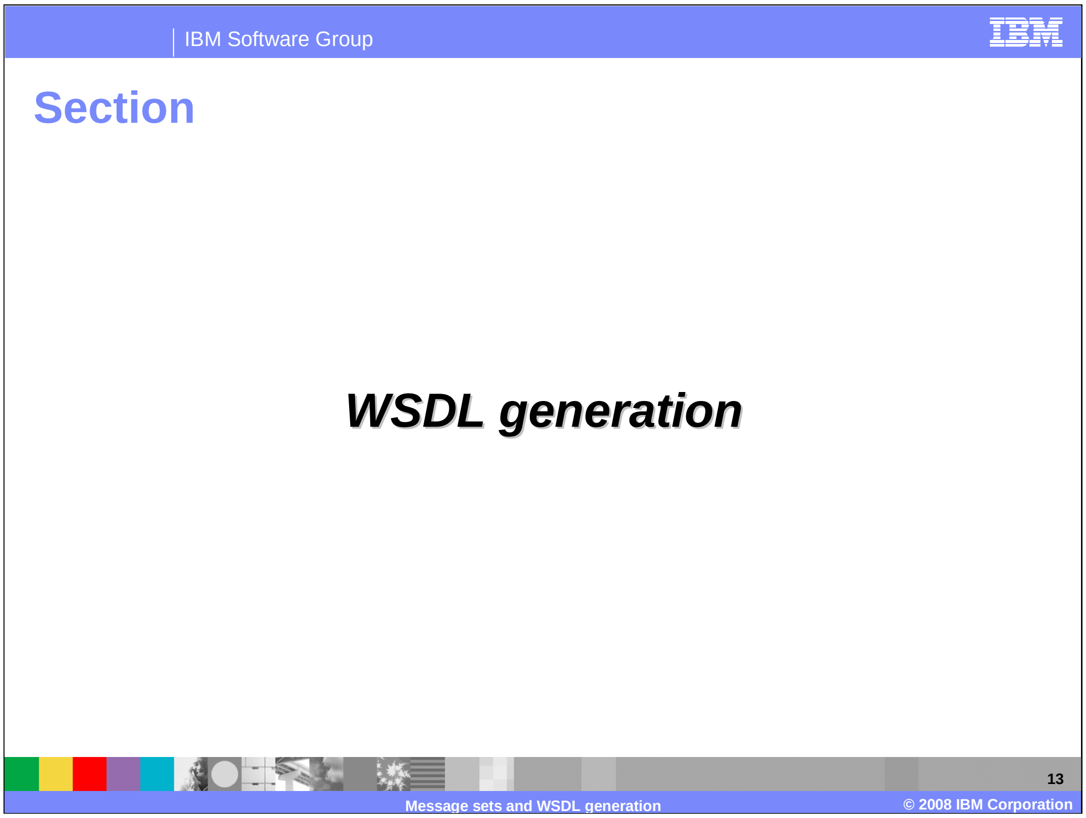
- Message Brokers IBM supplied Message Definitions
  - Soap 1.1 Encoding
  - Soap 1.2 Encoding
  - Soap 1.1 Envelope
  - Soap 1.2 Envelope
  - SOAP message tree
  - MIME Multipart header
  - MIME Nested Multipart header
  - SAP IDoc
  - Message Broker Timeout Request
  - CSV

Message definition for standard multipart MIME documents, such as SOAP with Attachments or RosettaNet.

☐ Create an appropriate physical format if one does not already exist

© 2008 IBM Corporation

Finally in this section, the “Create Message from IBM Supplied Message” wizard has been improved to show both a long and short description of the pre-supplied schemas, rather than just the file names. These are shown on this screen capture.



