MQSeries Integrator - Uuidgen Plug-In Version 1.1

11th September 2002

Mike Brady
Senior Consulting IT Specialist
IBM
Australia

mjbrady@au1.ibm.com

Take Note!

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

Second Edition, September 2002

This edition applies to Version 1.1 of *MQSeries Integrator – Uuidgen Plug-In* and to all subsequent releases and modifications unless otherwise indicated in new editions.

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:

- AIX
- IBM
- MQSeries
- MQSeries Integrator
- MQSI

The following terms are trademarks of other companies:

• HP-UX	Hewlett Packard Corporation
• Solaris	Sun Corporation
• Windows NT	Microsoft Corporation

Summary of Amendments

Date	Changes
12th July 2001	Initial internal release
11th August 2001	Added support for HPUX
3rd September 2001	Released as a SupportPac
15th November 2001	Added IncludeSeparators option
11th September 2002	Add preprocessor directives to allow source to build correctly against v2.1, v2.0.2 and v2.0.1 releases Shipped v2.0.1 and post v2.0.1 binaries

Preface

When building message flows with MQSeries Integrator V2, it is often desirable to be able to generate a unique value. Such a value might be used to initialise a MsgId or CorrelId field, or it could be used as the basis for a database key. The approach most customers have taken to solving this problem is to generate such values with an stand-alone application.

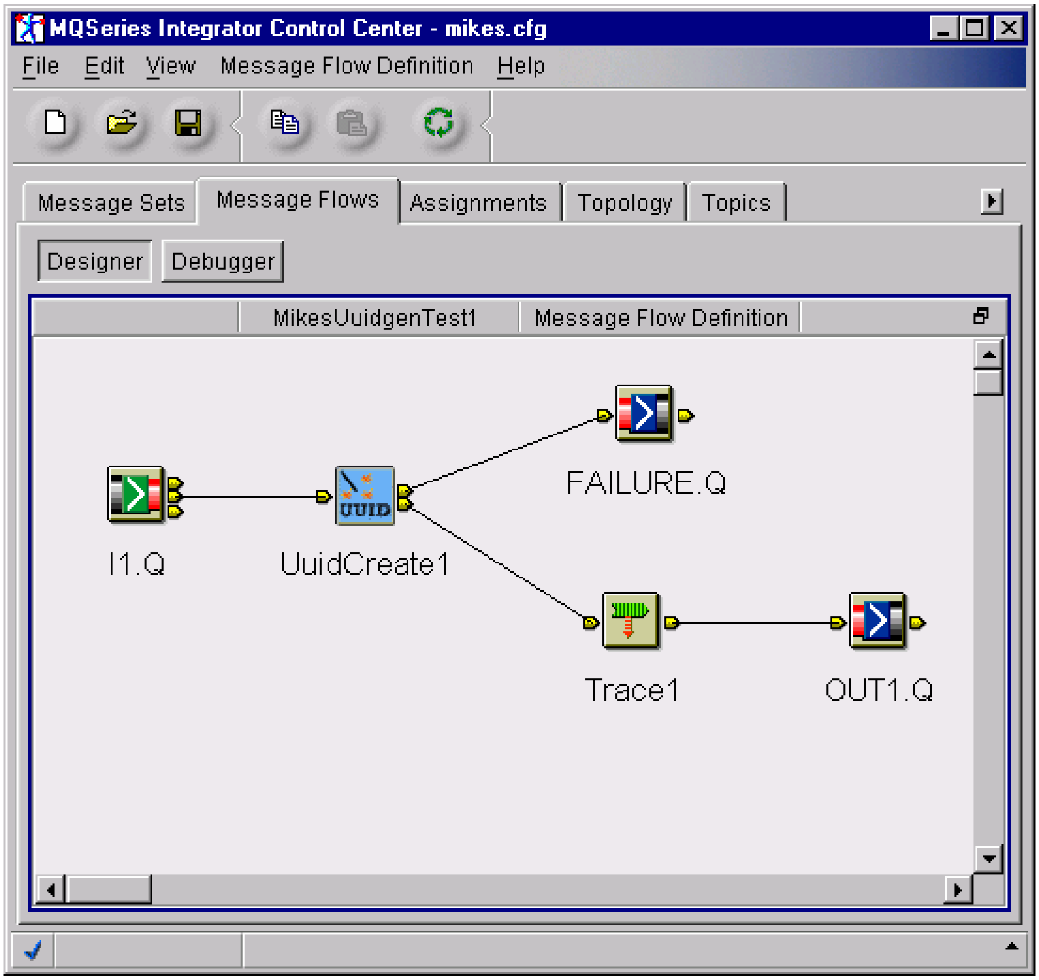
The MQSI V2 Uuidgen function provides an alternative mechanism for generating a unique value within the broker environment. The UUID (Universally Unique Identifier) value generated is guaranteed to be unique among any other values also generated by the Uuidgen plug-in. The UUID value is a 16 byte (binary) value. The Uuidgen implementation is thread-safe and will work equally well for deployments where Message Flows have been configured with multiple worker threads.

