

WebSphere MQ Integrator Version 2.1 - XML Schema message format importer

Version 1.0

17 May, 2002

WebSphere MQ Integrator Development
IBM United Kingdom Laboratories
Hursley Park, Hursley
Winchester SO21 2JN
United Kingdom

Property of IBM

Take Note!

Before using this report be sure to read the general information under "Notices".

First Edition, May 2002

This edition applies to Version 1.0 of *WebSphere MQ Integrator XML Schema Importer* and to all subsequent releases and modifications unless otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2002**. All rights reserved. Note to US Government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

Table of Contents

| | |
|---------------------------------------|----|
| Notices | 7 |
| Acknowledgments | 8 |
| Summary of Amendments..... | 9 |
| Preface..... | 10 |
| Bibliography..... | 11 |
| Chapter 1. Getting Started | 12 |
| Introduction | 12 |
| Installation..... | 12 |
| Chapter 2. Using the Importer..... | 14 |
| Overview | 14 |
| Example of using the Importer | 14 |
| Overview of the Import operation..... | 17 |
| mqsiimpxmlschema command syntax | 18 |
| Required parameters | 18 |
| -f parameters-file..... | 18 |
| Optional parameters | 18 |
| -n MRM_DB | 18 |
| -u userid | 18 |
| -p password | 18 |
| -o filename | 18 |
| -s messageSet | 18 |
| -x XMLTabName..... | 18 |
| -l directoryPath | 19 |
| -c | 19 |
| -r..... | 19 |
| -v | 19 |
| -t..... | 19 |
| -d..... | 19 |

| | |
|--|----|
| | |
| Chapter 3. XML Schema Support | 23 |
| Introduction | 23 |
| Simple types | 23 |
| Complex types | 23 |
| Attributes | 23 |
| Local and Global elements and attributes | 23 |
| Import, Include and Redefine..... | 23 |
| Recursion | 23 |
| Substitution groups and abstract elements | 24 |
| Nillable elements..... | 24 |
| Annotation, documentation and appinfo elements | 24 |
| Namespaces | 24 |
| Chapter 4. Troubleshooting | 25 |
| Gathering additional information | 25 |
| Logging a call with IBM | 25 |
| Chapter 5. An example of using the Importer | 26 |
| Introduction | 26 |
| The command line used to Import the XML Schema | 26 |
| The Schema files | 27 |
| Customer.xsd..... | 27 |
| Address.xsd | 28 |
| The parameter file | 29 |
| Customer.txt | 29 |
| The log file generated by the Import operation | 30 |
| Customer.log..... | 30 |
| A screen shot showing the imported message set | 31 |
| A Sample Instance Document | 32 |
| Customer.xml..... | 32 |

| | |
|---|----|
| Sample EQSL to copy the message in a compute node | 32 |
| A Sample output message | 33 |
| Appendices..... | 34 |
| Appendix A..... | 34 |
| BIP9351-BIP9423 – XML Schema Importer Messages..... | 35 |

Notices

The following paragraph does not apply in any country where such provisions are inconsistent with local law.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product.

Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS-IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Trademarks and service marks

The following terms, used in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- MQSeries
- WebSphere MQ
- WebSphere MQ Integrator
- WMQI

The following terms are trademarks of other companies:

- Windows NT, Visual Studio Microsoft Corporation

Acknowledgments

I would like to pay special thanks to Jim MacNair for his invaluable contribution to this supportPac. The importer is based on the code written by Jim for the DTD and XML Schema importer released as supportPac id04. Jim also did much of the core work to extend the functionality of the importer for this release.

Summary of Amendments

| Date | Changes |
|-------------|-----------------|
| 17 May 2002 | Initial release |

Preface

This SupportPac provides a command line utility to import XML Schemas into the WebSphere MQ Integrator Message Repository Manager. Throughout this document the WebSphere MQ Integrator V2.1 XML Schema message format importer is referred to as the Importer.

Bibliography

⑩ *WebSphere MQ Integrator Working with Messages Version 2.1*, IBM Corporation. SC34-6039.

⑩ *WebSphere MQ Integrator ESQL Reference Version 2.1*, IBM Corporation, SC34-5923

Chapter 1. Getting Started

Introduction

The Importer imports W3C standard XML Schema into the WebSphere MQ Integrator Message Repository Manager (MRM). The message sets that are created by the Importer can then be used to parse XML instance messages that conform to the XML schema. The message sets are used by MRM domain with Wire Format XML NOT the Generic XML Domain as in SupportPac id04. The imported message sets can be deployed to WebSphere MQ Integrator V2.1 brokers. The message sets can also be used to for drag and drop when writing ESQL within compute nodes. Please refer to the WebSphere MQ Integrator Working with Messages V2.1 manual for details of how to use the Message Repository Manager.

The Importer must be installed on a Windows NT or Windows 2000 system with the Configuration Manager installed. Both the system on which the Importer is installed and any system running the brokers to which the message sets are to be deployed must be running WebSphere MQ Integrator V2.1 (CSD2 or later). The systems running the brokers are not restricted to Windows NT and Windows 2000.

The MRM is not able to model all of the features available within XML Schema. Therefore some features of XML Schema have only limited support and others are not supported. The list of restrictions are listed in the *XML Schema Support* section of this guide. Please read this section carefully to check whether the Importer is able to import your XML schema

Installation

This section describes how to install the SupportPac. The SupportPac must be installed on a Windows NT or Windows 2000 system running WebSphere MQ Integrator V2.1 (CSD2 or later) with the Configuration Manager installed.

Installing the Importer

This SupportPac is supplied as a single compressed format file that is available from the MQSeries Web site, <http://www.ibm.com/software/mqseries/txppacs>

IO01.zip contains the following files:

| | |
|----------------------|--|
| WMQII001.properties | This includes the set of messages issued by this SupportPac. |
| mqsilmpXMLSchema.exe | The executable image that runs the importer. |
| IO01.pdf | This Document, a description of the SupportPac. |
| LIC_*.txt | Licensing files in different languages |
| *.txt, *.xsd, *.xml | Example files |

Installation on Windows NT and Windows 2000

This section describes how to install the remote commands on the Windows platform.

1. Locate the installation directory for the WebSphere MQ Integrator Configuration Manager. By default this would be "C:\Program Files\IBM\WebSphere MQ Integrator 2.1".
2. Copy file *IO01.zip* to this directory and decompress it.
3. Delete file *IO01.zip*.
4. Follow the instructions in the chapter "Using the Importer".

Installation Directories

The SupportPac files are installed in the following directories. Note that install_dir is the directory in which the WebSphere MQ Integrator V2.1 Configuration Manager is installed.

IO01.zip:

| Directory | Files |
|---------------------------|---|
| install_dir\bin | MqsImpXMLSchema.exe |
| install_dir\messages | WMQIO01.properties |
| install_dir\book\IO01 | IO01.pdf |
| install_dir\book\IO01 | Lic_da.txt, Lic_de.txt, Lic_en.txt, Lic_es.txt, Lic_fi.txt, Lic_fr.txt, Lic_it.txt, Lic_ja.txt, Lic_ko.txt, Lic_nl.txt, Lic_no.txt, Lic_pl.txt, Lic_pt.txt, Lic_ru.txt, Lic_sv.txt, Lic_zh.txt, Lic_zh_TW.txt |
| install_dir\examples\IO01 | address.xsd, customer.txt, customer.xml, customer.xsd, example.txt, example.xsd |

DB2 Log file size

The import utility will perform all database operations as a single unit of work. This may require that the size of the DB2 log files be increased, particularly if a large DTD is to be imported.

