`

WebSphere Message Broker v6

Importer for formatted text message models v2.0.0

Sanjay Nagchowdhury
WebSphere Message Broker Development
IBM United Kingdom Limited
Hursley Park
Winchester, UK
sanjay_nagchowdhury@uk.ibm.com

Property of IBM

Take Note!

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

Second Edition, November 2007
This edition applies to WebSphere Message Broker – Importer for formatted text message models v2.0.0 and to all subsequent releases and modifications unless otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2005, 2007. 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

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 be 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
- WebSphere MQ (WMQ)

- WebSphere Message Broker (WMB)

The following terms are trademarks of other companies:

- Windows 2000 Microsoft Corporation

Document Change History

Document Number	Date	Description of Changes
1.0.0	25th Sep 2006	Initial Release
2.0.0	19th Nov 2007	Added support for WMB 6.1 enhancements to MRM TDS: • Repeat Reference • Binary Data Added explicit 'message' objects: • Allows embedded (multipart) messages to be supported. • -msg parameter is no longer needed. Fixed a defect so that content=open and content=openDefined are now set correctly.

Preface

This SupportPac provides a utility called mqsicreatemsgdefsfromtds for creating WebSphere Message Broker message models that describe formatted text messages.

The mqsicreatemsgdefsfromtds utility is a WebSphere Message Brokers Toolkit V6 and V6.1 eclipse plugin similar in operation to mqsicreatemsgdefs.

You describe the formatted text message using a simple XML syntax that allows the specification of complex types and elements and their associated MRM Tagged/Delimited String (TDS) format physical properties.

The utility reads the XML syntax and creates a message definition file with populated Tagged/Delimited String (TDS) format properties.

You can use an existing message set or ask the utility to create one for you. An XML Schema is provided that defines the XML syntax, along with sample XML syntax files.

Possible uses

Currently, TDS models are created manually using the Message Definition editor provided by the WebSphere Message Broker toolkit.

The mqsicreatemsgdefsfromtds utility can be used to create TDS models without using the Message Definition editor provided in the toolkit. The utility creates the TDS model from TDS files which describe the formatted text message using XML syntax. This means that you could generate TDS models using automated tooling to create the TDS files and import them into WebSphere Message Broker using mqsicreatemsgdefsfromtds.

Prerequisites

This SupportPac requires either WebSphere Message Broker Toolkit version 6.0 with Fixpack 1 or higher, or WebSphere Message Broker Toolkit version 6.1. The utility can only be run on Windows®.

Installation

The SupportPac for WMB 6.0 has been packaged so that it can be installed using Rational Product Updater (RPU). For more information about Rational Product Updater, see:

Rational Product Updater Overview

The SupportPac for WMB 6.1 has been packaged so that it can be installed by unzipping and copying the files directly into the WMB 6.1 product directories.

Unzipping the SupportPac zip file

Two zip files are provided on the download site, ia7d_6.0.zip which is for WMB 6.0 and ia7d_6.1.zip which is for WMB 6.1.
After downloading the chosen SupportPac zip file, unzip it to a new local directory.

The ia7d_6.0.zip file contains the following:

- ia7d.pdf (this file).
- readme.html
- policy_ia7d.xml (used by Rational Product Updater to install the optional feature).
- install directory (containing files used by Rational Product Updater to install the optional feature)
- license directory containing license files
- samples directory containing WebSphere Message Broker message set projects. Each project contains a message set, a message definition file (.mxsd) and an ‘importFiles’ directory which has the .tds file used to create the message definition file.
- schema directory containing a copy of the XML Schema used by the mqsicreatemsgdefsfromtds utility to validate .tds files.

The ia7d_6.1.zip file contains the following:

- ia7d.pdf (this file).
- readme.html
- install directory containing mqsicreatemsgdefsfromtds.exe and the tds importer plugin
- license directory containing license files
- samples directory containing WebSphere Message Broker message set projects. Each project contains a message set, a message definition file (.mxsd) and an ‘importFiles’ directory which has the .tds file used to create the message definition file.
- schema directory containing a copy of the XML Schema used by the mqsicreatemsgdefsfromtds utility to validate .tds files.

