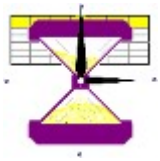
iCalendar OCX – a 32 bit ActiveX Date & Time Control

Features:

- **Point and Click Calendar, Date and Time Interface**
- **Comprehensive Date and Time Manipulation**
- **Date and Time arithmetic functions**
- **Web Ready**
- **Royalty FREE unlimited runtime distribution**



CompleteControl™

What is iCalendar

More About iCalendar

iCalendar Reference

System Requirements

Properties

Methods

Sample Applications

Examples

Contact Information

How to contact us

Obtaining Technical Support

Registration

Registering your purchase

Licensing and Distribution

License Agreement

Distributing iCalendar

Un-Installing iCalendar

iCalendar is a 32 bit ActiveX Control that provides an easy to use interface for date / time entry, a monthly calendar, comprehensive date / time manipulation and arithmetic.

iCalendar supports properties and methods to enable you to …

- Perform Date and Time calculations
- Display a monthly calendar for any year (from 0 to 9999)
- Provide fully validated Date and Time data entry fields in your application
- Convert Date and Time strings to practically any format

iCalendar is just one of many software components from our **CompleteControl** ™ range of products.

iCalendar is supported under the following 32 bit Microsoft® Windows operating systems on Intel® architecture machines.

Windows 95

Windows NT Server and Workstation 4.0

and future versions of these operating systems.

Note:

The current version of iCalendar is not supported on beta versions of the above operating systems

The current version of iCalendar is not supported on win32s.

The current version of iCalendar is not supported on non-Intel® architecture implementations of Microsoft® Windows.

For information about future versions and updates please register with Imagine IT Ltd

To register your copy of iCalendar please print out this page and send it to us after completing the details.
Alternatively you can also email the required information to us at: registration@imagineit.co.uk

Registered users automatically receive information about updates, new releases and future products from Imagine IT and are eligible for technical support.

Product: iCalendar.OCX 2.0

Last Name:

First Name:

Company Name:

Address:

Telephone Number:

Fax Number:

E-Mail Address:

Web URL:

Date of Purchase:

Purchased From:

---

please send to:

Imagine IT Limited, 3rd Floor, Hygeia Building, 66 College Road, Harrow HA1 1BE, United Kingdom

or email to:

registration@imagineit.co.uk

iCalendar is designed and developed by Imagine IT Limited.

Imagine IT specialises in object technologies and component software development.

Apart from creating great components we also help customers develop their own line-of-business objects and applications.

Contact us for more information about our products and services :

Write to us at:

Imagine IT Limited
3rd Floor,
Hygeia Building
66 College Road
Harrow, HA1 1FD
United Kingdom

Telephone:

from within the United Kingdom:	0181 324 1240
from other countries:	international access code + 44 + 181 324 1240
from other countries:	international access code + 44 + 181 324 1752

Fax:

from within the United Kingdom:	0181 324 1752
from other countries:	international access code + 44 + 181 324 1752

Email:

for general inquiries:	info@imagineit.co.uk
for technical support inquiries:	suport@imagineit.co.uk
for registration:	registration@imagineit.co.uk

World Wide Web:

find us at:	www.imagineit.co.uk

For the fastest possible response please direct all technical support enquiries via email to support@imagineit.co.uk.

Or fax your enquiry marked for the attention of “Technical Support”.

If you do not have access to email or fax then please call us at the above telephone number and ask for “Technical Support”

NOTE: We can only provide technical support to registered users.

IMPORTANT

THE USE OF THIS SOFTWARE IS SUBJECT TO THE TERMS OF THE LICENSE AGREEMENT PRINTED BELOW. PLEASE READ THE LICENSE AGREEMENT CAREFULLY. IF YOU DO NOT AGREE WITH ALL OF THE TERMS, THEN YOU SHOULD NOT USE THE SOFTWARE IN ANY WAY WHATSOEVER INCLUDING COPYING OR DISTRIBUTING THE SOFTWARE OR ANY PART OF THE SOFTWARE TO ANY THIRD PARTY. IF YOU USE THE SOFTWARE IN ANY WAY WHATSOEVER IT WILL BE DEEMED TO INDICATE YOUR ACCEPTANCE OF ALL THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT.

THE AGREEMENT

THIS DOCUMENT IS A LEGAL AGREEMENT BETWEEN IMAGINE IT LIMITED, UNITED KINGDOM, HEREAFTER REFERRED TO AS "IMAGINE IT" AND THE USER OF THE SOFTWARE HEREIN REFERRED TO AS "LICENSEE"

DEFINITIONS

"SOFTWARE" MEANS THE SOFTWARE KNOWN AS "iCalendar OCX" AND ALL ITS COMPONENTS, PARTS AND DOCUMENTATION, WHICH HAVE BEEN DEVELOPED BY IMAGINE IT

"END USER APPLICATION" MEANS ANY APPLICATION DEVELOPED WITH THE HELP OF OR BY USING THE SOFTWARE

PURPOSE

THE PURPOSE OF THIS AGREEMENT IS TO DEFINE THE RELATIONSHIP BETWEEN IMAGINE IT AND LICENSEE, THE TERMS, CONDITIONS, AND LIMITATIONS OF USE, THE RIGHTS OF BOTH PARTIES AND THE LIMITATION OF LIABILITY ARISING FROM THE USE OF THE SOFTWARE

