

# **MIDI 2.0 Specification Overview**

---

**MIDI 2.0 Specification Overview**  
**MIDI Capability Inquiry (MIDI-CI)**  
**Common Rules for MIDI-CI Profiles**  
**Common Rules for MIDI-CI Property Exchange**  
**Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol**

**Version 1.0**  
**February 20, 2020**

**Published By:**  
**Association of Musical Electronics Industry AMEI**  
**and**  
**MIDI Manufacturers Association MMA**

**M2-100-U**  
**<http://www.midi.org>**

## PREFACE

### M2-100-U MIDI 2.0 Specification Overview

This document defines the specific collection of MMA/AMEI specifications that collectively comprise the MIDI 2.0 Specification.

The document also defines minimum requirements for Devices to claim MIDI 2.0 compatibility.

©2018-2020 Association of Musical Electronics Industry (AMEI)(Japan)

©2018-2020 MIDI Manufacturers Association Incorporated (MMA)(Worldwide except Japan)

ALL RIGHTS RESERVED. NO PART OF THIS DOCUMENT MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING INFORMATION STORAGE AND RETRIEVAL SYSTEMS, WITHOUT PERMISSION IN WRITING FROM THE MIDI MANUFACTURERS ASSOCIATION.

<http://www.midi.org/>

## Table of Contents

|       |   |   |
|-------|---|---|
| 1.    | Introduction.....   | 1 |
| 1.1   | Background.....   | 1 |
| 1.2   | Related Documents .....                                       | 1 |
| 1.3   | Terminology.....  | 1 |
| 2.    | Core Components of MIDI 2.0 .....                             | 2 |
| 2.1   | Introduction.....   | 2 |
| 2.1.1 | Components of MIDI 2.0 .....                                  | 2 |
| 2.2   | MIDI Capability Inquiry (MIDI-CI).....                        | 2 |
| 2.3   | Common Rules for MIDI-CI Profiles .....                       | 4 |
| 2.4   | Common Rules for MIDI-CI Property Exchange.....               | 4 |
| 2.5   | Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol..... | 4 |
| 3.    | Compatibility Requirements of MIDI 2.0.....                   | 5 |

# 1. Introduction

## 1.1 Background

MIDI 2.0 is an extension of MIDI 1.0. It does not replace MIDI 1.0 but builds on the core principles, architecture, and semantics of MIDI 1.0.

MIDI has grown and continued to serve users well since its inception in 1983 as the worldwide standard method of musical communication and control. The MIDI industry, through the standardization work of the Association of Musical Electronics Industry (AMEI) and the MIDI Manufacturers Association (MMA), has greatly expanded the capabilities and market reach of MIDI in the years and decades since 1983.

MIDI 1.0 has some limitations for future expansion which are addressed by MIDI 2.0. A foundational architecture for MIDI 2.0 expansion is defined by the MIDI Capability Inquiry (MIDI-CI) specification.

MIDI-CI allows Devices with bidirectional communication to agree to use extended MIDI capabilities beyond those already defined in MIDI 1.0, while carefully protecting backward compatibility.

MIDI 2.0 is not a stand-alone specification. Manufacturers and developers must have a thorough understanding of MIDI 1.0 in order to implement MIDI 2.0.

This document defines the specific collection of MMA/AMEI specifications that collectively comprise the MIDI 2.0 Specification.

The document also defines minimum requirements for Devices to claim MIDI 2.0 compatibility.

## 1.2 Related Documents

*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/>.

*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/>.

*Common Rules for MIDI-CI Profiles, Version 1.0*, Association of Musical Electronics Industry, <http://www.amei.or.jp/>, and MIDI Manufacturers Association, <https://www.midi.org/>.

*Common Rules for MIDI-CI Property Exchange, Version 1.0*, Association of Musical Electronics Industry, <http://www.amei.or.jp/>, and MIDI Manufacturers Association, <https://www.midi.org/>.

*Universal MIDI Packet Format and MIDI 2.0 Protocol, Version 1.0*, Association of Musical Electronics Industry, <http://www.amei.or.jp/>, and MIDI Manufacturers Association, <https://www.midi.org/>.

## 1.3 Terminology

**Device:** A hardware unit or software component.

**MIDI-CI:** The MIDI Capability Inquiry (MIDI-CI) specification.

**MMA/AMEI:** MIDI Manufacturers Association and Association for Musical Electronics Industry.

**Profile:** A set of MIDI messages and defined responses to those messages. A Profile may have a defined minimum set of mandatory messages, along with some optional or recommended messages. General MIDI is one example of a profile (although in original form it does not quite meet all requirements of a MIDI-CI Profile). General MIDI defines a requirement for supporting GM messages on all 16 channels of 1 virtual cable (or stream). Future Profiles may define support on only 1 channel within a virtual cable, or support for more than 16 channels using multiple virtual cables or Groups.

**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.

**Protocol:** A defined data message structure that defines the semantics for MIDI control messages. In MIDI 1.0 the Protocol includes the Opcode of the control message being sent, system-wide messages, addressing for some messages in the form of 16 MIDI Channels, and for some types of message a value associated with the specific Opcode.