After extracting the contents of zip into a new local directory, install the SupportPac using the instructions below.

Installing the SupportPac for WMB v6.0

To install the SupportPac for WebSphere Message Broker v6.0, complete the following steps:

1. Ensure that no Rational products (including the Toolkit) are running.

2. Start the Rational Product Updater (RPU). You must have Administrator authority to run the RPU on Windows®.

   Select Start > Programs > IBM WebSphere Message Brokers 6.0 > Rational Product Updater.

   To start the RPU from the command line, run the file rpu.exe in directory inst_dir\updater\eclipse\ (where inst_dir is the directory in which the Toolkit is installed).

3. Select Preferences > Update Sites. A window is displayed in which you need to enter the path or URL of a new policy file. Click Browse to find the policy_ia7d.xml file in the directory to which you extracted the Toolkit update package, or type the path and file name into the entry field. Click OK.

4a) If you had previously installed version 1.0.0 of this SupportPac, click Find Updates. The RPU interrogates the policy file and displays a list of available updates in the Updates pane.

4b) If you are installing this SupportPac for the first time, click Find Optional Features. The RPU interrogates the policy file and displays a list of available optional feature in the Optional Features pane.

5) Check that "WebSphere Message Broker v6 - Importer for formatted text message models" is selected, click Install Updates (if you are in the Updates pane) or click Install Features (if you are in the Optional Features pane), and follow the instructions given. When requested, select 'I accept the terms in the license agreements' and click OK to agree to the terms and conditions in this fix pack.

6) When the update/optional feature is installed, close the RPU to complete the processing. The install may take about 10 minutes.

Uninstalling the SupportPac for WMB v6.0

1. Ensure that no Rational or IBM Software Development Platform products (including the Toolkit) are running.

2. Start RPU.

3. Click Rollbacks

4. In the Rollbacks pane, select "WebSphere Message Broker v6 - Importer for formatted text message models" and tick the check box.

5. Click Rollback Updates.

6. When the optional feature is rolled back, close the RPU to complete the processing. The rollback may take about 10 minutes.

Installing the SupportPac for WMB v6.1

To install the SupportPac for WebSphere Message Broker v6.1, complete the following steps:

1. Ensure that no Rational or IBM Software Development Platform products (including the Toolkit) are running.

2. Copy `unzip_dir\install\mqsicreatemsgdefsfromtds.exe`
   To `C:\Program Files\IBM\WMBT610`

   Note:
   - `unzip_dir` is the directory where you extracted the SupportPac zip file that you downloaded.]
   - `C:\Program Files\IBM\WMBT610` is the default location for the WMB 6.1 package group installation directory. If you have specified a different location for WMB 6.1, then you should copy mqsicreatemsgdefsfromtds.exe to your specified WMB 6.1 location. You will see the other WMB toolkit commands present in this directory, e.g `mqsicreatemsgdefs.exe, mqsicreatebar.exe.`

3. Copy
   `unzip_dir\install\plugins\com.ibm.etools.msg.importer.tds_6.1.0.200712101446.jar`
   To `C:\Program Files\IBM\WMBT610\plugins`

   (Please see note above)

4. Restart the Message Broker toolkit with the –clean option by running the following command :

```
"C:\Program Files\IBM\WMBT610\eclipse.exe" -product
com.ibm.etools.msgbroker.tooling.ide -clean
```

Uninstalling the SupportPac for WMB v6.1

1. Ensure that no Rational or IBM Software Development Platform products (including the Toolkit) are running.

2. Delete `C:\Program Files\IBM\WMBT610\plugins\com.ibm.etools.msg.importer.tds_6.1.0.200712101446.jar`

3. Delete `C:\Program Files\IBM\WMBT610\mqsicreatemsgdefsfromtds.exe`

4. Restart the Message Broker toolkit with the –clean option by running the following command :

```
"C:\Program Files\IBM\WMBT610\eclipse.exe" -product
com.ibm.etools.msgbroker.tooling.ide -clean
```

mqsicreatemsgdefsfromtds command

Supported platforms

- Windows®

Purpose

The **mqsicreatemsgdefsfromtds** command takes as a parameter a directory where source TDS model files (*.tds) are located and generates the equivalent message definition files (*.mxsd). The generated files are placed in the specified message set folder.

