

## AUDIN

Composants & systèmes d'automatisme  
7 bis rue de Tinquex - 51100 Reims - France  
Tel. +33(0)326042021 • Fax +33(0)326042820  
<http://www.audin.fr> • e-mail [info@audin.fr](mailto:info@audin.fr)

---

# OLE for Formula

Reference Manual





**OLE for Formula**  
**REFERENCE MANUAL**



DATALOGIC S.p.A.  
Via Candini 2  
40012 - Lippo di Calderara di Reno  
Bologna - Italy

OLE for Formula - Reference Manual

Vers.: 1.0.5.0 and later  
Ed.: 04/01  
Code: 820042330

**ALL RIGHTS RESERVED**

Datalogic reserves the right to make modifications and improvements without prior notification. Product names mentioned herein are for identification purposes only and may be trademarks and or registered trademarks of their respective companies.

© - 1997-2001 Datalogic S.p.A.

## DATALOGIC S.p.A. Software License Agreement

This legal document is an agreement between you, the end user and DATALOGIC S.p.A. BY INSTALLING THE SOFTWARE, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, which includes the SOFTWARE LICENSE, LIMITED WARRANTY and ACKNOWLEDGMENT.

**GRANT OF LICENSE.** DATALOGIC grants to you the right to use one copy of the enclosed DATALOGIC S.p.A., program (the "SOFTWARE") on a single terminal connected to a single computer (i.e.; with a single CPU). You may not network the SOFTWARE or otherwise use it on more than one computer or computer terminal at the same time.

**COPYRIGHT.** The SOFTWARE is owned by DATALOGIC S.p.A. or its suppliers and is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (i.e.; a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the software.

**NON PERMITTED USES:** Without the express permission of DATALOGIC S.p.A., you may not:

1. Use the software in a computer service business including rental, networking or time sharing software, nor may you use it for multiple users, or multiple computer system applications in the absence of individual network licenses with DATALOGIC S.p.A.
2. Use, Copy or modify, alter or transfer, electronically or otherwise, the software or documentation except as expressly allowed in this agreement.
3. Translate, reverse engineer, de-assemble, de-compile or create derivative works based on the written materials.
4. Sub-license or lease this program or its documentation.

**LIMITED WARRANTY:** This software and accompanying written materials are provided "as is" without warranty of any kind. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THIS SOFTWARE IS ASSUMED BY YOU. SHOULD THE SOFTWARE PROVE DEFECTIVE IN USE, YOU (AND NOT DATALOGIC OR ITS AGENTS) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION OF DATA. FURTHER, DATALOGIC DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF, OR THE RESULTS OF THE USE OF THE SOFTWARE, IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE AND YOU RELY ON THE SOFTWARE AND RESULTS AT YOUR OWN RISK.

DATALOGIC warrants to the original licensee that the CD-Rom on which the SOFTWARE is recorded is free from defects in materials and workmanship under normal use and service for a period of ninety (90) days from the date of delivery as evidenced by a copy of your receipt. DATALOGIC's entire liability and your exclusive remedy shall be at DATALOGIC's option, replacement of the CD-Rom which shall be returned to DATALOGIC with a copy of your receipt. If failure of the CD-Rom has resulted from accident, abuse or misapplication of the product, then DATALOGIC shall have no responsibility to replace the product under this warranty.

THE ABOVE IS THE ONLY WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, THAT IS MADE BY DATALOGIC ON THIS DATALOGIC PRODUCT. IN NO EVENT SHALL DATALOGIC OR ITS SUPPLIERS, NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION OR DELIVERY OF THIS PRODUCT, BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE, THE RESULTS OF USE, OR INABILITY TO USE SUCH PRODUCT, EVEN IF DATALOGIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**ACKNOWLEDGMENT:** You acknowledge that you have read this LICENSE and Limited Warranty, understand them and agree to be bound by their terms and conditions. You also agree that the LICENSE and Limited Warranty are the complete and exclusive statement between the parties and supersedes all other communications between the parties relating to the subject matter of the LICENSE or the Limited Warranty.

# CONTENTS

---

<b>1</b>	<b>GENERAL INFORMATION</b> .....	<b>1</b>
1.1	Scope of this Manual .....	1
1.2	Definition of OLE Custom Control .....	1
1.3	Description of OLE for Formula .....	2
1.4	Files Necessary to Use OLE for Formula .....	2
1.5	Examples .....	2
1.6	System Parameters .....	3
<b>2</b>	<b>USE AND FUNCTIONING</b> .....	<b>4</b>
2.1	Using OLE for Formula .....	4
2.2	Conventions Used in the Manual .....	5
2.3	Formula Constants .....	8
	Formula BarcodeCheckDigit Constants .....	8
	Formula BarcodeType Constants .....	9
	Formula BaudRate Constants .....	10
	Formula ByteSize Constants .....	10
	Formula CommPort Constants .....	11
	Formula Error Constant .....	11
	Formula ErrorExtended Constants .....	12
	Formula Parity Constants .....	12
	Formula RunMode Constants .....	13
	Formula Setup Constants .....	13
	Formula State Constants .....	13
2.4	Formula Properties .....	14
	Property AckTimeout (Formula object) .....	14
	Property BarcodeCheckDigit (Term object) .....	15
	Property BarcodeSize (Term object) .....	16
	Property Barcode Type (Term object) .....	17
	Property BaudRate (Formula object) .....	19
	Property ByteSize (Formula object) .....	20
	Property CheckHealthText (Term object, Sat object) .....	21
	Property Claimed (Formula object) .....	21
	Property CommPort (Formula object) .....	22
	Property ControlObjectDescription (Formula object) .....	23
	Property ControlObjectVersion (Formula object) .....	24
	Property DataEventEnabled (Formula object) .....	25
	Property Delay1, Delay2, Delay3, Delay4 (Sat object) .....	26
	Property DeviceData (Formula object) .....	27
	Property DeviceDescription (Term object, Sat object) .....	28
	Property DeviceEnabled (Term object, Sat object) .....	29
	Property DeviceName (Term object, Sat object) .....	30
	Property DeviceVersion (Term object, Sat object) .....	31

	Property DirectIORead (Term object, Sat object).....	32
	Property DirectIOText (Term object, Sat object).....	33
	Property FieldSeparator (Term object, Sat object).....	34
	Property FileName (Term object, Sat object).....	35
	Property FreezeEvents (Formula object).....	35
	Property Group (Sat object).....	36
	Property ID (Term object, Sat object).....	37
	Property OutputID (Formula object).....	37
	Property Parity (Formula object).....	38
	Property ResultCode (Formula object).....	39
	Property ResultCodeExtended (Formula object).....	41
	Property RunFromTerminal (Sat object).....	42
	Property RunMode (Sat object).....	43
	Property RunToTerminal (Sat object).....	44
	Property Satellites Collection, Sat object.....	45
	Property ScanData (Formula object).....	46
	Property ServiceObjectDescription (Formula object).....	47
	Property ServiceObjectVersion (Formula object).....	48
	Property Setup (Term object, Sat object).....	49
	Property Sleep (Formula object).....	50
	Property State (Formula object).....	51
	Property State (Term object, Sat object).....	52
	Property Terminals Collection, Term object.....	53
	Property TxSleep (Term object).....	54
2.5	Formula Events.....	55
	Event DataEvent (Formula object).....	55
	Event DirectIOEvent (Formula object).....	56
	Event ErrorEvent (Formula object).....	56
	Event OutputCompleteEvent (Formula object).....	57
	Event StatusUpdateEvent (Formula object).....	58
2.6	Formula Methods.....	59
	Method Action (Formula object).....	59
	Method Add (Terminals collection, Satellites collection).....	60
	Method CheckHealth (Term object, Sat object).....	61
	Method Claim (Formula object).....	61
	Method ClearInput (Formula object).....	62
	Method ClearOutput (Term object, Sat object).....	62
	Method Close (Formula object).....	63
	Method DataWrite (Term object, Sat object).....	64
	Method DirectIO (Term object, Sat object).....	65
	Method FileWrite (Term object, Sat object).....	66
	Method FindDevice (Formula object).....	67
	Method HexWrite (Formula object).....	69
	Method Initiate (Formula object).....	70
	Method Load (Formula object).....	71
	Method Open (Formula object).....	72

	Method Release (Formula object).....	72
	Method Save (Formula object).....	73
<b>A</b>	<b>ACTION METHOD</b> .....	<b>74</b>
A.1	Properties Window.....	74
A.1.1	General Properties.....	75
A.1.2	Terminals Properties.....	77
A.1.3	Satellites Properties.....	81
<b>B</b>	<b>PROGRAMMING EXAMPLES</b> .....	<b>87</b>
B.1	Loading the Examples.....	87
<b>C</b>	<b>COMMUNICATION PROTOCOL</b> .....	<b>93</b>
C.1	Communication Between Formula Terminals and Host Computer.....	93
C.2	Message.....	95
C.2.1	Message "Commands".....	95
C.2.2	Message "Data Received".....	97
C.2.3	Message "Replies".....	97
C.4	Compression and Decompression of Data.....	107



# Formula<sup>®</sup>

## 1 GENERAL INFORMATION

---

### 1.1 SCOPE OF THIS MANUAL

This manual was written by Datalogic S.p.A. and accompanies the EasyGen™ Application Generator Package.

It supplies alphabetically ordered specifications for the connection and management of Formula terminals and satellites of the OLE for Formula "object".

The manual is aimed at the operator who is familiar both with "object programming" and the Windows 95/98/2000/ME and Windows NT environments.

### 1.2 DEFINITION OF OLE CUSTOM CONTROL

OLE Custom Control is an "object" intended primarily as an extension either of the Visual Basic Toolbox or of an area providing the same function in other environments.

An OLE Custom Control supplies new functionality to a program when it is inserted because it becomes an integral part of the program. For example in Visual Basic there are OLE Custom Controls for graphic displays, database and multimedia management etc.

OLE Custom Control is a special form of Dynamic-Link Library containing one or more checks.

### 1.3 DESCRIPTION OF OLE FOR FORMULA

OLE for Formula is an OLE Custom Control whose properties, events and methods are supported by environments such as Visual Basic, Excel, Access and others that operate with Window 95/98/2000/ME and Windows NT.

OLE for Formula defines and manages the transmission protocol between PC and Formula terminal, thus relieving the user of the task.

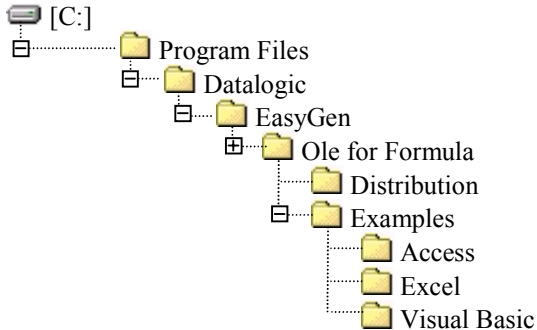
### 1.4 FILES NECESSARY TO USE OLE FOR FORMULA

2 files are required to use OLE for Formula:

- FORMULA.OCX
- FORMULA.DLL

### 1.5 EXAMPLES

The examples are installed in the directory:



OLE is supported by environments such as Visual Basic, Excel and Access. You must integrate OLE for Formula into the chosen environment as described in the appropriate manual.

## ❑ OLE for Formula icons

When an OLE Custom Control is added, a new icon normally appears in the Toolbox of the environment being used.

## ❑ OLE for Formula class

Every OLE Custom Control has its own classification. OLE for Formula belongs to the "Formula" class.

## 1.6 SYSTEM PARAMETERS

It is possible to define a session of the file WIN.INI with the name "OLE for Formula" and to insert the following system parameters:

MaxRetry	This parameter indicates the number of times that the host computer will try to send a command if the Ack. timeout expires (default = 2).
SatOffRetry	This parameter indicates the number of times that the host computer will try to send the command ESC 4\$ if the Ack. timeout expires. Each attempt is followed by a delay of 100ms (default = 10).
SatOffFlag	This parameter indicates whether or not to respond to an error resulting from disconnection of the satellite from the network (for example, if the power supply is interrupted). The error does not affect operation of other satellites remaining connected to the network (default = 1). SatOffFlag = 1 respond to error; SatOffFlag = 0 ignore error.

The session is not created automatically during installation but must be entered manually by the user.

# Formula<sup>®</sup>

## 2 USE AND FUNCTIONING

---

**ATTENTION**

*The user is responsible for all damage occurring due to misuse of the equipment and non-observance of the instructions laid down in the manual.*

*All necessary precautions must be observed during the data transmission phase to the host computer to avoid any interruptions, which could cause a partial or total loss of data.*

### 2.1 USING OLE FOR FORMULA

For every OLE Custom Control you can define properties, events and methods and therefore also for OLE for Formula:

- the properties of an OLE Custom Control are values that define its status. They are visible and can be changed in the “Properties Window” during the development of an application. Some properties are only accessible while the program is running. It is usually possible to configure them although not always possible to read them;
- the events are indications transmitted by the OLE Custom Controls to the application containing them;
- the methods are the operations that the object can carry out.

## 2.2 CONVENTIONS USED IN THE MANUAL

**Bold text** indicates a term or key word that should be used exactly as written. For example names of properties, methods and events should be written exactly as shown.

[ ] Indicates that the data in square brackets is optional.

{ | } Separates the parameters with a choice of either/or.

CAPITALS The words in capitals indicate names of files, routes, names of structures (also in bold) and constants.