If the maximum log size is exceeded, error messages should be produced in the application log in the Windows event log. The error messages can be viewed with the Windows event viewer.

To increase the size of the log file, start the DB2 control center and then open the views until the individual databases are visible. Select the MRM database and click on it using the right mouse button. Select the logs tab.

On the Configure Database notebook, on the Logs page, when you click on the Log File Size parameter, the Hint text says the range for NT is [4 - 4095] Pages (one page is 4K bytes). For NT, the range is actually from four to 65535 pages (one page is 4K bytes).

Potential problem parsing MRM XML messages

Under some circumstances in WMQI V2.1 CSD2 a broker can abend when parsing an MRM XML message. An e-fix is available for this problem at the following FTP location :

<ftp://ftp.software.ibm.com/software/mqseries/fixes/wmqiv21/IY30451> Alternatively please log a call with your IBM Support Center quoting reference IY30451.

Chapter 2. Using the Importer

Overview

The Importer is a command line utility that is invoked using the `mqsiiimportxmlschema` command. The command takes as input a parameter file that controls the import process. Additional command line options can also be specified which override some of the parameters in the parameter file. The detailed syntax of the command and the parameter file are documented in following subsections.

The following mandatory information must be specified on the command line or within the parameter file :

- The MRM database name
- The MRM database administrator user name
- The MRM database administrator user password
- The name of the MRM Message Set into which the XML Schema is to be imported. This message set will be created if it does not already exist.

The following mandatory information must be specified in the parameter file :

- The name of the top level XML schema file to be imported. It is possible for the XML schema to consist of multiple files but only the location of the top level file is mandatory.
- The name of the XML root element to be imported. This element is used as the name of the message within the MRM. Only XML constructs that are referenced directly or indirectly by this root element are imported into the MRM.

Example of using the Importer

This subsection shows how to use the importer via a simple example. The `example.txt` and `example.xsd` are provided in the `install_dir/examples\IO01` directory. The `example.txt` file will need to be modified to reflect your database name, username and password :

`example.xsd` (The XML schema to be imported)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="Example" type="ExampleType"/>
<xsd:complexType name="ExampleType">
  <xsd:sequence>
    <xsd:element name="field1" type="xsd:string"/>
    <xsd:element name="field2" type="xsd:integer"/>
    <xsd:element name="field3" type="xsd:decimal"/>
    <xsd:element name="field4" type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

`Example.txt` (The parameter file used for the import)

```
#
# Parameters file for MQSI V2.1 XML Schema importer
#
# Name of the message set to import the message into
#
MessageSet=ExampleMsgSet
#
# Name of the MRM database to import into
#
MRM_DB=MQSIMRDB
#
```

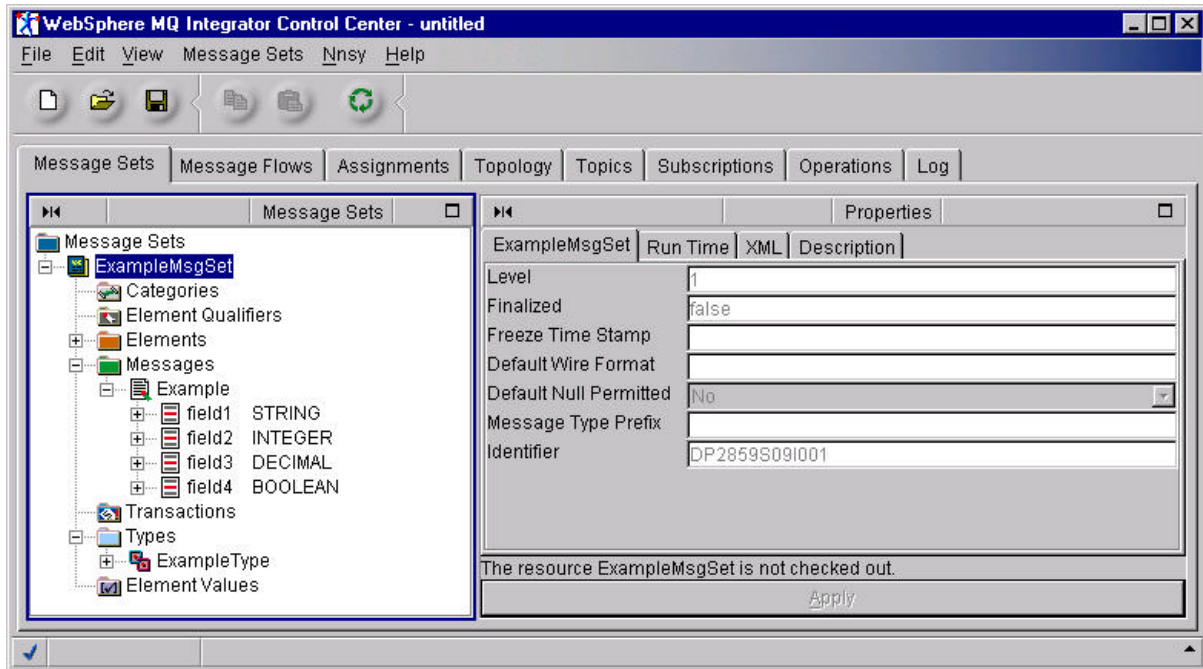
WebSphere MQ Integrator XML Schema Importer

```
# MRM Database User id - can be overridden on the command line
#
DB_User=db2admin
#
# MRM Database Password - can be overridden on the command line
#
DB_PW=password
#
# Name of the top level element in the MRM message
#
RootElement=Example
#
[FILELIST]
#
# Name of the XML Schema file
#
example.xsd
```

The command line to run the import.

```
Mqsiimpxmlschema -f example.txt
```

The structure of the imported message set as seen from the WMQI Control Center :



Overview of the Import operation

The Importer uses the Xerces Dom parser to parse the XML Schema files. If the XML Schema files do not contain well formed XML an error will be returned by the Importer. All of the XML Schema files must be available locally, the URI specifications are not used directly to locate the Schema files. The Importer is not intended to be an XML Schema validator. The Importer assumes that it will be passed valid XML Schema files but if it finds an error it will try to stop the import operation as soon as possible.

The Importer checks whether the Message Set into which the XML schema is to be imported already exists. If it does not exist the Importer will create the Message Set as part of the Import operation. If the Message Set already exists it will attempt to import the XML Schema into this Message Set. If the Message Set is finalized, frozen or there are components checked out an error will be returned by the Importer.