Note:

1. Ensure that only the files that are required for the command to run exist in the directory and subdirectory structure you specify. One of the actions the **mqsicreatemsgdefsfromtds** command performs is to copy all the files in the directory and subdirectories into the workspace prior to creating the message definition. This could include any files not associated with the message definitions you are trying to create.

2. You must specify the **-data** *workspace* parameter to specify the target workspace.

Syntax

```
>>-mqsicreatemsgdefsfromtds-- -p --<message set project name>---->

>-- -d --<pathname of source files folder>----------------------->

>-- -data –workspace path---------------------------------------->

>--+-----------------------------------------------------------+-->

   '- -base --<baseMessageSetProjectName>\<baseMessageSetName>-'

>--+--------+--+--------+--+-------+--+--------+---------------->

   '- -rmp -'  '- -rmd -'  '- -ns -'

>--+-----------------------------+----+------+----------------><

   '- -log --<report file name>-'    '- -v -'
```

Parameters

-p ***message set project name***

(Required) The name of the message set project. If the project does not exist, a new one is created.

-d ***pathname of source files folder***

(Required) The absolute or relative pathname of the directory of TDS model (*.tds) files.

All relevant files located in any subfolders under the source files folder are scanned and imported.

-data ***workspace path***

(Required) The path of the workspace in which your projects are created.

The workspace location is also the default location for projects. Relative paths are interpreted as being relative to the directory from which the command was started.

-base

(Optional) If a new message set is to be created, this is the existing message set project and message set, on which it is based.

-rmp

(Optional) Replaces the existing project of the same name.

-rmd

(Optional) Replaces an existing message definition file of the same name.

-ns

(Optional) If a new message set is to be created, the message set is enabled for namespace support

-v

(Optional) Verbose report.

-log ***report file pathname***

(Optional) Absolute or relative pathname of the report file. If omitted, the report is written to the default log file named mqsicreatemsgdefs.report.txt in the Eclipse current directory.

Examples

This example will create .mxsd files in message set project "newtds". If "newtds" does not exist it will be created. If "newtds" does exist it will replace any existing .mxsd files with the same name that already exist.

```
mqsicreatemsgdefsfromtds -p newtds -d c:\mydata\tdsmodels -data c:\myworkspace
-rmd
```

This example will create .mxsd files in message set project "newtds". If "newtds" already exists it will be deleted. A copy of message set "basetds" in message set project "basetds" will be used to create "newtds".

```
mqsicreatemsgdefsfromtds -p newtds -d c:\mydata\tdsmodels -data c:\myworkspace
-base basetds\basetds -rmp
```

TDS model file

A TDS model file is a simple XML file that specifies the logical structure and the TDS physical format properties of one or more tagged/delimited string messages.

The TDS model file allows the specification of the following objects:

- **Message**. Represents a message. It has an *elementRef* attribute that refers to an Element. The name of the Message is inherited from the referenced Element. A Message can have an alias for use when identifying embedded messages via a *messageAlias* attribute. There are no TDS properties for a Message.

- **Element**. Represents an element. It has a *name* attribute and a type. The type may be complex or simple. If the type is complex a *complexTypeRef* attribute refers to a ComplexType. If the type is simple a *simpleTypeRef* refers to an XML Schema built-in type. If the type is simple, the Element can additionally have a ConstraintGroup child element which restricts the built-in type further. An Element can have a default value provided via a *default* attribute, and can be flagged as capable of having a null value via a *nillable* attribute. The TDS properties for an Element are specified using a TDSPropertyGroup child element, the allowable properties being those defined in WMB V6 help topic "Global element TDS properties".