Four tables listing all the constants, properties, events and methods of OLE for Formula are shown below:

<b>FORMULA CONTROL CONSTANTS</b>
Formula <b>BarcodeCheckDigit</b> Constants
Formula <b>BarcodeType</b> Constants
Formula <b>BaudRate</b> Constants
Formula <b>ByteSize</b> Constants
Formula <b>CommPort</b> Constants
Formula <b>Error</b> Constants
Formula <b>ErrorExtended</b> Constants
Formula <b>Parity</b> Constants
Formula <b>RunMode</b> Constants
Formula <b>Setup</b> Constants
Formula <b>State</b> Constants

<b>OLE FOR FORMULA PROPERTIES</b>
Property <b>AckTimeout</b>
Property <b>BarcodeCheckDigit</b> (Term object)
Property <b>BarcodeSize</b> (Term object)
Property <b>BarcodeType</b> (Term object)
Property <b>BaudRate</b>
Property <b>ByteSize</b>
Property <b>CheckHealthText</b> (Term object, Sat object)
Property <b>Claimed</b>
Property <b>CommPort</b>
Property <b>ControlObjectDescription</b>
Property <b>ControlObjectVersion</b>
Property <b>DataEventEnabled</b>
Property <b>Delay1, Delay2, Delay3, Delay4</b> (Sat object)

<b>OLE FOR FORMULA PROPERTIES</b>
Property <b>DeviceData</b>
Property <b>DeviceDescription</b> (Term object, Sat object)
Property <b>DeviceEnabled</b> (Term object, Sat object)
Property <b>DeviceName</b> (Term object, Sat object)
Property <b>DeviceVersion</b> (Term object, Sat object)
Property <b>DirectIOText</b> (Term object, Sat object)
Property <b>FieldSeparator</b> (Term object, Sat object)
Property <b>FileName</b> (Term object, Sat object)
Property <b>FreezeEvents</b>
Property <b>Group</b> (Sat object)
Property <b>ID</b> (Term object, Sat object)
Property <b>OutputID</b>
Property <b>Parity</b>
Property <b>ResultCode</b>
Property <b>ResultCodeExtended</b>
Property <b>RunFromTerminal</b> (Sat object)
Property <b>RunMode</b> (Sat object)
Property <b>RunToTerminal</b> (Sat object)
Property <b>Satellites</b> Collection, <b>Sat</b> object
Property <b>ScanData</b>
Property <b>ServiceObjectDescription</b>
Property <b>ServiceObjectVersion</b>
Property <b>Setup</b> (Term object, Sat object)
Property <b>Sleep</b>
Property <b>State</b>
Property <b>State</b> (Term object, Sat object)
Property <b>Terminals</b> Collection, <b>Term</b> object
Property <b>TxSleep</b> (Term object)

<b>OLE FOR FORMULA EVENTS</b>
Event <b>DataEvent</b>
Event <b>DirectIOEvent</b>
Event <b>ErrorEvent</b>
Event <b>OutputCompleteEvent</b>
Event <b>StatusUpdateEvent</b>

<b>OLE FOR FORMULA METHODS</b>
Method <b>Action</b>
Method <b>Add</b> (Terminals collection, Satellites collection)
Method <b>CheckHealth</b> (Term object, Sat object)
Method <b>Claim</b>
Method <b>ClearInput</b>
Method <b>ClearOutput</b> (Term object, Sat object)
Method <b>Close</b>
Method <b>DataWrite</b> (Term object, Sat object)
Method <b>DirectIO</b> (Term object, Sat object)
Method <b>FileWrite</b> (Term object, Sat object)
Method <b>FindDevice</b>
Method <b>HexWrite</b>
Method <b>Initiate</b> (Term object, Sat object)
Method <b>Load</b>
Method <b>Open</b>
Method <b>Release</b>
Method <b>Save</b>

The OLE for Formula characteristics are described in the following pages.

## 2.3 FORMULA CONSTANTS

### *FORMULA BARCODECHECKDIGIT CONSTANTS*

#### *AVAILABLE VALUE*

---

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaCheckDigitStandard39	1	Enables a character check on Standard codes 3/9
FormulaCheckDigitInterleaved25	2	Enables a character check on Interleaved codes 2/5
FormulaCheckDigitMatrix25	4	Enables a character check on Matrix codes 2/5
FormulaCheckDigitDeltaAIBM	8	Enables a character check on Delta A IBM codes
FormulaCheckDigitMSI	16	Enables a character check on MSI codes



*FORMULA BARCODETYPE CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaBarcodeStandard39	1	Enables reading of Standard codes 3/9
FormulaBarcodeInterleaved25	2	Enables reading of Interleaved codes 2/5
FormulaBarcodeUPCEAN	4	Enables reading of UPC and EAN codes
FormulaBarcodeMonarch27	8	Enables reading of Monarch codes 2/7
FormulaBarcodeUPCOnly	16	Enables reading of UPC codes only
FormulaBarcodeUPCEANPlus2	32	Enables reading of extended UPC and EAN codes +2
FormulaBarcodeUPCEANPlus5	64	Enables reading of extended UPC and EAN codes +5
FormulaBarcodeExtended39	128	Enables reading of Extended 39 codes
FormulaBarcodePharmaceutical	256	Enables reading of pharmaceutical codes
FormulaBarcodeUPCEOnly	512	Enables reading of UPC E codes only
FormulaBarcodeEAN8Only	1024	Enables reading of EAN 8 codes only
FormulaBarcodeUPCAEAN13Only	2048	Enables reading of UPC E and EAN 13 codes only
FormulaBarcodePako	4096	Enables reading of Pako codes
FormulaBarcodeCode128	8192	Enables reading of 128 codes
FormulaBarcodeIndustrial 25	16384	Enables reading of industrial codes 2/5
FormulaBarcodeMatrix25	32768	Enables reading of Matrix codes 2/5
FormulaBarcodeDeltaAIBM	65536	Enables reading of Delta A IBM codes
FormulaBarcodeMSI	131072	Enables reading of MSI codes
FormulaBarcodeCode93	262144	Enables reading of 93 codes
FormulaBarcodeITF14	524288	Enables reading of ITF codes 1/4

*FORMULA BAUDRATE CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaBaud110	0	Transmission speed 110 baud
FormulaBaud300	1	Transmission speed 300 baud
FormulaBaud600	2	Transmission speed 600 baud
FormulaBaud1200	3	Transmission speed 1200 baud
FormulaBaud2400	4	Transmission speed 2400 baud
FormulaBaud4800	5	Transmission speed 4800 baud
FormulaBaud9600	6	Transmission speed 9600 baud
FormulaBaud14400	7	Transmission speed 14400 baud
FormulaBaud19200	8	Transmission speed 19200 baud
FormulaBaud38400	9	Transmission speed 38400 baud
FormulaBaud56000	10	Transmission speed 56000 baud
FormulaBaud57600	11	Transmission speed 57600 baud
FormulaBaud115200	12	Transmission speed 115200 baud
FormulaBaud128000	13	Transmission speed 128000 baud

*FORMULA BYTESIZE CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaByteSize7	0	7 data bits
FormulaByteSize8	1	8 data bits

*FORMULA COMMPORT CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaCommPort1	0	Serial port 1
FormulaCommPort2	1	Serial port 2
FormulaCommPort3	2	Serial port 3
FormulaCommPort4	3	Serial port 4
FormulaCommPort5	4	Serial port 5
FormulaCommPort6	5	Serial port 6
FormulaCommPort7	6	Serial port 7
FormulaCommPort8	7	Serial port 8

*FORMULA ERROR CONSTANT**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked
FormulaErrorNotClaimed	103	Device does not have exclusive access
FormulaErrorNoService	104	Requested service does not exist
FormulaErrorDisabled	105	Device disabled
FormulaErrorIllegal	106	Invalid operation
FormulaErrorNoHardware	107	Device requested does not exist or cannot be used
FormulaErrorOffline	108	Device off-line
FormulaErrorNoExists	109	Device does not exist
FormulaErrorExists	110	Device already exists
FormulaErrorFailure	111	Operation not successfully carried out
FormulaErrorTimeout	112	Maximum operation time exceeded
FormulaErrorBusy	113	Device busy
FormulaErrorExtended	114	Extended error in checking the property ResultCodeExtended

*FORMULA ERROR EXTENDED CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaErrorExtendedAddress	201	Invalid address
FormulaErrorExtendedFileOpened	202	Device's file invalid or in use
FormulaErrorExtendedNotOpen	203	Device's file was not opened
FormulaErrorExtendedMemoryFull	204	Device 's memory full
FormulaErrorExtendedRAM	205	RAM error in device
FormulaErrorExtendedEEPROM	206	EEPROM error in device
FormulaErrorExtendedRAM_EEPROM	207	RAM and EEPROM error in device
FormulaErrorExtendedPolling	208	Device's polling inactive

*FORMULA PARITY CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaParityNone	0	No parity
FormulaParityOdd	1	Odd parity
FormulaParityEven	2	Even parity
FormulaParityMark	3	Mark parity
FormulaParitySpace	4	Space parity

*FORMULA RUNMODE CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaRunModeNetwork	0	Network
FormulaRunModeTransmitImmediately	1	Immediate transmission
FormulaRunModeTestSiteSurvey	2	Test site survey
FormulaRunModeTestEcho	3	Test echo
FormulaRunModeTestRSSI	4	Test RSSI

*FORMULA SETUP CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSetupPolling	1	Enables polling during network startup
FormulaSetupDownloadFile	2	Downloads data onto a file
FormulaSetupDownloadRecord	4	Downloads data one record at time
FormulaSetupFileAppend	8	File append. If this is not set, the file is overwritten.
FormulaSetupSatPolling(Sat object)	65536	Enabled polling between satellite and terminals
FormulaSetupTermBeep(Tem object)	65536	Enables the terminal's beep

*FORMULA STATE CONSTANTS**AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaStateClosed	1	OLE for Formula inactive
FormulaStateIdle	2	OLE for Formula active
FormulaStateBusy	3	OLE for Formula active but carrying out another operation (for example downloading data)
FormulaStateError	4	OLE for Formula in error status

## 2.4 FORMULA PROPERTIES

PROPERTY *ACKTIMEOUT* (FORMULA OBJECT)

### *DESCRIPTION*

---

Sets or returns the waiting time for a reply from the terminals and/or satellites. The time is expressed in milliseconds.

### *SYNTAX*

---

[form.] Formula. AckTimeout [=integer]

### *AVAILABLE VALUE*

---

Not all the devices have the same response time. If you have a mixed network, you must insert the highest time. The default value is 500 milliseconds.

### *EXAMPLE*

---

Formula. AckTimeout = 500

### *DATA TYPE*

---

Integer

*PROPERTY **BARCODECHECKDIGIT** (TERM OBJECT)***DESCRIPTION**

Sets or returns the type of bar code, whose last character the terminal must recognize as a check digit.

**SYNTAX**

Object. BarcodeCheckDigit [= long]

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaCheckDigitStandard39	1	Enables a character check on Standard codes 3/9
FormulaCheckDigitInterleaved25	2	Enables a character check on Interleaved codes 2/5
FormulaCheckDigitMatrix25	4	Enables a character check on Matrix codes 2/5
FormulaCheckDigitDeltaAIBM	8	Enables a character check on Delta A IBM codes
FormulaCheckDigitMSI	16	Enables a character check on MSI codes

**EXAMPLE**

Set termX = Formula.Terminals.Add

```
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

**DATA TYPE**

Long

---

PROPERTY *BARCODESIZE* (TERM OBJECT)

*DESCRIPTION*

---

Sets or returns the maximum bar code length the terminal can read.

*SYNTAX*

---

Object. BarcodeSize [= integer]

*COMMENT*

---

The possible values of this property range from 0 (default value) to 32 characters. If the property is set at 0, it means there are no checks on the length of the bar codes.

*EXAMPLE*

---

```
Dim termX As term
Set termX = Formula.Terminals.Add
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

*DATA TYPE*

---

Integer



*PROPERTY BARCODE TYPE (TERM OBJECT)***DESCRIPTION**

Sets or returns the types of bar codes the terminal must read.

**SYNTAX**

object. BarcodeType [= long]

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaBarcodeStandard39	1	Enables reading of Standard codes 3/9
FormulaBarcodeInterleaved25	2	Enables reading of Interleaved codes 2/5
FormulaBarcodeUPCEAN	4	Enables reading of UPC and EAN codes
FormulaBarcodeMonarch27	8	Enables reading of Monarch codes 2/7
FormulaBarcodeUPCOnly	16	Enables reading of UPC codes only
FormulaBarcodeUPCEANPlus2	32	Enables reading of extended UPC and EAN codes +2
FormulaBarcodeUPCEANPlus5	64	Enables reading of extended UPC and EAN codes +5
FormulaBarcodeExtended39	128	Enables reading of Extended 39 codes
FormulaBarcodePharmaceutical	256	Enables reading of pharmaceutical codes
FormulaBarcodeUPCEOnly	512	Enables reading of UPC E codes only
FormulaBarcodeEAN8Only	1024	Enables reading of EAN 8 codes only
FormulaBarcodeUPCAEAN13Only	2048	Enables reading of UPC E and EAN 13 codes only
FormulaBarcodePako	4096	Enables reading of Pako codes
FormulaBarcodeCode128	8192	Enables reading of 128 codes
FormulaBarcodeIndustrial 25	16384	Enables reading of industrial codes 2/5
FormulaBarcodeMatrix25	32768	Enables reading of Matrix codes 2/5
FormulaBarcodeDeltaAIBM	65536	Enables reading of Delta A IBM codes
FormulaBarcodeMSI	131072	Enables reading of MSI codes
FormulaBarcodeCode93	262144	Enables reading of 93 codes
FormulaBarcodeITF14	524288	Enables reading of ITF codes 1/4

