

TrainTools™ Interface Description

Building your own visual interface to a model
railroad

Engine-Commander, ComputerDispatcher, TrainBasic, and TrainTools are trademarks of KAM Industries, all rights reserved.

Questions concerning the product can be EMAILED to:

traintools@kam.rain.com

You can also mail questions to:

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

Table of contents

1. OVERVIEW	1
2. TUTORIAL	2
2.1 Visual BASIC Example	2
3. IDL COMMAND REFERENCE	7
3.1 Introduction	7
3.2 Data Types	7
3.3 Commands to read and write configuration variables	7
KamCVGetValue	8
KamCVGetValueAll	8
KamCVPutValue	9
KamCVPutValueAll	9
KamCVGetEnable	10
KamCVGetEnableAll	10
KamCVPutEnable	11
KamCVPutEnableAll	11
KamCVGetName	12
KamCVGetMinRegister	12
KamCVGetMaxRegister	12
KamProgramCV	13
KamProgramReadDecoderToDataBase	13
KamProgramDecoderFromDataBase	13
3.4 Commands to control locomotive decoders	14
KamEngGetSpeed	14
KamEngPutSpeed	15

KamEngGetSpeedSteps	15
KamEngPutSpeedSteps	16
KamEngGetFunction	16
KamEngPutFunction.....	17
KamEngGetFunctionAll	17
KamEngPutFunctionAll.....	18
KamEngGetFunctionMax	18
KamEngGetName	18
KamEngPutName	19
KamEngGetFunctionName	19
KamEngPutFunctionName	20
KamEngPutAdd	20
KamEngPutDel	21
KamCmdCommand.....	21
3.5 Commands to control accessory decoders.....	22
KamAccGetFunction.....	22
KamAccGetFunctionAll.....	22
KamAccPutFunction	23
KamAccPutFunctionAll	23
KamAccGetFunctionMax	24
KamAccGetName	24
KamAccPutName	24
KamAccGetFunctionName	25
KamAccPutFunctionName	25
KamAccPutAdd	26
KamAccPutDel.....	26

3.6 Commands to control the command station	27
KamOprPutTurnOnStation	27
KamOprPutStartStation	27
KamOprPutClearStation	28
KamOprPutStopStation	28
KamOprPutPowerOn	28
KamOprPutPowerOff	29
KamOprPutHardReset	29
KamOprPutEmergencyStop	29
KamOprGetStationStatus	30
3.7 Commands to configure the command station communication port	31
KamPortPutRestrans	31
KamPortPutRate	32
KamPortPutParity	32
KamPortPutStop	33
KamPortPutWatchdog	33
KamPortPutFlow	34
KamPortPutData	34
KamPortGetName	35
KamPortPutDebug	35
KamPortPutMapController	36
KamPortGetMaxLogPorts	36
KamPortGetMaxPhysical	36
3.8 Commands that control command flow to the command station.....	37
KamCmdConnect	37
KamCmdDisConnect.....	37

3.9 Commands that access the decoder information database.....	38
KamDecoderGetMaxModels.....	38
KamDecoderGetModelName.....	38
KamDecoderSetModelToAddr.....	39
KamDecoderGetMaxAddress.....	39
KamDecoderChangeOldNewAddr.....	39
KamDecoderMovePort.....	40
3.10 Miscellaneous Commands.....	41
KamMiscGetErrorMsg.....	44
KamMiscGetClockTime.....	44
KamMiscPutClockTime.....	45
KamMiscGetInterfaceVersion.....	45
KamMiscSaveData.....	45
KamMiscGetControllerName.....	46
KamMiscMaxControllerID.....	46

Table of Figures

Figure 1: Controller ID to controller name mapping.....	31
---	----

1. OVERVIEW

This manual is divided into two sections, the Tutorial, and the IDL Command Reference. The tutorial shows the complete code for a simple Visual BASIC program that controls all the major functions of a locomotive. This program makes use of many of the commands described in the reference section. The IDL Command Reference describes each command in detail.