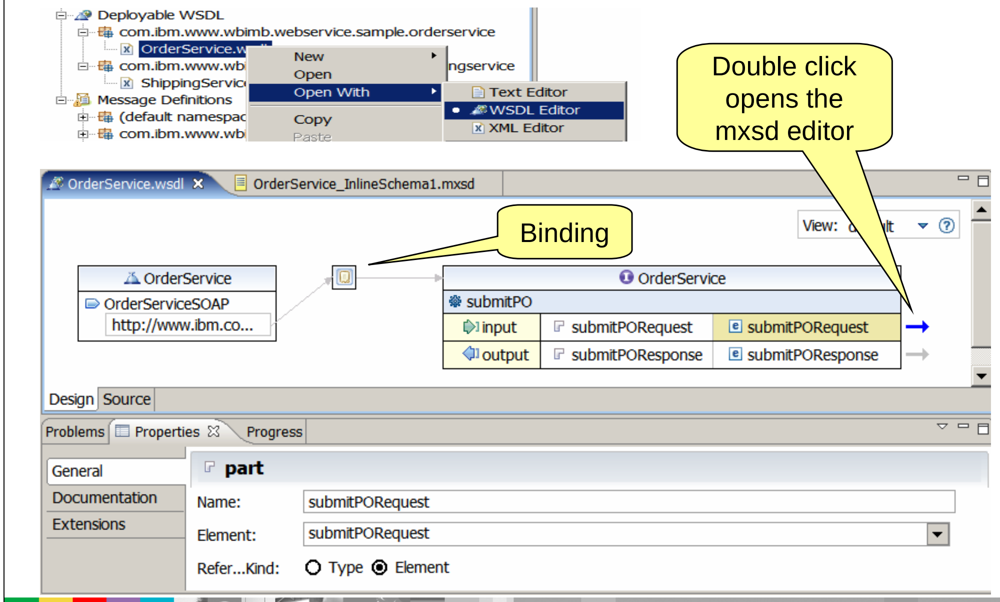
Section

WSDL generation

© 2008 IBM Corporation

The presentation will now look at improvements which relate to WSDL generation.