**EXAMPLE**

---

```
Dim termX As term
Set termX = Formula.Terminals.Add
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

**DATA TYPE**

---

Long

*PROPERTY* **BAUDRATE** (*FORMULA OBJECT*)**DESCRIPTION**

Sets or returns the transmission speed between the PC and the network.

**SYNTAX**

[form. ] Formula. BaudRate [= integer]

**COMMENT****Warning!!!**

Be careful in your choice of the transmission speed. Each time you must refer to the device you intend to connect.

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaBaud110	0	Transmission speed 110 baud
FormulaBaud300	1	Transmission speed 300 baud
FormulaBaud600	2	Transmission speed 600 baud
FormulaBaud1200	3	Transmission speed 1200 baud
FormulaBaud2400	4	Transmission speed 2400 baud
FormulaBaud4800	5	Transmission speed 4800 baud
FormulaBaud9600 (default)	6	Transmission speed 9600 baud
FormulaBaud14400	7	Transmission speed 14400 baud
FormulaBaud19200	8	Transmission speed 19200 baud
FormulaBaud38400	9	Transmission speed 38400 baud
FormulaBaud56000	10	Transmission speed 56000 baud
FormulaBaud57600	11	Transmission speed 57600 baud
FormulaBaud115200	12	Transmission speed 115200 baud
FormulaBaud128000	13	Transmission speed 128000 baud

**EXAMPLE**

---

Formula. CommPort = FormulaComm1  
 Formula. BaudRate = FormulaBaud9600  
 Formula. ByteSize = FormulaByteSize8  
 Formula. Parity = FormulaParityMark  
 Formula. Open

**DATA TYPE**

---

Integer

**PROPERTY *BYTESIZE* (FORMULA OBJECT)****DESCRIPTION**

---

Sets or returns the size of the character being transmitted between the PC and the network.

**SYNTAX**

---

[form. ] Formula. ByteSize [= integer]

**AVAILABLE VALUE**

---

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaByteSize7	0	7 data bits
FormulaByteSize8 (default)	1	8 data bits

**EXAMPLE**

---

Formula. CommPort = FormulaComm1  
 Formula. BaudRate = FormulaBaud9600  
 Formula. ByteSize = FormulaByteSize8  
 Formula. Parity = FormulaParityMark  
 Formula. Open

**DATA TYPE**

---

Integer

PROPERTY **CHECKHEALTHTEXT** (TERM OBJECT, SAT OBJECT)

**DESCRIPTION**

---

Displays the results of the last time the CheckHealth method command was given.

**SYNTAX**

---

object. CheckHealthText

**COMMENT**

---

The possible values are those sent back from the terminal or satellite after the command 5, such as <RUN> for example.

**EXAMPLE**

---

Formula. Terminals (1) .CheckHealth (0)  
MsgBox Formula. Terminals (1) .CheckHealthText

**DATA TYPE**

---

String

PROPERTY **CLAIMED** (FORMULA OBJECT)

**DESCRIPTION**

---

Not supported.

*PROPERTY **COMMPORT** (FORMULA OBJECT)**DESCRIPTION*

Sets or returns the serial port for transmission between the PC and the network.

*SYNTAX*

[form. ] Formula. CommPort [= integer]

*AVAILABLE VALUE*

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaCommPort1	0	Serial port 1
FormulaCommPort2	1	Serial port 2
FormulaCommPort3	2	Serial port 3
FormulaCommPort4	3	Serial port 4
FormulaCommPort5	4	Serial port 5
FormulaCommPort6	5	Serial port 6
FormulaCommPort7	6	Serial port 7
FormulaCommPort8	7	Serial port 8

*EXAMPLE*

Formula. CommPort = FormulaComm1  
 Formula. BaudRate = FormulaBaud9600  
 Formula. ByteSize = FormulaByteSize8  
 Formula. Parity = FormulaParityMark  
 Formula. Open

*DATA TYPE*

Integer

*PROPERTY CONTROLOBJECTDESCRIPTION (FORMULA OBJECT)*

***DESCRIPTION***

---

Returns the OCX description.

***SYNTAX***

---

[form. ] Formula. ControlObjectDescription

***COMMENT***

---

The property value is "Formula Service Control, (C) 2001 Datalogic S.p.A.". The property is always available and is in read-only format.

***EXAMPLE***

---

szRet = Formula .ControlObjectDescription

***DATA TYPE***

---

String

---

*PROPERTY CONTROLOBJECTVERSION (FORMULA OBJECT)*

***DESCRIPTION***

---

Returns the OCX version.

***SYNTAX***

---

[form. ] Formula. ControlObjectVersion

***COMMENT***

---

The version number is expressed with a fixed point. For example the number 1001123 is shown as version 1.001.123.  
This property is always available and is read-only format.

***EXAMPLE***

---

ret = Formula .ControlObjectVersion

***DATA TYPE***

---

Long



*PROPERTY DATAEVENTENABLED (FORMULA OBJECT)****DESCRIPTION***

---

Returns or sets the ability to receive the DataEvent event.

***SYNTAX***

---

[form. ] Formula. DataEventEnabled [= {True | False}]

***COMMENT***

---

If True is entered, when the PC has received the data, the DataEvent event is sent; whereas if False is entered, the computer will receive no event to indicate the arrival of data.

***EXAMPLE***

---

Formula. DataEventEnabled = True

***DATA TYPE***

---

Integer (boolean)

PROPERTY *DELAY1, DELAY2, DELAY3, DELAY4* (SAT OBJECT)

#### **DESCRIPTION**

---

Returns or sets the internal transmission times between the satellites and the terminals. The times are expressed in milliseconds.

#### **SYNTAX**

---

object. Delay1 [= integer], object. Delay2 [= integer],  
object. Delay3 [= integer], object. Delay4 [= integer]

#### **COMMENT**

---

##### **Warning!!!**

Be careful if changing these values. Each time you must refer to the device you intend to connect.

The default values for RF devices are: 12, 2, 7, 40.

The default values for IR devices are: 10, 7, 2, 40.

#### **EXAMPLE**

---

Formula. Satellites {101} . Delay1 = 12

Formula. Satellites {101} . Delay2 = 2

Formula. Satellites {101} . Delay3 = 7

Formula. Satellites {101} . Delay4 = 40

#### **DATA TYPE**

---

Integer

*PROPERTY **DEVICE**DATA (FORMULA OBJECT)****DESCRIPTION***

---

Returns the last data received from the devices.

***SYNTAX***

---

[ form ] .Formula. DeviceData

***COMMENT***

---

This property is set before receiving the DataEvent event. The property is always available and is in read-only format.

***EXAMPLE***

---

```
Sub Formula_DataEvent {ByVal OutputID As Integer}
    List.AddItem Formula.DeviceData
End Sub
```

***DATA TYPE***

---

String

PROPERTY *DEVICEDESCRIPTION* (TERM OBJECT, SAT OBJECT)

**DESCRIPTION**

---

Returns the description of the device.

**SYNTAX**

---

object.DeviceDescription

**COMMENT**

---

The property value can vary depending on the type of device. The property is always available and is in read-only format.

**EXAMPLE**

---

MsgBox Formula.Terminals (1) .DeviceDescription

**DATA TYPE**

---

String

PROPERTY *DEVICEENABLED* (*TERM OBJECT*, *SAT OBJECT*)

---

**DESCRIPTION**

Returns or sets if the downloading of the device's data is enabled.

---

**SYNTAX**

object.DeviceEnabled [= {True|False}]

---

**COMMENT**

If True (default) is entered, when the terminal or satellite has data to download, this is received by the PC, whereas if False is entered, the computer receives no data.

---

**EXAMPLE**

Formula.Terminals (1) .DeviceEnabled = False

---

**DATA TYPE**

Integer (boolean)

PROPERTY ***DEVICENAME*** (TERM OBJECT, SAT OBJECT)

***DESCRIPTION***

---

Returns the device's name.

***SYNTAX***

---

object.DeviceName

***COMMENT***

---

The property value can vary depending on the type of device. The property is always available in read-only format.

***EXAMPLE***

---

MsgBox Formula.Terminals (1) .DeviceName

***DATA TYPE***

---

String

PROPERTY *DEVICEVERSION* (TERM OBJECT, SAT OBJECT)

***DESCRIPTION***

---

Returns the device's version.

***SYNTAX***

---

object.DeviceVersion

***COMMENT***

---

The version number is expressed with a fixed point. For example the number 1001123 is shown as version 1.001.123.  
This property is always available in read-only format.

***EXAMPLE***

---

ret = Formula.Terminals (1) .DeviceVersion

***DATA TYPE***

---

Long

PROPERTY *DIRECTIOREAD* (TERM OBJECT, SAT OBJECT)

#### *DESCRIPTION*

---

See the result of the last call to the DirectIO method when the sent command requires a multiple reply.

#### *SYNTAX*

---

object. DirectIOText

#### *COMMENT*

---

The possible values vary depending on the direct commands sent to the device using the DirectIO method.

#### *EXAMPLE*

---

```
Dim termX As term
Set termX = Formula.Terminals (1)
ret = termX.DirectIO (Chr (27) + "5" + Chr (27))
do
a = (termix.DirectIORead (""))
list1.additem (a)
loop until a = "<EOF>"
```



PROPERTY *DIRECTIOTEXT* (TERM OBJECT, SAT OBJECT)

---

**DESCRIPTION**

Displays the result of the last time the DirectIO method command was given.

---

**SYNTAX**

object.DirectIOText

---

**COMMENT**

The possible values vary depending on the direct commands sent to the device using the DirectIO method.

---

**EXAMPLE**

```
Dim termX As term
Set termX = Formula.Terminals (1)
ret = termX.DirectIO (Chr (27) + "5" + Chr (27))
MsgBox termX.DirectIOText
```

---

**DATA TYPE**

String

PROPERTY **FIELDSEPARATOR** (TERM OBJECT, SAT OBJECT)

**DESCRIPTION**

---

Sets or returns the device's field separator.

**SYNTAX**

---

object.FieldSeparator [= integer]

**COMMENT**

---

The value must be entered as a decimal ASCII code (eg The character ! corresponds to ASCII 33).  
The default value is 32 (blank).

**EXAMPLE**

---

```
Dim termX As Term
Set termX = Formula.Terminals.Add
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

**DATA TYPE**

---

Integer

PROPERTY **FILENAME** (TERM OBJECT, SAT OBJECT)

---

**DESCRIPTION**

Sets or returns the name of the file on which the data received from the device is written.

---

**SYNTAX**

object.FileName [= string]

---

**COMMENT**

The filename must be valid and can also include the route.

---

**EXAMPLE**

```
Dim termX As Term
Set termX = Formula.Terminals.Add
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

---

**DATA TYPE**

String

PROPERTY **FREEZEEVENTS** (FORMULA OBJECT)

---

**DESCRIPTION**

Not supported.

---

*PROPERTY GROUP (SAT OBJECT)*

*DESCRIPTION*

---

Sets or returns the name of the group to which the satellite belongs.

*SYNTAX*

---

object.Group [= Integer]

*COMMENT*

---

The group must be between 1 and 6. The default group is 1. Consult the satellite operating manual if using this property.

*EXAMPLE*

---

Formula. Satellites (101). Group = 1

*DATA TYPE*

---

Integer

*PROPERTY ID (TERM OBJECT, SAT OBJECT)***DESCRIPTION**

---

Sets or returns identifying number of the terminal or satellite.

**SYNTAX**

---

object. ID [= Integer]

**COMMENT**

---

The device's identifying number is stored on the terminal or satellite. For terminals it is between 1 and 99 and for satellites it is between 101 and 199.

**EXAMPLE**

---

```
Dim termX As Term
Set termX = Formula.Terminals.Add
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

**DATA TYPE**

---

Integer

*PROPERTY OUTPUTID (FORMULA OBJECT)***DESCRIPTION**

---

Not supported.

*PROPERTY **PARITY** (FORMULA OBJECT)***DESCRIPTION**

Sets or returns the transmission parity between the PC and the network.

**SYNTAX**

[form. ] Formula. Parity [= Integer]

**COMMENT****Warning!!!**

Be careful in your choice of transmission parity. Each time you must refer to the device you intend to connect.

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaParityNone	0	No parity
FormulaParityOdd	1	Odd parity
FormulaParityEven	2	Even parity
FormulaParityMark	3	Mark parity
FormulaParitySpace	4	Space parity

**EXAMPLE**

Formula. CommPort = FormulaComm1  
 Formula. BaudRate = FormulaBaud9600  
 Formula. ByteSize = FormulaByteSize8  
 Formula. Parity = FormulaParityMark  
 Formula. Open

**DATA TYPE**

Integer

*PROPERTY RESULTCODE (FORMULA OBJECT)***DESCRIPTION**

Displays the last error code.

**SYNTAX**

[form. ] Formula. ResultCode

**COMMENT**

This property is configured from each of the OCX methods and indicates an error if one is found.

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked
FormulaErrorNotClaimed	103	Device does not have exclusive access
FormulaErrorNoService	104	Requested service does not exist
FormulaErrorDisabled	105	Device disabled
FormulaErrorIllegal	106	Invalid operation
FormulaErrorNoHardware	107	Device requested does not exist or cannot be used
FormulaErrorOffline	108	Device off-line
FormulaErrorNoExists	109	Device does not exist
FormulaErrorExists	110	Device already exists
FormulaErrorFailure	111	Operation not successfully carried out
FormulaErrorTimeout	112	Maximum operation time exceeded
FormulaErrorBusy	113	Device busy
FormulaErrorExtended	114	Extended error in checking the property ResultCodeExtended

**EXAMPLE**

---

```
Select Case Formula.ResultCode
  Case FormulaErrorClosed
  Case FormulaErrorFailure
  Case FormulaErrorExtended
    Select Case Formula.ResultCodeExtended
      Case FormulaErrorExtendedNoAddress
        MsgBox "Terminal address not valid"
    End Select
End Select
```

**DATA TYPE**

---

Long



*PROPERTY RESULTCODEEXTENDED (FORMULA OBJECT)***DESCRIPTION**

Indicates a specific error in OLE for Formula's OLE Custom Control.

**SYNTAX**

[form. ] Formula. ResultCodeExtended

**COMMENT**

When the ResultCode property has the value FormulaErrorExtended, ResultCodeExtended shows the error specific to the device.

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaErrorExtendedAddress	201	Invalid address
FormulaErrorExtendedFileOpened	202	Device's file invalid or in use
FormulaErrorExtendedNotOpen	203	Device's file was not opened
FormulaErrorExtendedMemoryFull	204	Device 's memory full
FormulaErrorExtendedRAM	205	RAM error in device
FormulaErrorExtendedEEPROM	206	EEPROM error in device
FormulaErrorExtendedRAM_EEPROM	207	RAM and EEPROM error in device
FormulaErrorExtendedPolling	208	Device's polling inactive

**EXAMPLE**

```

Select Case Formula.ResultCode
  Case FormulaErrorClosed
  Case FormulaErrorFailure
  Case FormulaErrorExtended
    Select Case Formula.ResultCodeExtended
      Case FormulaErrorExtendedNoAddress
        MsgBox "Terminal address not valid"
    End Select
  End Select
End Select

```

**DATA TYPE**

Long

---

PROPERTY ***RUNFROMTERMINAL*** (*SAT OBJECT*)

***DESCRIPTION***

---

Sets or returns the initial value of the range the satellite must manage, stipulating the first and last terminals IDs.

