MIDI-CI Property Exchange

Mode Resources: ModeList, Current Mode

Version 1.01
November 24, 2020

Document M2-106-UM
Published By:
Association of Musical Electronics Industry
http://www.amei.or.jp
and
The MIDI Association
https://www.midi.org

MIDI™

PREFACE

Property Exchange is part of the MIDI-CI specifications first released in 2018. Property Exchange is a method for sending JSON over SysEx between two devices to get and set device properties. Each MIDI device is unique and provides an experience different from another device. Property Exchange allows you to discover and use almost any device in a consistent way. This document describes the Property Data for these Resources. For information on how to transmit and receive Property Data over SysEx please see the MIDI-CI [MMA02] and Common Rules for MIDI-CI Property Exchange [MMA03].

https://www.midi.org
http://www.amei.or.jp

AMEI Association of Musical Electronics Industry

MIDI™ Association

Table of Contents

1. Introduction

1.1 Background

Property Exchange is part of the MIDI Capability Inquiry (MIDI-CI) [MMA02] specification and MIDI 2.0. Property Exchange is a method for getting and setting various data, called Resources, between two Devices. Resources are exchanged inside two payload fields of System Exclusive Messages defined by MIDI-CI, the Header Data field and Property Data field. This document defines only the contents of the Header Data and Property Data fields. For information on how to transmit and receive these Resource payloads inside MIDI-CI System Exclusive messages, see the MIDI Capability Inquiry specification [MMA02] and Common Rules for MIDI-CI Property Exchange specification [MMA03].

This specification defines two Resources, ModeList and CurrentMode. If a Property Exchange Device has Modes, then it should support the ModeList Resource and CurrentMode Resource.

1.2 Related Documents

[MMA01] ***The Complete MIDI 1.0 Detailed Specification, Document Version 96.1, Third Edition***, Association of Musical Electronics Industry, http://www.amei.or.jp/, and MIDI Manufacturers Association, https://www.midi.org/.

[MMA02] ***MIDI Capability Inquiry (MIDI-CI), Version 1.1***, Association of Musical Electronics Industry, http://www.amei.or.jp/, and MIDI Manufacturers Association, https://www.midi.org/.

[MMA03] ***Common Rules for MIDI-CI Property Exchange, Version 1.1***, Association of Musical Electronics Industry, http://www.amei.or.jp/, and MIDI Manufacturers Association, https://www.midi.org/.

1.3 Terminology

Data Set: A complete Property Exchange message whether sent in one System Exclusive message in a single Chunk or in multiple Chunks.

Device: An entity, whether hardware or software, which can send and/or receive MIDI messages.

List Resource: A specific type of Resource that provides a list of objects in a JSON array.

MIDI 1.0 Specification: [MMA01] Complete MIDI 1.0 Detailed Specification, Document Version 96.1, Third Edition

MIDI-CI: [MMA02] MIDI Capability Inquiry.

PE: Property Exchange.

Program: A set of Device parameters which is selectable by Bank Select and Program Change messages.

Property: A JSON key:value pair used by Property Exchange.

Property Data: A set of one or more Properties in a Device which are accessible by Property Exchange. Contained in the Property Data field of a MIDI-CI Property Exchange message.

Property Exchange: An AMEI/MMA specification, which is the basis for this specification, in which one Device may access Property Data from another Device.

Property Exchange Device: A Device which implements Property Exchange.

Property Key: the key in a JSON key:value pair used by Property Exchange.

Property Value: the value in a JSON key:value pair used by Property Exchange.

Resource: A defined Property Data with an associated inquiry for accessing the Property Data.

Simple Property Resource: A Resource that defines only a single Property which includes only a Property Value, without the Property Key, in the Property Data.

1.4 Reserved Words and Specification Conformance

In this document, the following words are used solely to distinguish what is required to conform to this specification, what is recommended but not required for conformance, and what is permitted but not required for conformance:

Table 1 Words Relating to Specification Conformance

Word	Reserved For	Relation to Spec Conformance
shall	Statements of requirement	Mandatory. A conformant implementation conforms to all 'shall' statements.
should	Statements of recommendation	Recommended but not mandatory. An implementation that does not conform to some or all 'should' statements is still conformant, providing all 'shall' statements are conformed to.
may	Statements of permission	Optional. An implementation that does not conform to some or all 'may' statements is still conformant, providing all 'shall' statements are conformed to.

By contrast, in this document, the following words are never used for specification conformance statements; they are used solely for descriptive and explanatory purposes:

Table 2 Words Not Relating to Specification Conformance

Word	Reserved For	Notes
must	Statements of unavoidability	Describes an action to be taken that, while not required (or at least not directly required) by this specification, is unavoidable. Not used for statements of conformance requirement (see 'shall' above).
will	Statements of fact	Describes a condition that as a question of fact is necessarily going to be true, or an action that as a question of fact is necessarily going to occur, but not as a requirement (or at least not as a direct requirement) of this specification. Not used for statements of conformance requirements (see 'shall' above).
can	Statements of capability	Describes a condition or action that a system element is capable of possessing or taking. Not used for statements of conformance permission (see 'may' above).
might	Statements of possibility	Describes a condition or action that a system element is capable of electing to possess or take. Not used for statements of conformance permission (see 'may' above).