2. TUTORIAL

2.1 Visual BASIC Example

```
' Copyright 1997, KAM Industries. All rights reserved.
',
' This is a demonstration program showing the intergration of VisualBasic and
' EngineCommadner(tm) TrainTools(tm) interface. You may use this application for
' non commerical usage.
',
' $Date: $
' $Author: $
' $Revision: $
' $Log: $

' Copyright 1997, KAM Industries. All rights reserved.
',
' This is a demonstration program showing the intergration of VisualBasic and
' EngineCommadner(tm) TrainTools(tm) interface. You may use this application for
' non commerical usage.
',

' This first command adds the reference to the TrainTools Interface object
Dim EngCmd As New EngComlfc

' Engine Commander uses the term Ports, Devices and Controllers
' Ports -> These are logical ids where Decoders are assigned to. TrainTools Interface
' supports a limited number of logical ports. You can also think of
' of ports as mapping to a command station type. This allows you to move
' decoders between command station without loosing any information about the decoder
' Devices -> These are communciations channels configured in your computer. You may have a
single
' device (com1) or mutple devices (COM 1 - COM30). You are required to map a port to
' a device to access a command station. Devices start from ID 0 -> max id
' (FYI; devices do not necessarily have to be serial channel. Always check the name
' of the device before you use it as well as the maximum number of devices supported
' Controler - These are command the command station like LENZ, Digitrax or Northcoast. It is
recommend that
' you check the command station ID before you use it.
',

' To Operate your layout you will need to perform a mapping between
' a Port (logical reference), Device (physical communications channle) and a Controller (commadn
station)
' for the program to work.
Dim iLogPort, iController, iComPort As Integer
',

' Note:
' Please follow the command order. Order is important for the beta product
' to work!

Private Sub Command_Click()
```

```

'Send the command from the interface to the command station
Dim iStatus, iAddr, iSpeed As Integer
If Not Connect.Enabled Then
    ' TrainTools interface is a caching interface. This means that you need to set up
    ' the CV's or other operations first; then execute the command.
    iAddr = Address.Text
    iSpeed = Speed.Text
    iStatus = EngCmd.KamEngPutFunction(iAddr, 0, F0.Value)
    iStatus = EngCmd.KamEngPutFunction(iAddr, 1, F1.Value)
    iStatus = EngCmd.KamEngPutFunction(iAddr, 2, F2.Value)
    iStatus = EngCmd.KamEngPutFunction(iAddr, 3, F3.Value)
    iStatus = EngCmd.KamEngPutSpeed(iAddr, iSpeed, Direction.Value)
    If iStatus = 0 Then iStatus = EngCmd.KamCmdCommand(iAddr)
    SetError (iStatus)
End If

End Sub

Private Sub Connect_Click()
    Dim iStatus, iMaxLogs, iMaxControllers As Integer

    iMaxLogs = EngCmd.KamPortGetMaxLogPorts()
    iLogPort = 1 ' use logical port 1; You need to check the max number of
                ' controllers that you can use. 1 is always a save bet

    ' These are the potential values for the command station. Please check the
    ' KamMiscMaxControllId for the maximum controller ID supported
    iMaxControllers = EngCmd.KamMiscMaxControllerId()
    ' UNKNOWN      0 // Unknown control type
    ' SIMULAT      1 // Interface simulator
    ' LENZ_1x      2 // Lenz serial support module
    ' LENZ_2x      3 // Lenz serial support module
    ' DIGIT_DT200  4 // Digitax direct drive support using DT200
    ' DIGIT_DCS100 5 // Digitax direct drive support using DCS100
    ' MASTERSERIES 6 // North Coast engineering master Series
    ' SYSTEMONE    7 // System One
    ' RAMFIX       8 // RAMFix system
    ' SERIAL       9 // NMRA Serial interafacce
    ' EASYDCC      10 // NMRA Serial interafacce
    ' MRK6050      11 // 6050 Marklin interface (AC and DC)
    ' MRK6023      12 // 6023 Marklin hybrid interface (AC)
    ' DIGIT_PR1    13 // Digitax direct drive support using PR1
    ' DIRECT       14 // Direct drive interface routine
    ' ZTC          15 // ZTC Systems ltd
    ' TRIX         16 // TRIX
    iController = 2 ' Use lenz controller vers 1.5 (ID 2)
    iComPort = 3 ' use COM4; 0 means com1 (digitrax must use Com1 or Com2)
    bConnected = False ' Set to disconenct state

    ' use these values; don't change
    iStatus = EngCmd.KamPortPutFlow(iLogPort, 0)
    iStatus = EngCmd.KamPortPutRetrans(iLogPort, 10) ' number of retries before packet is dropped
    ' 0: // Baud rate is 300
    ' 1: // Baud rate is 1200
    ' 2: // Baud rate is 2400
    ' 3: // Baud rate is 4800

```

```
' 4: // Baud rate is 9600
' 5: // Baud rate is 14.4
' 6: // Baud rate is 16.4
' 7: // Baud rate is 19.2
iStatus = EngCmd.KamPortPutRate(iLogPort, 4) ' 4-> 9600 baud
'   Stop bits 0,1,2 -> 1, 1.5, 2
iStatus = EngCmd.KamPortPutStop(iLogPort, 0)
'   Parity values 0-4 -> no, odd, even, mark, space
iStatus = EngCmd.KamPortPutParity(iLogPort, 0)
'   Date bits 0, 1 -> 7 bits, 8 bits
iStatus = EngCmd.KamPortPutData(iLogPort, 1)
iStatus = EngCmd.KamPortPutWatchdog(iLogPort, 2048) ' delay between packet retries; don't
change for serial link
```

```
' We need to set the appropriate debug mode for display..
' this command can only be sent if the following is true
' - Controller is not connected
' - port has not been mapped
,
```

```
' Write Display Log Debug
' File Win Level Value
' 1 + 2 + 4 = 7 -> LEVEL1 -- put packets into queues
' 1 + 2 + 8 = 11 -> LEVEL2 -- Status messages send to display windows
' 1 + 2 + 16 = 19 -> LEVEL3 --
' 1 + 2 + 32 = 35 -> LEVEL4 -- All system semaphores and critical sections
' 1 + 2 + 64 = 67 -> LEVEL5 -- If you don't know where to put it put it in level 5
' 1 + 2 + 128 = 131 -> COMMONLY -- Read comm write comm ports
,
```

```
' Trust me on this one, you prob only want to use values of 130. This will give you a display
' what is read or written to the controller. If you want to write the information to
' disk, use 131.
,
```

```
' Note: This does effect the performance of you system; 130 is a save value for debug
' display. Always set the key to 1
,
```

```
iDebugMode = 130
iStatus = EngCmd.KamPortPutDebug(iLogPort, iDebugMode, 1) '1 is the key
```

```
' 2-> Lenz controller, 3 is Com4
iStatus = EngCmd.KamPortPutMapController(iLogPort, iController, iComPort)
iStatus = EngCmd.KamCmdConnect(iLogPort)
iStatus = EngCmd.KamOprPutTurnOnStation(iLogPort)
If (iStatus) Then
    Connect.Enabled = True
Else
    Connect.Enabled = False
End If
SetError (iStatus)
```

```
' Function not implmented; disable
Clear.Enabled = False
Reset.Enabled = False
```

```
' Display the port and controller information
Port.Caption = EngCmd.KamPortGetName(iComPort)
Controller.Caption = EngCmd.KamMiscGetControllerName(iController)
```

```

MaxLogPort.Caption = iLogPort

End Sub

Private Sub DCCAddr_Click()
    Dim iAddr, iStatus As Integer
    If Not Connect.Enabled Then
        ' All addresses must be match to a logical port to operate
        If (Address.Text > 1) Then iStatus = EngCmd.KamEngPutAdd(Address.Text, iLogPort)
        SetError (iStatus)
        Result.Caption = Str(iStatus)
        Command.Enabled = True ' turn on the command button to send speed...
    End If
End Sub

Private Sub Disconnect_Click()
    Dim iStatus As Integer
    If Not Connect.Enabled Then
        iStatus = EngCmd.KamCmdDisconnect(iLogPort)
        SetError (iStatus)
        Connect.Enabled = True
    End If
End Sub

Private Sub SetError(iError As Integer)
    Dim szError As String
    ' This shows how to retrieve a smaple error message from the interface
    ' for the status received.
    szError = EngCmd.KamMiscGetErrorMsg(iError)
    ErrorMessage.Caption = szError
End Sub

Private Sub GetVers_Click()
    Dim strVer As String
    Dim iStatus As Integer
    strVer = EngCmd.KamMiscGetInterfaceVersion
    Version.Caption = strVer

    'Display the port and controller information
    Port.Caption = EngCmd.KamPortGetName(iComPort)
    Controller.Caption = EngCmd.KamMiscGetControllerName(iController)
    MaxLogPort.Caption = EngCmd.KamPortGetMaxLogPorts
    MaxComPort.Caption = EngCmd.KamPortGetMaxPhysical

    ' Function not implemented; disable
    Clear.Enabled = False
    Reset.Enabled = False

End Sub

Private Sub Off_Click()
    Dim iStatus As Integer