***SYNTAX***

---

object.RunFromTerminal [= integer]

***COMMENT***

---

The device's identifying number is stored on the terminal. It must be between 1 and 99.

***EXAMPLE***

---

```
Dim satX As Sat
Set satX = Formula.Satellites.Add
satX.ID = 1
satX.RunFromTerminal = 1
satX.RunToTerminal = 5
satX.RunMode = FormulaRunModeNetwork
satX.FieldSeparator = 32
satX.FileName = "SATELLITE.DAT"
satX.Setup = FormulaSetupPolling + FormulaSetupDownloadFile
```

***DATA TYPE***

---

Integer

*PROPERTY RUNMODE (SAT OBJECT)***DESCRIPTION**

Sets or returns the satellite's operating mode.

**SYNTAX**

object.RunMode [= integer] Dim satX As Sat

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaRunModeNetwork	0	Network
FormulaRunModeTransmitImmediately	1	Immediate transmission
FormulaRunModeTestSiteSurvey	2	Test site survey
FormulaRunModeTestEcho	3	Test echo
FormulaRunModeTestRSSI	4	Test RSSI

**EXAMPLE**

```
Dim satX As Sat

Set satX = Formula.Satellites.Add
satX.ID = 1
satX.RunFromTerminal = 1
satX.RunToTerminal = 5
satX.RunMode = FormulaRunModeNetwork
satX.FieldSeparator = 32
satX.FileName = "SATELLITE.DAT"
satX.Setup = FormulaSetupPolling + FormulaSetupDownloadFile
```

**DATA TYPE**

Integer

---

PROPERTY ***RUNTO**TERMINAL* (*SAT* OBJECT)

***DESCRIPTION***

---

Sets or returns the final value of the range the satellite must manage, stipulating the first and last terminals IDs.

***SYNTAX***

---

object. RunToTerminal [= integer]

***COMMENT***

---

The device's identifying number is stored on the terminal. It must be between 1 and 99.

***EXAMPLE***

---

```
Dim satX As Sat
Set satX = Formula.Satellites.Add
satX.ID = 1
satX.RunFromTerminal = 1
satX.RunToTerminal = 5
satX.RunMode = FormulaRunModeNetwork
satX.FieldSeparator = 32
satX.FileName = "SATELLITE.DAT"
satX.Setup = FormulaSetupPolling + FormulaSetupDownloadFile
```

***DATA TYPE***

---

Integer

*PROPERTY SATELLITES COLLECTION, SAT OBJECT***DESCRIPTION**

The satellites collection contains one or more Sat objects. The Sat object contains all the properties and methods that allow you to use the satellites.

**SYNTAX**

[form. ] .Formula. Satellites  
 [form. ] .Formula. Satellites (index)

**PROPERTY**

*CheckHealthText (Sat object)	*DeviceName (Sat object)	*RunFromTerminal (Sat object)
Count	*DeviceVersion (Sat object)	*RunMode (Sat object)
*Delay1	*DirectIOText (Sat object)	*RunToTerminal (Sat object)
*Delay2 (Sat object)	*FieldSeparator (Sat object)	*Satellites
*Delay3 (Sat object)	*FileName (Sat object)	*Setup (Sat object)
*Delay4 (Sat object)	*Group (Sat object)	*State (Sat object)
*DeviceDescription (Sat object)	*ID (Sat object)	
*DeviceEnabled(Sat object)	Index (Sat object)	

**METHOD**

Add	*DataWrite (Sat object)	*Initiate (Sat object)
*CheckHealth (Sat object)	*DirectIO (Sat object)	Item
*ClearOutput (Sat object)	*FileWrite (Sat object)	Remove

**COMMENT**

\* Properties and methods added to the satellites collection.

**EXAMPLE**

```
Dim satX As Sat
Set satX = Formula. Satellites. Add
satX. ID = 1
satX. RunFromTerminal = 1
satX. RunToTerminal = 5
satX. RunMode = FormulaRunModeNetwork
satX. FieldSeparator = 32
satX. FileName = "SATELLITE.DAT"
satX. Setup = FormulaSetupPolling + FormulaSetupDownloadFile
```

---

PROPERTY *SCANDATA* (FORMULA OBJECT)

*DESCRIPTION*

---

Returns the last data received from the devices.

*SYNTAX*

---

[form. ] .Formula. ScanData

*COMMENT*

---

This property should be set before receiving the DataEvent event. The property is always available in read-only format.

*EXAMPLE*

---

```
Sub Formula_DataEvent {ByVal OutputID As Integer}
    List.AddItem Formula.ScanData
End Sub
```

*DATA TYPE*

---

String

---

*PROPERTY SERVICEOBJECTDESCRIPTION (FORMULA OBJECT)*

---

*DESCRIPTION*

---

Returns the Service Object description.

---

*SYNTAX*

---

[form. ] Formula.ServiceObjectDescription

---

*COMMENT*

---

The property's value is "Formula Service Driver, © 2001 Datalogic S.p.A.". The property is always available in read-only format.

---

*EXAMPLE*

---

szRet = Formula.ServiceObjectDescription

---

*DATA TYPE*

---

String

---

PROPERTY ***SERVICEOBJECTVERSION*** (FORMULA OBJECT)

***DESCRIPTION***

---

Returns the Service Object version.

***SYNTAX***

---

[form. ] Formula.ServiceObjectVersion

***COMMENT***

---

The version number is expressed with a fixed point. For example the number 1001123 is shown as version 1.001.123.  
This property is always available in read-only format.

***EXAMPLE***

---

ret = Formula.ServiceObjectVersion

***DATA TYPE***

---

Long



*PROPERTY SETUP (TERM OBJECT, SAT OBJECT)***DESCRIPTION**

Sets or returns the operating mode of the terminal or satellite.

**SYNTAX**

object. Setup [= Long]

**AVAILABLE VALUE**

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSetupPolling	1	Enables polling during network start-up.
FormulaSetupDownloadFile	2	Downloads data onto a file.
FormulaSetupDownloadRecord	4	Downloads data one record at a time.
FormulaSetupFileAppend	8	File append. If this is not set then it is file overwrite.
FormulaSetupSatPolling (Sat object)	65536	Enables polling between satellite and terminals.
FormulaSetupTermBeep (Term object)	65536	Enables the terminal's beep.

**EXAMPLE**

```
Dim satX As Sat
Set satX = Formula.Satellites.Add
satX.ID = 1
satX.RunFromTerminal = 1
satX.RunToTerminal = 5
satX.RunMode = Formula.RunModeNetwork
satX.FieldSeparator = 32
satX.FileName = "SATELLITE.DAT"
satX.Setup = FormulaSetupPolling + FormulaSetupDownloadFile
```

**DATA TYPE**

Long

---

PROPERTY ***SLEEP*** (FORMULA OBJECT)

***DESCRIPTION***

---

Sets or returns the waiting time for a reply from every device. The time is expressed in milliseconds.

***SYNTAX***

---

[form. ] Formula.Sleep [= integer]

***COMMENT***

---

The value of this property is normally 0. Each time you must refer to the devices you intend to connect.

***EXAMPLE***

---

ret = Formula.Sleep

***DATA TYPE***

---

Integer

*PROPERTY STATE (FORMULA OBJECT)***DESCRIPTION**

---

Indicates the network's state.

**SYNTAX**

---

[form. ] Formula.State

**COMMENT**

---

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaStateClosed	1	OLE for Formula inactive
FormulaStateIdle	2	OLE for Formula active
FormulaStateBusy	3	OLE for Formula active but carrying out another operation (for example downloading data)
FormulaStateError	4	OLE for Formula in error status

**EXAMPLE**

---

ret = Formula.State

See also "Event StatusUpdateEvent (Formula object)" described on page 58.

**DATA TYPE**

---

Long

PROPERTY **STATE** (*TERM OBJECT, SAT OBJECT*)

**DESCRIPTION**

---

Indicates the device's state.

**SYNTAX**

---

object.State

**COMMENT**

---

<b>CONSTANT</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaStateClosed	1	Device inactive or not responding.
FormulaStateIdle	2	Device active
FormulaStateBusy	3	Device active but carrying out another operation (for example downloading data)
FormulaStateError	4	Device in error status

**EXAMPLE**

---

ret = Formula.Terminals (TermID). State

ret = Formula.Satellite (SatID). State

See also "Event StatusUpdateEvent (Formula object)" described on page 58.

**DATA TYPE**

---

Long

*PROPERTY TERMINALS COLLECTION, TERM OBJECT***DESCRIPTION**

The terminals collection contains one or more Term objects. The Term object contains all the properties and methods that allow you to use the terminals.

**SYNTAX**

[form. ] .Formula. Terminals  
[form. ] .Formula. Term (index)

**PROPERTY**

*BarcodeCheckDigit (Term object)	*DeviceEnabled (Term object)	*ID (Term object)
*BarcodeSize (Term object)	*DeviceName (Term object)	Index (Term object)
*BarcodeType (Term object)	*DeviceVersion (Term object)	*Setup (Term object)
*CheckHealthText (Term object)	*DirectIOText (Term object)	*State (Term object)
Count	*FieldSeparator (Term object)	*Terminals
*DeviceDescription (Term object)	*FileName (Term object)	*TxSleep (Term object)

**METHOD**

Add	*DataWrite (Term object)	*Initiate (Term object)
*CheckHealth (Term object)	*DirectIO (Term object)	Item
*ClearOutput (Term object)	*FileWrite (Term object)	Remove

**COMMENT**

\* Properties and methods added to the terminals collection.

**EXAMPLE**

```
Dim termX As Term
Set termX = Formula.Terminals.Add
termX.ID = 1
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

---

PROPERTY ***TxSLEEP*** (*TERM OBJECT*)

***DESCRIPTION***

---

Sets or returns the transmission delay, expressed in milliseconds.

***SYNTAX***

---

object.TxSleep [= integer]

***COMMENT***

---

The default value of this property is 2. Each time you must refer to the devices you intend to connect.

***EXAMPLE***

---

Formula.Terminals(1).TxSleep = 2

***DATA TYPE***

---

Long

## 2.5 FORMULA EVENTS

*EVENT DATAEVENT (FORMULA OBJECT)*

### DESCRIPTION

---

It is activated every time the PC receives new data from a device.

### SYNTAX

---

```
Private Sub Formula_DataEvent (ByVal OutputID As Integer)
```

### COMMENT

---

If Download Record has been enabled in the device's Setup property, every record will be reported by this event.

OutputID identifies which device sent the record. If the number is below 100, a terminal is identified; if it is over 100, a satellite is identified.

The DeviceData and/or ScanData properties store the record.

### EXAMPLE

---

```
Private Sub Formula_DataEvent (ByVal OutputID As Integer)
    1stList.AddItem Formula.DeviceData
    1stList.ListIndex = 1stList.NewIndex
End Sub
```

---

*EVENT **DIRECTIOEVENT** (FORMULA OBJECT)****DESCRIPTION***

---

Not supported.

*EVENT **ERROR EVENT** (FORMULA OBJECT)****DESCRIPTION***

---

Indicates an error has occurred.

***SYNTAX***

---

```
Private Sub Formula_ErrorEvent (ByVal ResultCode As Long,
ByVal ResultCodeExtended As Long, ByVal ErrorLocus As Long,
ErrorResponse As Long)
```

***COMMENT***

---

This event appears every time an error is discovered. It shows the last value of the ResultCode, the extended error in the ResultCodeExtended property and indicating the operation during which the error was detected.

***EXAMPLE***

---

```
Private Sub Formula_ErrorEvent (ByVal ResultCode As Long
ByVal ResultCodeExtended As Long, ByVal ErrorLocus As Long,
ErrorResponse As Long)
    MsgBox "Error Formula: " + CStr (ResultCode)
End Sub
```



*EVENT OUTPUTCOMPLETEEVENT (FORMULA OBJECT)****DESCRIPTION***

---

Indicates that one device has terminated its downloading.

***SYNTAX***

---

```
Private Sub Formula_OuputCompleteEvent (ByVal OutputID  
As Integer)
```

***COMMENT***

---

If the device's Setup property Download File has been enabled, the completion of the downloading will be reported by this event. OutputID identifies which device sent the record. If the number is below 100 a terminal is identified; if it is over 100 a satellite is identified.

***EXAMPLE***

---

```
Private Sub Formula_OutputCompleteEvent (ByVal OutputID  
As Integer)  
MsgBox "Download to the device completed" + CStr (OutputID)  
End Sub
```

*EVENT STATUSUPDATEEVENT (FORMULA OBJECT)***DESCRIPTION**

Indicates the Custom control has changed status.

**SYNTAX**

```
Private Sub Formula_StatusUpdateEvent (ByVal Data As Long)
```

**COMMENT**

This reports on how the OCX has changed the State property. For example, when transmission is initiated and successfully carried out, the event displays the information FormulaStateIdle.

**EXAMPLE**

