



# ***Interoperability Development Kit Reference Guide***

---

**For Velocity 3.1**

**IDK001-02010**

IDK001-02010 February, 2010

Revision D

Copyright© 2010 Hirsch Electronics Corporation. All rights reserved. ScramblePad® and ScrambleProx® are registered trademarks of Hirsch Electronics Corporation. DIGI\*TRAC™, MATCH™, ScrambleCard™, SCRAMBLE\*NET™ (abbreviated S\*NET), X\*NET, and Velocity™ are all trademarks of Hirsch Electronics Corporation.

Microsoft, MS and MS-DOS are registered trademarks, and Windows, Windows 2000, and Windows XP are trademarks of Microsoft Corporation.

Hirsch Electronics Corporation  
1900-B Carnegie Avenue  
Santa Ana, CA 92705-5520

Phone:949-250-8888 or 888-809-8880 (toll-free)  
Fax:949-250-7372  
Web:[www.HirschElectronics.com](http://www.HirschElectronics.com)



---

## Getting Help

If you encounter a problem that is not discussed in this guide and you need technical support, do the following:

1. Contact your local dealer or the provider of this product.
2. If your dealer is not available, contact Hirsch Technical Support directly. This can be done in a number of ways:

---

Mail: Hirsch Electronics Corporation  
1900-B Carnegie Avenue  
Santa Ana, CA 92705-5520

Attn: Professional Services

---

Phone: 877-HIRSCHX (877-447-7249) toll-free

---

Fax: (949) 250-7362

---

Email: support@HirschElectronics.com

---

WWW: www.HirschElectronics.com

---

Whenever you call your local dealer or Hirsch, be sure to have your registration material, serial number and software version number available.

For future reference, record these numbers here.

Serial Number: \_\_\_\_\_

Version Number: \_\_\_\_\_

Dealer: \_\_\_\_\_

Dealer Phone #: \_\_\_\_\_

CCM Rev/Version #: \_\_\_\_\_



# Contents

Getting Help .....	iii
<b>Introduction.....</b>	<b>1</b>
<b>Installing IDK.....</b>	<b>5</b>
<b>HirschVMO API .....</b>	<b>7</b>
VMO Class Properties .....	9
Credential .....	13
Parameter .....	13
Properties .....	13
Methods .....	15
Events .....	15
Remarks .....	16
CredentialCollection .....	17
Properties .....	17
Methods .....	17
Events .....	17
Remarks .....	17
CredentialTemplate .....	19
Properties .....	19
Methods .....	19
Events .....	19
Remarks .....	19
CredentialTemplateCollection .....	20
Properties .....	20
Methods .....	20
Events .....	20
Remarks .....	20
ImagePlaceholder .....	21
Properties .....	21

Methods .....	21
Events .....	21
ImagePlaceholderCollection .....	22
Properties .....	22
Methods .....	22
Events .....	22
Remarks .....	22
Person .....	24
Properties .....	24
Methods .....	25
Events .....	25
Remarks .....	26
PersonCollection .....	27
Properties .....	27
Methods .....	28
Events .....	28
Remarks .....	28
PersonImage .....	31
Properties .....	31
Methods .....	31
Events .....	32
Remarks .....	32
PersonImageCollection .....	33
Properties .....	33
Methods .....	33
Events .....	33
Remarks .....	33
Selection .....	35
Properties .....	35
SelectionCollection .....	36
Properties .....	36
Methods .....	36
Events .....	36

---

UserDefinedField .....	37
Properties .....	37
Methods .....	37
Events .....	37
Remarks .....	38
UserDefinedFieldCollection .....	39
Properties .....	39
Methods .....	39
Events .....	39
Remarks .....	39
UserDefinedFieldSetup .....	40
Properties .....	40
Methods .....	41
Events .....	41
Remarks .....	41
UserDefinedFieldSetupCollection .....	42
Properties .....	42
Methods .....	42
Events .....	42
Remarks .....	42
VelocityServer .....	43
Properties .....	43
Methods .....	44
Events .....	45
Remarks .....	51
VB Calls Sample .....	52
<b>XML Interoperability Services.....</b>	<b>57</b>
XML Overview.....	57
Remote Procedure Calls .....	59
Acknowledging and Clearing Alarms.....	59
Acknowledge .....	60
Clear .....	60
SQL Statement Execution (Insert, Update, Delete, Query, etc.) .....	60

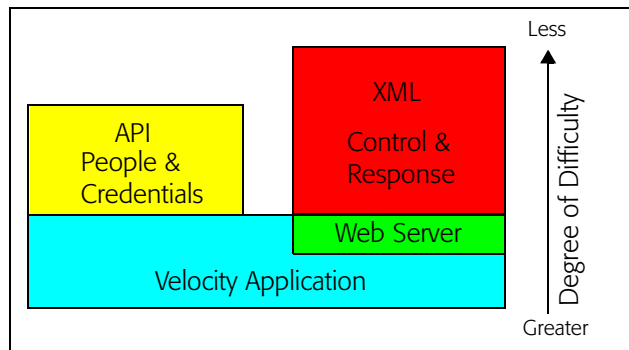
---

Channel Responses .....	61
Error Handling .....	61
Discovery .....	62
Default Supported Lists .....	63
Receiving Alarms & Events .....	64
Alarm & Event Messaging.....	64
Non-Alarm Event .....	64
Alarm.....	65
Supported Remote Procedure Calls.....	66
Communicating with Velocity Web Server .....	67
Executing Remote Procedure Calls.....	67

# Introduction

Hirsch Electronics has undertaken a new multi-faceted initiative to support interoperability. This initiative is based on the latest version of Hirsch's Velocity Security Management System application software.