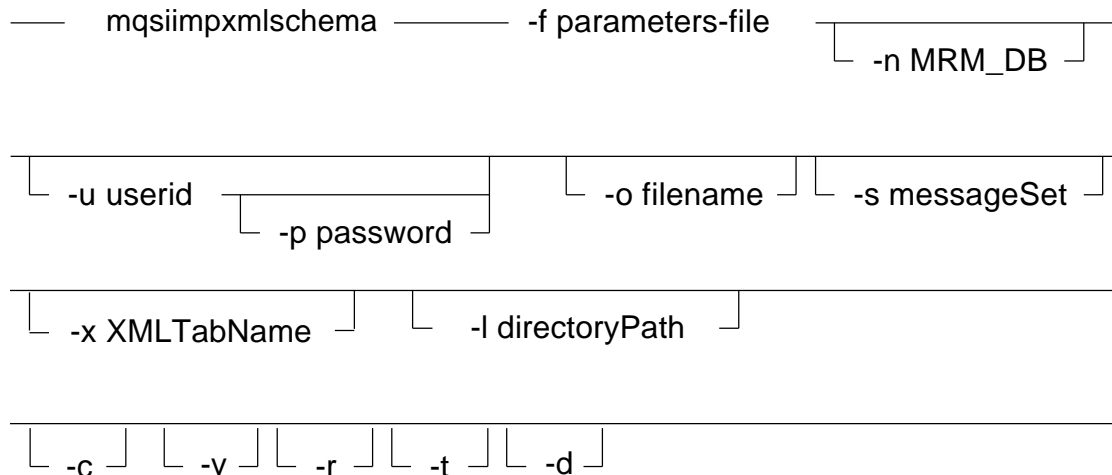
If importing into an existing Message Set and the Message corresponding to the XML root tag to be imported already exists the Importer will return an error. It is not possible to use the Importer to update an existing Message definition. If you wish to update an existing Message definition you must either delete the entire Message Set before commencing the import or you must use the reset option of the Importer. The reset option will remove all existing information from the Message Set before the import of the XML Schema is started. This has the advantage over deletion in that the Message Set Identifier will remain the same so the Input node or RFH2 header will not need to be updated to reflect a new Message Set Identifier. The reset option is specified using the `-c` command line flag or the `ResetMsgSet` parameter. By default the Message Set will not be reset.

It is possible to Import multiple root elements into the same Message Set from the same set of XML Schema files by running the Importer multiple times. On each run of the Importer specify a different root element name in the parameter file. It is not advisable to import from separate sets of XML Schema files into the same Message Set because there may be conflicts of XML element and attribute names.

The Importer does all updates within a single Unit of Work. If any errors are detected during the import of an XML Schema the Unit of Work is not committed and the MRM database is not updated. If you wish to try an import operation but not commit the changes to the MRM database the importer can be run in report mode.

mqsiiimpxmlschema command syntax

This subsection describes the detailed syntax of the mqsiiimpxmlschema command. The use of some of the options is described in more detail in subsequent sections. The option flags can be specified in upper or lower case.



Required parameters

-f parameters-file

The name of the XML Schema Importer parameters file.

Optional parameters

-n MRM_DB

The name of the MRM repository database. The database name can be specified in the parameter file instead of on the command line.

-u userid

The username to access the MRM database. The user must have permissions to modify the MRM database. The username can be specified in the parameters file instead of on the command line.

-p password

The password of the user to access the MRM database. The password can be specified in the parameters file instead of on the command line.

-o filename

If this option is specified a line will be written to this file containing the Message Set name followed by the Message Set Identifier, separated by an '=' character. If the file already exists the line will be appended to this file.

-s messageSet

The name of the message set to be created. The Message Set can be specified in the parameters file instead of on the command line.

-x XMLTabName

The name to be given to the XML wire format tab. The XMLTabName can be specified in the parameters file instead of on the command line. If no value is specified on the command line or in the parameters file the value will default to XML.

-l directoryPath

The directory path to be placed before XML Schema file names. The directory path is applied to schema file names that are obtained from schemaLocation attributes in the schema files and the those specified in the [FILELIST] and [FILELOCATION] sections of the parameter file. To use this parameter all the schema files must be located in the same directory or subtree. The directory path can also be specified in the parameter file instead of on the command line.

-c

This option will (clean) remove all content from an existing MRM message set before importing the XML Schema. If creating a new message set this flag will have no effect. The advantage of using this method rather than deleting the message set before the import is that the Message Set Identifier will remain unchanged.

The following options are intended to be used for debugging purposes

-r

This option runs the Importer in report mode only. i.e. It does not update the database. This allows you to check whether any errors are reported by the Importer without making any permanent changes to the MRM database.

-v

This option instructs the Importer to run in verbose mode. This will cause more detailed messages to be reported during a run of the Importer.

-t

This option instructs the Importer to display trace information. This information will be useful for IBM support personnel to troubleshoot problems.

-d

This option instructs the importer to dump the Schema tags as processed and to dump internal tables. This information will be useful for IBM support personnel to troubleshoot problems.

The format of the Importer parameter file

The Import parameter file is split into the 4 sections. The start of each section in the parameter file is indicated by the section name in square brackets. Comment lines can be specified in the parameters file by starting lines with "#", ";", or "*". In the [HEADER] section a number of parameters can be specified. The parameter names are not case sensitive and must be separated from their values by an "=" character. Only one parameter is allowed per line. Any options specified on the command line will override their equivalent value in the parameter file. The sections and parameters are described below :

[HEADER] section

The header section contains a number parameters as described in the list below.

MessageSet=MsgSetName

The name of the Message Set to create into which to import the message. If the message set does not already exist the Importer will create the Message Set. This is the equivalent of the -s command line option.

XMLTabName=name

The name to be given to the XML wire format layer. This is the equivalent of the -x command line option.

RootElement=name

The name of the root element within the XML Schema. The message in the MRM will be given this name.

MRM_DB=DBName

The name of the MRM database. This is the equivalent of the -n command line option.

DB_USER=UserName

The name of the database user. This is the equivalent of the -u command line option.

DB_PW=Password

The password of DB_USER. This is the equivalent of the -p command line option.

MsgSetFileName=filename

If this parameter is specified a line will be written to the file containing the Message Set name followed by the Message Set Identifier, separated by an '=' character. If the file already exists the line will be appended to the file. This is the equivalent of the -o command line option.

FilePath=directoryPath

The directory path to be placed before XML Schema file names. The directory path is applied to schema file names that are obtained from schemaLocation attributes in the schema files and the those specified in the [FILELIST] and [FILELOCATION] sections of the parameter file. To use this parameter all the schema files must be located in the same directory or subtree. This is the equivalent of the -l command line option.
Eg. FilePath="[c:\direc\subdirec](#)"

ResetMsgSet=(Y|N)

This parameter will (clean) remove all content from an existing MRM message set before importing the XMLSchema. If creating a new message set this flag will have no effect. The advantage of using this method rather than deleting the message set before the import is that the Message Set Identifier will remain unchanged. This is the equivalent of the -c command line option.

DefNameSpace=NameSpaceURI

The URI of the default namespace. If this parameter is omitted there is no default namespace. An xmlns attribute will be created under the message for the default namespace.

xsiPrefix=prefix

The prefix to be used for the XML Schema Instance Namespace. By convention this prefix is normally set to xsi. This will be used to write the xmlns attribute under the message tag.

Eg. XsiPrefix=xsi.

SchemaLoc="namespec location"

The Namespace and Location of the XML Schema file to be referenced in an instance document. An xsi:schemaLocation attribute will be created under the message tag for the schema location.

Eg. SchemaLoc=" http://www.ibm.com/example example.xsd"

NoNamespaceLoc="location"

The location of an XML Schema file to be referenced by an instance document when the XML Schema file has no target namespace. An xsi:noNamespaceschemaLocation attribute will be created under the message tag for the no namespace schema location.

Eg. NoNamespaceLoc="example.xsd"

DateStringOnly=(Y|N)

If this parameter is set to Y all XML date and time data types are modeled using the MRM STRING data type not the MRM DATETIME data type. The default value for this parameter is N.

InclTimeFrac=(Y|N)

If this parameter is set to Y fractions of a second will be included in the MRM DATETIME format for the XML data types dateTime and time. The default value for this parameter is N.