- **ComplexType.** Represents a complex type and its content. It has a *name* attribute and one or more TypeMember child elements. A ComplexType can specify whether its content is a sequence group, a choice group, an all group, an orderedSet group, an unorderedSet group or a message group, via a *composition* attribute. A ComplexType can specify whether its content is a closed group, an open group or an openDefined group, via a *content* attribute. The TDS properties for a ComplexType are specified using a TDSPropertyGroup child element, the allowable properties being those defined in WMB V6 help topic "Complex type TDS properties".

- **TypeMember**. Represents a member of the content of a complex type. If the member of the ComplexType is an element then an *elementRef* attribute refers to an Element. If the member of the ComplexType is a directly embedded group then a *complexTypeRef* attribute refers to another ComplexType. If the member of the Complex Type is an embedded message then a messageRef attribute refers to a Message. If the member repeats then *minOccurs* and *maxOccurs* attributes are used to specify the minimum and maximum number of occurrences. An optional member is specified by *minOccurs*=0. An unbounded member is specified by *maxOccurs*=-1. The TDS properties for a member are specified using a TDSPropertyGroup child element, the allowable properties being those defined in WMB V6 help topics "Element reference TDS properties" or “Group reference TDS properties”.

- **ConstraintGroup.** The allowable constraints on the simple type of an Element are Enum child elements and *length, maxLength, minLength, maxInclusive, minInclusive, pattern, totalDigits* and *fractionDigits* attributes.

- TDSPropertyGroup. This contains attributes that provide the values of the TDS properties that are permitted on the parent object. The names of the attributes are the actual English names of TDS properties as they appear in the .mxsd file, but with the initial letter in upper-case. The values of the attributes are the actual values of TDS properties as they appear in the .mxsd file. Only TDS properties that differ from the default need to be specified.

What is created by the mqsicreatemsgdefsfromtds command

Each TDS model file (*.tds) results in a new message definition file (*.mxsd) of the same name. The TDS model file is processed as described below. You are strongly recommended to examine the log file after the command has completed as this will indicate any problems that were encountered.

- If no TDS physical format exists in the message set, then a warning is issued and only logical model information is created in the .mxsd file.

- If one or more TDS physical formats exist in the specified message set, then all TDS physical formats are populated. For each TDS format, each .mxsd file object created gets a set of default TDS properties, which are overridden by the TDS properties in any .tds TDSPropertyGroup element associated with the object.

- The .tds *name* object attribute provides the name for the object in the .mxsd file.

- Any occurrence of the optional .tds *description* object attribute populates the *Documentation* tab for the object in the .mxsd file.

- Each .tds ComplexType becomes an xsd:complexType and an associated xsd:group in the .mxsd file. The xsd:complexType that is created in this way can be local to an xsd:element or can be global. The default is local, but it becomes global if any one of the following conditions is true:
    1. The ComplexType is unreferenced by an Element.
    2. The ComplexType is referenced by more than one Element
    3. The *global* attribute is specified.

  The xsd:group that is created in this way can be local to the xsd:complexType or can be global. The default is local, but it becomes global if the ComplexType is referenced by another ComplexType TypeMember.
  Any .tds ComplexType *composition* attribute sets the *composition* of the xsd:group.
  Any .tds ComplexType *content* attribute sets the *content* of the xsd:group.

- Each .tds Element becomes an xsd:element in the mxsd file. The xsd:element that is created in this way can be local to an xsd:complexType or can be global. The default is local, but becomes global if any one of the following conditions is true:
    1. The Element is unreferenced
    2. The Element is referenced in more than one ComplexType TypeMember
    3. The *global* attribute is specified.
    4. The element is referenced by a Message.

  The type of the xsd:element that is created can be:
    - A global xsd:complexType
    - An anonymous local xsd:complexType
    - A schema built-in xsd:simpleType
    - An anonymous xsd:simpleType that restricts a schema built-in xsd:simpleType.

Any ConstraintGroup associated with a .tds Element of simple type causes the creation of an anonymous xsd:simpleType.