```

```
If Not Connect.Enabled Then
    iStatus = EngCmd.KamOprPutPowerOff(iLogPort)
End If
End Sub
```

```
Private Sub ON_Click()
    If Not Connect.Enabled Then
        Status = EngCmd.KamOprPutPowerOn(iLogPort)
    End If
End Sub
```

```
Private Sub Throttle_Click()
    If Not Connect.Enabled Then
        If (Throttle.Value > 0) Then
            Speed.Text = Throttle.Value
        End If
    End If
End Sub
```

3. IDL COMMAND REFERENCE

3.1 Introduction

This document describes the IDL interface to the KAM industries Engine Commander COM server. This server handles all the background details of controlling your railroad. You write simple, front end programs in a variety of languages such as BASIC or C++ to provide the visual interface to the user while the server handles the details of communicating with the command station, etc.

3.2 Data Types

Data is passed to and from the IDL interface using a several primitive data types. Arrays of these simple types are also used. The exact type passed to and from your program depends on the programming language your are using.

The following primitive data types are used:

IDL Type	BASIC Type	C++ Type	Java Type	Description
short	short	short	short	Short signed integer
int	int	int	int	Signed integer
BSTR	BSTR	BSTR	BSTR	Text string

3.3 Commands to read and write configuration variables

This section describes the commands that read and write configuration variables (CVs). These CVs are stored in the decoder and control many of its characteristics such as its address. For efficiency, a copy of each CV value is also stored in the server. Commands such as **KamCVGetValue** and **KamCVPutValue** communicate only with the server, not the actual decoder. You should initially transfer a copy of the decoder CVs to the server using the **KamProgramReadDecoderToDataBase** command. You can then read and modify this server copy of the CVs. Finally, you can program one or more CVs into the decoder using the **KamProgramCV** and **KamProgramDecoderFromDataBase** commands respectively.

KamCVGetValue				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVReg	int	1-1024 ²	In	CV register
pCVValue	int *	³	Out	Pointer to CV value
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Range is 1-1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ CV Value pointed to has a range of 0 to 255.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVGetValue takes the decoder address and configuration variable (CV) number as parameters. It sets the memory pointed to by <i>pCVValue</i> to the value of the server copy of the configuration variable.				

KamCVGetValueAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVRegStart	int	²	In	Starting CV register
iCVRegEnd	int	²	In	Ending CV register
lpCVValueArray	int *	³	Out	Pointer to CV array
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Range is 1-1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ Maximum array size is 1024. Maximum for this decoder is given by KamCVGetMaxRegister .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVGetValueAll takes the decoder address and a pointer to a CV value array as parameters. It copies the specified range of CV values from the server copy of the CV array to the array pointed to by <i>lpCVValueArray</i> .				

KamCVPutValue				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVReg	int	1-1024 ²	In	CV register
iCVValue	int	0-255	In	CV value
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum CV is 1024. Maximum CV for this decoder is given by KamCVGetMaxRegister .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVPutValue takes the decoder address, configuration variable (CV) number, and a new CV value as parameters. It sets the server copy of the specified decoder CV to <i>iCVValue</i> .				

KamCVPutValueAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVRegStart	int	²	In	Starting CV register
iCVRegEnd	int	²	In	Ending CV register
lpCVValueArray	int *	³	Out	Pointer to CV array
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Range is 1-1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ Maximum array size is 1024. Maximum for this decoder is given by KamCVGetMaxRegister .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVPutValueAll takes the decoder address and a pointer to a CV value array as parameters. It sets the specified range of server CV values to match the values in the array pointed to by <i>lpCVValueArray</i> .				

KamCVGetEnable				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVReg	int	1-1024 ²	In	CV number
pEnable	int *	³	Out	Pointer to enable state
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum CV is 1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ <i>pEnable</i> should point to a boolean variable containing the value TRUE or FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVGetEnable takes the decoder address, configuration variable (CV) number, and a pointer to store the enable flag as parameters. It sets the location pointed to by <i>pEnable</i> TRUE if the given server copy of the CV is enabled or FALSE otherwise.				

KamCVGetEnableAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVRegStart	int	²	In	Starting CV register
iCVRegEnd	int	²	In	Ending CV register
lpCVEnableArray	int *	³	Out	Pointer to CV enable array
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Range is 1-1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ Maximum array size is 1024. Maximum for this decoder is given by KamCVGetMaxRegister .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVGetEnableAll takes the decoder address and a pointer to a CV enable array as parameters. It sets the specified range of server enable states in the <i>lpCVEnableArray</i> to TRUE if the corresponding decoder CV is enabled or FALSE otherwise.				

KamCVPutEnable				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVReg	int	1-1024 ²	In	CV number
iEnable	int	³	In	Enable state
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum CV is 1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ Enable is boolean TRUE and disable is boolean FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVPutEnable takes the decoder address, configuration variable (CV) number, and a new enable state as parameters. It sets the server copy of the enable state to <i>iEnable</i> .				

KamCVPutEnableAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVRegStart	int	²	In	Starting CV register
iCVRegEnd	int	²	In	Ending CV register
lpCVEnableArray	int *	³	In	Pointer to CV enable array
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Range is 1-1024. Maximum CV for this decoder is given by KamCVGetMaxRegister . ³ Maximum array size is 1024. Maximum for this decoder is given by KamCVGetMaxRegister .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCVPutEnableAll takes the decoder address and a pointer to a CV enable array as parameters. It sets the specified range of server enable states to match the states in <i>lpCVEnableArray</i> . The possible enable states are TRUE and FALSE .				

KamCVGetName				
Parameter List	Type	Range	Direction	Description
iCV	int	1-1024	In	CV number
Return Value	Type	Range	Description	
iCVName	BSTR	1	CV name	
¹ Exact return type depends on language. It is CString for C++.				
KamCVGetName takes a configuration variable (CV) number as a parameter. It returns a BSTR set to the name of the CV as defined in NMRA Recommended Practice RP 9.2.2.				

KamCVGetMinRegister				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	1	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iMinRegister	int	1 - 1024	Minimum CV register number	
KamCVGetMinRegister takes a decoder address as a parameter. It returns the minimum possible CV register number for the specified decoder.				

KamCVGetMaxRegister				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	1	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iMaxRegister	int	1 - 1024	Maximum CV register number	
KamCVGetMaxRegister takes a decoder address as a parameter. It returns the maximum possible CV register number for the specified decoder.				

KamProgramCV				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCVReg	int	²	In	CV number
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum CV is 1024. Maximum CV for this decoder is given by KamCVGetMaxRegister .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamProgramCV takes the decoder address and configuration variable (CV) number as parameters. It programs the specified decoder CV using the server copy of the CV as source data.				

KamProgramReadDecoderToDataBase				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamProgramReadDecoderToDataBase takes the decoder address as a parameter. It reads all CV values from the decoder and stores them in the server database.				

KamProgramDecoderFromDataBase				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamProgramDecoderFromDataBase takes the decoder address as a parameter. It programs all decoder CV values using the server copy of the CVs as source data.				

3.4 Commands to control locomotive decoders

This section describes the commands that control locomotive decoders. These commands control things such as locomotive speed and direction. For efficiency, a copy of all the engine variables such speed is stored in the server. Commands such as **KamEngGetSpeed** communicate only with the server, not the actual decoder. You should first make any changes to the server copy of the engine variables. You can send all changes to the engine using the **KamCmdCommand** command.

KamEngGetSpeed				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
lpSpeed	int *	²	Out	Pointer to locomotive speed
lpDirection	int *	³	Out	Pointer to locomotive direction
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Speed range is dependent on whether the decoder is set to 14,18, or 128 speed steps and matches the values defined by NMRA S9.2 and RP 9.2.1. 0 is stop and 1 is emergency stop for all modes. ³ Forward is boolean TRUE and reverse is boolean FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngGetSpeed takes the decoder address and pointers to locations to store the locomotive speed and direction as parameters. It sets the memory pointed to by <i>lpSpeed</i> to the locomotive speed and the memory pointed to by <i>lpDirection</i> to the locomotive direction.				

KamEngPutSpeed				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iSpeed	int	²	In	Locomotive speed
iDirection	int	³	In	Locomotive direction
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Speed range is dependent on whether the decoder is set to 14,18, or 128 speed steps and matches the values defined by NMRA S9.2 and RP 9.2.1. 0 is stop and 1 is emergency stop for all modes. ³ Forward is boolean TRUE and reverse is boolean FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutSpeed takes the decoder address, new locomotive speed, and new locomotive direction as parameters. It sets the locomotive database speed to <i>iSpeed</i> and the locomotive database direction to <i>iDirection</i> . Note: This command only changes the locomotive database. The data is not sent to the decoder until execution of the KamCmdCommand command.				

KamEngGetSpeedSteps				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
lpSpeedSteps	int *	14,28,128	Out	Pointer to number of speed steps
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngGetSpeedSteps takes the decoder address and a pointer to a location to store the number of speed steps as a parameter. It sets the memory pointed to by <i>lpSpeedSteps</i> to the number of speed steps.				

KamEngPutSpeedSteps				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iSpeedSteps	int	14,28,128	In	Locomotive speed steps
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutSpeedSteps takes the decoder address and a new number of speed steps as a parameter. It sets the number of speed steps in the locomotive database to <i>iSpeedSteps</i> . Note: This command only changes the locomotive database. The data is not sent to the decoder until execution of the KamCmdCommand command.				

KamEngGetFunction				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iFunctionID	int	0-8 ²	In	Function ID number
lpFunction	int *	³	Out	Pointer to function value
¹ 1-127 for short addresses, 1-10239 for long addresses.				
² FL is 0. F1-F8 is 1-8 respectively. Maximum for this decoder is given by KamEngGetFunctionMax .				
³ Function active is boolean TRUE and inactive is boolean FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngGetFunction takes the decoder address, a function ID, and a pointer to the location to store the specified function state as parameters. It sets the memory pointed to by <i>lpFunction</i> to the specified function state.				

KamEngPutFunction				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iFunctionID	int	0-8 ²	In	Function ID number
iFunction	int	³	In	Function value
¹ 1-127 for short addresses, 1-10239 for long addresses. ² FL is 0. F1-F8 is 1-8 respectively. Maximum for this decoder is given by KamEngGetFunctionMax . ³ Function active is boolean TRUE and inactive is boolean FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutFunction takes the decoder address, a function ID, and a new function state as parameters. It sets the specified locomotive database function state to <i>iFunction</i> . Note: This command only changes the locomotive database. The data is not sent to the decoder until execution of the KamCmdCommand command.				

KamEngGetFunctionAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
lpFunction	int *	²	Out	Pointer to function state array
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum array size is 9. Maximum for this decoder is given by KamEngGetFunctionMax + 1.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngGetFunctionAll takes the decoder address and a pointer to a boolean function state array as parameters. It sets all function states in the <i>lpFunction</i> array to TRUE if the corresponding decoder function is enabled or FALSE otherwise.				

KamEngPutFunctionAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
lpFunction	int *	²	In	Pointer to function state array
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum array size is 9. Maximum for this decoder is given by KamEngGetFunctionMax + 1.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
<p>KamEngPutFunctionAll takes the decoder address and a pointer to a function state array as parameters. It sets all decoder function enable states to match the states in <i>lpFunction</i>. The possible enable states are TRUE and FALSE. The data is not sent to the decoder until execution of the KamCmdCommand command.</p>				

KamEngGetFunctionMax				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iMaxFunction	int	0-8	Maximum function number	
<p>KamEngGetFunctionMax takes a decoder address as a parameter. It returns the maximum possible function number for the specified decoder.</p>				

KamEngGetName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
bsEngName	BSTR	¹	Locomotive name	
¹ Exact return type depends on language. It is Cstring for C++.				
<p>KamEngGetName takes a decoder address as a parameter. It returns a BSTR set to the name of the locomotive.</p>				

KamEngPutName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
bsEngName	BSTR	²	Out	Locomotive name
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Exact parameter type depends on language. It is LPCSTR for C++.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutName takes a decoder address and a BSTR as parameters. It sets the symbolic locomotive name to <i>bsEngName</i> .				

KamEngGetFunctionName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iFunctionID	int	0-8 ²	In	Function ID number
¹ 1-127 for short addresses, 1-10239 for long addresses. ² FL is 0. F1-F8 is 1-8 respectively. Maximum for this decoder is given by KamEngGetFunctionMax .				
Return Value	Type	Range	Description	
bsFcnNameString	BSTR	¹	Function name	
¹ Exact return type depends on language. It is Cstring for C++.				
KamEngGetFunctionName takes a decoder address and a function ID as parameters. It returns a BSTR set to the symbolic name of the specified function.				

KamEngPutFunctionName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iFunctionID	int	0-8 ²	In	Function ID number
bsFcnNameString	BSTR	³	In	Function name
¹ 1-127 for short addresses, 1-10239 for long addresses. ² FL is 0. F1-F8 is 1-8 respectively. Maximum for this decoder is given by KamEngGetFunctionMax . ³ Exact parameter type depends on language. It is LPCSTR for C++.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutFunctionName takes a decoder address, function ID, and a BSTR as parameters. It sets the specified symbolic function name to <i>bsFcnNameString</i> .				

KamEngPutAdd				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iLogicalPortID	int	1-65535 ²	In	Logical port ID
¹ 1-127 for short addresses, 1-10239 for long addresses. ² Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutAdd takes a decoder address as a parameter. It creates a new locomotive object in the locomotive database using <i>iDCCAddr</i> as a key.				

KamEngPutDel				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	1	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamEngPutDel takes a decoder address as a parameter. It deletes the locomotive object specified by <i>iDCCAddr</i> from the locomotive database.				

KamCmdCommand				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	1	In	Decoder address
¹ 1-127 for short addresses, 1-10239 for long addresses.				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCmdCommand takes the decoder address as a parameter. It sends all state changes from the server database to the specified locomotive or accessory decoder.				

3.5 Commands to control accessory decoders

This section describes the commands that control accessory decoders. These commands control things such as accessory decoder activation state. For efficiency, a copy of all the engine variables such speed is stored in the server. Commands such as **KamAccGetFunction** communicate only with the server, not the actual decoder. You should first make any changes to the server copy of the engine variables. You can send all changes to the engine using the **KamCmdCommand** command.

KamAccGetFunction				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
iFunctionID	int	0-7 ¹	In	Function ID number
lpFunction	int *	²	Out	Pointer to function value
¹ By convention, bit 0 is used to distinguish between a pair of outputs the accessory decoder is controlling. Bits 1 and 2 are used to indicate which of the 4 pairs of outputs is being selected. The maximum for this decoder is given by KamAccGetFunctionMax . ² Function active is boolean TRUE and inactive is boolean FALSE .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamAccGetFunction takes the decoder address, a function ID, and a pointer to the location to store the specified function state as parameters. It sets the memory pointed to by <i>lpFunction</i> to the specified function state.				

KamAccGetFunctionAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
lpFunctionArray	int *	¹	Out	Pointer to function state array
¹ Maximum array size is 8. Maximum for this decoder is given by KamAccGetFunctionMax + 1.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg). KamAccGetFunctionAll takes the decoder address and a pointer to a boolean function state array as parameters. It sets all function states in the <i>lpFunctionArray</i> to TRUE if the corresponding decoder function is enabled or FALSE otherwise.				

KamAccPutFunction				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
iFunctionID	int	0-7 ¹	In	Function ID number
iFunction	int	²	In	Function value
<p>¹ By convention, bit 0 is used to distinguish between a pair of outputs the accessory decoder is controlling. Bits 1 and 2 are used to indicate which of the 4 pairs of outputs is being selected. The maximum for this decoder is given by KamAccGetFunctionMax.</p> <p>² Function active is boolean TRUE and inactive is boolean FALSE.</p>				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
<p>¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).</p> <p>KamAccPutFunction takes the decoder address, a function ID, and a new function state as parameters. It sets the specified accessory database function state to <i>iFunction</i>. Note: This command only changes the accessory database. The data is not sent to the decoder until execution of the KamCmdCommand command.</p>				

KamAccPutFunctionAll				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
lpFunctionArray	int *	¹	In	Pointer to function state array
<p>¹ Maximum array size is 8. Maximum for this decoder is given by KamAccGetFunctionMax + 1.</p>				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
<p>¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).</p> <p>KamAccPutFunctionAll takes the decoder address and a pointer to a function state array as parameters. It sets all decoder function enable states to match the states in <i>lpFunctionArray</i>. The possible enable states are TRUE and FALSE. The data is not sent to the decoder until execution of the KamCmdCommand command.</p>				

KamAccGetFunctionMax				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
Return Value	Type	Range	Description	
iMaxFunction	int	0-7	Maximum function number	
<p>KamAccGetFunctionMax takes a decoder address as a parameter. It returns the maximum possible function number for the specified decoder.</p>				

