MIDI-CI Property Exchange

ProgramList Resource

Version 1.01
November 24, 2020

Document M2-107-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

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 the ProgramList Resource. If a Property Exchange Device has Programs selectable by Bank Select and Program Change messages, then it should support the ProgramList 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.

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.

Foundational Resource: A Resource which provides core Properties of a Device which are critical or highly valuable to properly implement numerous other Resources.

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.

Program Collection: A grouping of Programs with some common trait (bank, category, instrument, synthesis engine, presets, etc).

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.

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. ProgramList Resource

2.1 Introduction

ProgramList is a List Resource which provides the list of Programs available in a Program Collection. A Program Collection is a grouping of Programs with some common trait (bank, category, instrument, synthesis engine, presets, etc).

2.1.1 Dependency on the ChannelList Resource

The ProgramList Resource requires a "resId" Property in the Header Data, to allow selection from the list of available Program Collections. The list of available Program Collections is retrieved from the "link" Property in a ChannelList Payload Data. Therefore, Property Exchange Devices which implement ProgramList must also implement ChannelList. See [MMA04] MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema.

2.1.2 Selecting a Program Collection

After the Initiator discovers a list of available Program Collections, several actions might occur to proceed to use a ProgramList Resource. Examples include:

- To use a complete concatenation of all available Programs, the Initiator might access all of the separate ProgramLists for each of the Program Collections, using multiple ProgramList inquiries, one for each of the available Program Collections.
- To use only the Programs from just one Program Collection, the Initiator might expose a list of available Program Collections as options for the user to select one specific Program Collection for access via ProgramList.

2.1.3 Pagination of ProgramList

The ProgramList Resource may be paginated, and therefore shall have the appropriate Header data Properties added as defined in [MMA03] Common Rules for MIDI-CI Property Exchange, Section 6.6.

2.2 Getting ProgramList Property Data

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

Initiator Sends Inquiry: Get Property Data Message

Header Data	`{"resource":"ProgramList","resId":"GMVoices","offset":0,"limit":20}`
Property Data	*none*

Responder that supports ProgramList 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
`title`	string (required)	Human-readable name of the Program
`bankPC`	array of 3 numbers (integers 0-127, required)	This a 3 item array containing the Bank MSB, Bank LSB, and Program Change for the current Program. All Bank/PC messages are 0-based. If the Device does not recognize Bank Select, set to [0,0,pc] where pc = the valid Program Change number.
`category`	array of strings	This is an array of the top level categories that best describe this program. Some common categories are defined in Appendix A, but the array is not limited to only those in Appendix A. The Device may also define its own categories.
`tags`	array of strings	This is an array of manufacturer and user defined meta tags for further description or classification of the program. Manufacturers are encouraged to use human-readable words for their meta tags. Some Devices may allow users to define their own meta tags.

Responder Sends Reply to Get Property Data Message

Header Data	`{"status":200,"totalCount":128}`

Property Data	`[`
	`{`
	`"title": "Acoustic Grand Piano",`
	`"bankPC": [121,0,1],`
	`"category": ["Piano","Keys"],`
	`"tags": ["grand","acoustic"]`
	`},`
	`{`
	`"title": "Bright Acoustic Piano",`
	`"bankPC": [121,0,2],`
	`"category": ["Piano","Keys"],`
	`"tags": ["upright","bright","acoustic"]`
	`},`
	`{`
	`"title": "60s Strings",`
	`"bankPC": [121,2,49],`
	`"category": ["Ensemble","Strings"],`
	`"tags": ["pad","bright"]`
	`},`
	`...`
	`]`

2.3 Using Bank Select and Program Change from ProgramList

Each entry in the ProgramList represents a Program on the Device. Each Program may be recalled by sending Bank Select and Program Change messages which are declared for each entry. The Channel used for the Bank Select and Program Change messages comes from the ChannelList where this Program Collection was declared (in a "links" Property.) See Section 4.3 of [MMA04] MIDI-CI Property Exchange Foundational Resources: DeviceInfo, ChannelList, JSONSchema.

In the example in Section 2.2, to select the "60s Strings" Program, send Bank Select Control Change MSB #0 (0x00) with a value of 121 (0x79), Bank Select Control Change LSB #32 (0x20), with a value of 2 (0x02), followed by Program Change with a value of 49 (0x31).

2.4 "ResourceList" integration for ProgramList

Minimal entry in ResourceList:

Property Data

```
[
   {"resource": "ProgramList"}
]
```

Full version with default settings:

Property Data

```
[
   {
      "resource": "ProgramList",
      "canGet": true,
      "canSet": "none",
      "canSubscribe": false,
      "canPaginate": true,
      "requireResId": true,
      "schema":{
         "type": "array",
         "title": "Program List",
         "$ref": "http://schema.midi.org/property-exchange/M2-107-S_v1-0_ProgramList.json"
      },
      "columns":[
         {"property": "title", "title": "Program Name"},
         {"property": "category", "title": "Categories"},
         {"property": "tags", "title": "Tags"}
      ]
   }
]
```

Appendix A - Program Categories

Suggested Categories for Musical Instruments for use in the "category" Property of objects in the ProgramList Property Data.

Program Categories	Notes
`Piano`	acoustic pianos: grand, upright, etc.
`Keys`	clavinet, harpsichord, electric pianos
`Organ`	including reed organs, accordion, and harmonica
`Guitar`	plucked : guitar, mandolin, banjo, shamisen, oud, etc.
`Bass`	bass instruments
`Drums`	drums and percussive instruments, generally with no distinct pitch
`Percussion`	pitched such as xylophone and tympani
`Vocal`	includes single, choir, vocoder
`Strings`	orchestral strings and bowed instruments, solo and ensemble
`Brass`	brass instruments including trumpets, trombones, french horn, etc.
`Winds`	woodwinds including sax, clarinet, oboe, flute, duduk, shakuhachi, etc.
`Ensemble`	full orchestra or combination of multiple instrument types
`SynBass`	synthesizer bass
`SynLead`	melodic synthesizer
`SynComp`	typically has a fast attack
`SynPad`	typically has a slow attack
`MFX`	musical effects, pitched/harmonic
`SFX`	sound effects, non-pitched/anharmonic
`Other`	

Devices are not limited to these suggested categories. The array of categories is flexible to allow any categories to suit any device type.

Revision History

Date	Version	Changes
Nov. 17, 2020	1.01	Initial Version

https://www.midi.org

AMEI
AMEI Association of Musical Electronics Industry

MIDI™ Association