Chapter 1. Overview

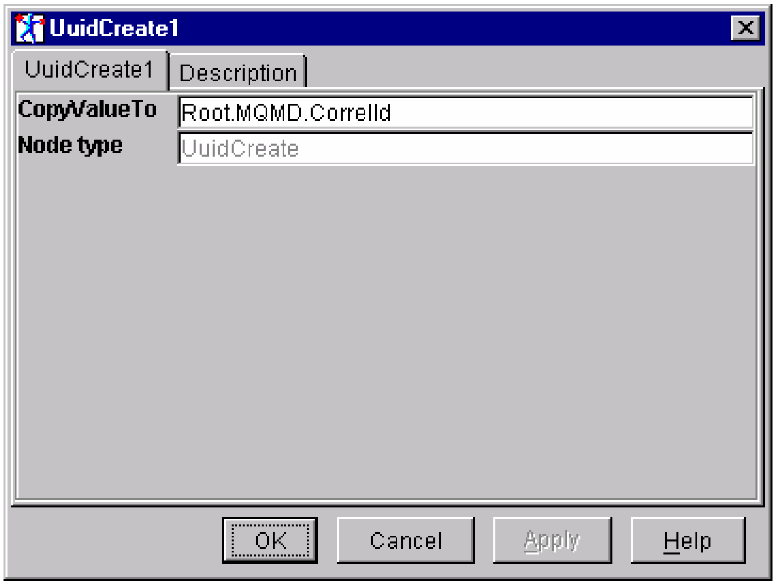
The UuidCreate node is responsible for creating a new UUID value and copying it into the current message at the specified syntax element location.

The following simple message flow shows an instance of a UuidCreate node being used. The value is stored in a copy of the input message, and if successful, the new message is propagated to OUT1.Q (via the UuidCreate 'out' terminal). Any errors result in the original message being propagated to the 'failure' terminal and a message being logged to the system log.

The trace node is being used simply to verify that the CorrelId field contains the UUID value following processing by the UuidCreate node.



The properties window for the UuidCreate node (UuidCreate1) is shown in the following image. The CopyValueTo attribute specifies the message location into which the UUID value should be stored. In this case, the UUID will be stored in the message's CorrelId field. If the location does not exist, a syntax element is created with an element type of NAME-VALUE and the UUID value is stored as CHAR for XML and BYTEARRAY for non-XML messages.



CopyValueTo: Root.MQMD.CorrelId

Node type: UuidCreate

It should be noted that the location specified for CopyValueTo must be compatible with the binary data type of the UUID. Generally this means that the syntax element data type must be BYTEARRAY or BITARRAY, however CHAR is also acceptable. In the latter case, the UUID value is first converted to a sequence of hexadecimal characters before copying to the message. The UUID value will contain hyphens as separators if the IncludeSeparators option was specified.

