



SURFACE VEHICLE RECOMMENDED PRACTICE

J2534™-5_0500

JAN2022

Issued

2022-01

Pass-Thru Interface - Alternate Platforms for API Version 05.00

RATIONALE

SAE J2534 was originally published in 2002 and supported the dominate platform of that time: personal computers running 32-bit Microsoft Windows®. Since then, computer technology has advanced and new operating systems have gained market share. The automotive industry needs to be able to select the platform that best suit their needs, without having to define an entirely new interface.

This document details the SAE J2534 (API Version 05.00) requirements to support alternate platforms and to allow applications to move between API versions. This will expand the number of products that are available to maintain and repair vehicles, which are becoming increasingly complex and technologically advanced.

TABLE OF CONTENTS

1.	SCOPE.....	3
2.	REFERENCES.....	3
2.1	Applicable Documents	3
2.1.1	SAE Publications.....	3
3.	DEFINITIONS	4
4.	DOCUMENT CONVENTIONS.....	4
5.	SWITCHING PLATFORMS AND/OR API VERSIONS WITH THE SAME PASS-THRU INTERFACE.....	4
6.	SAE J2534 (API VERSION 05.00) FOR 64-BIT MICROSOFT WINDOWS®	4
6.1	Application Programming Interface.....	4
6.1.1	DEFINES/TYPEDEFS.....	4
6.1.2	Structures.....	10
6.1.3	Function Prototypes	12
6.2	DLL Installation and Registry	12
6.2.1	Registry Key.....	12
6.2.2	Naming of Files	12
6.2.3	WIN64 Registry	13
6.2.4	Export Library Definition File.....	14
7.	NOTES.....	14
7.1	Revision Indicator.....	14

SAE Executive Standards Committee Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2022 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
http://www.sae.org

SAE WEB ADDRESS:

Full document details: SAE J2534-5_5_00

For more information on this standard, visit
https://www.sae.org/standards/content/J2534-5_5_00

Documents

Document Sharing Hub

Figure 1	WIN64 registry structure	13
Table 1		4
Table 2	WIN64 registry values.....	13
Table 3	WIN64 registry example.....	14

1. SCOPE

This SAE Recommended Practice details the changes needed to support the SAE J2534 (API Version 05.00) on operating systems other than 32-bit Microsoft Windows®.

2. REFERENCES

2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise indicated, the latest issue of the SAE publication shall apply.

2.1.1 SAE Publications

SAE J2534-1_0500 (API Version 05.00)

SAE J2534-2/BA (API Version 05.00)

SAE J2534-2/RE (API Version 05.00)

SAE J2534-2/1 (API Version 05.00)

SAE J2534-2/2 (API Version 05.00)

SAE J2534-2/3 (API Version 05.00)

SAE J2534-2/4 (API Version 05.00)

SAE J2534-2/5 (API Version 05.00)

SAE J2534-2/6 (API Version 05.00)

SAE J2534-2/7 (API Version 05.00)

SAE J2534-2/8 (API Version 05.00)

SAE J2534-2/9 (API Version 05.00)

SAE J2534-2/10 (API Version 05.00)

SAE J2534-2/11 (API Version 05.00)

SAE J2534-2/13 (API Version 05.00)

SAE J2534-2/15 (API Version 05.00)

SAE J2534-2/16 (API Version 05.00)

SAE J2534-2/17 (API Version 05.00)

SAE J2534-2/18 (API Version 05.00)

3. DEFINITIONS

3.1 OPERATING SYSTEM

The software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals.

3.2 REGISTRY

A mechanism within Microsoft's Windows operating systems to handle hardware and software configuration information.

4. DOCUMENT CONVENTIONS

This document details only the changes from the designated API Version. Unless otherwise noted, all SAE J2534 functionality remains as it was originally defined.

In the document body and in tables:

- All SAE J2534 API functions shall be in bold print (for example, **PassThruOpen**).
- All SAE J2534 API variables shall be enclosed in "<>" (for example, <DeviceID>).
- All SAE J2534 API DEFINES shall be in upper-case print (for example, STATUS_NOERROR).
- All number shall be in decimal unless otherwise indicated. Hexadecimal values shall have a leading "0x"; for example, 0x7AF or 0x800 are hexadecimal values.