Any Enum elements and any *length, maxLength, minLength, maxInclusive, minInclusive, pattern, totalDigits* and *fractionDigits* attributes are used to create xsd:facets to restrict the base schema built-in xsd:simpleType.

Any .tds Element *default* attribute sets the *default* attribute of the xsd:element.

Any .tds Element *nillable* attribute sets the *nillable* attribute of the xsd:element.

- Each .tds Message becomes a message in the .mxsd file. Any .tds *messageAlias* attribute sets the messageAlias property of the message in the .mxsd file.

Example TDS Model File

The following TDS model file illustrates a model for a CSV message.

```
<?xml version='1.0' encoding='UTF-8' ?>
<Model xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="http://www.ibm.com/etools/msg/importer/tdsmodel.xsd">

    <ComplexType name="CSVSimple"
                 global="true"
                 composition="sequence"
                 content="closed">

        <TDSPropertyGroup
            dataElementSeparation="AllElementsDelimited"
            delimiter="&lt;CR&gt;&lt;LF&gt;"/>

         <TypeMember minOccurs="0"
                     maxOccurs="-1"
                     elementRef="customer">

             <TDSPropertyGroup repeatingElementDelimiter="&lt;CR&gt;&lt;LF&gt;"/>

         </TypeMember>

    </ComplexType>

    <ComplexType name="t_customer"
                 global="true"
                 composition="sequence"
                 content="closed">

          <TDSPropertyGroup dataElementSeparation="AllElementsDelimited"
                            delimiter=","/>

          <TypeMember elementRef="firstname"/>

          <TypeMember elementRef="lastname"/>

   </ComplexType>

   <Element name="CSV_1" complexTypeRef="CSVSimple" />
   <Element name="customer" complexTypeRef="t_customer" />
   <Element name="firstname" description="customer first name" simpleTypeRef="string"/>
   <Element name="lastname" description="customer last name" simpleTypeRef="string"/>

   <Message elementRef="CSV_1"/>

</Model>
```

This could represent formatted text message like:

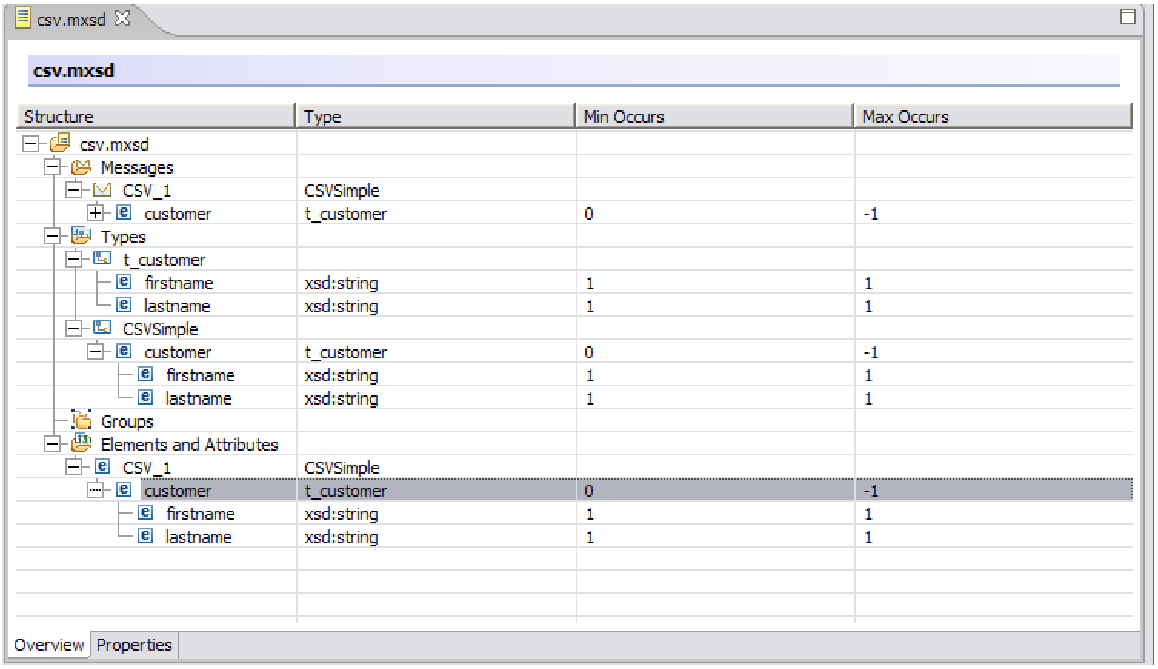
```
Bob,Dyson<CRLF>
Jimmy,Pace<CRLF>
Ralf,McCartney<CRLF>
Elvis,John<CRLF>
Rick,Clayton
```