WSDL – Open with WSDL editor



Name: submitPORequest

Element: submitPORequest

Refer...Kind: ○ Type ● Element

Double click opens the mxsd editor

Binding

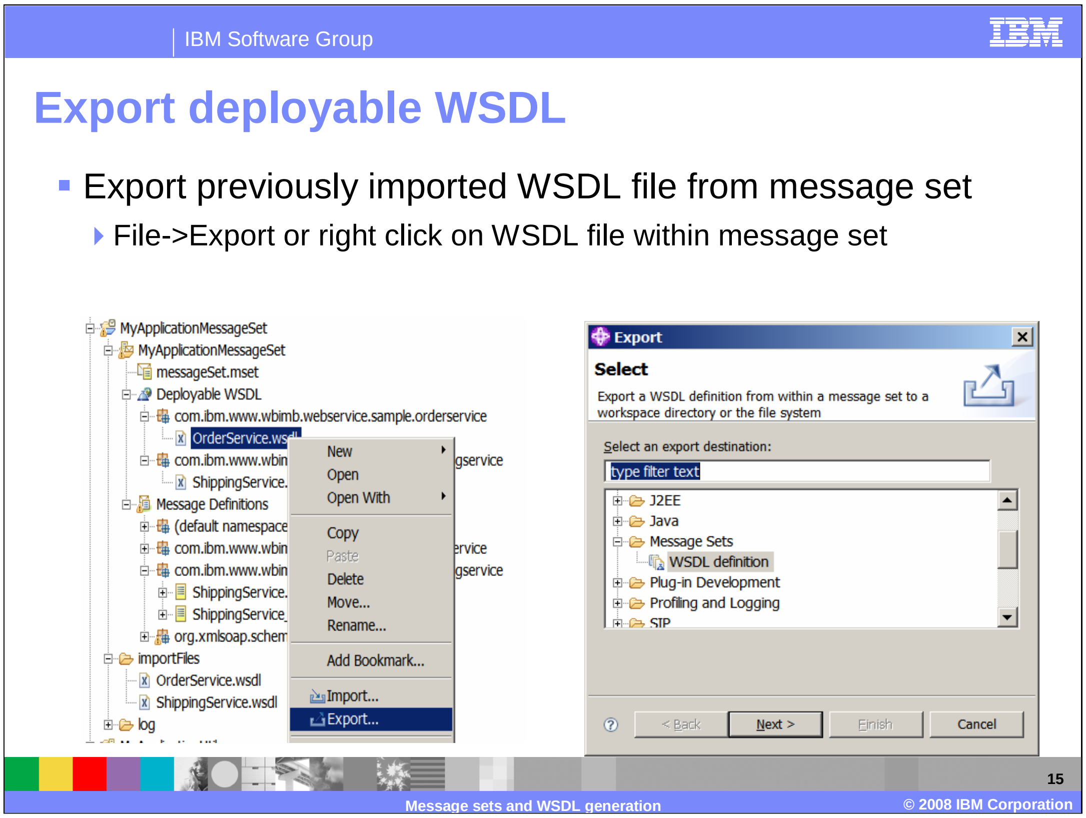
© 2008 IBM Corporation

To start with, once a WSDL file has been imported, opening that WSDL file can be performed using the WSDL editor. This shows the various properties, such as service name, port and binding details. Clicking on the arrow to the right of the element will take you to the message definition editor.

The view can be switched to the XML source editor by selecting the Source view.

Export deployable WSDL

- Export previously imported WSDL file from message set
  - File->Export or right click on WSDL file within message set



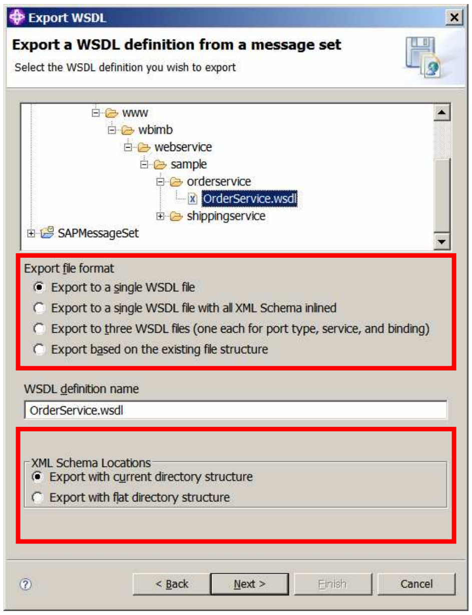
Export a WSDL definition from within a message set to a workspace directory or the file system

Deployable WSDL that has been previously imported can also be exported. The normal Toolkit techniques are used to achieve this.

IBM Software Group

Export deployable WSDL (2)

- WSDL export options
  - Single or 3 files (service, binding, interface) format
  - In-line all schema type definitions irrespective of how the original WSDL referenced schema definition
  - Keep referenced schema definitions location based on directory structure within the message set (based on nsURIs)
  - Put WSDL and referenced schema definitions within a single directory
  - Export WSDL to external directory or workspace



Export a WSDL definition from a message set

Select the WSDL definition you wish to export

Export file format

● Export to a single WSDL file

○ Export to a single WSDL file with all XML Schema inlined

○ Export to three WSDL files (one each for port type, service, and binding)

○ Export based on the existing file structure

WSDL definition name: OrderService.wsdl

XML Schema Locations

● Export with current directory structure

○ Export with flat directory structure

16

Message sets and WSDL generation

© 2008 IBM Corporation

When exporting the WSDL, there are several choices that you can make to control this export. These options are shown in the middle part of the screen capture on this slide.

First, the WSDL can be exported to a single WSDL file. This format is widely understood by external applications and SOAP toolkits.

Second, the WSDL can be exported to three WSDL files, one each for port type, service, and binding. The WSDL definition is split into multiple files. This format offers better reuse of the component files.

The third option is to export to a single WSDL file with all XML schema in-lined. The WSDL definition is written to a single file. This format is widely understood by external applications and SOAP toolkits.