The way in which the plug-in copies the UUID value into the message depends upon the target field for the value. For XML messages and non-XML messages where the target field data type is CHAR, the UUID value is converted to hexadecimal characters and then copied into the specified element. In this case, the length of the data is 32 characters, excluding any terminating null (36 characters if the IncludeSeparators options is set to true).

For non-XML messages where the target location is a non-CHAR data type, the plug-in checks to ensure that the UUID value will not exceed the length of the existing value, truncating the UUID if necessary. If the data type is binary (BYTEARRAY or BITARRAY), and its length is longer than the UUID, then the UUID value is padded with binary zeroes. In the example message flow, the length of the CorrelId field following the UuidCreate node will still be 24 bytes (The final 8 of which will contain binary zeroes).

When specifying a location in an XML message, the 'XML' qualifier can be omitted. For example, the CopyValueTo attribute settings 'Root.XML.outer.tag1' and 'Root.outer.tag1' will locate/create the same syntax element in the message.

When the target field is defined to hold character data (XML messages and non-XML messages with a target field of data type CHAR), the IncludeSeparators option controls the format of the UUID value. If set to 'true', the UUID will be formatted to include hyphens as separators, e.g., '36e3f290-d91f-11d5-a3e6-204c4f4f5020'.

Chapter 2. Installation

SupportPac contents

The supplied zip file should be unzipped in a temporary directory. The following files and sub-directories will be created.

/source

Makefile.NT

Makefile.AIX

Makefile.Solaris

Makefile.HPUX

uuidgen.c

plugin.c

create.c

retcodes.c

utilities.c

uuid.c

uuidgen.h

uuid.h

uuidgen_retcodes.h

uuidgen_constants.h

plugin.h

unicode.h

threads.h

rc.h

trace_publ.h

trace_defs.h

trace_func.h

trace_data.h

uuidgenTest.c

samples.mak.NT

samples.mak.AIX

samples.mak.Solaris

samples.mak.HPUX

/NT/help

MessageProcessingNodeType_UuidCreate.htm

/NT/bin

uuidgen.lil

v201/uuidgen.lil

uuidgenTest.exe

/NT/messages

MQSIV2_uuidgen.msg

MQSIV2_ uuidgen _msg.h

MQSIV2_ uuidgen.dll

Makefile

/NT/objects

traceinit.obj

trace.obj

error.obj

unicode.obj

threads.obj

uuid.obj

/NT/config

UuidCreate

UuidCreate.wdp

/NT/images

UuidCreate.gif

UuidCreate 30.gif

UuidCreate 42.gif

UuidCreate 58.gif

UuidCreate 84.gif

/AIX

/AIX/bin

uuidgen.lil
v201/uuidgen.lil
uuidgenTest

/AIX/messages

MQSIV2_uuidgen.cat
MQSIV2_ uuidgen.msg
MQSIV2_ uuidgen _msg.h

/AIX/objects

traceinit.o
trace.o
error.o
threads.o
unicode.o
uuid.o

/Solaris

/Solaris/bin

uuidgen.lil
v201/uuidgen.lil
uuidgenTest

/Solaris/messages

MQSIV2_uuidgen.cat
MQSIV2_ uuidgen.msg
MQSIV2_ uuidgen _msg.h

/Solaris/objects

traceinit.o
trace.o
error.o
threads.o
unicode.o
uuid.o

/HPUX

/HPUX/bin

uuidgen.lil
v201/uuidgen.lil

uuidgenTest

/HPUX/messages

MQSIV2_uuidgen.cat

MQSIV2_ uuidgen.msg

MQSIV2_ uuidgen _msg.h

/HPUX/objects

traceinit.o

trace.o

error.o

threads.o

unicode.o

uuid.o

license2.txt

ia0l.pdf

Prerequisites

This SupportPac provides a Plug-in node to be used with the IBM MQSeries Integrator for Windows NT or 2000 - V2.0.1, IBM MQSeries Integrator for AIX - V2.0.1, IBM MQSeries Integrator for Solaris - V2.0.1 and above and IBM MQSeries Integrator for HP-UX - V2.0.2. For normal use, there are no other pre-requisite products other than those required by MQSeries Integrator V2.1 itself.