2. ModeList Resource

2.1 Introduction

ModeList is a List Resource which describes the different Modes available in the Device. A Mode is a fundamental configuration of a Device. A change of Mode might change the response to MIDI messages, might change the number of active MIDI Channels, and might change the contents of Payload Data on the Device for any supported Resource.

Examples:

1. A Device might declare it has a "Single MIDI Channel Mode" and a "Multiple MIDI Channel Mode".
2. A Device might declare it has a "Song Mode" and a "Loop Mode"

The Property Value of the "modeId" Property is used with the CurrentMode Resources.

It is strongly recommended that all Property Exchange Devices that use multiple Modes should implement this Resource. If the Device does not implement multiple Modes (only has one Mode), then this Resource should not be supported by the Device.

All Property Exchange Devices which implement the ModeList Resource shall implement the CurrentMode Resource. See Section 3.

2.2 Getting ModeList Property Data

An Initiator may request the "ModeList" Resource from a Responder using an Inquiry: Get Property Data message.

Initiator Sends Inquiry: Get Property Data Message

Header Data	`{"resource":"ModeList"}`
Property Data	*none*

Responder that supports ModeList Resource shall return an array of objects in the Property Data using a Reply to Get Property Data Message.

Each object contains the following Properties:

Property Key	Property Value Type	Description
`modeId`	string (required, max 36 chars)	This is the identifier for the Mode. The Property Value is the Property Data used by the "CurrentMode" Resource.
`title`	string, required	The title of the Mode being described.
`description`	string, commonmark	A description of the Mode.

Responder Sends Reply to Get Property Data Message

Header Data	{"status":200}

Property Data	[
	  {
	  "modeId": "singleChannelMode",
	  "title": "Single MIDI Channel Mode",
	  "description": "This describes a single Program that plays one source."
	  },
	  {
	  "modeId": "multiChannelMode",
	  "title": "Multiple MIDI Channel Mode",
	  "description": "This describes a Performance Mode made up many sources played on many channels."
	  }
	]

2.3 "ResourceList" Integration for ModeList

Minimal entry in ResourceList:

Property Data	[{"resource": "ModeList"}]

Full version default settings:

Property Data	[
	  {
	  "resource": "ModeList",
	  "canGet": true,
	  "canSet": "none",
	  "canSubscribe": false,
	  "canPaginate": false,
	  "schema":{
	  "type": "array",
	  "title": "Modes Available",
	  "$ref": " http://schema.midi.org/property-exchange/M2-106-S_v1-0_ModeList.json"
	  },
	  "columns":[
	  {"property": "title", "title": "Mode"},
	  {"property": "description", "title": "Description"}
	  ]
	  }
	]

3. CurrentMode Resource

3.1 Introduction

The CurrentMode is a Simple Property Resource used to get or set the current Mode. The list of Modes available is retrieved using the ModeList Resource.

The Initiator can assemble a list of available Modes by retrieving the Property Value from the "modeId" Property in each entry of the ModeList Payload Data. See Section 2.2.

All Property Exchange Devices which implement the ModeList Resource shall implement the CurrentMode Resource. If the Device does not implement multiple Modes (only has one Mode), then this Resource should not be supported by the Device.

3.2 Getting CurrentMode Property Data

An Initiator may request the CurrentMode Resource from a Responder using an Inquiry: Get Property Data message.

Initiator Sends Inquiry: Get Property Data Message

Header Data	`{"resource":"CurrentMode"}`
Property Data	*none*

Responder Sends Reply to Get Property Data Message

Header Data	`{"status":200}`
Property Data	`"multiChannelMode"`

3.3 Setting CurrentMode Property Data

An Initiator may send the Property Data to a Responder for the CurrentMode Resource using an Inquiry: Set Property Data message.

Initiator Sends Inquiry: Set Property Data Message

Header Data	`{"resource":"CurrentMode"}`
Property Data	`"singleChannelMode"`

Responder Sends Reply to Set Property Data Message

Header Data	`{"status":200}`
Property Data	*none*

3.4 "ResourceList" integration for CurrentMode

Minimal entry in ResourceList:

Property Data	[{"resource": "CurrentMode"}]

Full version with default values:

Property Data	[
	{
	"resource": "CurrentMode",
	"canGet": true,
	"canSet": "full",
	"canSubscribe": false,
	"schema":{
	"type": "string",
	"title": "Current Mode",
	"maxLength": 36
	}
	}
	]

Revision History

Date	Version	Changes
Nov. 17, 2020	1.01	Initial Version

https://www.midi.org
http://www.amei.or.jp

MIDI™ Association