Finally, you can export based on the existing file structure. The WSDL definition is written to a single file. This format is also widely understood by external applications and SOAP toolkits.

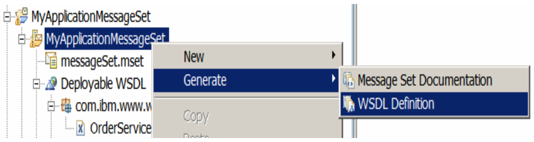
In addition to specifying the type of WSDL file format, you can also specify where to export the WSDL.

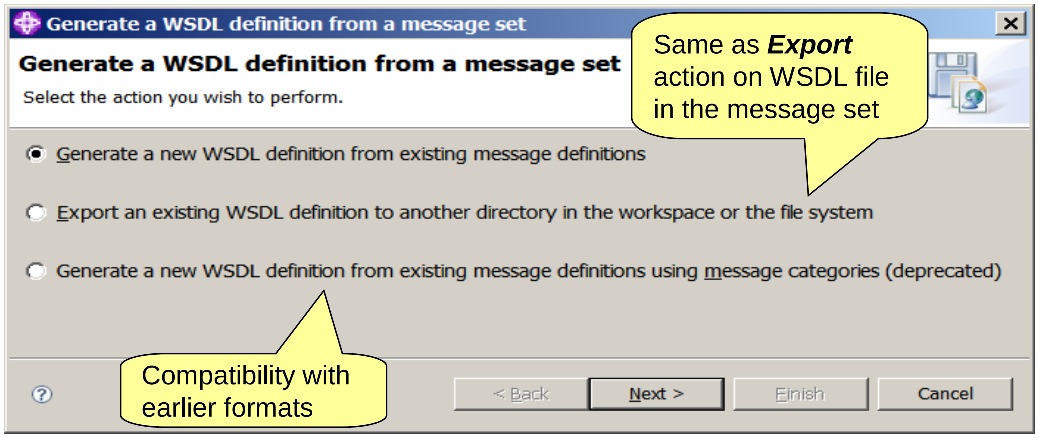
You can export to a workspace directory. The structure of the workspace is displayed. Click the folder that you want the WSDL definition to be exported to.

Or you can export to an external directory. In this case, you specify the name of the external directory that you want the WSDL definition to be exported to.

Generate WSDL without categories

Popup menu on the message set folder ***Generate->WSDL Definition***





Generate a WSDL definition from a message set

Generate a WSDL definition from a message set

Select the action you wish to perform.

● Generate a new WSDL definition from existing message definitions

○ Export an existing WSDL definition to another directory in the workspace or the file system

○ Generate a new WSDL definition from existing message definitions using message categories (deprecated)

Same as ***Export*** action on WSDL file in the message set

Compatibility with earlier formats

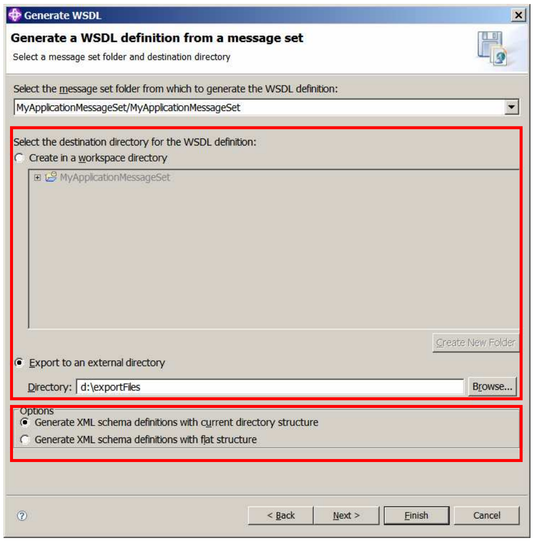
To generate WSDL from a Message Set definition, you should use the a pop-up menu on the message set folder. Right-clicking on this folder allows you to generate the WSDL definition directly from this definition.

