

# Jacobsen Declaration Exhibit AW

# **Train Server<sup>®</sup> Interface Description Volume I**

---

Building your own visual interface to a model  
railroad

Copyright 1992-1998 KAM Industries.

**Computer Dispatcher™, Engine Commander™, The Conductor™, Train Server™, and Train Tools™**  
are Trademarks of KAM Industries, all Rights Reserved.

Questions concerning the product can be EMAILED to:

trainserver@kam.rain.com

You can also mail questions to:

KAM Industries  
2373 NW 185<sup>th</sup> Avenue Suite 416  
Hillsboro, Oregon 97124  
FAX - (503) 291-1221

---

# Table of contents

1. OVERVIEW .....	1
1.1 System Architecture .....	1
1.2 COM Interfaces .....	1
1.3 API Implementation.....	1
1.4 API Version Information .....	2
1.5 Tutorial Samples .....	2
2. COMMAND REFERENCE .....	3
2.1 Introduction.....	3
2.2 Data Types .....	3
2.3 Decoder Classes & Types.....	4
2.4 Commands to access the server configuration variable cache.....	5
DccCVGetValue .....	5
DccCVSetValue.....	5
DccCVGetStatus .....	6
DccCVSetStatus.....	6
DccCVGetName.....	7
DccCVGetMinRegister .....	7
DccCVGetMaxRegister .....	8
2.5 Commands to program configuration variables.....	9
DccProgramSetMode.....	9
DccProgramGetMode .....	10
DccProgramGetStatus .....	10
DccProgramReadCV.....	11
DccProgramWriteCV.....	11
DccProgramReadDecoderToCache .....	11

DccProgramWriteDecoderFromCache .....	12
2.6 Commands to control all decoder types.....	13
DccDecoderGetMaxModels .....	13
DccDecoderGetModelName .....	13
DccDecoderTranslate.....	14
DccDecoderSetModelToObject.....	14
DccDecoderGetMaxAddress.....	15
DccDecoderChangeAddress .....	15
DccDecoderCheckAddrInUse .....	16
DccDecoderGetModelFacility.....	17
DccDecoderGetObjectCount.....	18
DccDecoderGetObjectAtIndex .....	18
DccDecoderAddAddr.....	19
DccDecoderReconnectObject.....	19
DccDecoderDel .....	20
DccDecoderGetMfgName.....	20
DccDecoderGetPowerMode.....	21
DccDecoderGetMaxSpeed .....	21
DccDecoderGetErrorState .....	22
DccDecoderCopySpeedTbIToObject.....	22
2.7 Commands to control locomotive decoders.....	23
DccEngGetSpeed.....	23
DccEngSetSpeed .....	24
DccEngGetSpeedSteps .....	24
DccEngSetSpeedSteps.....	25
DccEngGetFunction .....	25
DccEngSetFunction.....	26

DccEngGetFunctionMax .....	26
DccEngGetName .....	27
DccEngSetName.....	27
DccEngGetFunctionName .....	28
DccEngSetFunctionName.....	28
DccEngConsistGetMax .....	29
DccEngConsistSetParent.....	29
DccEngConsistAddUnit .....	30
DccEngConsistRemoveUnit.....	30
DccEngConsistGetParent .....	30
2.8 Commands to control accessory decoders .....	31
DccAccGetFunction .....	31
DccAccGetFunctionAll .....	31
DccAccSetFunction.....	32
DccAccSetFunctionAll.....	32
DccAccGetFunctionMax .....	33
DccAccGetName.....	33
DccAccSetName .....	34
DccAccGetFunctionName.....	34
DccAccSetFunctionName .....	35
DccAccSetResetOrScan .....	35
2.9 Commands to set speed tables.....	36
DccSpeedGetMaxTableNumber .....	36
DccSpeedGetTableValue.....	36
DccSpeedSetTableValue .....	37
DccSpeedGetTableName .....	37
DccSpeedSetTableName.....	38

2.10	Commands to control the command station .....	39
	DccOprTurnOnStation.....	39
	DccOprStartStation .....	39
	DccOprClearStation .....	40
	DccOprStopStation.....	40
	DccOprPowerOn .....	40
	DccOprPowerOff .....	41
	DccOprHardReset .....	41
	DccOprEmergencyStop .....	41
	DccOprGetStationStatus.....	42
2.11	Commands to configure the command station communication port .....	43
	DccPortSetConfig.....	44
	DccPortGetConfig .....	45
	DccPortGetName .....	45
	DccPortSetMapController.....	46
	DccPortGetMapController .....	46
	DccPortGetMaxLogPorts .....	47
	DccPortGetMaxPhysical .....	47
2.12	Commands that control command flow to the command station .....	48
	DccCmdConnect .....	48
	DccCmdDisconnect.....	48
	DccCmdCommand.....	49
2.13	Cab Control Commands .....	50
	DccCabReadMessage.....	50
	DccCabWriteMessage .....	50
	DccCabGetDccObject .....	51
	DccCabSetDccObject .....	51