The following parameters are intended to be used for debugging purposes

ReportOnly=(Y|N)

This is the equivalent of the -r command line option.

Verbose=(Y|N)

This is the equivalent of the -v command line option.

Trace=(Y|N)

This is the equivalent of the -t command line option.

ShowSchema=(Y|N)

This is the equivalent of the -d command line option.

[FILELIST] section

This section contains a line with the name of the master XML schema file to import.

[NAMESPACEPREFIX] section

This section contains mappings of Namespace URI specifications to Namespace prefixes. Each line contains a URI followed by its corresponding prefix. If there is no Namespace prefix defined in the parameter file for a Namespace the prefix specified in the xmlns attribute in the XML schema file will be used. For more information see the section on namespaces.

Eg. `http://www.ibm.com/example exmpl`

[FILELOCATION] section

This section contains mappings between Namespace URI specifications to local filenames containing the XML schema definitions. Each line contains a URI followed by its corresponding filename. If there is no definition for a namespace in this section the value of the SchemaLocation attribute of the import or include statement within the XML schema will be used.

Eg. `http://www.ibm.com/example example.xsd`

Chapter 3. XML Schema Support

Introduction

Although the MRM contains a rich set of Message Modeling features it is not able to model all of the constructs that can be used in XML Schema. Therefore the Importer is not able to fully support all XML Schema features. This section describes the XML Schema features that are supported and those for which there is limited or no support.

Simple types

The Importer supports simple data types. There are more XML Schema built-in data types than the 7 MRM built-in data types. Therefore the Importer maps an XML Schema built-in data type to its nearest MRM built-in data type. See appendix A for the mappings.

XML Schema simple data types can be restricted using facets. Some of the facets can be modeled by the MRM using value constraints but others have no direct equivalent in the MRM so they are ignored. The facets enumeration, maxInclusive, minInclusive, maxLength and minLength are supported using value constraints. The maxExclusive and minExclusive facets are modelled using the maxInclusive and minInclusive value constraints. If the maxExclusive or minExclusive facets are encountered by the Importer a warning message is displayed. If the datatype of the element being restricted is an integer the facet value will be decremented by 1 for the maxExclusive facet and incremented by 1 for the minExclusive facet. For all other data types the value will remain unchanged.

Complex types

Complex types are supported in the MRM by using compound types. The Simple, Element, Mixed and empty content types are supported. The model groups sequence, choice and all are supported. Complex type extensions and restrictions are supported.

Attributes

XML attributes are modeled in the MRM using Elements. The Importer sets the Element Render and Member Render properties to XMLAttribute for these elements.

Local and Global elements and attributes

In XML Schema elements and attributes can be defined to be local or global. However in the MRM all elements are global. It is possible for there to be more than 1 local element or attribute with the same name which could cause a name conflict within the MRM. To prevent any name conflicts the MRM Identifiers of elements are prefixed with their type identifier. This guarantees the uniqueness of Identifiers. The prefix is separated from the element name by a caret (^) character in the Identifier name. When generating ESQLE using the Control Center only the data following caret character should be used for the element identifier names. This is done automatically when using the drag and drop facility.

References to Global elements within element declarations using the ref attribute are supported.

Import, Include and Redefine

The importer supports the import and include attributes to access schema files within other schema files. The redefine attribute is not supported.

Recursion

Recursion is supported by XML but not by the MRM. If the Importer finds an element being used recursively the element will be replaced in the MRM with an embedded compound type of composition empty and content open. This will allow the XML parser to parse the data as self defining.

Substitution groups and abstract elements

It is not possible to model substitution groups within the MRM. If the substitutiongroup attribute is found by the Importer it is ignored and a warning is displayed. If the Importer finds an element with the 'abstract' attribute set to true it will replace the element with an embedded compound type of composition empty and content open. This will allow the XML parser to parse the data as self defining.

Nilable elements

The nilable attribute is supported on simple elements by using the Null Permitted value constraint within the MRM. The nilable attribute is not supported on complex elements. If the Importer finds a nilable attribute on a complex element it is ignored.

Annotation, documentation and appinfo elements

The annotation, documentation and appinfo elements are ignored by the Importer. Any data under these elements is also ignored.

Namespaces

The MRM does not have any direct support for namespaces. Therefore the Importer only provides limited support for the import of XML Schemas that use namespaces. If using namespaces all elements in the instance documents both local and global must be prefixed with a namespace prefix. It is not possible to specify the namespace of an element in the instance document by changing the default namespace. The prefix to be used in the instance document must be the same as that specified by the Importer. The reason for this is that the Importer sets the XML name to be 'prefix:element' and at run time the parser will use the XML name to identify the elements in the instance document. By default the prefix used by the importer will be the prefix specified against the xmlns attribute for the namespace, in the XML Schema file. However this can be overridden by the specifying the prefix of a namespace in the [NAMESPACEPREFIX] section of the parameter file.

The importer always qualifies global elements and attributes with the namespace prefix. The Importer determines whether local elements and attributes are to be qualified by determining whether the form attribute has been specified for an element or attribute. If no form attribute has been specified it checks whether the attributes elementFormDefault for an element or attributeFormDefault for an attribute have been specified against the schema. If none have been specified the default for both local elements and local attributes is unqualified.

Each namespace used in the XML Schema will be in a separate file. By default the importer uses the schemaLocation attribute of the import element to locate a schema file. However the location can be overridden by specifying a file location for a namespace in the [FILELOCATION] section of the parameters file. Similarly for XML Schema files that are included the Importer uses the schemaLocation attribute of the include element.

If the defNamesSpace parameter is specified in the parameter file the xmlns attribute for the default namespace is created under the message. Many instance documents specify an xmlns attribute for the XML Schema Instance namespace but this is not specified in the Schema file. The prefix for the XML Schema Instance namespace can be specified using the xsiPrefix parameter in the parameter file. The value specified for this would normally be 'xsi'. Instance documents also specify the location of the XML Schema file to which it corresponds using the xsi:schemaLocation attribute. This can be specified using the schemaLoc parameter in the parameters file. Similarly the location that corresponds to the xsi:noNamespaceSchemaLocation attribute can be specified using the noNamespaceLoc parameter. The prefix used for these 2 attributes will be that specified for the xsiPrefix parameter. Although elements are created in the model for these attributes the values will not be defaulted in the output message. They must be either copied from the input message or set using ESQL in a compute node.

Chapter 4. Troubleshooting

Gathering additional information

If an error occurs during the import of an XML schema check any errors that are reported by the Importer. The descriptions of the messages that can be generated are listed at the end of the document. By default all errors are written to the screen. To write the output to a file put the '> filename' flag on the end of the command line.

There are options that can be specified either on the command line or in the parameters file to provide more diagnostic information. The -v command line option or the verbose parameter displays more detailed information as the import proceeds. The -t command line option or the trace parameter gives detailed information about the internal functions that are called. This option is primarily intended for IBM support. The -d command line option or the showschema parameter dumps out details of internal tables used by the Importer. This is also primarily intended for use by IBM support.

Logging a call with IBM

If you need to log a call with IBM support please provide the following information :

- The parameter file used for the import
- All of the XML Schema files
- If importing into an existing message set extract the message set to a file using the mqiimpexpmsgset command line utility.
- If the XML schema imports without error but the message instances are not parsed correctly at run time, provide a copy of the instance document(s) and the message flow that is being used.