Selecting “WSDL Definition”, as shown on this slide, opens a window with three choices. Select the appropriate choice, and complete the wizard.

Note the use of Categories is only provided for backward compatibility, and is no longer required in Version 6.1

Generate WSDL without categories (2)

- WSDL generation options
  - Destination for generated WSDL -> workspace or External directory
  - Generate schema definitions location based on directory structure within the message set (based on nsURIs)
  - Put WSDL and referenced schema definitions within a single directory



Generate WSDL

Generate a WSDL definition from a message set

Select a message set folder and destination directory

Select the message set folder from which to generate the WSDL definition:

MyApplicationMessageSet/MyApplicationMessageSet

Select the destination directory for the WSDL definition:

○ Create in a workspace directory

MyApplicationMessageSet

● Export to an external directory

Directory: d:\exportFiles

Options

● Generate XML schema definitions with current directory structure

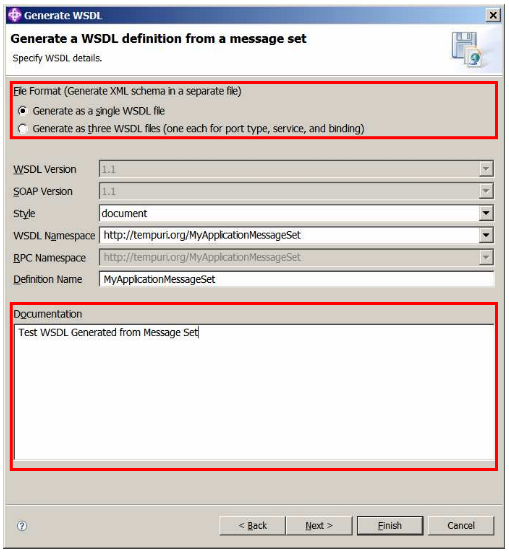
○ Generate XML schema definitions with flat structure

This slide shows the next screen of the wizard, when the first, default option is selected. The user nominates the location for the generated WSDL, either workspace or external directory.

You should also select whether the generated WSDL should be based on a directory structure, or whether the WSDL and referenced schemas should be generated within a single directory.

Generate WSDL without categories (3)

- WSDL generation options
  - Single or three files (service, binding, interface) format
  - Documentation for generated WSDL added using <wsdl:documentation> tags
  - Other options same as in 6.0.2



Generate WSDL

Generate a WSDL definition from a message set

Specify WSDL details.

File Format (Generate XML schema in a separate file)

● Generate as a single WSDL file

○ Generate as three WSDL files (one each for port type, service, and binding)

