MIDI-CI Property Exchange

Channel Resources:

ChannelMode, BasicChannelRx, BasicChannelTx

Version 1.01
November 24, 2020

Document M2-108-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 document defines three Property Exchange Resources: ChannelMode, BasicChannelRx, and BasicChannelTx. These Resources are used to Get and Set information related to the choice of MIDI Channels which are actively in use by a Device.

There is an additional Property Resource defined to address the use of MIDI Channels, ChannelList, which is not defined in this document but is in MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema [MMA04]. Devices which have complex usage of MIDI Channels should also implement the ChannelList 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/.

[MMA04] ***MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema, Version 1.0***, Association of Musical Electronics Industry, http://www.amei.or.jp/, and MIDI Manufacturers Association, https://www.midi.org/.

1.3 Terminology

Basic Channel: When a Device is operating on multiple MIDI channels, the Basic Channel is one where MIDI messages can control parameters across multiple channels of the Device. For example, a Program Change sent on the Basic Channel could select sounds on multiple MIDI channels of the Device.

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

MIDI-CI: [MMA02] MIDI Capability Inquiry, a specification published by MMA and AMEI.

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. ChannelMode Resource

2.1 Introduction

"ChannelMode" is a Simple Property Resource which is used to describe a Responder's current Omni On/Off and Poly/Mono Mode. For more information please the MIDI 1.0 specification [MMA01], "RECEIVERS MODE (OMNI ON/OFF & POLY/MONO)" Section.

To change the Receiver's Mode, use the existing Control Change messages as described in The Complete MIDI 1.0 Detailed Specification [MMA01].

ChannelMode is only useful for Devices which have only a single Basic Channel. These Devices shall not use Multi-Mode as described in The Complete MIDI 1.0 Detailed Specification [MMA01], page 7, CHANNEL MODES Section. If a Device has a more complex channel arrangement, the Device should instead use the ChannelList Resource to report its use of MIDI channels in more detail (See ChannelList in MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema [MMA04]).

2.2 Initiator Requests Data from a Responder Using an Inquiry: Get Property Data

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

Initiator Sends Inquiry: Get Property Data Message

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

Responder Sends Reply to Get Property Data Message

Header Data	`{"status":200}`
Property Data	`3`

The ChannelMode Resource values are as follows:
1 = Mode 1: Omni On, Poly
2 = Mode 2: Omni On, Mono
3 = Mode 3: Omni Off, Poly
4 = Mode 4: Omni Off, Mono

2.3 "ResourceList" Integration for ChannelMode

Example minimal entry in ResourceList:

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

Example full version with default settings:

Property Data	[
	{
	"resource": "ChannelMode",
	"canGet": true,
	"canSet": "no",
	"canSubscribe": false,
	"requireResId": false,
	"schema": {
	"title": "Channel Mode",
	"type": "number",
	"min": 1,
	"max": 4,
	"multipleOf": 1,
	"description": "This is the Channel Mode value. It is one of the following values:\n1 = Mode 1 Omni On Poly\n2 = Mode 2 Omni On Mono\n3 = Mode 3 Omni Off Poly\n4 = Mode 4 Omni Off Mono"
	}
	}
	]

3. BasicChannelRx Resource

3.1 Introduction

An instrument can receive MIDI messages on more than one channel. The channel on which it receives its main instructions, such as which program number to be on and what mode to be in, is referred to as its Basic Channel. "BasicChannelRx" is a Simple Property Resource which is used to manage the Responder's Basic Channel for receiving MIDI data. Basic Channel is described further in The Complete MIDI 1.0 Detailed Specification [MMA01], "THE BASIC CHANNEL OF AN INSTRUMENT" Section.

BasicChannelRx is only useful for Devices which have only a single Basic Channel. These Devices shall not use Multi-Mode as described in The Complete MIDI 1.0 Detailed Specification [MMA01], page 7, CHANNEL MODES Section. If a Device has a more complex channel arrangement, the Device should instead use the ChannelList Resource to report its use of MIDI channels in more detail (See ChannelList in MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema [MMA04]).

3.2 Initiator Requests Data from a Responder Using an Inquiry: Get Property Data

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

Initiator Sends Inquiry: Get Property Data Message

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

Responder Sends Reply to Get Property Data Message

Header Data	`{"status":200}`
Property Data	`1`

3.3 Request using Inquiry: Set Property

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

Initiator Sends Inquiry: Set Property Data Message

Header Data	`{"resource":"BasicChannelRx"}`
Property Data	`2`

Responder Sends Reply to Set Property Data Message

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

3.4 "ResourceList" Integration for BasicChannelRx

Example minimal entry in ResourceList:

Property Data	`[`
	`  {"resource": "BasicChannelRx"}`
	`]`

Example full version with default settings:

Property Data	`[`
	`  {`
	`  "resource": "BasicChannelRx",`
	`  "canGet": true,`
	`  "canSet": "full",`
	`  "canSubscribe": false,`
	`  "requireResId": false,`
	`  "schema": {`
	`  "title": "Basic Channel Receive",`
	`  "type": "number",`
	`  "min": 1,`
	`  "max": 16,`
	`  "multipleOf": 1`
	`  }`
	`  }`
	`]`

4. BasicChannelTx Resource

4.1 Introduction

An instrument can transmit MIDI messages on more than one channel. The channel on which it transmits its main instructions, such as which program number to be on and what mode to be in, is referred to as its "Basic Channel". The "BasicChannelTx" Simple Property Resource is used to manage the Responder's Basic Channel for transmitting MIDI data. Basic Channel is described further in The Complete MIDI 1.0 Detailed Specification [MMA01], "THE BASIC CHANNEL OF AN INSTRUMENT" Section.

This BasicChannelTx Simple Property Resource is only useful for Devices which have only a single Basic Channel. These Devices shall not use Multi-Mode as described in The Complete MIDI 1.0 Detailed Specification [MMA01], page 7, CHANNEL MODES Section. If a Device has a more complex channel arrangement, the Device should instead use the ChannelList Resource to report its use of MIDI channels in more detail. (See ChannelList in MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema [MMA04])

4.2 Initiator Requests Data from a Responder Using an Inquiry: Get Property Data

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

Initiator Sends Inquiry: Get Property Data Message

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

Responder Sends Reply to Get Property Data Message

Header Data	`{"status":200}`
Property Data	`1`

4.3 Request using Inquiry: Set Property

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

Initiator Sends Inquiry: Set Property Data Message

Header Data	`{"resource":"BasicChannelTx"}`
Property Data	`2`

Responder Sends Reply to Set Property Data Message

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

4.4 "ResourceList" Integration for BasicChannelTx

Example minimal entry in ResourceList:

Property Data	`[`
	`  {"resource": "BasicChannelTx"}`
	`]`

Example full version with default settings:

Property Data	`[`
	`  {`
	`  "resource": "BasicChannelTx",`
	`  "canGet": true,`
	`  "canSet": "full",`
	`  "canSubscribe": false,`
	`  "requireResId": false,`
	`  "schema": {`
	`  "title": "Basic Channel Receive",`
	`  "type": "number",`
	`  "min": 1,`
	`  "max": 16,`
	`  "multipleOf": 1`
	`  }`
	`  }`
	`]`

Revision History

Date	Version	Changes
Nov. 17, 2020	1.01	Initial Version

https://www.midi.org

MIDI™ Association