**Receiver:** MIDI 1.0 Device or MIDI 2.0 Device that receives MIDI Messages from a Sender and parses them.

**Resource:** A defined set of properties that comprise a set of Property Data.

**Sender:** MIDI 1.0 Device or MIDI 2.0 Device that creates MIDI Messages and transmits them to a Receiver.

## 2. Core Components of MIDI 2.0

### 2.1 Introduction

MIDI 2.0 includes four main documents that expand on the capabilities of MIDI 1.0. These four documents collectively, along with this MIDI 2.0 Specification document, define the core architecture of MIDI 2.0 and its connection to MIDI 1.0. None of these documents are stand-alone but are extensions of MIDI 1.0.

#### 2.1.1 Components of MIDI 2.0

MIDI 2.0 features are defined in the following documents:

1. MIDI Capability Inquiry (MIDI-CI)
2. Common Rules for MIDI-CI Profiles
3. Common Rules for MIDI-CI Property Exchange
4. Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol

### 2.2 MIDI Capability Inquiry (MIDI-CI)

MIDI-CI is the fundamental enabler for the expansion of MIDI.

MIDI-CI defines an architecture that allows Devices with bidirectional communication to agree to use the extended capabilities of MIDI 2.0 that are beyond those already defined in MIDI 1.0, while carefully protecting backward compatibility. MIDI-CI features “fall back” mechanisms so that if a Device does not support new features, MIDI continues to work as defined by MIDI 1.0.

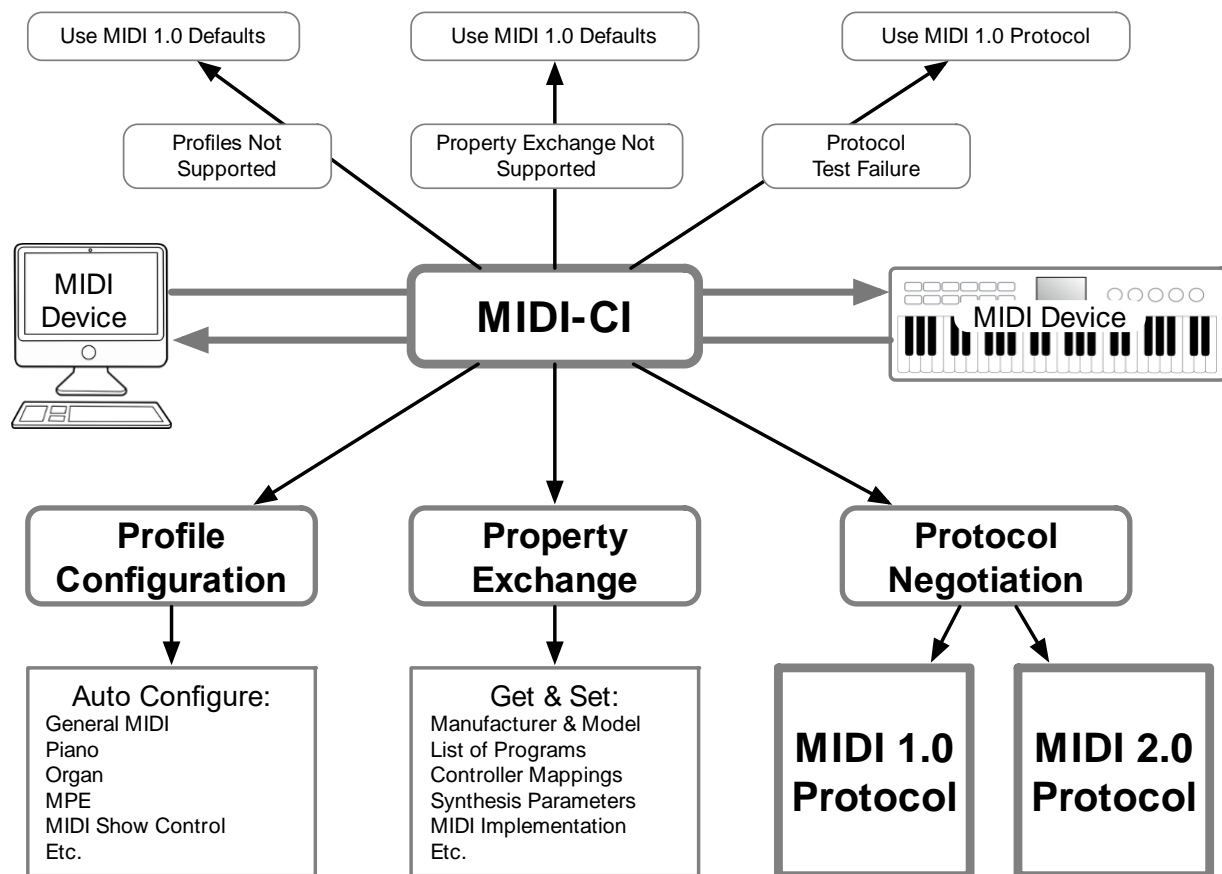
Goals of MIDI-CI design:

- Extend MIDI while protecting backward compatibility: supports continued MIDI 1.0 functionality for any Devices that do not recognize extended MIDI 2.0 features.
- Allow a Sender to know the capabilities of a Receiver.
- Enable easier configuration between Devices using auto-configuration Profiles.
- Define method for Getting and Setting a wide range of Device properties.
- Define method for negotiating choice of Protocol between Devices.