WSDL Version: 1.1

SOAP Version: 1.1

Style: document

WSDL Namespace: http://tempuri.org/MyApplicationMessageSet

RPC Namespace: http://tempuri.org/MyApplicationMessageSet

Definition Name: MyApplicationMessageSet

Documentation: Test WSDL Generated from Message Set

© 2008 IBM Corporation

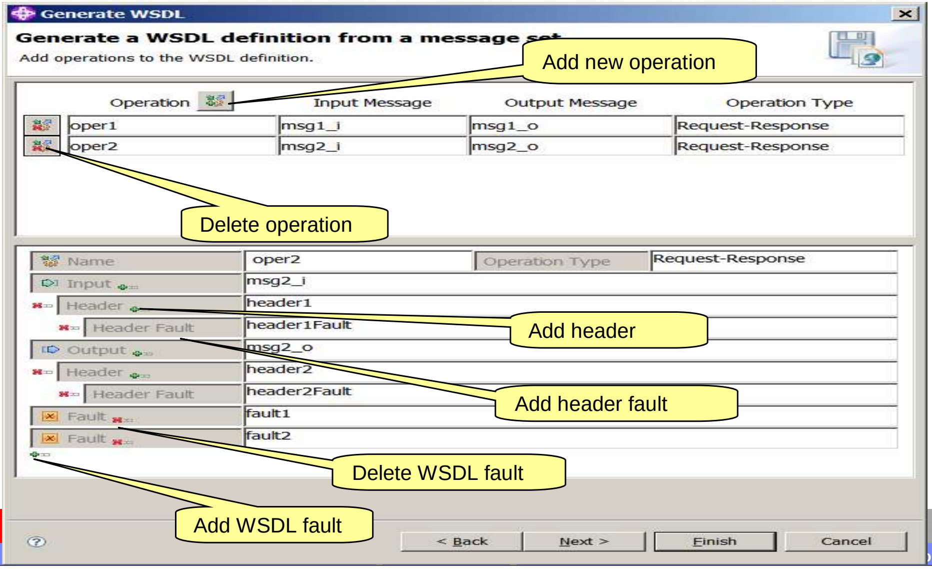
On this screen you enter the WSDL generation options. The format options are the same as those available in the WDSL export.

Any documentation that is required here will be added using WSDL documentation tags. The other options such as style and namespace are the same as in Message Broker Version 6.0.2.

Generate WSDL without categories (4)

Simple wizard to specify WSDL operation and its parameters which are ***messages*** within the message set

- This information was captured through categories in 6.0.2 release



Generate a WSDL definition from a message set

Add operations to the WSDL definition.

Operation	Input Message	Output Message	Operation Type
oper1	msg1_i	msg1_o	Request-Response
oper2	msg2_i	msg2_o	Request-Response

Name: oper2

Operation Type: Request-Response

Input: msg2_i

Header: header1

Header Fault: header1Fault

Output: msg2_o

Header: header2

Header Fault: header2Fault

Fault: fault1

Fault: fault2

Add new operation

Delete operation

Add header

Add header fault

Delete WSDL fault

Add WSDL fault

On this screen you define the operations that you want to add to the WSDL definition.

The top pane shows the operations which have already been defined. You can add further operations on the bottom pane by specifying certain information.

The key pieces of information are:

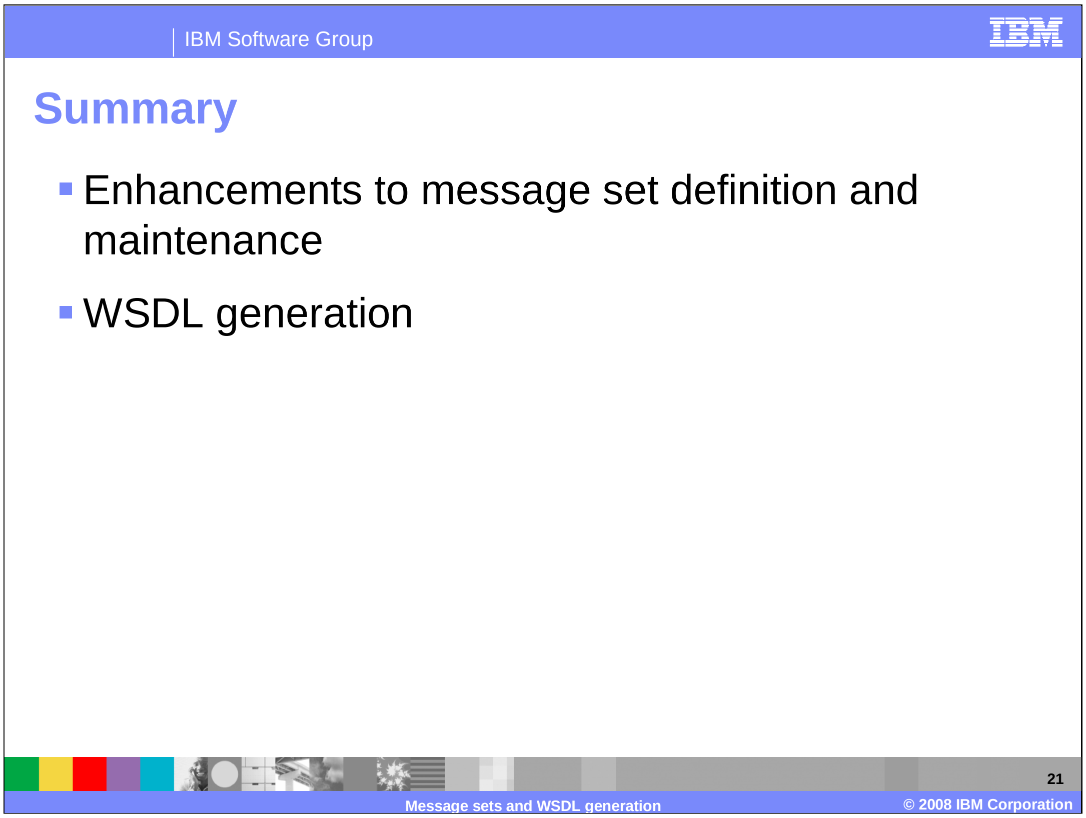
1. The name that you have given to the operation.