KamAccGetName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
Return Value	Type	Range	Description	
bsAccNameString	BSTR	¹	Accessory name	
<p>¹ Exact return type depends on language. It is Cstring for C++.</p> <p>KamAccGetName takes a decoder address as a parameter. It returns a BSTR set to the name of the accessory.</p>				

KamAccPutName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
bsAccNameString	BSTR	¹	In	Accessory name
<p>¹ Exact parameter type depends on language. It is LPCSTR for C++.</p>				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
<p>¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).</p> <p>KamAccPutName takes a decoder address and a BSTR as parameters. It sets the symbolic accessory name to <i>bsAccName</i>.</p>				

KamAccGetFunctionName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
iFunctionID	int	0-7 ¹	In	Function ID number
¹ By convention, bit 0 is used to distinguish between a pair of outputs the accessory decoder is controlling. Bits 1 and 2 are used to indicate which of the 4 pairs of outputs is being selected. The maximum for this decoder is given by KamAccGetFunctionMax .				
Return Value	Type	Range	Description	
bsFcnNameString	BSTR	¹	Function name	
¹ Exact return type depends on language. It is CString for C++.				
KamAccGetFuncntionName takes a decoder address and a function ID as parameters. It returns a BSTR set to the symbolic name of the specified function.				

KamAccPutFunctionName				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
iFunctionID	int	0-7 ¹	In	Function ID number
bsFcnNameString	BSTR	²	In	Function name
¹ By convention, bit 0 is used to distinguish between a pair of outputs the accessory decoder is controlling. Bits 1 and 2 are used to indicate which of the 4 pairs of outputs is being selected. The maximum for this decoder is given by KamAccGetFunctionMax .				
² Exact parameter type depends on language. It is LPCSTR for C++.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamAccPutFunctionName takes a decoder address, function ID, and a BSTR as parameters. It sets the specified symbolic function name to <i>bsFcnNameString</i> .				

KamAccPutAdd				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamAccPutAdd takes a decoder address as a parameter. It creates a new accessory object in the accessory database using <i>iDCCAddr</i> as a key.				

KamAccPutDel				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	0-511	In	Decoder address
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamAccPutDel takes a decoder address as a parameter. It deletes the accessory object specified by <i>iDCCAddr</i> from the accessory database.				

3.6 Commands to control the command station

This section describes the commands that control the command station. These commands do things such as controlling command station power.

KamOprPutTurnOnStation				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutTurnOnStation takes a logical port ID as a parameter. It performs the steps necessary to turn on the command station.				

KamOprPutStartStation				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutStartStation takes a logical port ID as a parameter. It performs the steps necessary to start the command station.				

KamOprPutClearStation				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutClearStation takes a logical port ID as a parameter. It performs the steps necessary to clear the command station queue.				

KamOprPutStopStation				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutStopStation takes a logical port ID as a parameter. It performs the steps necessary to stop the command station.				

KamOprPutPowerOn				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutPowerOn takes a logical port ID as a parameter. It performs the steps necessary to apply power to the track.				

KamOprPutPowerOff				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutPowerOff takes a logical port ID as a parameter. It performs the steps necessary to remove power from the track.				

KamOprPutHardReset				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutHardReset takes a logical port ID as a parameter. It performs the steps necessary to perform a hard reset of the command station.				

KamOprPutEmergencyStop				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamOprPutEmergencyStop takes a logical port ID as a parameter. It performs the steps necessary to broadcast an emergency stop command to all decoders.				

KamOprGetStationStatus				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
bsStationStatus	BSTR	¹	Command station status	
¹ Exact return type depends on language. It is CString for C++.				
KamOprGetStationStatus takes a logical port ID as a parameter. It returns a BSTR containing the command station status. The exact format of the status BSTR is vendor dependent.				

3.7 Commands to configure the command station communication port

This section describes the commands that configure the command station communication port. These commands do things such as setting BAUD rate.

Several of the commands in this section use the numeric controller ID (*iControllerID*) to identify a specific type of command station controller. The following table shows the mapping between the controller ID (*iControllerID*) and controller name (*bsControllerName*) for a given type of command station controller.

iControllerID	bsControllerName	Description
0	UNKNOWN	Unknown controller type
1	SIMULAT	Interface simulator
2	LENZ_1x	Lenz version 1 serial support module
3	LENZ_2x	Lenz version 2 serial support module
4	DIGIT_DT200	Digitrax direct drive support using DT200
5	DIGIT_DCS100	Digitrax direct drive support using DCS100
6	MASTERSERIES	North coast engineering master series
7	SYSTEMONE	System one
8	RAMFIX	RAMFixx system
9	SERIAL	NMRA serial interface
10	EASYDCC	CVP Easy DCC
11	MRK6050	Marklin 6050 interface (AC and DC)
12	MRK6023	Marklin 6023 interface (AC)
13	DIGIT_PR1	Digitrax direct drive using PR1
14	DIRECT	Direct drive interface routine
15	ZTC	ZTC system ltd
16	TRIX	TRIX controller

Figure 1: Controller ID to controller name mapping

KamPortPutRestrans				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	10-255	In	Number of retransmits
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutRestrans takes a logical port ID and number of retransmits as parameters. It sets the number of retransmits attempted before failure.				

KamPortPutRate				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	²	In	BAUD rate
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² 0 - 300 BAUD, 1 - 1200 BAUD, 2 - 2400 BAUD, 3 - 4800 BAUD, 4 - 9600 BAUD, 5 - 14400 BAUD, 6 - 16400 BAUD, 7 - 19,200 BAUD .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutRate takes a logical port ID and BAUD rate as parameters. It sets the BAUD rate of the port.				

KamPortPutParity				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	0-4 ²	In	Parity type
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² 0 - NONE, 1 - Odd, 2 - EVEN, 3 - MARK, 4 - SPACE.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutParity takes a logical port ID and BAUD rate as parameters. It sets the parity type of the port.				

KamPortPutStop				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	0-2 ²	In	Number of stop bits
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² 0 - 1 bit, 1 - 1.5 bits, 2 - 2 bits.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutStop takes a logical port ID and number of stop bits as parameters. It sets the number of stop bits of the port.				