MIDI-CI delivers on these goals with 3 main areas of functionality:

- **Protocol Negotiation:** choice of Protocol between Devices.
- **Profile Configuration:** use auto-configuration Profiles.
- **Property Exchange:** Get & Set Device Property Data using JSON.

Devices default to using MIDI 1.0 and only use extended MIDI features after two Devices have agreed via MIDI-CI to use those extended features.



See the **MIDI Capability Specification (MIDI-C)** for more details.

## 2.3 Common Rules for MIDI-CI Profiles

Profiles are a beneficial component in enabling intelligent auto-configuration.

A Profile is a defined set of MIDI messages and implementation rules to achieve a particular purpose or to suit a particular application. In addition to defining responses to MIDI messages, a Profile may optionally also define other Device functionality requirements. In general, Profiles define Receiver implementation. A Profile definition may also imply or require MIDI implementation of a Sender.

The most successful legacy MIDI feature that is similar to a Profile in the first three decades of MIDI has been General MIDI. GM allows Devices to "know" that a defined set of sounds is available at particular Program Change locations, that the Device receives on all 16 MIDI channels, that a Drum set is on channel 10, and that there is a defined response to a chosen set of MIDI messages.

This kind of knowledge shared between Devices allows those Devices to configure a more integrated level of control with increased predictability of the results that will come from sending related MIDI messages. MIDI-CI Profiles are also intended to allow more integrated cooperation between Devices.

While GM is the best model of a successful profile concept prior to MIDI-CI, it does not take MIDI-CI or a two way communication into account. There is a "GM On" message but no reply from the Receiver. New MIDI-CI Profiles take advantage of two way communication. This document defines how specific Profile specifications should be written and how Devices that are compatible with MIDI-CI Profile Configuration should use Profiles.

**See the Common Rules for MIDI-CI Profiles for more details.**

## 2.4 Common Rules for MIDI-CI Property Exchange

Property Exchange is a set of mechanisms to get and set Device property data using JSON transmitted via MIDI-CI Universal System Exclusive messages.

Property Exchange can allow Devices to auto map controllers, choose programs, change state and also provide visual editors to DAW's without any prior knowledge of the Device or specially crafted software. This means that Devices could work on a wide range of systems such as desktop operating systems, mobile devices, and web browsers and may provide tighter integrations with DAWs and hardware controllers.

Property Exchange provides a common way for Devices to work together by providing defined schema for describing how Property Data is transferred.

Specially written software might provide a more unique experience by relying on custom software and SysEx. However, by using Property Exchange, the longevity and accessibility of equipment is no longer restricted by underlying system upgrades or new platforms that appear in the marketplace.

**See the Common Rules for MIDI-CI Property Exchange for more details.**

## 2.5 Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol

Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol specification defines a new data format for MIDI 1.0 Protocol messages and MIDI 2.0 Protocol messages.

This new data format adds 16 Groups, each containing an independent set of System Messages and 16 Channels that are equivalent to MIDI 1.0 Channels. The Universal MIDI Packet format adds a Jitter Reduction Timestamp mechanism. A Timestamp can be prepended to any MIDI message for improved timing accuracy.

MIDI 2.0 Protocol Specification defines the MIDI 2.0 Protocol with extended data resolution for all Channel Voice Messages. MIDI 2.0 makes some messages easier to use, by aggregating combination messages into one atomic message. It adds new properties to several Channel Voice Messages. New Channel Voice Messages are added to provide increased Per-Note control and musical expression.

New data messages include System Exclusive 8 and Mixed Data Set. The System Exclusive 8 message is very similar to MIDI 1.0 System Exclusive but with 8 bit data format. The Mixed Data Set Message is used to transfer large data sets, including non-MIDI data.

The Universal MIDI Packet format includes a large reserved space for future extensibility.

**See the Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol Specification for more details.**

### 3. Compatibility Requirements of MIDI 2.0

Devices may claim MIDI 2.0 compatibility if they implement MIDI-CI\* plus any one or more of the following features:

- One or more Profiles controllable by the use of MIDI-CI Profile Configuration messages.
- Any Property Data exchange by the use of MIDI-CI Property Exchange messages.
- MIDI 2.0 Channel Voice Messages as defined by the Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol.
- Jitter Reduction Timestamps as defined by the Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol.
- System Exclusive 8 as defined by the Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol.
- Mixed Data Set as defined by the Universal MIDI Packet (UMP) Format and MIDI 2.0 Protocol.

\*Note: See Appendix A of the *MIDI Capability Inquiry (MIDI-CI) Version 1.1* specification for a list of the minimum requirements to implement MIDI-CI.