```
Private Sub Formula_StatusUpdateEvent (ByVal OutputID As Integer)
    If OutputID <100 then
        Select Case Formula.Terminals (OutputID). State
            Case FormulaStateClosed
                MsgBox "Terminal No." & OuputID & "Closed!"
            Case FormulaStateIdle
                MsgBox "Terminal No." & OuputID & "Idle!"
            Case FormulaStateError
                MsgBox "Terminal No." & OuputID & "Error!"
            Case FormulaStateBusy
                MsgBox "Terminal No." & OuputID & "Busy!"
        End Select
    Else
        Select Case Formula.Satellites (OutputID). State
            Case FormulaStateClosed
                MsgBox "Satellite No." & OuputID & "Closed!"
            Case FormulaStateIdle
                MsgBox "Satellite No." & OuputID & "Idle!"
            Case FormulaStateError
                MsgBox "Satellite No." & OuputID & "Error!"
            Case FormulaStateBusy
                MsgBox "Satellite No." & OuputID & "Busy!"
        End Select
    End if
End Sub
```

## 2.6 FORMULA METHODS

*METHOD ACTION (FORMULA OBJECT)*

---

*DESCRIPTION*

Displays the OLE for Formula properties dialog box.

---

*SYNTAX*

[form.] Formula.Action ()

---

*COMMENT*

This methods allows you to see the properties window while the program is running, rather than in the development phase as normally happens.

---

*EXAMPLE*

Formula.Action ()

Refer to appendix A on page 74 to see the properties window displayed while using this method.

---

**METHOD *ADD*** (*TERMINALS COLLECTION, SATELLITES COLLECTION*)***DESCRIPTION***

---

Adds a new terminal or satellite to the collection.

***SYNTAX***

---

object.Add ([ID As Integer]) As Object

***COMMENT***

---

To add a new terminal or satellite you must use this method or add it to the properties window.

***EXAMPLE***

---

```
Dim termX As term
Set termX = Formula.Terminals.Add
termX.BarcodeType = FormulaBarcodeStandard39 +
FormulaBarcodeUPCEAN
termX.BarcodeCheckDigit = FormulaCheckDigitStandard39
termX.BarcodeSize = 13
termX.FieldSeparator = 32
termX.FileName = "TERMINAL.DAT"
```

*METHOD CHECKHEALTH (TERM OBJECT, SAT OBJECT)***DESCRIPTION**

Tests the device's operation.

**SYNTAX**

object.CheckHealth (Level As Long) As Long

**COMMENT**

The Level parameter is not currently in use. The result of the operation's check is shown in the CheckHealthText property.

**EXAMPLE**

Formula.Terminals (1). CheckHealth (0)  
MsgBox Formula.Terminals (1). CheckHealthText

**RETURN VALUE**

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device closed
FormulaErrorClaimed	102	Device jammed

*METHOD CLAIM (FORMULA OBJECT)***DESCRIPTION**

Not supported.

*METHOD CLEARINPUT (FORMULA OBJECT)***DESCRIPTION**

Not supported.

*METHOD CLEAROUTPUT (TERM OBJECT, SAT OBJECT)***DESCRIPTION**

Deletes the last data sent by the FileWrite method.

**SYNTAX**

object.ClearOutput ( ) As Long  
 object.ClearOutput (ID As Integer) As Long

**COMMENT**

If you are using the ClearOutput method on the Sat object, you must specify the ID of the terminal to which the data refers.

**EXAMPLE**

```
ret = Formula.Satellites (101).ClearOutput (1)
```

**RETURN VALUE**

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked
FormulaErrorNoExists	109	Device does not exist
FormulaErrorFailure	111	Device not connected
FormulaErrorBusy	113	Device busy
FormulaErrorExtended	114	Extended error check
FormulaErrorExtendedAddress	201	Invalid address

*METHOD CLOSE (FORMULA OBJECT)**DESCRIPTION*

---

Close OLE for Formula.

*SYNTAX*

---

[form. ] .Fomula .Close ( ) As Long

*COMMENT*

---

Closes transmission with terminals and/or satellites. The method also configures the State property to FormulaStateClosed.

*EXAMPLE*

---

ret = Formula. Close ( )

*RETURN VALUE*

---

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked

*METHOD DATAWRITE (TERM OBJECT, SAT OBJECT)***DESCRIPTION**

Sends a record to the terminal or satellite.

**SYNTAX**

object.DataWrite (Data As String) As Long  
 object.DataWrite (ID As Integer, Data As String) As Long

**COMMENT**

If you are using the DataWrite method on the Sat object, you must specify the terminal ID to which the record refers.

**EXAMPLE**

```
ret = Formula.Satellites (101).DataWrite (1, "4902030060442|35000)
```

**RETURN VALUE**

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked
FormulaErrorFailure	111	Device not connected
FormulaErrorExtended	114	Extended error check
FormulaErrorExtendedAddress	201	Invalid address
FormulaErrorExtendedFileOpened	202	Device's file invalid or in use
FormulaErrorExtendedNotOpen	203	Device's file was not opened
FormulaErrorExtendedMemoryFull	204	Device's memory full



*METHOD* **DIRECTIO** (*TERM OBJECT, SAT OBJECT*)

---

**DESCRIPTION**

Allows an output operation using the device's native protocol.

---

**SYNTAX**

object.DirectIO (Data As String) As Long

---

**COMMENT**

This method allows you to access the device directly using the low level escape sequences. It is normally used to program the device's special functions.

The DirectIOText property contains the device's response after DirectIO.

---

**EXAMPLE**

```
Dim termX As Term
Set termX = Formula.Terminals.(1)
ret = termX.DirectIO (Chr (27) + "5" + Chr (27))
MsgBox termX.DirectIOText
```

---

**RETURN VALUE**

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked
FormulaErrorNoService	104	General error
FormulaErrorTimeout	112	No reply from device

---

**METHOD FILEWRITE** (TERM OBJECT, SAT OBJECT)

---

**DESCRIPTION**

Sends a file to the terminal or satellite.

---

**SYNTAX**

object.FileWrite (FileName As String) As Long  
 object.FileWrite (ID As Integer, FileName As String) As Long

---

**COMMENT**

If the FileWrite method is used on the Sat object, you must specify the terminal ID to which the record refers.

---

**EXAMPLE**

ret = Formula.Satellites (101) . FileWrite (1, "Program.prg")

---

**RETURN VALUE**

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is blocked
FormulaErrorFailure	111	Device not connected
FormulaErrorExtended	114	Extended error check
FormulaErrorExtendedAddress	201	Invalid address
FormulaErrorExtendedFileOpened	202	Device's file invalid or in use
FormulaErrorExtendedNotOpen	203	Device's file was not opened
FormulaErrorExtendedMemoryFull	204	Device 's memory full

*METHOD FINDDEVICE (FORMULA OBJECT)****DESCRIPTION***

---

Checks that the requested device exists.

***SYNTAX***

---

[form] .Formula. FindDevice (ID As Integer) As Boolean

***COMMENT***

---

The use of this method is linked to the use of the Open method (see page 72).

If the method returns the message True, the device has been found. If the message False appears, check the ResultCode for more details of the operation.

***EXAMPLE***

---

```
Formula.Load
Formula.Open
If Formula. FindDevice (101) = True Then
    MsgBox "Satellite found"
Else
    MsgBox "Satellite not found or No ResultCode answer" &
        Formula.ResultCode & ", Result Code Extended:" &
        Formula.ResultCodeExtended
End If
Formula.Close
```

---

*RETURN VALUE*

---

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is jammed
FormulaErrorNoHardWare	107	Device cannot be used
FormulaErrorNoExists	109	Device does not exist
FormulaErrorExtended	114	Extended error check
FormulaErrorExtendedRAM	205	RAM error in device
FormulaErrorExtendedEEPROM	206	EEPROM error in device
FormulaErrorExtendedRAM_EEPROM	207	RAM and EEPROM error in device
FormulaErrorExtendedPolling	208	Device's polling inactive

*METHOD **HEXWRITE** (FORMULA OBJECT)***DESCRIPTION**

Sends a program with a .HEX extension to the terminal.

**SYNTAX**

[form] .Formula. HexWrite (FileName As String) As Long

**COMMENT**

This is a complicated procedure and may involve a delay of several minutes.

**EXAMPLE**

```
If Formula.HexWrite ("F730.HEX") = FormulaSuccess Then
    MsgBox "Program loaded correctly"
Else
    MsgBox "Errors during program loading"
End If
```

**RETURN VALUE**

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is jammed
FormulaErrorNoExists	109	Device does not exist
FormulaErrorFailure	111	Device not connected

*METHOD INITIATE (FORMULA OBJECT)**DESCRIPTION*

Initializes the device.

*SYNTAX*

```
object.Initiate ([BaudRate As Integer], [ByteSize As Integer], [Parity As Integer]) As Long
object.Initiate ([ID As Integer], [BaudRate As Integer]) As Long
```

*COMMENT*

This method allows you to customize the base parameter of the terminal or satellite.

- When dealing with the terminal you can set the transmission parameters the terminal should respond to after an Initiate. If not otherwise specified, use the general ones.
- For the satellite, however, you can set both the current ID and also the transmission parameters the satellite should respond to after an Initiate. If not otherwise specified, use the general ones.

*EXAMPLE*

```
ret = Formula.Satellites (101). Initiate (100, FormulaBaud38400)
```

*RETURN VALUE*

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is jammed
FormulaErrorIllegal	106	Invalid parameters

*METHOD LOAD (FORMULA OBJECT)****DESCRIPTION***

---

Lloads all the configuration parameters from the disk.

***SYNTAX***

---

[form] .Formula.Load (Path As String) As Long

***COMMENT***

---

If the route is not shown, the file will be searched for in the current directory. This method is used in conjunction with the Save method. The configuration filename is FORMULA3.CFG. It can be requested at any time.

***EXAMPLE***

---

```
ret = Formula. Load ()
```

***RETURN VALUE***

---

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorFailure	111	Cannot read file from disk

**METHOD OPEN** (FORMULA OBJECT)**DESCRIPTION**


---

Opens OLE for Formula.

**SYNTAX**


---

[form] .Formula.Open () As Long

**COMMENT**


---

Opens transmission with the terminals and/or satellites. This method also sets the State property to FormulaStateIdle.

**EXAMPLE**


---

```
ret = Formula.Open ()
```

**RETURN VALUE**


---

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorClosed	101	Device is closed
FormulaErrorClaimed	102	Device is jammed
FormulaErrorIllegal	106	Invalid parameters
FormulaErrorNoHardware	107	Transmission port cannot be opened

**METHOD RELEASE** (FORMULA OBJECT)**DESCRIPTION**


---

Not supported.



*METHOD SAVE (FORMULA OBJECT)****DESCRIPTION***

---

Saves all configuration parameters onto disk.

***SYNTAX***

---

[form] .Formula.Save ([Path As String]) As Long

***COMMENT***

---

If the route is not shown, the file will be saved in the current directory. This method is used in conjunction with the Load method. The configuration filename is FORMULA3.CFG. It can be requested at any time.

***EXAMPLE***

---