KamPortPutWatchdog				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	500-65535 ²	In	Watchdog timer value
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² Value in milliseconds. Recommended value is 2048 (2.048 seconds).				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutWatchdog takes a logical port ID and watchdog timer value as parameters. It sets the time the server will wait for a response from the command station before attempting a retransmit.				

KamPortPutFlow				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	0-3	In	Flow control type
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² 0 - NONE, 1 - XON/XOFF, 2 - RTS/CTS, 3 - BOTH XON/XOFF and RTS/CTS.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutFlow takes a logical port ID and flow control type as parameters. It sets the flow control type of the port.				

KamPortPutData				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iValue	int	0-1	In	Number of data bits
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² 0 - 7 bits, 1 - 8 bits.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutData takes a logical port ID and number of data bits as parameters. It sets the number data bits used by the port.				

KamPortGetName				
Parameter List	Type	Range	Direction	Description
iPhysicalPortID	int	1-65535 ¹	In	Physical port number
¹ Maximum value for this server given by KamPortGetMaxPhysical .				
Return Value	Type	Range	Description	
bsPortName	BSTR	¹	Physical port name	
¹ Exact return type depends on language. It is CString for C++.				
KamPortGetName takes a physical port ID number as a parameter. It returns a BSTR containing the physical port name such as "COMM1."				

KamPortPutDebug				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iMode	int	²	In	Mode
iKey	int	0-1 ³	In	Key
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
² <i>iMode</i> is a bit mask where setting a bit enables a function. Bit 1 sends messages to the debug file. Bit 2 sends messages to the screen. Bit 3 shows queue data. Bit 4 shows UI status. Bit 5 is reserved. Bit 6 shows semaphores and critical sections. Bit 7 shows miscellaneous messages. Bit 8 shows comm port activity.				
³ 0 - disable debug messages, 1 - enable debug messages.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutDebug takes a logical port ID, mode bit mask, and enable flag as parameters. It sends the debug messages specified by the <i>iMode</i> bit mask to the debug file, screen, or both. The debug file path is C:\Temp\Debug{PORT}.txt where {PORT} is the physical comm port ID.				

KamPortPutMapController				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
iControllerID	int	1-65535 ²	In	Command station type ID
iCommPortID	int	1-65535 ³	In	Physical comm port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts . ² See Figure 1: Controller ID to controller name mapping for values. Maximum value for this server is given by KamMiscMaxControllerID . ³ Maximum value for this server given by KamPortGetMaxPhysical .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamPortPutMapController takes a logical port ID, a command station type ID, and a physical communications port ID as parameters. It maps <i>iLogicalPortID</i> to <i>iCommPortID</i> for the type of command station specified by <i>iControllerID</i> .				

KamPortGetMaxLogPorts				
Parameter List	Type	Range	Direction	Description
NONE				
Return Value	Type	Range	Description	
iMaxLogicalPorts	int	1-65535	Maximum logical port ID	
KamPortGetMaxLogPorts takes no parameters. It returns the maximum logical port ID.				

KamPortGetMaxPhysical				
Parameter List	Type	Range	Direction	Description
NONE				
Return Value	Type	Range	Description	
iMaxPhysical	int	1-65535 ¹	Maximum physical port ID	
¹ The maximum value depends on the number of ports supported by the PC.				
KamPortGetMaxPhysical takes no parameters. It returns the maximum physical port ID.				

3.8 Commands that control command flow to the command station

This section describes the commands that control the command flow to the command station. These commands do things such as connecting and disconnecting from the command station.

KamCmdConnect				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCmdConnect takes a logical port ID as a parameter. It connects the server to the specified command station.				

KamCmdDisconnect				
Parameter List	Type	Range	Direction	Description
iLogicalPortID	int	1-65535 ¹	In	Logical port ID
¹ Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCmdDisconnect takes a logical port ID as a parameter. It disconnects the server to the specified command station.				

3.9 Commands that access the decoder information database

This section describes the commands that access the decoder information database. These commands do things such as getting the maximum address a given type of decoder supports.

KamDecoderGetMaxModels				
Parameter List	Type	Range	Direction	Description
NONE				
Return Value	Type	Range	Description	
iMaxModels	int	1-65535	Maximum decoder type ID	
KamDecoderGetMaxModels takes no parameters. It returns the maximum decoder type ID.				

KamDecoderGetModelName				
Parameter List	Type	Range	Direction	Description
iModel	int	1-65535 ¹	In	Decoder type ID
¹ Maximum value for this server given by KamDecoderGetMaxModels .				
Return Value	Type	Range	Description	
bsModelName	BSTR	¹	Decoder name	
¹ Exact return type depends on language. It is CString for C++.				
KamPortGetModelName takes a decoder type ID as a parameter. It returns a BSTR containing the decoder name.				

KamDecoderSetModelToAddr				
Parameter List	Type	Range	Direction	Description
iModel	int	¹	In	Decoder model ID
iDCCAddr	int	²	In	Decoder address
¹ Maximum value for this server given by KamDecoderGetMaxModels . ² 1-10239 for locomotive decoders. 0-511 for accessory decoders.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamDecoderSetModelToAddr takes a decoder ID and decoder address as parameters. It sets the decoder model type of the decoder at address <i>iDCCAddr</i> to the type specified by <i>iModel</i> .				

KamDecoderGetMaxAddress				
Parameter List	Type	Range	Direction	Description
iModel	int	¹	In	Decoder type ID
¹ Possible values are 127 or 10239 for locomotive decoders, 511 for accessory decoders.				
Return Value	Type	Range	Description	
iMaxAddress	int	Model dependent	Maximum decoder address	
KamDecoderGetMaxAddress takes a decoder type ID as a parameter. It returns the maximum address supported by the specified decoder.				

KamDecoderChangeOldNewAddr				
Parameter List	Type	Range	Direction	Description
iOldAddr	int	¹	In	Old address
iNewAddr	int	¹	In	New address
¹ 1-10239 for locomotive decoders. 0-511 for accessory decoders.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamDecoderChangeOldNewAddr takes an old address and a new address as parameters. It moves the specified locomotive or accessory decoder from <i>iOldAddr</i> to <i>iNewAddr</i> .				