The key features of this formatted text message are:
The <CRLF> is the "Repeat Element Delimiter" of the "customer" element
The comma (,) is the delimiter within the "customer" type

You import it into the Message Broker Toolkit, by running:

```
mqsicreatemsgdefsfromtds -p TEST_CSV -d C:\test -data "C:\Documents and
Settings\Administrator\IBM\wmbt6.0\workspace" -rmd -base
base_MSGSET\base_MSGSET -rmp -log C:\tds.log -v
```

After the import is complete, you will see a message definition created which looks like:



Structure	Type	Min Occurs	Max Occurs
csv.mxsd			
Messages			
CSV_1	CSVSimple		
customer	t_customer	0	-1
Types			
t_customer			
firstname	xsd:string	1	1
lastname	xsd:string	1	1
CSVSimple			
customer	t_customer	0	-1
firstname	xsd:string	1	1
lastname	xsd:string	1	1
Groups			
Elements and Attributes			
CSV_1	CSVSimple		
customer	t_customer	0	-1
firstname	xsd:string	1	1
lastname	xsd:string	1	1

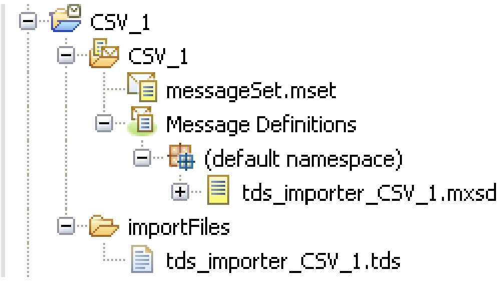
The mqsicreatemsgdefsfromtds utility uses an XML Schema to validate the TDS file. The schema file can be found under unzip_dir\schema\tdsmodel.xsd where unzip_dir is the directory where you extracted the SupportPac zip file that you downloaded. You may refer to this schema to check syntax in your TDS files.

A sample is provided which shows examples of values used to specify TDS properties.
This is located in:
unzip_dir\samples\ TDS_FULL_EXAMPLE_602 \importFiles\ full_example_602.tds
or
unzip_dir\samples\ TDS_FULL_EXAMPLE_61 \importFiles\ full_example_61.tds

Using the provided samples

Sample message set projects are provided in unzip_dir\samples (where unzip_dir is the directory to which you extracted the zip file).

Each message set project contains a message set, a message definition file (.mxsd) and an 'importFiles' directory which has the .tds file used to create the message definition file:



```
CSV_1
  CSV_1
    messageSet.mset
    Message Definitions
      (default namespace)
        tds_importer_CSV_1.mxsd
  importFiles
    tds_importer_CSV_1.tds
```

The following sample message set projects are provided:

- base_MSGSET: a message set used as a base project for importing the TDS files. It does not contain a message definition file or a .tds file.