2. The type of operation, such as, Request-response, One-way, Solicit-response, Notification.

3. The name of the input message. This is omitted for a Notification operation.

4. The name of the output message. This is omitted for a One-way operation.

And 5, the name of the fault message. This is omitted for a Notification operation. Otherwise, you can specify one or more fault messages.

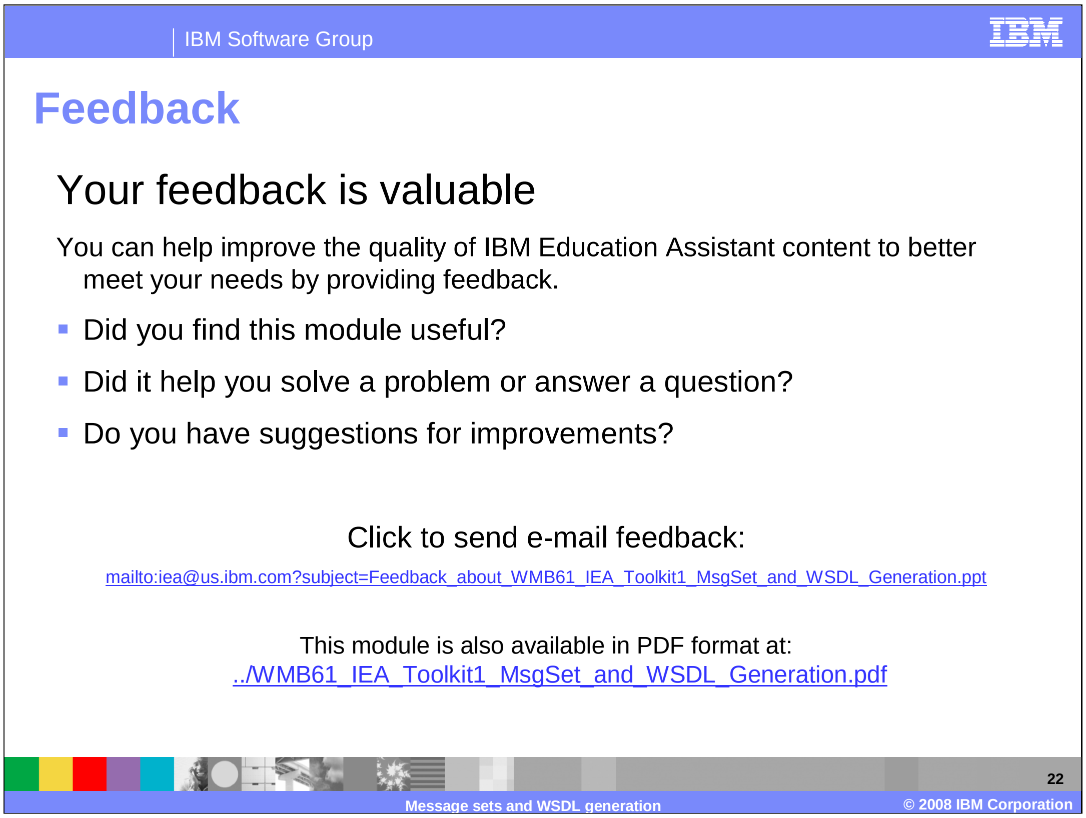


Summary

- Enhancements to message set definition and maintenance
- WSDL generation

© 2008 IBM Corporation

In summary, this session looked at the toolkit enhancements in version 6.1 and has discussed changes to message set definition, and to the generation of WSDL.