Chapter 5. An example of using the Importer

Introduction

The following example is more a complex example of using the Importer than that given in chapter 2. In this example there are multiple namespaces and the schema is split into multiple files. The main schema holds customer details and this schema file imports another schema file which holds address details. The example uses simple types, complex types, simple content, facets, sequences, choices and extensions. The XML schema defines customer details and the customer address details in separate namespaces. The customer details are in namespace `http://www.ibm.com/customer`, which is in file `customer.xsd` and the prefix used for the namespace is `cust`. The customer address details are in namespace `http://www.ibm.com/address`, which is in file `address.xsd` and the prefix used for the namespace is `addr`. The Importer obtains the prefixes from the XML Schema. The location of `customer.xsd` is specified in the [FILELOCATION] section of the parameter file and the location of `address.xsd` is obtained from the `xsd:import` element of `customer.xsd`. The XML Schema files, the parameter file, the log of the Import, a screen shot of the imported message set as seen in the Control Center and a sample instance document are shown below. Also shown is some sample ESQL to copy the fields of the message in a compute node and the output document from a flow containing the compute node and using the sample instance document as input.

The `customer.txt`, `customer.xsd`, `address.xsd` and `customer.xml` are provided in the `install_dir\examples\IO01` directory. The `customer.txt` file will need to be modified to reflect your database name, username and password.

The command line used to Import the XML Schema

MqsilmpXMLSchema -f Customer.txt > Customer.log

The Schema files

Customer.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:cust="http://www.ibm.com/customer"
            targetNamespace="http://www.ibm.com/customer"
            xmlns:addr="http://www.ibm.com/address"
            elementFormDefault="qualified">
  <xsd:import namespace="http://www.ibm.com/address" schemaLocation="address.xsd"/>
  <xsd:element name="customerDetails" type="cust:customerType"/>
  <xsd:complexType name="customerType">
    <xsd:sequence>
      <xsd:element name="salutation">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="letters" type="xsd:string" use="optional"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="firstName">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:minLength value="1"/>
            <xsd:maxLength value="20"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="lastName">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:minLength value="1"/>
            <xsd:maxLength value="30"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element ref="addr:addressDetails"/>
    </xsd:sequence>
    <xsd:attribute name="sex">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Male"/>
          <xsd:enumeration value="Female"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:schema>

```

Address.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.ibm.com/address"
  xmlns:addr="http://www.ibm.com/address"
  elementFormDefault="qualified">
<xsd:element name="addressDetails" type="addr:addressType" />
<xsd:complexType name="addressType">
  <xsd:sequence>
    <xsd:element name="house">
      <xsd:complexType>
        <xsd:choice>
          <xsd:element name="no">
            <xsd:simpleType>
              <xsd:restriction base="xsd:integer">
                <xsd:minInclusive value="1"/>
                <xsd:maxInclusive value="9999"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="name">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="20"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="addressLine" type="xsd:string" minOccurs="2" maxOccurs="4" />
    <xsd:element name="postCode">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:minLength value="6"/>
          <xsd:maxLength value="8"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

The parameter file

Customer.txt

```
# Parameters file for WQMI V2.1 XML Schema importer utility
#
#
# Name of the message set to import the message into
#
MessageSet=Customer
#
# Name of the MRM database to import into
#
MRM_DB=MQSIMRDB
#
# MRM Database User id - can be overridden on the command line
#
DB_User=db2admin
#
# MRM Database Password - can be overridden on the command line
#
DB_PW=*****
#
# Name of the top level element in the MRM message
#
RootElement=customerDetails
#
# List of schema files appears after the [FILELIST] section header
# Each file name must be qualified if necessary and on a separate line
#
# The prefix to be used for the XML Schema Instance namespace
#
xsiPrefix=xsi
#
# The location of the schema to output in an instance document
#
SchemaLoc="http://www.ibm.com/customer customer.xsd"
#
# The filename of the main schema file
#
[FILELIST]
customer.xsd
#
# The prefixes to be used by the name spaces
#
[NAMESPACEPREFIX]
```

The log file generated by the Import operation**Customer.log**

```

XML schema Import Utility V1.00 for WMQI V2.1 - Copyright (C) IBM Corp.
2001 (Apr 12 2002)
Execution date and time 2002-04-12-15.26.28
Processing parameters file customer.txt
Connection successful MQSIMRDB
Accessing MRM database: MQSIMRDB using: db2admin schema:
BIP9404I: The Identifier for Message Set Customer is DP2859S0BE001.
Checking for duplicate message customerDetails in MessageSet Customer
reading in the schema files
Parsing the main XML schema file
BIP9406I: The Xerces parser has been successfully initialized.
Informational message - no action required.
BIP9408I: The Document Object Model (DOM) Tree was built successfully for
file 'customer.xsd'.
Informational message - no action required.
BIP9408I: The Document Object Model (DOM) Tree was built successfully for
file 'address.xsd'.
Informational message - no action required.
total number of XML tags found in Schema files 48

Totals
element tags found in schema 11
schema tags 2
attribute tags found in schema 2
attributeGroup tags found in schema 0
group tags found in schema 0
complexType tags found in schema 4
simpleType tags found in schema 6
complexContent tags found in schema 0
simpleContent tags found in schema 1
restriction tags found in schema 6
extension tags found in schema 1

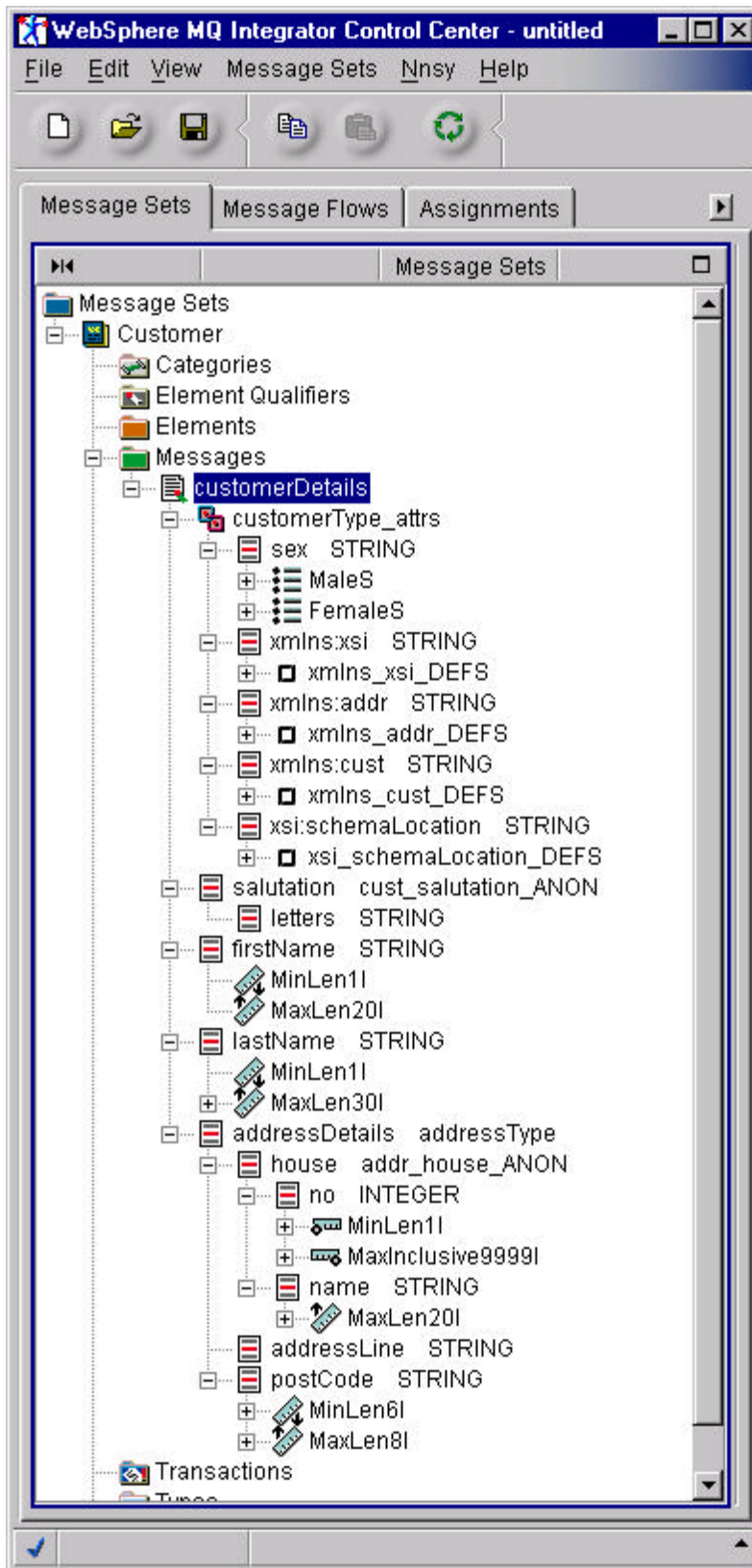
root element (customerDetails) found as element 3
XML Wire format layer XML successfully added
items added to tables in repository database MQSIMRDB
M_MESSAGE          - 1
M_ELEMENT          - 15
M_VALUE           - 12
VALUE_MEMBER      - 15
M_TYPE            - 5
TYPE_MEMBER       - 16
M_ATTRIBUTE       - 27
MEMBER_ATTRIBUTE  - 21