DccCabAdd .....	52
DccCabDelete .....	52
DccCabLookupCabObject.....	53
DccCabTranslate.....	53
2.14 Miscellaneous Commands.....	54
DccMiscGetErrorMsg .....	54
DccMiscGetClockTime.....	54
DccMiscSetClockTime .....	55
DccMiscGetInterfaceVersion.....	55
DccMiscSaveData.....	56
DccMiscGetControllerName.....	56
DccMiscGetControllerNameAtPort .....	57
DccMiscGetCommandStationValue .....	57
DccMiscSetCommandStationValue.....	58
DccMiscGetCommandStationIndex.....	58
DccMiscMaxControllerID.....	59
DccMiscGetControllerFacility .....	59
DccMiscGetServerInfo .....	60
DccMiscGetMaxFeedbackBus.....	60
DccMiscGetFeedbackBusName .....	61
3. END OF DOCUMENT .....	62



## Table of Figures

Figure 1: Train Server™ Functional Diagram.....	1
Figure 2: Basic data types .....	3
Figure 3: Boolean types.....	3
Figure 4: Decoder classes .....	4
Figure 5: Basic Locomotive Decoder Types .....	4
Figure 6: Basic Accessory Decoder Types.....	4
Figure 7: Controller ID to controller name mapping .....	43
Figure 8: Controller configuration Index values.....	44

# 1. OVERVIEW

This manual is divided into two sections, an Overview and a Command Reference. The Overview gives general information about the interface. The Command Reference describes each command in detail.

---

## 1.1 System Architecture

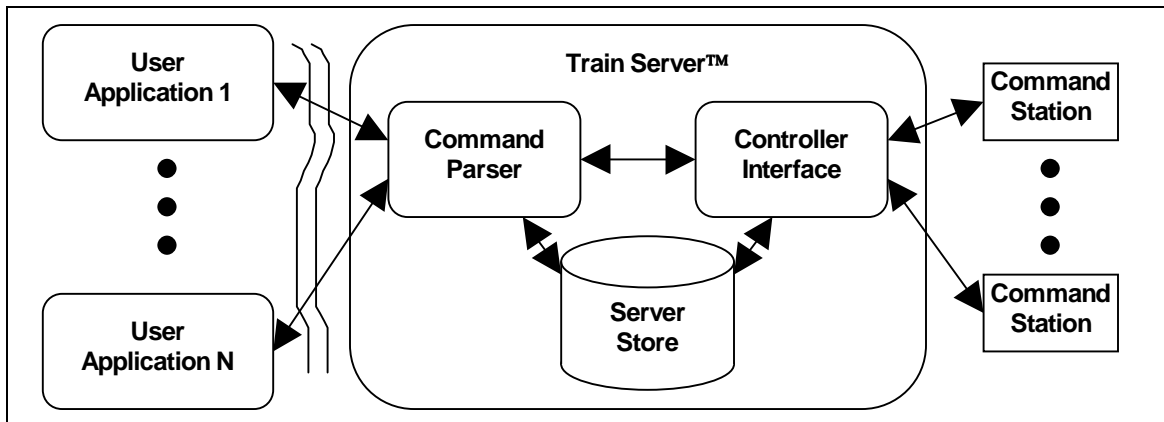


Figure 1: Train Server™ Functional Diagram

**Train Tools™** is the applications interface that uses the **Train Server™** to send commands to and from the command stations. The **Train Server™** uses a patented cache model to maintain decoder state, provide updates, and to establish a fault tolerant command execution sequence to the command station.

This interface specification allows the programmer to select functionality to meet end user expectations. Sample applications are constantly being generated to utilize these features. Sample application software is available at KAM's website at <http://www.kamind.com> and on the **Train Tools™** CD. These samples include throttles, fast clocks, universal programmers, and event feedback.

---

## 1.2 COM Interfaces

The **Train Server™** interface is based on two COM object interfaces. The primary interface is **IEngComIfc** that sends commands from clients to the **Train Server™** program. Other interfaces such as an interface that provides asynchronous feedback interface from the **Train Server™** to the client can be added by extending the interface.

---

## 1.3 API Implementation

All APIs defined in this specification have an associated API ID along with an implementation state. The state of an API is determined via the command **DccMiscGetApiName**. See the miscellaneous command section for details on this command.

---

## 1.4 API Version Information

The API implementation will have changes and additions over time. The version of the API is determined by the command **DccMiscGetInterfaceVersion**. See the miscellaneous command section for details on this command.

---

## 1.5 Tutorial Samples

This interface specification allows the programmer to select functionality to meet end user expectations. Sample applications are constantly being generated to utilize these features. Sample application software is available at KAM's website at <http://www.kamind.com> and on the **Train Tools™** CD. These samples include throttles, fast clocks, universal programmers, and event feedback.

Programmers can also consult **Developing Train Server® Applications using the Proposed NMRA Application Programming Interface** for a tutorial on developing applications.