It is easiest to understand how Velocity 'exposes' its application software by thinking of two programs running on the Velocity application. First, there is an API for People and Credentials. This is the information generally stored in the Velocity server. Second, there is a web server and an XML interface for Control and Response functions. Both the API and XML enable interoperability to extend through the application software to the connected controllers for distributed execution.



A company might use the People and Credentials API to interface Human Resources operations with the physical security application. This allows a new employee's personal and departmental information to be transferred from HR to Security in order to automatically assign the new employee credential. The credential is immediately authorized for specific doors and is printed as a photo badge in the HR department to hand to the new hire with their other orientation materials. Conversely, terminations are sent from HR to Security to immediately disable the employee's access privileges throughout company facilities.

All this is possible because Velocity has an API available to third-party applications that efficiently interacts with person and credential records. These records can be added, modified, and deleted through the third-party software. For example, applications such as visitor management can force a download of visitor credentials to the Velocity controller network for accountability records that are far superior to illegible sign-in logbooks commonly found at reception desks.

Technically, the Velocity API is a DLL (Dynamic Link Library), *HirschVMO.DLL*, based on a programmable COM (Component Object Model) interface. Using *HirschVMO.DLL*, third parties can programmatically connect to the Velocity SQL database and interact with the Person and Credential records without touching the data directly.

The Velocity API and XML have a full SDK to support developers of interoperable interfaces. It is far easier to develop an interface to Velocity using an API than to obtain and learn the underlying application code and database structure.

XML provides a different strategy for simplifying interface development. In this approach, commands required to trigger certain Velocity functions from a third-party application can be sent to Velocity via XML. For example, severe weather alarms could unlock doors. Velocity might also send triggers to elevator equipment to activate floor buttons for an individual presenting a card or activating a unique keypad code. Transactional information can notify a nurse's station concerning which doctors are currently in the building. Alarms and arming status can be posted to a custom display in the command center of a central utilities district.

Velocity includes a web server that enables it to receive authenticated commands from an outside application and broadcast selected information to specific outside sources. An XML-RPC (Remote Procedure Call) is used to communicate with the Velocity web server. Velocity uses this web server to communicate over the internet using HTTP.

The Velocity web server recognizes the HTTP POST and GET functions. A third-party application can issue an HTTP POST to the Velocity web server that consists of an XML document containing the correct information to request a List or issue a Command. The Velocity web server then responds to a properly formatted and authenticated POST by allowing the third-party application to GET an XML document that provides the requested information. Each POST must be properly authenticated by including a name, password, security domain, and workstation ID. In effect, the Velocity web server considers the third-party application to be a client and enforces the same NT authentication and Active Directory policy as is applied to a Velocity operator on a client terminal. Velocity manages Port 80, processing POST and GET messages in XML format. By convention, Port 80 allows communication across firewalls for web browsers.

Velocity also supports a form of 'discovery' by enabling a POST to obtain a list of related items or even a report which is a list of lists. A list provides the command name of an item as well as its unique identifier assigned by the SQL Server database. This information can be used to issue a specific command to Velocity—such as 'Unlock the Lobby Door'—or to develop a transform between two applications.

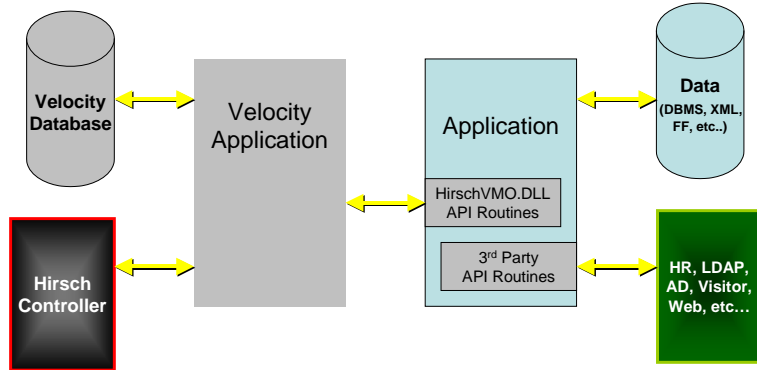
As an add-on server extension, the Velocity web server can also broadcast serial data over a user-defined port via its XML Writer. The XML Writer enables the customer to choose what specific types of alarms, events, and transactions are sent to which port at what time. In this way one or more target applications can receive only the information they need, providing a sort of subscription service. The target application typically parses or transforms the data from the Velocity XML Writer so that is usable locally.

The new Hirsch Velocity IDK consists of two modules:

- Hirsch Velocity Management Object (HirschVMO.DLL) API routines
- Velocity XML Interoperability Services

These two modules enable the qualified programmer to customize their application for interaction with Velocity.

As shown in the following diagram, Velocity can now exchange specific types of data using HirschVMO.DLL API routines and export data in XML form.



Through HirschVMO.DLL, a programmer can access data on persons and credentials within the Velocity database, reformat that data as required, even modify the data and return it to the Velocity database. Third-party applications, such as human resource or accounting spreadsheets, can then share security information for seamless exchanges of data. In this way, for example, an employee who HR terminates, can have all security privileges simultaneously suspended if the programmer customizes the HR software to interoperate with Velocity through HirschVMO.DLL.

For information on installing IDK software, see "Installing IDK," starting on page 5.

For more on HirschVMO.DLL, see "HirschVMO API" starting on page 7.

Velocity's new XML interoperability services enable a qualified programmer to receive XML data from Velocity through Velocity's XML Writer. This data can then be reinterpreted as required for a third-party application. Since XML Writer outputs raw XML files, the programmer must be familiar with the manipulation of XML: essentially, how to cross-match tags using XML schemas.

For more on this, see "XML Interoperability Services" starting on page 57.