elements reused from existing entries in MessageSet - 0
types reused from existing entries in MessageSet   - 0
values reused from existing entries in MessageSet   - 1

duplicate elements reused in MessageSet - 0
duplicate values reused in MessageSet   - 0
Total elapsed time = 4.7
BIP9350I: The message 'customerDetails' has been successfully imported into
Message Set 'Customer', Identifier DP2859S0BE001.
Informational message - no user action required.

```

A screen shot showing the imported message set



A Sample Instance Document

Customer.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<cust:customerDetails xmlns:cust="http://www.ibm.com/customer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:addr="http://www.ibm.com/address"
  xsi:schemaLocation="http://www.ibm.com/customer customer.xsd"
  sex="Male">
  <cust:salutation letters="Bsc Hons">Mr</cust:salutation>
  <cust:firstName>Fred</cust:firstName>
  <cust:lastName>Bloggs</cust:lastName>
  <addr:addressDetails>
    <addr:house>
      <addr:no>20</addr:no>
    </addr:house>
    <addr:addressLine>The Street</addr:addressLine>
    <addr:addressLine>The Area</addr:addressLine>
    <addr:addressLine>The Town</addr:addressLine>
    <addr:postCode>AA1 ZZ9</addr:postCode>
  </addr:addressDetails>
</cust:customerDetails>
```

Sample ESQL to copy the message in a compute node

```
DECLARE I INTEGER;
SET I = 1;
WHILE I < CARDINALITY(InputRoot.*[ ]) DO
  SET OutputRoot.*[I] = InputRoot.*[I];
  SET I=I+1;
END WHILE;
SET "OutputRoot"."MRM"."sex" = "InputBody"."sex";
SET "OutputRoot"."MRM"."xmlns_xsi" = "InputBody"."xmlns_xsi";
SET "OutputRoot"."MRM"."xmlns_cust" = "InputBody"."xmlns_cust";
SET "OutputRoot"."MRM"."xmlns_addr" = "InputBody"."xmlns_addr";
SET "OutputRoot"."MRM"."xsi_schemaLocation" = "InputBody"."xsi_schemaLocation";
SET "OutputRoot"."MRM"."salutation" = "InputBody"."salutation";
SET "OutputRoot"."MRM"."salutation"."letters" = "InputBody"."salutation"."letters";
SET "OutputRoot"."MRM"."firstName" = "InputBody"."firstName";
SET "OutputRoot"."MRM"."lastName" = "InputBody"."lastName";
SET "OutputRoot"."MRM"."addressDetails"."house"."no" =
  "InputBody"."addressDetails"."house"."no";
SET "OutputRoot"."MRM"."addressDetails"."addressLine"[1] =
  "InputBody"."addressDetails"."addressLine"[1];
SET "OutputRoot"."MRM"."addressDetails"."addressLine"[2] =
  "InputBody"."addressDetails"."addressLine"[2];
SET "OutputRoot"."MRM"."addressDetails"."addressLine"[3] =
  "InputBody"."addressDetails"."addressLine"[3];
SET "OutputRoot"."MRM"."addressDetails"."postCode" =
  "InputBody"."addressDetails"."postCode";
-- Enter SQL below this line. SQL above this line might be regenerated, causing
any modifications to be lost.
```

A Sample output message

The elements and attributes have been split into separate lines for readability.

```
<?xml version="1.0"?>
<!--MRM Generated XML Output on :Thu Mar 28 16:37:31 2002-->
<cust:customerDetails
  sex="Male"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cust="http://www.ibm.com/customer"
  xmlns:addr="http://www.ibm.com/address"
  xsi:schemaLocation="http://www.ibm.com/customer customer.xsd">

  <cust:salutation letters="Bsc Hons">Mr</cust:salutation>
  <cust:firstName>Fred</cust:firstName>
  <cust:lastName>Bloggs</cust:lastName>
  <addr:addressDetails>
    <addr:house>
      <addr:no>20</addr:no>
    </addr:house>
    <addr:addressLine>The Street</addr:addressLine>
    <addr:addressLine>The Area</addr:addressLine>
    <addr:addressLine>The Town</addr:addressLine>
    <addr:postCode>AA1 ZZ9</addr:postCode>
  </addr:addressDetails>
</cust:customerDetails>
```


Appendices

Appendix A

The following tables shows the MRM data types to which the XML Schema built-in data types are mapped by the Importer. For the XML data types that map to the MRM DATETIME data type the corresponding date time templates are shown.

| XML Schema Built-in Data Type | MRM data Type |
|---|---|
| boolean | BOOLEAN |
| hexBinary base64Binary | BINARY |
| decimal | DECIMAL |
| double float | FLOAT |
| byte int integer long negativeInteger nonNegativeInteger nonPositiveInteger positiveInteger Short unsignedByte unsignedInt unsignedLong unsignedShort | INTEGER |
| date dateTime Time gDay gMonth gMonthDay gYear gMonthYear | DATETIME yyyy-MM-dd yyyy-MM-dd'T'HH:mm:ssZZZ or yyyy-MM-dd'T'HH:mm:ss.SSSZZZ HH:mm:ss or HH:mm:ss.SSS ---dd --MM --MM-dd yyyy yyyy-MM |
| All other data types | STRING |

BIP9351-BIP9423 – XML Schema Importer Messages

Reading variables in the message

Some messages display text or numbers that vary according to the circumstances giving rise to the message; these are known as *message variables*. The message variables are indicated in this book by the use of the '&' symbol and a number: &1, &2, and so on.

In some cases a message may have variables in the Explanation or User action. Find the values of the message variables by looking in the error log. The complete message, including the Explanation and the User action, is recorded there.