```
ret = Formula. Save ()
```

***RETURN VALUE***

---

<b>SETTING</b>	<b>VALUE</b>	<b>DESCRIPTION</b>
FormulaSuccess	0	Operation completed successfully
FormulaErrorFailure	111	File cannot be written onto disk

# Formula<sup>®</sup>

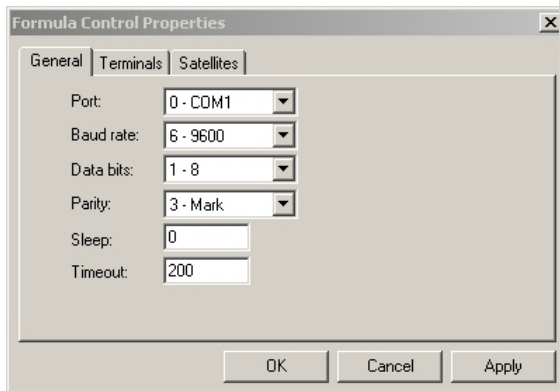
## A ACTION METHOD

---

### A.1 PROPERTIES WINDOW

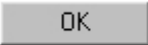


While the program is running, OLE for Formula **"Action Method"** displays the properties window normally used during the development phase.

This window is shown below:

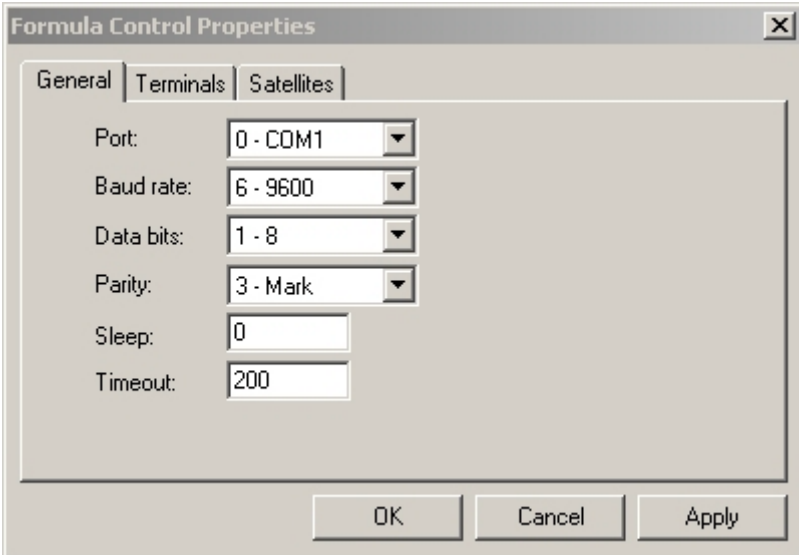


- Properties - Check Formula window is divided into three folders:
- **General Properties:** allows you to define general transmission parameters between host computer and devices;
- **Terminal properties:** allows you to define the networked terminals and their transmission mode;
- **Satellite Properties:** allows you to define the networked satellites and their transmission mode;

□ **Buttons in the Properties - Formula Control dialog box**

	Allows you to save selected properties and exit the open dialog box.
	Allows you to exit the open dialog box without saving the selected properties.
	Allows you to store the selected properties without exiting the open dialog box.

### A.1.1 General Properties



The screenshot shows the 'Formula Control Properties' dialog box with the 'General' tab selected. The dialog has three tabs: 'General', 'Terminals', and 'Satellites'. The 'General' tab contains the following settings:

- Port: 0 - COM1
- Baud rate: 6 - 9600
- Data bits: 1 - 8
- Parity: 3 - Mark
- Sleep: 0
- Timeout: 200

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Apply'.

Allows to configure:

**Port**

Host serial port used to connect the terminals.

**Baud Rate**

Transmission speed between host computer and terminals.

**Data bits**

Number of bits making up the transmitted data.

**Parity**

Type of control parity.

**Sleep**

The waiting time (expressed in milliseconds) for communication between one terminal and the next.

A value of 0 ensures the fastest polling but slows down other programs considerably. You are therefore advised to insert a Sleep time of several hundredths of a millisecond.

**Timeout**

This is the maximum waiting time for a reply from terminals or satellites, expressed in milliseconds.

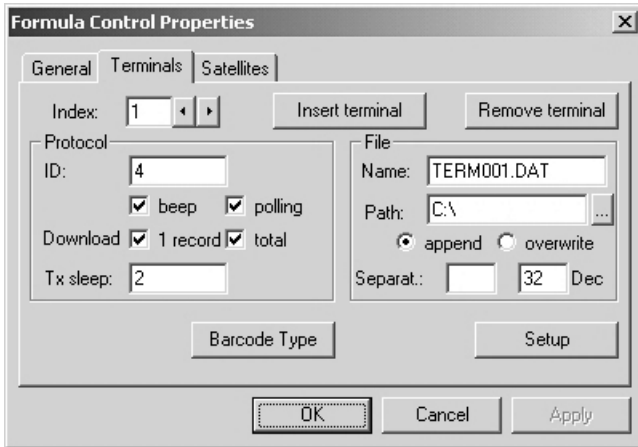
The default value is 200 ms. Depending on the specifications of the equipment in use, this value can be reduced to speed communication.



**CAUTION**

*When satellites are also being used, the choice of Baud Rate and Timeout is restricted to the values they can support. These are stated in the relevant documentation.*

## A.1.2 Terminals Properties



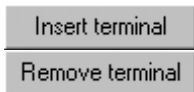
Allows you to define the networked terminals and their transmission mode.

### □ Index

Shows the terminal that the displayed data refer to.



buttons allow you to scroll the list of the networked terminals.



Enable you to update the list of programmed terminals.

### □ Protocol

**ID** Identifying number (from 1 to 99) given to each terminal.

**Beep** Allows you to enable or disable the beep within the terminal.

**Polling** Allows you to enable or disable the downloading of data from the terminal.

**Download** Allows you to define how the terminal should download data onto the host computer - one record at a time (1 record) or all the data contained in it (total).

**Tx sleep** Allows you to define, in milliseconds, the transmission delay between host computer and terminal. Generally, the suggested value should not be altered.

**□ File**

It allows to define the way to save the data received from each terminal.

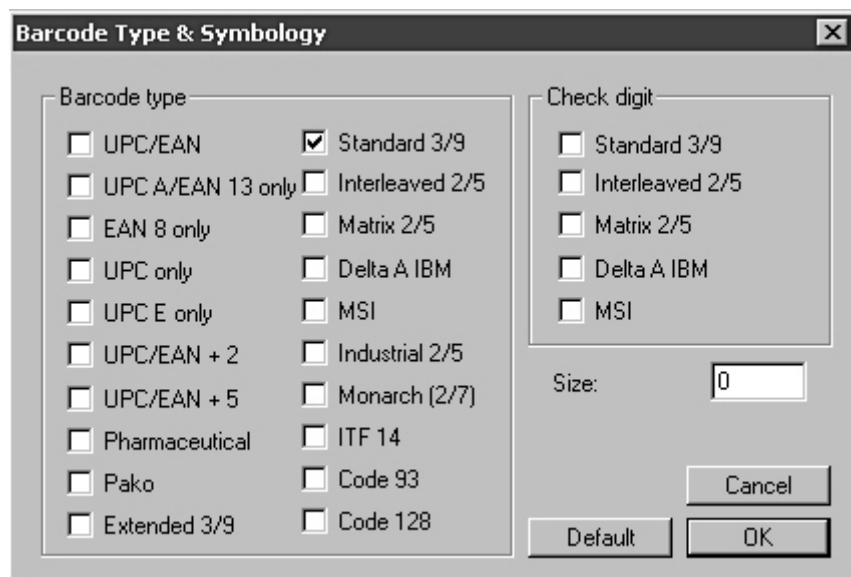
<b>Name</b>	enables you to define the filename.
<b>Path</b>	enables you to define the folder where the file will be saved.
<b>Append</b>	allows you to add new data to the file without having to remove the data already saved.
<b>Overwrite</b>	allows you to add new data to the file, replacing the data already saved.
<b>Separat.</b>	allows you to choose the character that will be used to separate the different fields of each record saved to a file. It is possible to insert the character directly or through its corresponding ASCII codification.

**CAUTION**

*If the filename is blank, no file is generated and the data must be generated absolutely with the DataEvent event.*

Barcode Type

If this button is selected, the following dialog box will appear.



The **Barcode Type & Symbology** window allows you to define:

 **Barcode type**

Types of codes used for receiving data.

 **Check digit**

This can only be available for some types of code. It checks the correct reading of the code's constituent characters.

 **Size**

Limits the readable codes to their fixed dimensions.

This limit is cancelled by the 0 value, thus allowing the reading of any code whose length comes within the maximum value of 32 characters.

### □ Buttons of the Bar Code dialog box

Cancel

allows you to return to the Terminals Properties window without saving the options selected.

OK

Saves the selected parameters and returns you to the Terminals Properties dialog box.

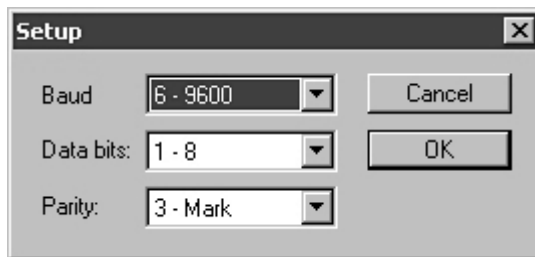
Default

Reverts the parameters of this window to the default values.

Setup

Once the terminal configuration is set in the "Terminals" folder, this button allows to send it to the relative terminal.

The displayed dialog box allows to confirm or not (for special use) the transmission parameters.

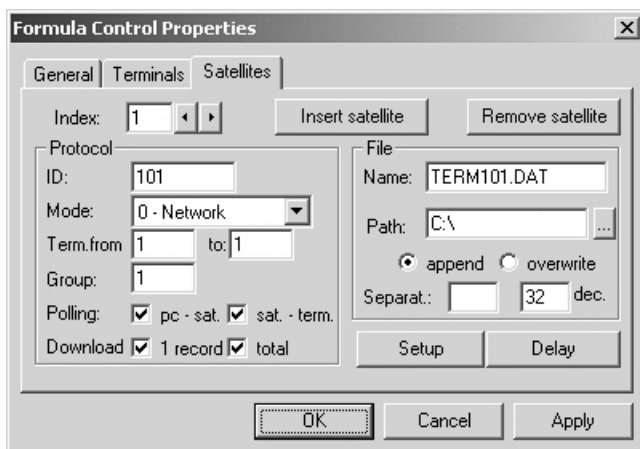


#### ATTENTION

*To program a correct initial Setup, ensure that the terminal has been previously set to the default values. This Setup occurs if a Standard or Enhanced application program is sent to the terminal or by reading the "return to default parameters" barcode in the "Menu & Commands booklet". Then, ensure that the terminal is the only one accessible in the network.*



### A.1.3 Satellites Properties



It allows to define the networked satellites and their transmission mode.

#### □ Index

It shows the satellites that the displayed data refer to .



buttons allow you to scroll the list of the networked satellites.

Insert satellite

Remove satellite

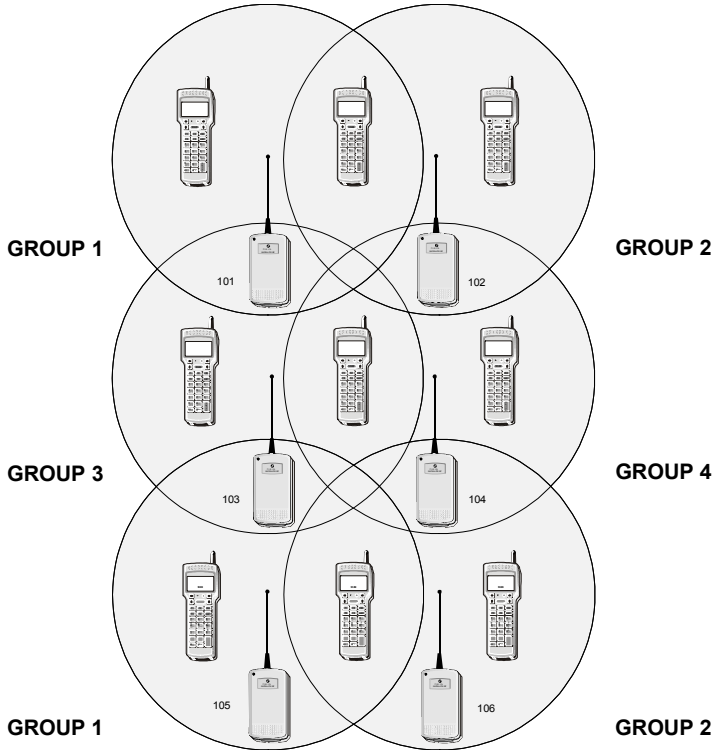
Enable you to update the list of programmed satellites.

## □ Protocol

<b>ID</b>	Identifying number (from 101 to 199) given to each satellite.
<b>Mode</b>	<p>Allows you to define the method of connection to the system:</p> <p>0 - Network = RS-485 connection with Sysnet network;</p> <p>1 - Transmit Immediate = Single RS-232 connection without the support of a transmission protocol;</p> <p>2 - Test site survey = Reserved for the testing procedure;</p> <p>3 - Test echo = Reserved for the testing procedure</p> <p>4 - Test RSSI = Reserved for the testing procedure.</p>
<b>Term. from... to</b>	<p>is necessary to indicate the first and last identifying number in the group of terminals managed by each satellite.</p> <p>If one terminal must transfer data within the coverage range of different satellites, its identifying number should be included in each group.</p>
<b>Group</b>	<p>Numerical value from 1 to 6 which allows management of different satellites (communicating with the same terminal) to be distinguished, so that no problems arise in the overlapping coverage zones.</p> <p>The polling operation requires the system to know which satellites share or not radio overlapping coverage zones. To get this information it is necessary that the satellites are divided into different groups. The satellites of each group must not share overlapping coverage and communicate with the same terminals at the same time.</p> <p>This procedure allows to the different groups to be activated alternatively, so that they never contain two satellites polling the same terminal at the same time. Furthermore, by giving the same group number to satellites communicating with different terminals, the communication can take place simultaneously.</p>
<b>Polling</b>	<p>Enables the terminals to communicate. To receive data, either transmission between the host computer and the satellite (pc - sat) or between the satellite and the terminals (sat - term) must be initiated.</p>
<b>Download</b>	<p>Allows you to define how the terminal should download data onto the host computer - one record at a time (1 record) or all the data contained in it (total).</p>

**EXAMPLE**

The following example shows, in a simplified form, how the Group command can be usefully carried out.



In the example, the coverage range of each satellite intersects that of the other three. To not create transmission problems, different Group numbers are given to adjacent satellites. As a result, when a data transmission takes place between satellites 101 and 105 (which are not adjacent) and the respective terminals, the transmission between satellites 102, 103, 104 and 106 is disabled; it is then enabled from 102 and 105 and disabled from the other satellites, and so on.

**□ File**

It allows to define the way to save the data received through each satellite.

- Name** enables you to define the filename.
- Path** enables you to define the folder where the file will be saved.
- Append** allows you to add new data to the file without having to remove the data already saved.
- Overwrite** allows you to add new data to the file, replacing the data already saved.
- Separat.** allows you to choose the character which will be used to separate the different fields of each record saved to a file. It is possible to insert the character directly or through its corresponding ASCII codification.

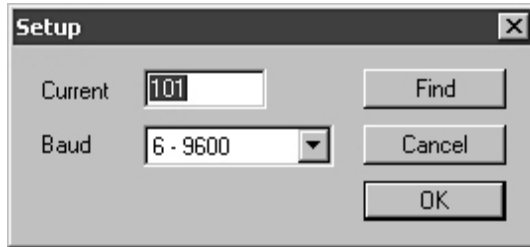
**ATTENTION**

*If the "Name" file is null, no file is generated and the data must be generated using the DataEvent event.*



Setup

When this button is selected the following dialog box will appear, requiring you to insert the confirm the transmission parameters defined in the General Properties file.



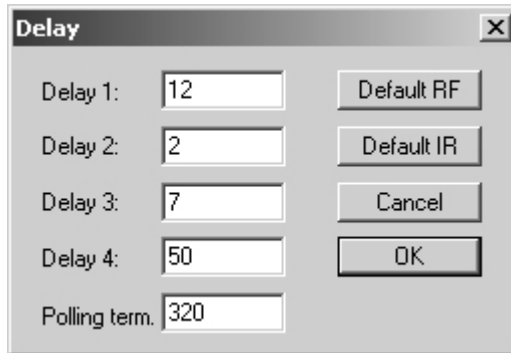
Find

If the same properties have to be given to several satellites, this button initiates a sequential search for an identifying number from 101 to 199. The current properties will automatically be attributed to the identifier which searched. Due to the long time required to complete this operation it is advised to use this function only when the satellite address is unknown.



Delay

When this button is selected, the following dialog box will appear:



The dialog box titled "Delay" contains the following fields and buttons:

Delay 1:	<input type="text" value="12"/>	<input type="button" value="Default RF"/>
Delay 2:	<input type="text" value="2"/>	<input type="button" value="Default IR"/>
Delay 3:	<input type="text" value="7"/>	<input type="button" value="Cancel"/>
Delay 4:	<input type="text" value="50"/>	<input type="button" value="OK"/>
Polling term:	<input type="text" value="320"/>	

The window lists the hardware parameters preset by the radio frequency satellite (preset as RF) and by the infrared satellite (preset as IR).

**ATTENTION**

*The values of these parameters can only be altered on special instruction from Datalogic or the distributor.*

# Formula<sup>®</sup>

## B PROGRAMMING EXAMPLES

---

### B.1 LOADING THE EXAMPLES

The examples described below are copied into the appropriate directory during the installation of EasyGen<sup>™</sup>.

They show how OLE for Formula can be employed in the most commonly used environments such as Visual Basic 6.0 and in the office applications Access 97/2000 and Excel 97/2000.

To use these examples, your terminal must be loaded with Example 1 (Code+Quantity) from EasyGen<sup>™</sup>.

To load the examples you can double-click on the file icon displayed in Windows Explorer (you must check that the correct environment has been installed).

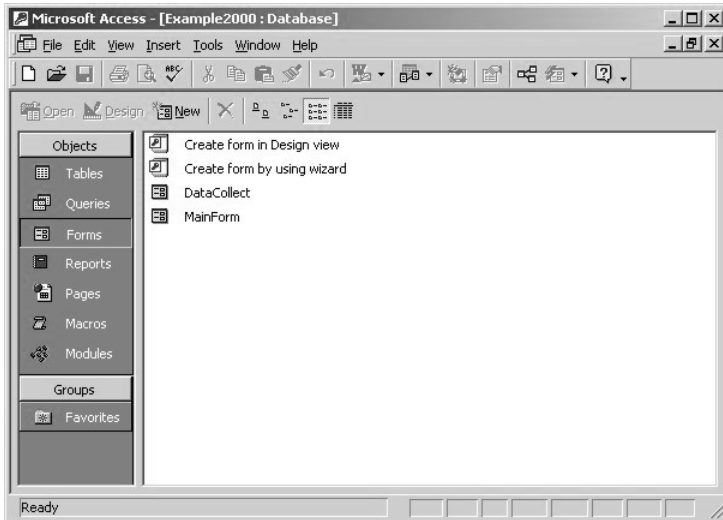
To load data from the terminal by using these examples you must:

- set the DataEventEnabled property to "True";
- tick the "1 record" check box in the Download option of the Terminal and Satellites properties (see par. A.1.2 and par. A.1.3).

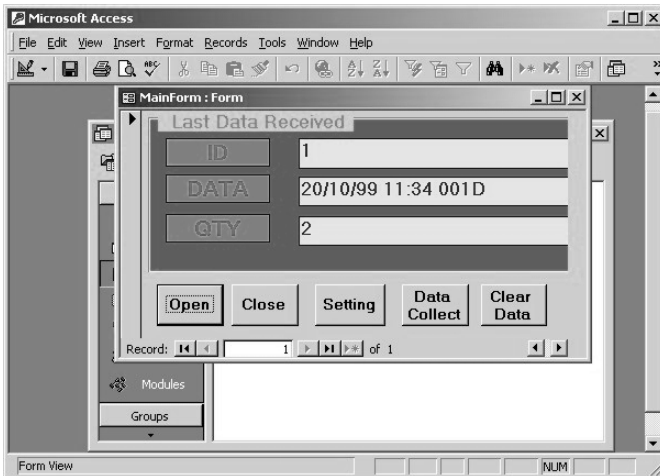
### ❑ “ACCESS” Directory - “example.mdb” Installation file

For this example ACCESS 97/2000 must be installed.

After loading the EXAMPLE.MDB file, you can use all the ACCESS 97/2000 commands as described in the relevant manual. The screen below appears:



To display the application simply press the <Open> button and the screen below appears:





A description of the buttons functions is shown below:



This button initiates transmission with all devices configured for polling (download 1 record);



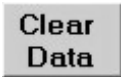
This button interrupts transmission.



This button starts up the devices' configuration and applies the Action Method, as described in the section entitled "Action Method" on page 59;



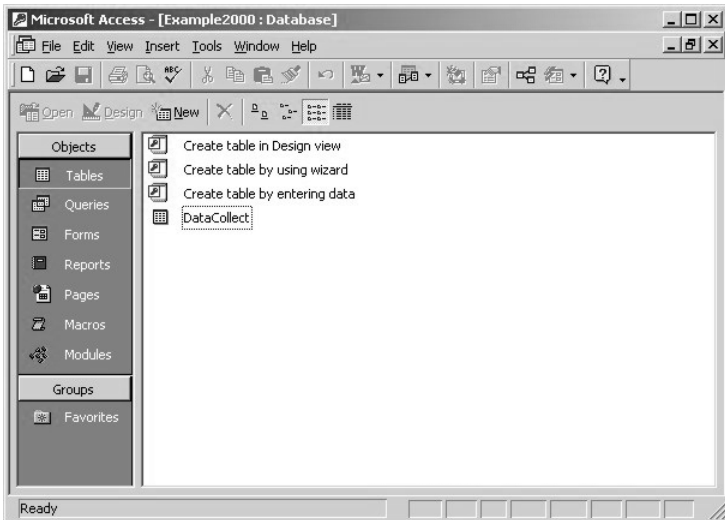
This button opens a new dialog box showing a list of all the downloaded data.



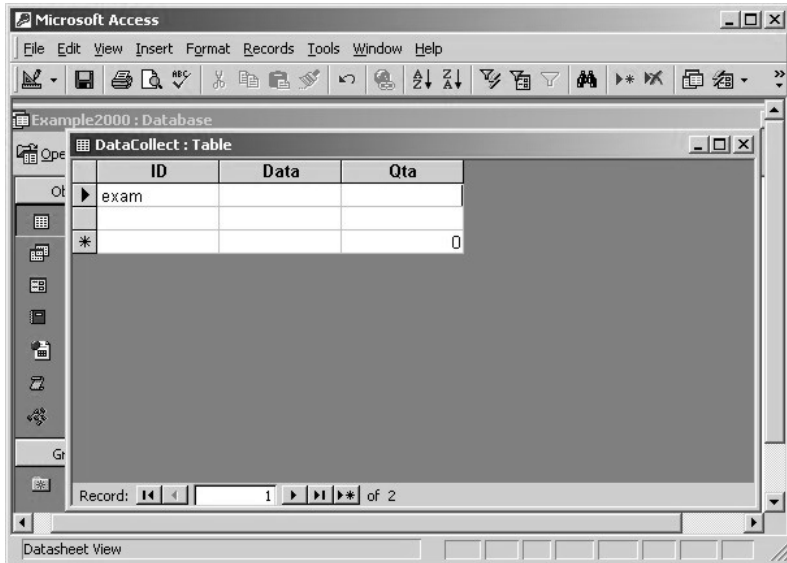
This button clears the table containing the data.

After pressing the <Open> button, you can download data at any time. Any data downloaded is saved in the "Data Collect" table, while in the "Last Data Received" dialog box you can see each data coming from the .OCX file.

Once the transmission has been interrupted by pressing the <Close> button, you can see the data ordered in a table by selecting the "Tables" item. The screen below appears:



Now you must press the <Open> button. The screen below appears:

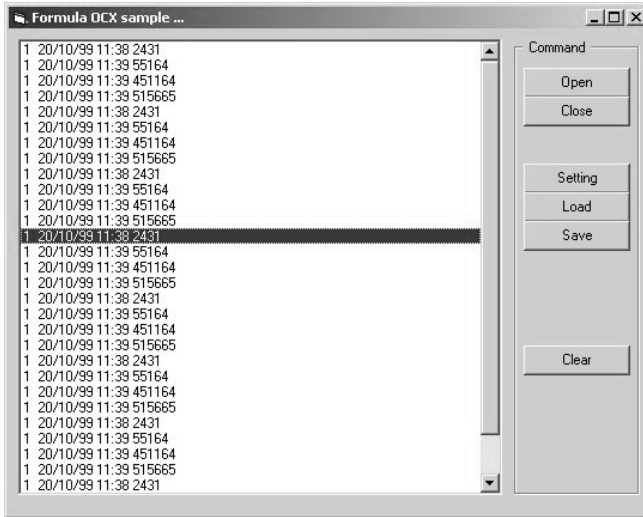


The table will contain all the downloaded data.

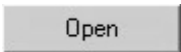
### □ “Visual Basic” Directory - “Sample.EXE” Installation File

For this example Visual Basic 6.0 must be installed.

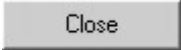
Once the installation file has loaded the dialog box below appears:



A description of the buttons in this dialog box is shown below:



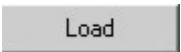
This button initiates transmission with all devices configured for polling (download 1 record);



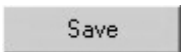
This button interrupts transmission.