KamDecoderMovePort				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iLogicalPortID	int	1-65535 ²	In	Logical port ID
¹ 1-10239 for locomotive decoders. 0-511 for accessory decoders. ² Maximum value for this server given by KamPortGetMaxLogPorts .				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamDecoderMovePort takes a decoder address and logical port ID as parameters. It moves the decoder specified by <i>iDCCAddr</i> to the controller specified by <i>iLogicalPortID</i> .				

3.10 Cab Control Commands

This section describes commands that control the cabs attached to a command station.

KamCabGetMessage				
Parameter List	Type	Range	Direction	Description
iCabAddress	int	1-65535 ¹	In	Cab address
¹ Maximum value is command station dependent.				
Return Value	Type	Range	Description	
bsMsg	BSTR	¹	Cab message string	
¹ Exact return type depends on language. It is CString for C++.				
KamCabGetMessage takes a cab address as a parameter. It returns a BSTR set to present cab message.				

KamCabPutMessage				
Parameter List	Type	Range	Direction	Description
iCabAddress	int	¹	In	Cab address
bsMsg	BSTR	²	Out	Cab message string
¹ Maximum value is command station dependent.				
² Exact parameter type depends on language. It is LPCSTR for C++.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCabPutMessage takes a cab address and a BSTR as parameters. It sets the cab message to <i>bsMsg</i> .				

KamCabGetCabAddr				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCabAddress	int	1-65535 ²	In	Cab address
¹ 1-10239 for locomotive decoders. 0-511 for accessory decoders. ² Maximum value is command station dependent.				
Return Value	Type	Range	Description	
iCabAddress	int	1-65535 ¹	Cab address	
¹ Maximum value is command station dependent.				
KamCabGetCabAddr takes a decoder address as a parameter. It returns the address of the cab attached to the specified decoder.				

KamCabPutAddrToCab				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCabAddress	int	1-65535 ²	In	Cab address
¹ 1-10239 for locomotive decoders. 0-511 for accessory decoders. ² Maximum value is command station dependent.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCabPutAddrToCab takes a decoder address and cab address as parameters. It attaches the decoder specified by <i>iDCCAddr</i> to the cab specified by <i>iCabAddress</i> .				

KamCabREmoveAddr				
Parameter List	Type	Range	Direction	Description
iDCCAddr	int	¹	In	Decoder address
iCabAddress	int	1-65535 ²	In	Cab address
¹ 1-10239 for locomotive decoders. 0-511 for accessory decoders. ² Maximum value is command station dependent.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamCabRemoveAddr takes a decoder address and cab address as parameters. It removes the decoder specified by <i>iDCCAddr</i> to the cab specified by <i>iCabAddress</i> .				

3.11 Miscellaneous Commands

This section describes miscellaneous commands that do not fit into the other categories.

KamMiscGetErrorMsg				
Parameter List	Type	Range	Direction	Description
iError	int	0-65535 ¹	In	Error flag
¹ <i>iError</i> = 0 for success. Nonzero indicates an error.				
Return Value	Type	Range	Description	
bsErrorString	BSTR	¹	Error string	
¹ Exact return type depends on language. It is Cstring for C++.				
KamMiscGetErrorMsg takes an error flag as a parameter. It returns a BSTR containing the descriptive error message associated with the specified error flag.				

KamMiscGetClockTime				
Parameter List	Type	Range	Direction	Description
lpHours	int *	0-23	Out	Hours
lpMinutes	int *	0-59	Out	Minutes
lpRatio	int *	¹	Out	Fast clock ratio
¹ Real time clock ratio.				
Return Value	Type	Range	Description	
iError	short	¹	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamMiscGetClockTime takes pointers to locations to store the hours, minutes, and fast clock ratio as parameters. It sets the memory pointed to by <i>lpHours</i> to the fast clock hours, sets the memory pointed to by <i>lpMinutes</i> to the fast clock minutes, and the memory pointed to by <i>lpRatio</i> to the fast clock ratio.				

KamMiscPutClockTime				
Parameter List	Type	Range	Direction	Description
iHours	int	0-23	In	Hours
iMinutes	int	0-59	In	Minutes
iRatio	int	1	In	Fast clock ratio
¹ Real time clock ratio.				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamMiscPutClockTime takes the fast clock hours, the fast clock minutes, and the fast clock ratio as parameters. It sets the fast clock using specified parameters.				

KamMiscGetInterfaceVersion				
Parameter List	Type	Range	Direction	Description
NONE				
Return Value	Type	Range	Description	
bsInterfaceVersion	BSTR	1	Interface version string	
¹ Exact return type depends on language. It is Cstring for C++.				
KamMiscGetInterfaceVersion takes no parameters. It returns a BSTR containing the interface version string.				

KamMiscSaveData				
Parameter List	Type	Range	Direction	Description
NONE				
Return Value	Type	Range	Description	
iError	short	1	Error flag	
¹ <i>iError</i> = 0 for success. Nonzero is an error number (see KamMiscGetErrorMsg).				
KamMiscSaveData takes no parameters. It saves all server data to permanent storage. This command is run automatically whenever the server stops running.				

KamMiscGetControllerName				
Parameter List	Type	Range	Direction	Description
iControllerID	int	1-65535 ¹	In	Command station type ID
¹ See Figure 1: Controller ID to controller name mapping for values. Maximum value for this server is given by KamMiscMaxControllerID .				
Return Value	Type	Range	Description	
bsControllerName	BSTR	1	Command station type name	
¹ Exact return type depends on language. It is CString for C++.				
KamMiscGetControllerName takes a command station type ID as a parameter. It returns a BSTR containing the command station type name.				

KamMiscMaxControllerID				
Parameter List	Type	Range	Direction	Description
NONE				
Return Value	Type	Range	Description	
iMaxControllerID	int	1-65535 ¹	Maximum decoder type ID	
¹ See Figure 1: Controller ID to controller name mapping for a list of controller ID values.				
KamMiscMaxControllerID takes no parameters. It returns the maximum controller type ID.				