Supported platforms

This SupportPac has been developed for and tested on

- Microsoft Windows NT environment
- IBM AIX environment
- Sun Solaris environment
- HP-UX environment

Installing the plug-in node on Windows NT

1. Unzip the packaged files into a temporary directory.
2. Copy the message catalogue file **NT\messages\MQSIV2_uuidgen.dll** to a directory of your choice, e.g. <MQSI root>\messages.
3. Add an entry for the message catalogue to the registry. Use regedit to add an entry to the registry under…

```
HKEY_LOCAL_MACHINE
    SYSTEM
      CurrentControlSet
        Services
          EventLog
```

```
                Application
```

Create a new entry with the following details

```
MQSIV2_uuidgen
  (default)             (value not set)
  EventMessageFile   <fully qualified name of MQSIV2_uuidgen.dll>
  TypesSupported        0x00000007 (7)
```

4. Copy the file **NT\bin\uuidgen.lil** to the MQSeries Integrator bin directory, e.g. <MQSI_root>\bin.

5. Restart the broker and check for plug-in initialisation messages in Event log

Installing the plug-in node on AIX

1. Unzip the packaged files into a temporary directory.

2. Copy the message catalogue file **AIX\bin\MQSIV2_uuidgen.cat** to a directory specified by the NLSPATH setting, e.g. <MQSI_root>/messages.

3. Copy the file **AIX\bin\uuidgen.lil** to the MQSeries Integrator lil directory (<MQSI_root>/lil).

4. Restart the broker and check for plug-in initialisation messages in the syslog.

Installing the plug-in node on Solaris

1. Unzip the packaged files into a temporary directory.

2. Copy the message catalogue file **Solaris\messages\ MQSIV2_uuidgen.cat** to a directory specified by the NLSPATH setting, e.g. /usr/lib/locale/<*locale*>/LC_MESSAGES where <*locale*> is the locale under which the machine is running or ‘C’ if none is set.

3. Copy the file **Solaris\bin\uuidgen.lil** to the MQSeries Integrator lil directory (<MQSI_root>/lil).

4. Restart the broker and check for plug-in initialisation messages in the syslog.

Installing the plug-in node on HP-UX

1. Unzip the packaged files into a temporary directory.

2. Copy the message catalogue file **HPUX\messages\ MQSIV2_uuidgen.cat** to a directory specified by the NLSPATH setting, e.g. /usr/lib/locale/<*locale*>/LC_MESSAGES where <*locale*> is the locale under which the machine is running or ‘C’ if none is set.

3. Copy the file **HPUX\bin\uuidgen.lil** to the MQSeries Integrator lil directory (<MQSI_root>/lil).

4. Restart the broker and check for plug-in initialisation messages in the syslog.

Integrating the plug-in node into the Windows Control Center

1. Unzip the packaged files into a temporary directory.

2. Change to the **NT\images** directory and copy its contents to <MQSI_root>\Tool\images

3. Change to the **NT\config** directory and copy its contents to <MQSI_root>\tool\repository\private\<machine name>\<Queue Manager name>\MessageProcessingNodeType

4. Change to the **NT\help** directory and copy its contents to <MQSI_root>\tool\help\com\isv

5. Start the MQSeries Integrator Control Center and display the MessageFlows panel. Right click on IBM Primitives and select **Add to Workspace**, then **Message Flow**. Select the UuidCreate node from the displayed list and add it to the palette.

6. Check in the new node type

Installation verification

Create a message flow that is similar to the one shown in the Overview section.