This button starts up the devices configuration and applies the Action Method, as described in the section entitled “Action Method” on page 59;



This button loads the Control configuration from the Formula3.cfg file.



This button saves the Control configuration in the Formula3.cfg file, which is stored in the same directory containing the .EXE file.



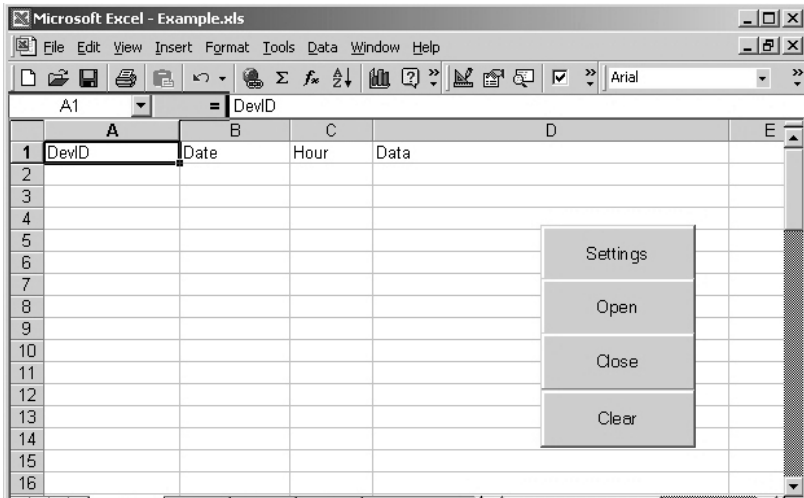
This button clears the list of data appearing in the list box.

After pressing the <Open> button you can download data from the terminal’s memory at any time. The downloaded data is displayed, in real time, in the white area to the left of the buttons.

❑ **“Excel” Directory - “Example.XLS” Installation file**

For this example Excel 97/2000 must be installed.

Once the installation file is loaded, you can use all the EXCEL 97/2000 commands as described in the relevant manual. The dialog box below appears:



A description of the buttons in this dialog box is shown below:

Settings	This button starts up the devices configuration and applies the Action Method, as described in the section entitled “Action Method” on page 59;
Open	This button initiates transmission with all devices configured for polling (download 1 record);
Close	This button interrupts transmission.
Clear	This button clears the table.

After pressing the <Open> button you can download data from the terminal's memory at any time. The downloaded data is displayed, in real time, in the white area of the sheet.

# Formula<sup>®</sup>

## C COMMUNICATION PROTOCOL

---

### C.1 COMMUNICATION BETWEEN FORMULA TERMINALS AND HOST COMPUTER

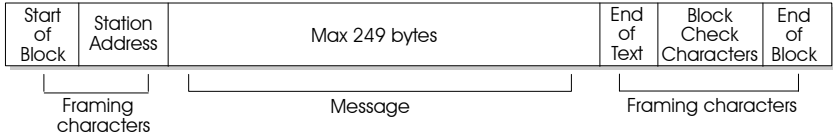
Communication between OLE for Formula and the terminals inserted in the cradle and/or satellites always takes place on an asynchronous RS-232 serial line which for compatibility between all connectable systems is set to 9600 bit/sec, 8 bits and Mark parity as shown in the table:

Baud Rate:	9600 baud (default) 19.200 baud 38.400 baud
Parity:	Mark
Data Bits:	8
Stop Bit:	1

The communication protocol employed is fixed and is as described in the terminals and satellite manuals. It is summarized here for convenience.

For more specific use, please refer to the manual of the product concerned.

Communication takes place by means of communication frames - see the figure on the following page - which are either confirmed by the single character ACK or not confirmed by the single character NAK when the Frame is received correctly. Otherwise no character is sent.



**Communication Frame**

Framing characters	<b>Start of Block:</b>	length	1 byte
		Value	02Hex (ASCII: STX)
Framing characters	<b>Station Address:</b>	length	1 byte
		Values	from 100 to 199 decimal
Message	Field containing data, commands or messages from satellite (maximum size 249 bytes); a data compress/decompress algorithm is used. You will find an example of data compression/decompression on paragraph "Compression and Decompression of Data" on page 107.		
Framing characters	End of Text	Length	1 byte
		Value	03Hex (ASCII: ETX)
	Block Check Characters	Size two bytes (the first byte contains the upper part of the checksum, the second byte the lower part); converted the corresponding ASCII values.	
Framing characters	End of Block	Length	1 byte
		Value	0DHex (ASCII: CR)

#### □ Calculation of checksum 256

To calculate the checksum 256 the sum of all the bytes of the frame on one byte by starting from the Start of Block byte up to End of Text byte included.

The result of the sum is logic ANDed with the value 0FFHex.

From this value the least significant nibble is extracted and converted to the equivalent ASCII character.

The same operation is performed with the most significant nibble.

#### EXAMPLE

*You calculate the checksum:*

$$2+101+27+53+27+3 = 213 = 0D5Hex$$

*The result of the sum is logic ANDed with the value 0FFHex:*

$$0D5Hex \text{ AND } 0FFHex = 0D5Hex$$

*In this case you will obtain 0D5Hex again.*

From this value the least significant nibble is extracted and converted to the equivalent ASCII character:

$$05\text{Hex} + 30\text{Hex} = 35\text{Hex} = '5' \text{ (second byte)}$$

The same operation is performed with the most significant nibble:

$$0D\text{Hex} + 37\text{Hex} = 44\text{Hex} = 'D' \text{ (first byte)}$$

## C.2 MESSAGE

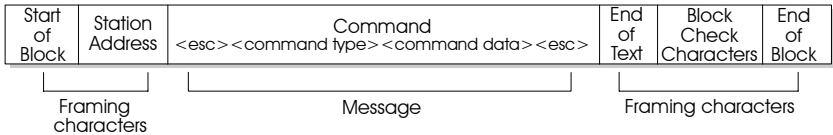
The message has different forms depending on the context in which it is used: we have “commands” from the OLE to the terminals or satellites, “data received”, and “replies”.

These messages must be sent with the DirectIO method (see page 65) and are received with the DirectIOText property (see page 33).

### C.2.1 Message “Commands”

To send a command to the terminals or satellites, OLE uses a message according to the figure below.

The commands are groups of characters whose beginning and end are marked by the ASCII character 1B Hex <esc>. The general format of the command inserted in the communication frame is the following:



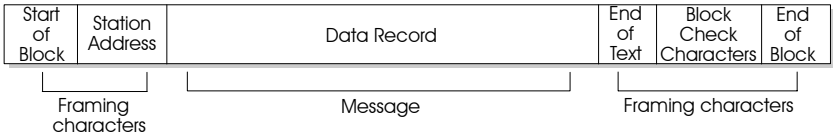
*Command type:*

ASCII CHARACTER	DESCRIPTION
0	Download data
1	Delete
2	Load record
3	Activate RF Polling
4	Deactivate RF polling
5	State request
7	Display Message
8	Terminal information
9	Open/Close File
B	Start application
C	Closing of a Data File
D	Cancellation of a data file sent by the host computer
F	Definition of attributes of the file fields
L	Satellite Programming
M	Operation Mode
O	Opening a data file
R	File size definition
S	Data file state
Z	Clock tasks
W	Loading a file with only one record

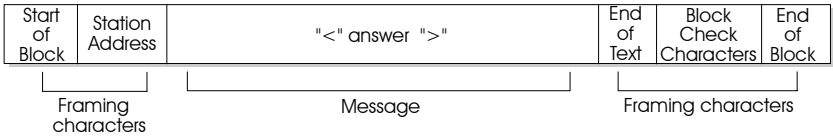
*Command data:* This command has alphanumeric values according to the instruction sent to the terminal.



### C.2.2 Message “Data Received”



### C.2.3 Message “Replies”



#### Message Explanations

##### □ 0 - Download data

Use this command when you want to download a data file or program file.

*Command data* can take the following values:

- filename* Requests all records of specified file, if it exists. The file name must always consist of 8 characters;
- \** Requests all records of the existing files;
- +* Requests download of current record, record pointer is not updated;
- +nnnnnn* Requests download of *nnnnnn* records beginning with the current one. The record pointer is not updated;
- #* Requests zero-setting of downloaded indices;

The answers are:

- “EOF” End of file.
- “FNF” File not found.
- “EOT” End of transmission.
- “ACK” Reset executed.

The data that arrive are in compressed format according to the specifications described in the paragraph "Compression and Decompression of Data" on page 107.

**□ 1 - Delete**

This command deletes/clears one or more data or program files.

*Command data* can take the following values:

<i>filename</i>	The specified file is deleted, whether it is a data file or a program file;
*	All output file records are deleted;
-	Deletes the current record and moves the pointer by one position.

The answers are:

"DEL"	Records deleted.
"CLR"	Records deleted.
"WAI"	Wait, deleting.

**□ 2 - Load record**

This command is used to send a record to the terminal or the satellite.

In the case of the terminal, it is followed by the record; in the case of the satellite, it is necessary to add as a trailer to the end of record (Dest. Terminal Addr.) OR 80Hex.

The responses to these commands are "Reply" messages with an answer equal to the value:

"ACK"	Record stored.
"SYS"	Record does not conform to definition or the file is full.
"FULL"	If the satellite's memory is full.
"OPEN"	If the file on the satellite must still be open. In this case the data are not accepted.

This command is repeated until the data file is completely transferred.

### □ 3 - Activate RF polling

This command refers to the satellite only and permits activation of polling of the terminals.

If it is not followed by the "\$" character, record this operating mode in the EEPROM.

If the command data is present, it is made up of 2 bytes, the first with the minimum address of the terminals on which to carry out polling, the second with the maximum address in the following format: (First Dest.Term. Address) OR 80 Hex (Last Dest.Term. Address) OR 80 Hex .

In response to these commands the satellite sends "Reply" messages with an answer equal to the value:

"ACK"                      Operation executed.

### □ 4 - Deactivate RF polling

This command refers to the satellite only and permits deactivation of polling of the terminals.

If it is not followed by the "\$" character, record this operating mode in the EEPROM.

In response to these commands the satellite sends "Reply" messages with an answer equal to the value:

"ACK"                      Polling disabled.

"SYS"                      Data Transfer in progress: the deactivation of polling will take place automatically as soon as the current transaction is finished.

### □ 5 - State request

Every time that the terminal or satellite receives this command, it sends the host computer a message that defines its internal state.

There is no command data.

In response to these commands the satellite sends "Reply" messages with an answer equal to the value:

"RUN"                      Fully operational;

"STOP"                     Satellite with suspended RF polling;

"RAM"                     Memory error, satellite blocked;

"EEPROM"                 Satellite not programmed and blocked;

"REE"                     General error; satellite blocked.

### □ 7 - Display message

This command refers to the terminal only and permits the host computer to send, at any moment, a text to be displayed.

The data command takes the value of the text to be sent, and is made up of ASCII characters included in the range between ASCII 20<sub>16</sub> and ASCII 7E<sub>16</sub>. The ASCII character 07<sub>16</sub> (BEL) is translated into an acoustic signal.

The responses to this command are “reply” messages with an answer equal to the value:

“BSY”	Terminal busy.
“WAI”	Terminal with active scrolling.
“ACK”	Display carried out.

### □ 8 - Terminal information

This command allows the host computer to identify the type of Formula terminal.

*Command data* can take the following values:

no value	Formula terminal transmits the name of the Terminal in use.
\$	Formula terminal sends version and firmware release to the host computer in the format v.X/r.Y
*	Formula terminal sends the RAM extension in the format XXX Kbyte.

The answers are:

<i>name</i>	Name of the terminal.
v.X / r.Y	X is the version and Y is the release of the terminal in use.
XXX	RAM memory of the terminal.
“NAK”	Wrong command.

**❑ 9 - Open/Close File**

This command allows the host computer to open or close any existing data file.

*Command data* can take the following values:

<i>filename</i>	Formula terminal opens the specified file, if it exists;
\$	Formula terminal sends the name of the currently open file to the host computer.

The answers are:

"FNF"	File does not exist.
"RUN"	Filename opened or closed.
None	Name of the opened file.

**❑ B - Start application**

This command allows the host computer to send the application program start command.

*Command data* does not take up any value.

**❑ C - Closing of a Data File**

This command refers to the satellite and permits a previously opened data file to be closed.

The command data must have the address of the terminal in the form (Dest. Terminal Addr.) OR 80Hex.

The replies to these commands are messages "Reply" with answers that have the value:

"ACK"	Operation executed.
-------	---------------------

### ❑ D - Cancellation of a data file sent by the host computer

This command concerns the satellite and allows a previously sent file data to be cancelled.

The command data must have the address of the terminal in the form (Dest. Terminal Addr.) OR 80Hex.

The replies to these commands are messages "Reply" with answers that have the value:

"SYS"	File in use. Cancellation not possible.
"DEL"	Deletion completed.
"ACK"	File non present: can be already been transmitted, and therefore cancelled by the Satellite, or never loaded
"NAK"	Wrong address.

### ❑ F - Definition of attributes of the file fields

This command enables the terminal to acquire the attributes of each field of the record to transfer from the host computer.

The command data is made up of a series of fields:

"Field Number, Field Name, 'Length', 'Filling Character', 'Lower character limit check', 'Upper character limit check', 'Alignment' "

Having the following values:

Field number :	from 0 to 99 (50 for 730/734), two characters aligned right with zero fillings. For 730/734 from 0 to 49
Field name:	10 ASCII characters ( from 20H to z except for  ), aligned flush left with blank fillings
Length:	from 1 to 250 (a number that depends on the protocol because also the characters of the transmission's Header and Trailer are considered), 3 characters aligned flush left with blank fillings
Filling Character:	Blank20H* Zero30H
Lower control character limit	[20H to 7AH (from Blank to z)]
Upper control character limit	[20H to 7AH (from Blank to z)]
Alignment:	Right30H* Left31H



### □ M - Operation Mode

This command concerns the satellite and permits it to define the operation mode. The command data with Value "0" makes the Satellite work with the Sysnet protocol. The command data with Value "1" makes the Satellite work directly i.e. with automatic transmission to the host computer of the data arriving from the RF network. Must be used only with an RS-232 interface.



**ATTENTION**

*This command is not compatible with OLE for Formula.*

The replies to these commands are messages "Reply" with answers that have the value:

"ACK"            Configuration executed

### □ O - Opening a data file

This command concerns the satellite and allows a data file to be opened. The command data must have the address of the terminal in the form (Dest. Terminal Addr.) OR 80Hex.

The replies to these commands are messages "Reply" with answers that have the value:

"ACK"            If the File has been opened successfully

"OPENED"        If the File is already open or in use by the RF manager



### □ R - File size definition

This command enables the terminal to define the dimensions of the file. The command data is made up of a series of fields:

```
" 'Table type' 'Table name' 'Table length' Number of records'
'Number of fields' 'Table number' "
```

Having the following values:

Table type:	Input/output table with a fixed 33H value
Table name:	8 Dos-format characters aligned left with blank fillings
Table length:	Number of records
*	Record length (to calculate), seven characters aligned flush right with zero fillings
Number of records:	from 1 to a maximum defined by the program or table, 7 characters aligned right with zero fillings
Number of fields:	from 0 to 99, two characters aligned right with zero fillings, the sum of the lengths of all the fields must be no greater than 250, (a number that depends on the protocol because also the transmission Header and Trailer characters are considered). For 730/734 the number of fields is from 0 to 49.
Table number:	Progressive from 0 (30H) to 14 (3EH).

The replies to these commands are messages "Reply" with answers that have the value:

"ACK"	Operation executed
"NAK"	Wrong format

### □ S - Data file state

This command concerns the satellite and allows the state of the file to be known. The command data must have the address of the terminal in the form (Dest. Terminal Addr.) OR 80Hex.

The replies to these commands are messages "Reply" with answers that have the value:

"ACK"	File present in memory.
"FNF"	File not found.
"NAK"	Wrong address.

### □ W - Loading a file with only one record

This command concerns the satellite and allows a file made up of only 1 record to be loaded. This command allows a field to be sent directly without requiring the data file to be opened and closed before and afterward.

The command data must have the address of the terminal in the form (Dest. Terminal Addr.) OR 80Hex followed by the record having a maximum length of 245 bytes.

The replies to these commands are messages "Reply" with answers that have the value:

"OPENED"	If data already exist in the Sysnet memory which are destined to that hand-held terminal.
"FULL"	If the memory is complete.
"ACK"	If The data have been correctly saved.

### □ Z - Clock tasks

This command allows the host computer to carry out special tasks on the internal clock of the Formula terminal.

Command data can take the following values:

yymmddhhmmss	date and time setting;
?	request date and time of Formula terminal.

The answers are:

"ACK"	Operation ready.
"NAK"	Wrong format.
Yymmddhhmmss	Year, month, day, time, minute, second.

### C.4 COMPRESSION AND DECOMPRESSION OF DATA

The data from the Formula terminal are compressed. They must be decompressed by the host computer.

The Formula terminals are enabled for decompress the compressed data from the satellite SAT/RF.

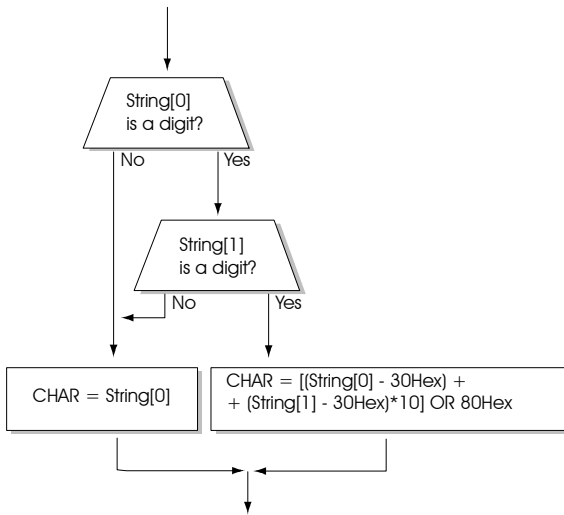
The compression is carried out exclusively on numeric characters which are those usually used with bar codes (characters from value 30Hex (ASCII: 0) to 39Hex (ASCII: 9)).

In one byte you can insert two consecutive numeric characters. If you are transmitting only numeric characters, you will reach a maximum compression ratio of 2:1.

#### □ Compression

Below is shown the flow chart of the data compression algorithm.

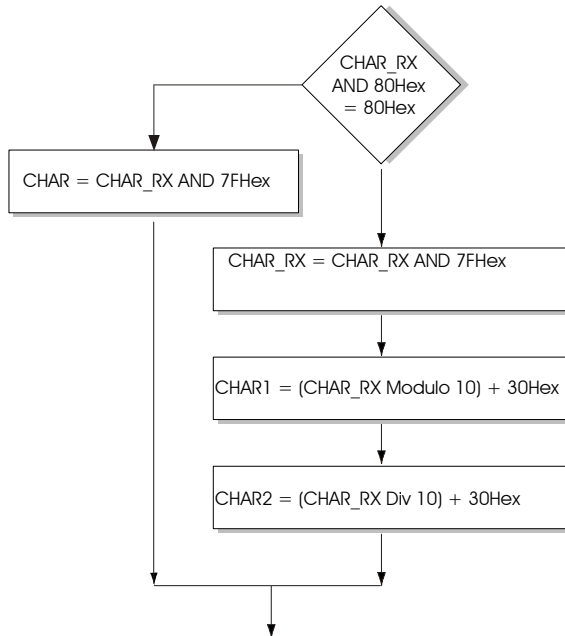
The "String" parameter is a string containing the numerical values to be compressed:



### □ Decompression

Char\_Rx = Character to decompress  
 Char1 = First character decompressed  
 Char2 = Second character decompressed  
 Char = Non compressed character

The following diagram illustrates the decompression procedure.



### EXAMPLE

Suppose we wish to have received the character "0E3Hex".

0E3Hex AND 80Hex	= 80Hex	= one compressed byte
E3Hex AND 7Fhex	= 63Hex	= 99 decimal
99 MOD 10	=	9
9 + 30Hex	= 39Hex	= '9' first character
99 DIV 10	=	9
9 + 30Hex	= 39Hex	= '9' second character