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?

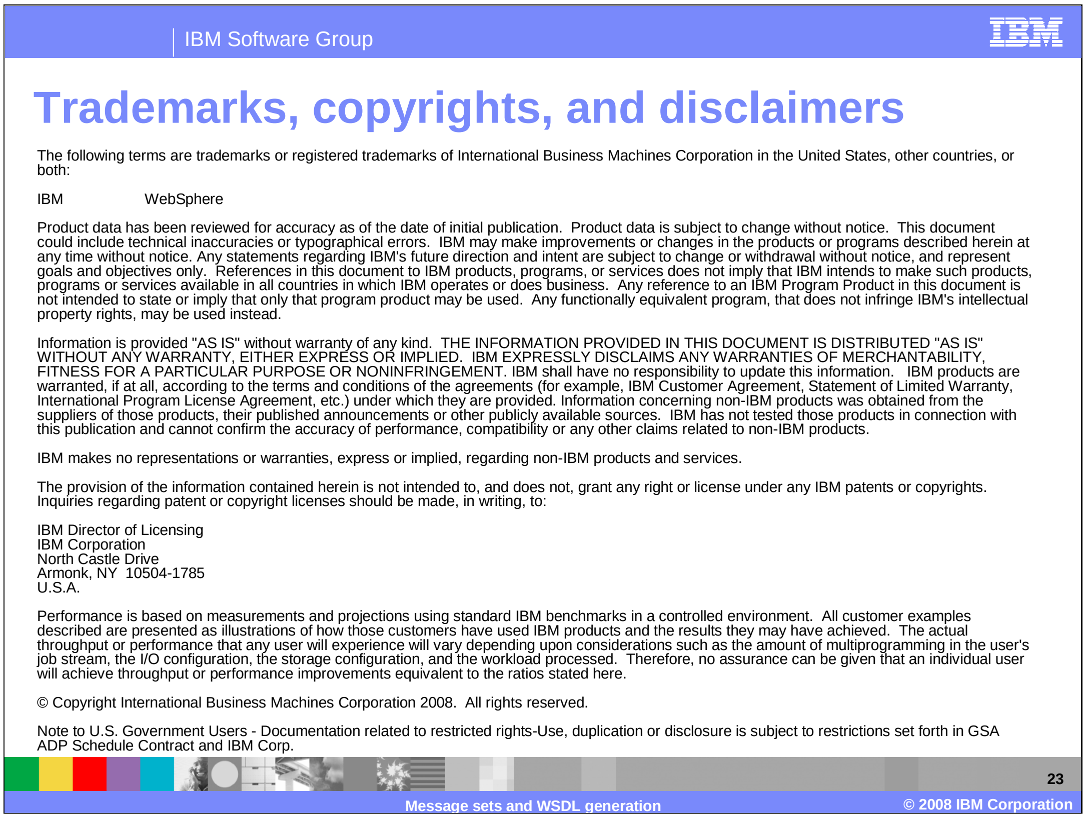
Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB61_IEA_Toolkit1_MsgSet_and_WSDL_Generation.ppt

This module is also available in PDF format at:
../WMB61_IEA_Toolkit1_MsgSet_and_WSDL_Generation.pdf

© 2008 IBM Corporation

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



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 WebSphere

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 or changes in the products or programs 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 (for example, 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 2008. 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.