- CSV_1 - CSV_9: several CSV message sets which describe the message data samples in the CSV sample provided with WebSphere Message Broker 6.0.2

  Or CSV_1 - CSV_6: several CSV message sets which describe the message data samples in the CSV sample provided with WebSphere Message Broker 6.1

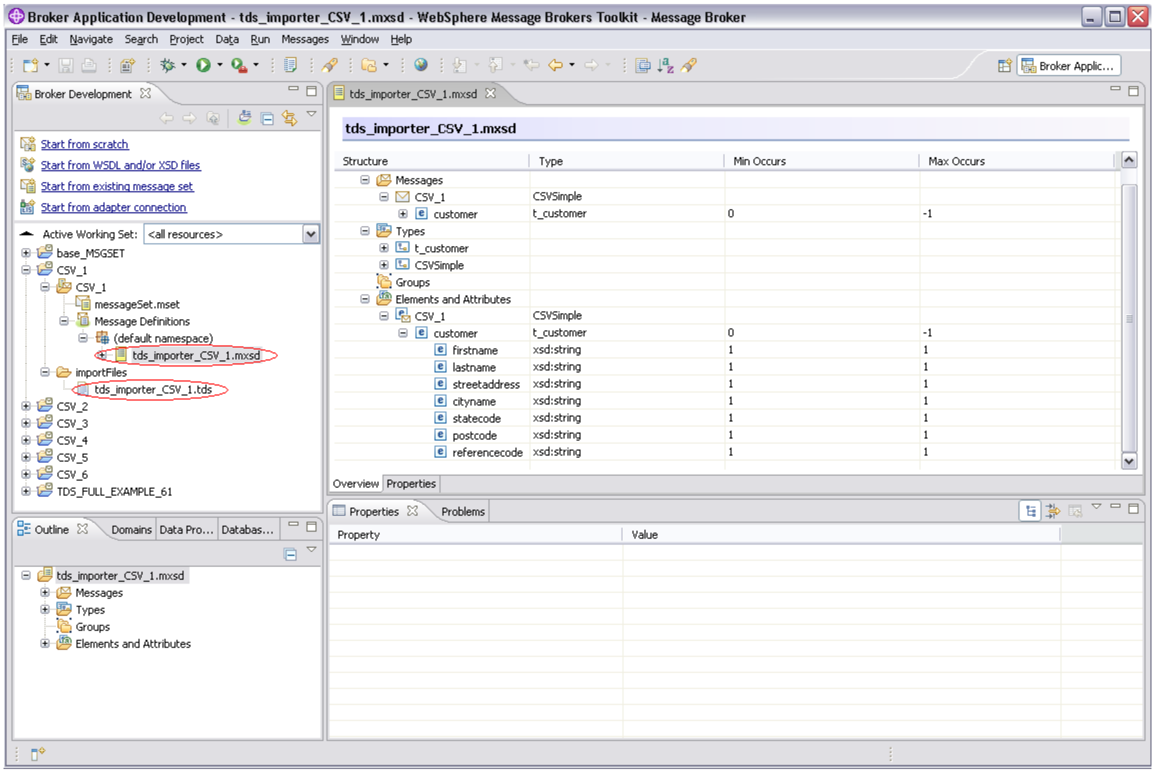
- TDS_FULL_EXAMPLE_602 (or TDS_FULL_EXAMPLE_61) : A message set containing examples of values used to specify TDS properties.

To import the projects:
Select File->Import…
Select Existing Project into Workspace
Browse to the directory where the sample projects are located
(e.g. unzip_dir\samples\CSV_1 where unzip_dir is the directory where you extracted the SupportPac zip file that you downloaded).

After importing each project, you will see the message set, message definition file and the .tds import file that was used to create the message model.

To see the sample TDS files, double click the .tds file under 'importFiles'. You should compare this tds file to the corresponding CSV sample described in the CSV message data sample in the Samples Gallery. You should also compare this file to the contents of the .mxsd file that was generated.