The sample program uuidgenTest puts a simple message on the I1.Q. The message has a null CorrelId value, i.e. it is NOT set. It then reads the message from the OUT1.Q and prints out the contents of the CorrelId which has been set by the UuidCreate node. By default, this will use the following queues for its processing, I1.Q and OUT1.Q, however, you can specify your own queue manager and queues by invoking the application in the following way

```
timerTest <queueManagerName> <putQueue> <getQueue>
```

The source for the uuidgenTest program is supplied along with an appropriate makefile for the target Operating System, e.g. samples.mak.NT to enable you to easily rebuild it.

Chapter 3. Nodes Reference

UuidCreate Node

The UuidCreate node is responsible for creating a new UUID value and storing it in the current message at the location specified by the CopyValueTo attribute.

Each message processed by a UuidCreate node will cause the UuidCreate node to perform the following actions:

- Create a new UUID value
- Locate the syntax element specified by CopyValueTo property. If the element is not found, create it.
- Store the UUID at the located (or created) syntax element ensuring that a hexadecimal string version is stored for CHAR type data.

If the node detects an error, the message will be routed to the **failure** terminal.

UuidCreate node terminals	
Terminal	**Description**
In	The input terminal that accepts a message for processing by the node
Out	The output terminal to which the message is normally routed
Failure	The output terminal to which the message is routed if there is an error during the node processing

UuidCreate node properties

These properties are displayed when you right click a UuidCreate node entry in the Message Flow Types pane, and click **Properties**. The values displayed are the default properties for this instance of the node. They cannot be edited when displayed from the Message Flow Types pane.

CopyValueTo

This mandatory attribute identifies the location in the message which will hold the UUID value. It must begin with '**Root.**' and if it references an existing element, then that element must be a suitable type to contain data. For example, it cannot be a NAME type element.

IncludeSeparators

An optional value which controls the format of the UUID value. If set to true, the UUID value is formatted with separators to look like a standard DCE-type UUID. For example, the value '36e3f290d91f11d5a3e6204c4f4f5020' would be formatted as '36e3f290-d91f-11d5-a3e6-204c4f4f5020'.

Note: this option is only valid for XML format messages or non-XML messages where the target field is defined as data type CHAR. For all other message formats it is ignored.

Configuring the UuidCreate node

For a description of the properties of the UuidCreate node and their possible values, see"UuidCreate node properties" above.

To configure a UuidCreate node:

1. In the Message Flow Definition pane, right click the symbol of the UuidCreate node you want to configure and click **Properties**. The **UuidCreate node** dialog is displayed.

2. In the **UuidCreate node** dialog, type values for those properties that you want to set.

3. If you want to provide a description of this instance of the UuidCreate node (which is recommended if you want other Control Center users to be able to make use of it), click the **Description** tab of the **UuidCreate node** dialog. Type a short description, or a long description, or both.

4. Click **OK** to finish configuring this UuidCreate node.

Tracing the uuidgen plug-in node

To trace execution of the plug-in nodes, set the UUIDGEN_PLUGIN_TRACE environment variable (system variable on NT) and reboot machine **before** restarting broker. Settings for trace are as follows.

UUIDGEN_PLUGIN_TRACE =

-f *traceOutputFileName* - name of file to write trace to

-t - include time stamp on trace entries.

-i - include process and thread id on entries

-c - commit (flush) entries to file after every write

-l **-** trace level to output (see trace values below)

-a - append trace to existing trace file

Valid trace level settings are

- TRACE_NONE
- TRACE_ENTRY_EXIT
- TRACE_ERROR
- TRACE_WARNING
- TRACE_INFO
- TRACE_SYSTEM_ALL
- TRACE_ALL

The following setting results in comprehensive tracing and will be sufficient in most cases.

```
UUIDGEN_PLUGIN_TRACE=-f <name of output file> -i –c -l TRACE_SYSTEM_ALL
```

End of Document