BIP9350 The message '&1' has been successfully imported into Message Set '&2', Identifier &3.

Explanation:

User Response: Informational message - no user action required.

BIP9351 Errors were detected - import of message set '&1' was unsuccessful - please make corrections and rerun the import

Explanation:

User Response: Examine the preceding error messages, correct the problem(s) and rerun the import.

BIP9352 No errors were detected in import of message set '&1', but the report only switch was set, so no changes were made

Explanation:

User Response: Rerun the import without the report only switch set.

BIP9353 Import file does not start with schema element

Explanation:

User Response: Imported XML schema files must have a top level Schema element - correct the imported schema file.

BIP9354 Incorrect type table index (&1)

Explanation:

User Response: This indicates an internal error within the schema importer. Contact IBM support.

BIP9355 Internal error - nameType (&1) is out of range

Explanation:

User Response: This is an internal error in the importer. Contact IBM support.

BIP9356 Internal error - idValue (element or attribute id) is zero for value element (&1)

Explanation:

User Response: This is an internal error. Contact IBM support.

BIP9357 Root name '&1' not found in schema

Explanation:

User Response: Check that the name of the root element specified in the parameters file is correct. It must be a global element in the schema.

BIP9358 No changes were made to the repository

Explanation:

User Response: Examine the preceding messages, correct the problem(s) and rerun the import.

BIP9359 Attribute '&1' mrm id '&2' was found in repository and reused

Explanation:

User Response: Existing entry in the MRM repository was found and reused. No action is required.

BIP9360 Type '&1' mrm id '&2' was found in the repository and reused

Explanation:

User Response: Existing entry in the MRM repository was found and reused. No action is required.

BIP9361 Element '&1' mrm id '&2' was found in repository and reused

Explanation:

User Response: Existing entry in the MRM repository was found and reused. No action is required.**BIP9362 Imported file '&1' has no entries except schema.**

Explanation:

User Response: Warning message - make sure the imported files are correct.**BIP9363 Unable to open input import/include file '&1'**

Explanation:

User Response: An import or include file could not be located. Check the schemaLocation in the include or import statement and the entries in the [FILELOCATION] section of the parameters file. If necessary, add a file location entry to the parameters file.**BIP9364 Unrecognized type entry value '&1'**

Explanation:

User Response: Internal error within the schema importer. Contact IBM support**BIP9365 Unable to resolve reference in id='&1' name '&2'**

Explanation:

User Response: This can be an error in the schemas. Please locate the schema element that is using a reference and ensure it is a valid reference. If name spaces are being used, make sure the name is properly qualified. If necessary, validate the schema using a schema verification tool.**BIP9366 Schema element not expected in &1 at this point xmlId '&2' child '&3' elemType '&4'**

Explanation:

User Response: This can be an error in the schemas. Please locate the schema element and make sure it is valid when used at this point. If necessary, validate the schema using a schema verification tool.**BIP9367 Schema '&1' xmlId '&2' has no name**

Explanation:

User Response: An entry (element or attribute) in the schema has no name. Check the schema. Rerun the import with the dump tables switch (-d) to locate the schema item by using the XML ID.**BIP9368 '&1' '&2' has no type definition**

Explanation:

User Response: A type definition for the attribute or element could not be found. Attributes and elements require a type definition, either with a type attribute or anonymous.**BIP9369 No base type for attribute '&1'**

Explanation:

User Response: The importer could not resolve the type of an attribute to a simple type. When a simple type is derived from another type, it must eventually resolve to a simple type. Please check the schema.**BIP9370 No restriction or extension found for complexContent XML id='&1'**

Explanation:

User Response: A complexContent element must have a restriction or extension child. Please check the schema element.**BIP9371 Unable to find base type for '&1' xml ID='&2'**

Explanation:

User Response: Unable to find the base type referenced by a complex type with complex content. Check the schema to make sure the base type is valid. It must be a global type.**BIP9372 Base type for XMLid='&1' must be complexType**

Explanation:

User Response: The base type does not point to a complex type definition. The base type for a complex content derivation must be a complex type. Please check the schema for errors.

BIP9373 '&1' facet found and ignored

Explanation:

User Response: The MRM does not support this type of facet. The current entry has been ignored. No user action is required.

BIP9374 Facet '&1' found and conversion to '&2' facet attempted initial value '&3' final value '&4'

Explanation:

User Response: A facet conversion has been attempted. Please ensure that the conversion was valid.

BIP9375 No schema element found for entry with XML id '&1' name '&2'

Explanation:

User Response: No schema element was found as a direct ancestor of the type. This could be an error in the schema or an internal error in the importer.

BIP9376 Unable to locate type name '&1' xml id '&2'

Explanation:

User Response: The importer could not locate a global type that matches the given name. This could be due to an incorrect schema, including name space problems or an internal error within the the importer.

BIP9377 Anonymous type without element or attribute as parent XMLid='&1'

Explanation:

User Response: The parent of an anonymous type must be an element or an attribute. This is probably an error within the schema.

BIP9378 Simple type '&1' from union xmlid '&2' not found.

Explanation:

User Response: A union must be based on simple types. Verify that the schema is valid. If the schema is valid, rerun the import with the dump tables (-d) switch to identify the schema element that the xml id is referring to.

BIP9379 Unable to resolve prefix '&1' for xmlid '&2'

Explanation:

User Response: Verify that a prefix has been declared in the schema and/or in the parameters file. This can be caused by an incorrect or missing targetNamespace parameter in a schema as well.

BIP9380 Could not find schema element for prefix '&1' id '&2' '&3'

Explanation:

User Response: A prefix could not be resolved to a schema element. This can be due to an incorrect or missing targetNamespace entry in the schema or due to an incorrect prefix declaration in either one of the schemas or in the parameters file.

BIP9381 Element '&1' xml id '&2' is being used recursively ('&3')

Explanation:

User Response: The MRM repository does not support recursion. If a recursion is detected, an empty open type will be substituted for the recursive element. This should allow instance documents that include a recursion to be processed.

BIP9382 Maximum structure depth (50) exceeded '&1' Element list ('&2')

Explanation:

User Response: The importer supports a maximum structure depth of 50. This is believed to be adequate for all known applications. The limitation is intended to catch runaway situations and exit gracefully rather than exhausting the stack.

BIP9383 Unable to locate typeEntry for '&1' id '&2'

Explanation:

User Response: This is probably an internal error in the importer. Contact IBM support.

BIP9384 File '&1' is zero length. The reference can be in the parameters file or in an import or include statement.

Explanation:

User Response: The file will be ignored. Verify that the file should be of zero length.

BIP9385 '&1' parameter is too long

Explanation:

User Response: An input parameter is longer than the maximum length allowed. Please correct the parameter that is in error and rerun the import.

BIP9386 No schema file name was found in the [FILELIST] section of the parameters file

Explanation:

User Response: The parameters file must point to the main schema file.

BIP9387 Prefix '&1' has incorrect first character '&2'

Explanation:

User Response: A prefix name must be a valid XML local name. It must start with a letter or an underscore. Please correct the prefix name and rerun the import.

BIP9388 '&1' parameter '&2' has been specified previously, new value '&3' used

Explanation:

User Response: The same parameter has been found more than once in the parameters. The new value will be used in place of the previous value. Please verify that the correct value was used.

BIP9389 Value for '&1' parameter ('&2') is incorrect - must be 'Y' or 'N'

Explanation:

User Response: The parameter value must be 'Y' or 'N' (without the quotes). Correct the parameter and rerun the import utility. A '1' can also be used for 'Y' and a '0' for 'N' (without the quotes).