5. SWITCHING PLATFORMS AND/OR API VERSIONS WITH THE SAME PASS-THRU INTERFACE

Applications interested in switching between API versions should consult SAE J2534-2/BA_0500 in the "Switching Platforms and/or API Versions with the Same Pass-Thru Interface" section.

6. SAE J2534 (API VERSION 05.00) FOR 64-BIT MICROSOFT WINDOWS®

6.1 Application Programming Interface

6.1.1 DEFINES/TYPEDEFS

In general, unless otherwise specified, Table 1 should be used when mapping types from the WIN32 API to the WIN64 API:

Table 1

WIN32 Type	WIN64 TYPE
char *	uint8_t *
unsigned char	uint8_t
unsigned char *	uint8_t *
integer	int32_t
integer *	int32_t *
unsigned integer	uint32_t
unsigned integer *	uint32_t *
long	int32_t
long *	int32_t *
unsigned long	uint32_t
unsigned long *	uint32_t *

```
enum e_protocol_id : uint32_t
{
    // J2534-1
    J1850VPW = 1,
    J1850PWM = 2,
    ISO9141 = 3,
    ISO14230 = 4,
    CAN = 5,
    J2610 = 7,
    ISO15765_LOGICAL = 0x200,

    //J2534-2
    SW_CAN = 0x8008,
    GM_UART = 0x8009,
    UART_ECHO_BYTE = 0x800A,
    HONDA_DIAGH = 0x800B,
    J1708 = 0x800D,
    FT_CAN = 0x800F,
    FD_CAN = 0x8011,
    ETHERNET_NDIS = 0x8013,
    ETHERNET = 0x8015,
    J1939_LOGICAL = 0x8200,
    TP2_0_LOGICAL = 0x8201,
    IP4_LOGICAL = 0x8203,
    IP6_LOGICAL = 0x8204,
    UDP_LOGICAL = 0x8205,
    TCP_CLIENT_LOGICAL = 0x8206,
    ANALOG_IN = 0xC000
};
typedef e_protocol_id protocol_id_t;

enum e_ioctl_id : uint32_t
{
    GET_CONFIG = 0x01,
    SET_CONFIG = 0x02,
    READ_PIN_VOLTAGE = 0x03,
    FIVE_BAUD_INIT = 0x04,
    FAST_INIT = 0x05,
    CLEAR_TX_QUEUE = 0x07,
    CLEAR_RX_QUEUE = 0x08,
    CLEAR_PERIODIC_MSGS = 0x09,
    CLEAR_MSG_FILTERS = 0x0A,
    CLEAR_FUNCT_MSG_LOOKUP_TABLE = 0x0B,
    ADD_TO_FUNCT_MSG_LOOKUP_TABLE = 0x0C,
    DELETE_FROM_FUNCT_MSG_LOOKUP_TABLE = 0x0D,
    READ_PROG_VOLTAGE = 0x0E,
    BUS_ON = 0x0F,

    // J2534 2_V0404
    START_REPEAT_MESSAGE = 0x8004,
    QUERY_REPEAT_MESSAGE = 0x8005,
    STOP_REPEAT_MESSAGE = 0x8006,
    GET_DEVICE_CONFIG = 0x8007,
    SET_DEVICE_CONFIG = 0x8008,
    PROTECT_J1939_ADDR = 0x8009,
    REQUEST_CONNECTION = 0x800A,
    TEARDOWN_CONNECTION = 0x800B,
    GET_DEVICE_INFO = 0x800C,
    GET_PROTOCOL_INFO = 0x800D,
    NS_AUTO_CHANGE_MSG = 0x8011,
    HS_AUTO_CHANGE_MSG = 0x8012,
    NON_BLOCKING_FIVE_BAUD_INIT = 0x8013,
    NON_BLOCKING_FAST_INIT = 0x8014,
    GET_ANALOG_CHANNEL_INFO = 0x8015,
    SET_MAC_ADDRESS = 0x8020,
    GET_MAC_ADDRESS = 0x8021,
    ETH_ACTIVATION_PULLUP = 0x8022,
    GET_SOCKET_STATE = 0x8023,
    GET_DATALENGTH_IN_FIRST_PTMSG = 0x8024,
    IP4_GET_ADDRESS_INFO = 0x8025,
    IP6_GET_ADDRESS_INFO = 0x8026,
    GET_DEVICE_NAME = 0x8027
};
typedef e_ioctl_id ioctl_id_t;

enum e_ioctl_config : uint32_t
```