# MIDI Model Description

# Test Utility 1.0

# Contents

# History

## Version 1.0

Initial release.

# MMD Test Suites

A test suite for an MMD is a sequence of test cases. Each test case exercises the MMD under test in a given scenario. Typically, the suite includes multiple test cases that exercise the same MMD functions under many different scenarios.

A test suite is described by a Lua table as follows:

```
{ -- test suite for the XYZ MMD:
   { -- test case #1
      <test case parameters>
   },
   { -- test case #2
      <test case parameters>
   },

   ...
}
```

Each test case is described by a sequence of parameter 'name=value' pairs, for example:

```
{
   ...

   { -- test case #14
      item = "program",
      config = { unit = 14 },
      slot = 0,
      command = "edit-buffer-dump-request.syx",
      dump = "edit-buffer-dump.syx"
   }

   ...
}
```

The **item** parameter is required for every test case and identifies the type of test to perform; other required and optional parameters depend on the type of test.

## Program Dump/Load Test

A program dump/load test is identified by **item** = **"program"** in test case parameters. This test verifies the functions of the MMD under test for:

- generating the MIDI messages to command a device to dump a program from its edit buffer, or from a stored program slot;
- decoding messages from the device to extract program data records; and
- encoding program data records into MIDI messages for loading the program to the edit buffer or a stored program slot in the same device.

## Synopsis

**item = "program",**
**config** = *<device configuration>*,
**slot** = *<non-negative integer>*,
**command** = *<file name>*,
**dump** = *<file name>*,
**load** = *<file name>*,
**load_slot** = *<non-negative integer>*

## Parameters

**config**: the device configuration to use for this test case. Refer to the MMD specification for the description of configuration attributes;

**slot**: the slot number of a program to retrieve from the device. '0' is the edit buffer; positive values designate stored program slots, the numbering of which is defined by the MMD under test;

**command**: the name of a file that contains the expected sequence of SysEx messages (one or more) to be generated by the MMD **dump_program_command()** function for the requested **slot** and device with the given **config**. This parameter is omitted if the device does not support such commands, and the program dump must be initiated manually by the user;

**dump**: the name of a file that contains the sequence of SysEx messages that would be transmitted by the device when commanded to dump the program (for accuracy of testing, it is recommended and may be necessary to obtain the **dump** file from an actual device);

**load**: the name of a file that contains the sequence of MIDI messages to reload the program onto the device. This parameter is required if the sequence for reloading the program is different from the sequence dumped by the device;

**load_slot**: the program slot number to reload the program to, if different from the original **slot**.

## Test Sequence

1. If **command** is specified in test case parameters, the MMD **dump_program_command()** function is called with the given device configuration 'config' and program slot number 'slot' to generate the sequence of MIDI messages to command the device to dump the requested program. The output from **dump_program_command()** is then compared with the contents of the **command** file for correctness;
2. The MMD **decode()** function is called with the contents of the 'dump' file to extract program data records. The MMD **load_program_command()** function is then called with the output from **decode()** to construct the sequence of MIDI messages to restore the same program onto a device with the same configuration **config**.

   If **load** is specified in test case parameters, the message sequence output by **load_program_command()** is compared for correctness against the contents of the **load** file. Otherwise, the sequence is compared against the original **dump** file;

If **load_slot** is specified in test case parameters, the MMD **load_program_command()** function is called with this as the program's destination slot number. Otherwise, the program is encoded for reloading into the original **slot**.

## Globals Dump/Load Test

A globals dump/load test is identified by **item = "globals"** in test case parameters. This test verifies the functions of the MMD under test for:

- generating the MIDI messages to command a device to dump its globals;
- decoding messages from the device that contain globals data records; and
- encoding globals data records into MIDI messages for loading the globals back to the same device.

### Synopsis

**item = "globals",**
**config** = *<device configuration>***,**
**globals** = *<name>***,**
**command** = *<file name>***,**
**dump** = *<file name>*

### Parameters

config: the device configuration to use for this test case. Refer to the MMD specification for the description of configuration attributes;

**globals**: the name of the globals to retrieve. For devices that organize globals in multiple categories that can be dumped and restored individually, this identifies the category for this test case. Globals category names are defined by the MMD in accordance with the MIDI implementation from the device manufacturer;

**command**: the name of a file that contains the expected sequence of SysEx messages (one or more) to be generated by the MMD **dump_globals_command()** function for the requested **globals** and device with the given **config**. This parameter is omitted if the device does not support such commands, and the globals dump must be initiated manually by the user;

**dump**: the name of a file that contains the sequence of SysEx messages that would be transmitted by the device when commanded to dump the globals (for accuracy of testing, it is recommended and may be necessary to obtain the **dump** file from an actual device);

### Test Sequence

1. If **command** is specified in test case parameters, the MMD **dump_globals_command()** function is called with the given device configuration **config** and category name **globals** to generate the sequence of MIDI messages to command the device to dump the requested globals. The output from **dump_globals_command()** is then compared with the contents of the **command** file for correctness;

2. The MMD **decode()** function is called with the contents of the **dump** file to extract globals data records. The MMD **load_globals_command()** function is then called with the output from **decode()** to construct the sequence of MIDI messages to restore the same globals onto a device with the same configuration **config**. If **load** is specified in test case parameters, the message sequence output by **load_globals_command()** is compared for correctness against the contents of the **load** file. Otherwise, the sequence is compared against the original **dump** file.

# Running a Test Suite

An MMD test suite is performed by running **verify,** for example:

```
$ lua verify.lua -s <MMD directory> [-q] -- <MMD name>
```

**verify** is a Lua 5.3 program with the following purpose:

1. load the MMD with the given name from *<MMD directory>*;
2. load the test case table from a Lua file named *<MMD directory>***/test/cases.lua**;
3. execute the test suite, output results to the console.

The **-q** option reduces the verbosity of the output. Omit it get more details.