BIP9390 Unrecognized value ('&1') for Parser parameter. Parser parameter must be MRM or XML

Explanation:

User Response: The value of the parser parameter must be either 'MRM' or 'XML' (without the quotes). Please correct the parameter and rerun the import.

BIP9391 Unrecognized parameter '&1' found in parameters file, value '&2'

Explanation:

User Response: An incorrect parameter was found in the parameters file. Remove or correct the parameter and rerun the import.

BIP9392 Missing or blank value for parameter '&1' found in parameters file

Explanation:

User Response: The parameter is ignored.

BIP9393 A file system error was encountered on file ('&1'). The return code was '&2'.

Explanation:

User Response: Determine the cause of the error and rerun the import. This can be caused by an incorrect file name, a file that is marked read only, a permissions problem or some other I/O error.

BIP9394 Unable to allocate storage for buffer for '&1'

Explanation:

User Response: A storage allocation request was unsuccessful. This may indicate an internal error. The allocation can be for storage to hold a file or for another purpose. There is no obvious cause for this problem. Make sure the system is not short of paging space.

BIP9395 '&1' value missing from -'&2' parameter on command line

Explanation:

User Response: A command line parameter argument was entered on the command line but no value was given for the argument. Correct the command line argument and rerun the import utility.

BIP9396 Unrecognized command line argument found ('&1')

Explanation:

User Response: User error. An incorrect command line argument value was found. Please correct the argument and rerun the import.

BIP9397 Command usage is: &1 < -n MRM_DB < -u DB_Userid < -p DB_Password > > > < -s MessageSet > -f ParameterFileName < -I FilePath > < -x XMLTabName > < -o MSGSETFILENAME > < -r > < -v > < -t > < -d > < -c > The database, Userid ,Password, message set file name, schema file path and XML tab names are optional. They can also be specified in the parameters file. If they are specified on the command line, they will override any values found in the parameters file. The -r (report only) switch will produce the output messages, but will not change the repository database. The -c (clear) switch will remove all current entries from the message set (except the wire format tabs) before beginning the import process. The -x (XMLTabName) switch allows the name of the XML tab that will be automatically created to be specified. The -v (verbose), -t (trace) and -d (dump internal tables) switches are primarily for debugging purposes.

Explanation:

User Response: This message contains the command line help. This is for information only.

BIP9398 An error occurred whilst accessing the MRM database. The error occurred in Function &1 calling &2. The return code = &3. The meaning of the error code is : &4

Explanation:

User Response: Check that the XML Schema being imported is valid. The meaning of the error code displayed will help determine the cause of the error. More diagnostic information can be obtained from the XML schema importer by using the verbose and trace options. If the cause of the error cannot be determined please contact your IBM support center.

BIP9399 Unable to open the Message Set file : &1

Explanation:

User Response: Check that a valid file name has been specified for the -o command line option or the MsgSetFileName parameter file option.

BIP9400 The Message &1 already exists in Message Set &2.

Explanation: It is not possible to import a Message that already exists within the message set. The Message name is specified by the RootName option in the parameters file.

User Response: Change the RootName in the options file and XML Schema file so it does not match an existing Message in the Message Set. Alternatively the existing contents can be removed from the Message Set by specifying the -c command line option or the parameter file ResetMsgSet option. This alternative should only be used if none of the existing information in the Message Set is required.

BIP9401 There are one or more components checked out in Message Set &1.

Explanation: It is not possible import an XML Schema into a message set that has any components checked out.

User Response: Check in or Unlock the components using the Control Center and then retry the import.

BIP9402 Message Set &1 is Frozen.

Explanation: It is not possible to import an XML Schema into a message set that is Frozen.

User Response: UnFreeze the Message Set using the Control Center and then retry the import.

BIP9403 Message Set &1 is Finalized.

Explanation: It is not possible to import an XML Schema into a message set that is Finalized.

User Response:

BIP9404 The Identifier for Message Set &1 is &2.

Explanation:

User Response:

BIP9405 The XML Schema Importer cannot be run on this version of WMQI.

Explanation:

User Response: The XML Schema Importer is designed to run against specific versions of WMQI. The version of WMQI that you are currently running is not compatible with this version of the XML Schema Importer.

BIP9406 The Xerces parser has been successfully initialized.

Explanation:

User Response: Informational message - no action required.

BIP9407 An attempt to initialize the Xerces Parser was unsuccessful.

Explanation: The Xerces parser reported the following error: '&1'

User Response: Examine the error message reported by the Xerces parser and respond accordingly.

BIP9408 The Document Object Model (DOM) Tree was built successfully for file '&1'.

Explanation:

User Response: Informational message - no action required.

BIP9409 The DOMParser reported an error during parsing.

Explanation: The DOMParser produced the following error message: '&1'

User Response: Examine your schema for errors, with the aid of the error message supplied by the DOMParser error handler.

BIP9410 The DOMParser reported an error during parsing.

Explanation: The DOMParser was unable to determine the cause of the parsing error.

User Response: Examine your schema for errors. The error is likely to be "unusual" as it was not caught by the DOMParser error handler.

BIP9411 Unrecognized XML tag name found in schema.

Explanation:

User Response: Examine your schema for illegal tag names.

BIP9412 The identifier &1 is not valid.

Explanation: The identifier specified is not a valid MRM identifier.

User Response: Check that the element and attribute names in the XML Schema files are valid XML names.

BIP9413 The MRM attribute value &1 is not valid.

Explanation:

User Response: Check that the element and attribute names in the XML Schema files are valid XML names.

BIP9414 Incorrect value for XML attribute.

Explanation: The XML attribute was: '&1' The attribute value was: '&2'

User Response: Correct the attribute value and try your schema again.

BIP9415 Unrecognized XML attribute '&1' found in schema.

Explanation:

User Response: Examine the XML attributes in your schema for errors.

BIP9416 Unable to find prefix for: '&1'

Explanation:

User Response: Examine your schema for errors.

BIP9417 XML name creation only valid for elements and attributes - xmlns= '&1'

Explanation:

User Response: Compare the value of xmlId against the trace output to locate the problem.

BIP9418 The element &1 already exists in the message set but is of a different type.

Explanation:

User Response: Modify the XML Schema so that it does not conflict with the existing elements in the message set and then retry the import.

BIP9419 Restriction of simple content (xmlId '&1') must have either a base type or an anonymous simple type

Explanation:

User Response: Rerun the import with the dump tables (-d) option and compare the value of xmlIds against the trace output to locate the problem.

BIP9420 base type (xmlId '&1') of restriction/extension (xmlId '&2') must be complexType that matches

Explanation:

User Response: Rerun the import with the dump tables (-d) option and compare the value of xmlIds against the trace output to locate the problem.

BIP9421 XML name creation only valid for elements and attributes - xmlId= '&1'

Explanation:

User Response: Compare the value of xmlId against the trace output to locate the problem.

BIP9422 The DOMParser reported an error during parsing.

Explanation: The DOMParser produced the following error message: Line Number '&1', Column Number '&2', Error Text: '&3'

User Response: Examine your schema for errors, with the aid of the error message supplied by the DOMParser error handler.

BIP9423 The DOMParser reported an "XMLException" type error during parsing.

Explanation: The DOMParser produced the following error message: '&1'

User Response: Examine your schema for errors, with the aid of the error message supplied by the DOMParser error handler.

End of Document