The following screen capture shows what you should see after importing the message set projects. The message definition file and the .tds file that was imported are circled.



tds_importer_CSV_1.mxsd

Structure	Type	Min Occurs	Max Occurs
Messages			
CSV_1	CSVSimple		
customer	t_customer	0	-1
Types			
t_customer			
CSVSimple			
Groups			
Elements and Attributes			
CSV_1	CSVSimple		
customer	t_customer	0	-1
firstname	xsd:string	1	1
lastname	xsd:string	1	1
streetaddress	xsd:string	1	1
cityname	xsd:string	1	1
statecode	xsd:string	1	1
postcode	xsd:string	1	1
referencecode	xsd:string	1	1

To see a description of the CSV message data samples:

Start WebSphere Message Broker Toolkit
Click Help ->Samples Gallery
When the Samples Gallery is shown, click Technology samples on the left hand side.
Click Message Brokers
Click Comma Separated Value (CSV) Message Set
On the right hand side, click "Read about the sample". An overview of the sample is shown.
To read details about each message data sample, scroll to the bottom of the overview information and click "Exploring the CSV sample"

To run the mqsicreatemsgdefsfromtds command using the provided .tds files:

1) Import the base_MSGSET message set project that is provided in the unzip_dir/samples directory.
2) Close the Message Broker Toolkit
3) Open a WebSphere Message Broker command console
4) In the command console, type:

```
mqsicreatemsgdefsfromtds -p CSV_1 -d
C:\ia7d_unzip_dir\samples\CSV_1\importFiles -data " C:\Documents and
Settings\Administrator\IBM\wmbt61\workspace" -rmd -rmp -base
base_MSGSET\base_MSGSET  -log c:\tds.log -v
```

After running, the command, you should see:

```
BIP0225I Command completed.

 Refer to file "c:\tds.log" for details of what was imported.
```

5) Start the Message Broker Toolkit, and you will see a message set project named CSV_1 created with a message definition.

TDS Properties

The following table shows the TDS properties that can be specified as described by the WMB toolkit help topic "**TDS format physical properties for message model objects**" (ad06740_):

Note:
Property and Enum values which are marked with + represent 6.0 only
Property and Enum values which are marked with * represent 6.1 only

Property	Data type	Enum values	Allowed on
dataElementSeparation	Enum	TaggedDelimited TaggedFixedLength TaggedEncodedLength AllElementsDelimited VariableLengthElementsDelimited FixedLength FixedLengthAL3 UseDataPattern Undefined	ComplexType
groupIndicator	String		ComplexType
groupTerminator	String		ComplexType
delimiter	String		ComplexType
suppressAbsentElementDelimiters	Enum	EndOfType Never	ComplexType
observeElementLength	Boolean		ComplexType
tagDataSeparator	String		ComplexType
tagLength	Int		ComplexType
lengthOfEncodedLength	Int		ComplexType
extraCharsInEncodedLength	Int		ComplexType
repeatingElementDelimiter	String		TypeMember
dataPattern	String		TypeMember and Element
lengthReference	Ref		TypeMember
tag	String		Element
interpretElementValue	Enum	None EDIFACTServiceString X12ServiceString EDIFACTSyntaxLevelIdentifier HL7ServiceString HL7FieldSeparator	Element
physicalType	Enum	Characters + Text * NullTerminatedString * LengthEncodedString1 * LengthEncodedString2 * Integer * PackedDecimal * ExternalDecimal * Float * TimeSeconds * TimeMilliSeconds * Binary * TLOGSpecific * MessagingStandardAlternate +	Element
length	Integer		Element
justification	Enum	LeftJustify RightJustify NotApplicable	Element
paddingCharacter	String		Element
encodingNull	Enum	NullPadFill NullLogicalValue NullLiteralValue NullLiteralFill *	Element
encodingNullValue	String		Element

signOrientation	Enum	Leading + Trailing + Leading Separate * Trailing Separate * Leading Overpunched * Trailing Overpunched *	Element
positiveSign	String		Element
negativeSign	String		Element
virtualDecimalPoint **Note:** Any leading + will be stripped	Int		Element
precision **Note:** Enum in help, but integer in .tds file	Int	Valid values are any non-negative integer plus the following special values: -1 = “All Significant Digits” -2 = “Explicit Decimal Point” -3 = “Exponential Notation”	Element
format	String		Element
repeatReference *	Ref		TypeMember
lengthUnits *	Enum	Characters Bytes	Element
signEBCDICCustomOverpunched *	Boolean		Element