EFFECTIVE DATE

THIS AGREEMENT IS DEEMED TO HAVE BEEN MADE EFFECTIVE ON THE FIRST DATE AT WHICH LICENSEE OPENS THE PACKAGE CONTAINING THE SOFTWARE AND ITS ASSOCIATED DOCUMENTATION, OR IF THE SOFTWARE IS DISTRIBUTED ELECTRONICALLY, THE FIRST DATE ON WHICH THE LICENSEE OBTAINS THE SOFTWARE

GRANT OF LIMITED LICENSE

I. EVALUATION LICENSE

IF THE LICENSEE HAS NOT PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN IMAGINE IT GRANTS THE LICENSEE A NON EXCLUSIVE, NON TRANSFERABLE, PERSONAL LICENSE TO USE THE SOFTWARE AND ITS DOCUMENTATION FOR EVALUATION PURPOSES ONLY AND FOR NO OTHER PURPOSE. THE LICENSE DOES NOT PERMIT THE DEVELOPMENT AND DISTRIBUTION OF ANY END USER APPLICATIONS USING THE SOFTWARE

II. FULL DEVELOPMENT LICENSE

IF THE LICENSEE HAS PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN IMAGINE IT GRANTS THE LICENSEE A NON EXCLUSIVE, NON TRANSFERABLE, PERSONAL LICENSE TO USE THE SOFTWARE AND ITS DOCUMENTATION FOR THE PURPOSE OF DEVELOPING AND DISTRIBUTING ANY NUMBER OF END USER APPLICATIONS AND TO COPY AND DISTRIBUTE ANY PARTS OF THE SOFTWARE DEFINED BELOW UNDER REDISTRIBUTABLE COMPONENTS TOGETHER WITH AND AS PART OF THE END USER APPLICATION, SUBJECT TO ALL THE TERMS AND CONDITIONS AND RESTRICTIONS IN THIS AGREEMENT.

RESTRICTIONS

A USER OF ANY END USER APPLICATIONS MAY NOT FURTHER USE PARTS OF THE SOFTWARE FOR SOFTWARE DEVELOPMENT, COPYING OR DISTRIBUTION. THE LICENSEE MUST ENFORCE THIS IN A SEPARATE AGREEMENT WITH THE USER OF ANY END USER APPLICATIONS.

THE LICENSEE MAY USE ONLY ONE COPY OF THE SOFTWARE AT ANY TIME ON ONE IBM PC OR COMPATIBLE SYSTEM.

ADDITIONAL PRODUCT KEYS FOR THE SOFTWARE MUST BE PURCHASED IF IT IS REQUIRED TO BE USED IN A MULTI USER NETWORKED ENVIRONMENT IN QUANTITIES OF ONE KEY FOR EACH PERSON HAVING ACCESS TO AND USING THE SOFTWARE.

THE LICENSEE MAY MAKE A SINGLE BACKUP COPY OF THE SOFTWARE.

THE LICENSEE MAY NOT USE, COPY, MODIFY, REVERSE ENGINEER, DISASSEMBLE, SELL, TRANSFER, HIRE, LEND OR OTHERWISE DISTRIBUTE THE SOFTWARE OR ANY OF ITS DOCUMENTATION OR COMPONENTS IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS AGREEMENT.

THE END USER APPLICATION MUST NOT BE ANOTHER DEVELOPMENT TOOL DESIGNED TO BE USED FOR CREATING OTHER END USER APPLICATIONS USING THE SOFTWARE. THE LICENSEE MUST ENFORCE THIS IN A SEPARATE AGREEMENT WITH THE USER OF THE END USER APPLICATION.

THE LICENSEE HAS NO RIGHTS TO THE ORIGINAL SOURCE CODE OF THE SOFTWARE WHATSOEVER.

THE LICENSEE MUST INFORM IMAGINE IT OF THE NATURE OF EACH AND EVERY NEW END USER APPLICATION IT DEVELOPS WITH THE USE OF THE SOFTWARE.

THE LICENSEE SHALL COMPLY WITH ALL LOCAL AND INTERNATIONAL LAWS AND EXPORT / IMPORT REGULATIONS WHEN DISTRIBUTING PARTS OF THIS SOFTWARE AS PROVIDED FOR BELOW ALONG WITH ANY END USER APPLICATIONS

REDISTRIBUTABLE COMPONENTS

THE SOFTWARE, BY ITS NATURE, INCLUDES COMPONENTS WHICH ARE REQUIRED TO BE DISTRIBUTED WITH ANY END USER APPLICATIONS IN ORDER FOR THAT END USER APPLICATION TO FUNCTION.

I. EVALUATION LICENSE

IF THE LICENSEE HAS NOT PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THEN THERE ARE NO DISTRIBUTABLE COMPONENTS. THE LICENSEE SHALL NOT DISTRIBUTE ANY COMPONENTS OF THE SOFTWARE

II. FULL DEVELOPMENT LICENSE

IF THE LICENSEE HAS PURCHASED A FULL PRODUCT KEY FOR THE SOFTWARE THENTHE FOLLOWING FILES DEFINE THE COMPONENTS WHICH THE LICENSEE MAY DISTRIBUTE PROVIDED THAT THEY ARE DISTRIBUTED AS PART OF THE END USER APPLICATION AND PROVIDED THAT THEY ARE DISTRIBUTED WITHOUT ANY MODIFICATION AND ARE COPIED FROM THE ORIGINAL DISKS OR CDS. ALL FILES LISTED BELOW MUST BE DISTRIBUTED WITH THE END USER APPLICATION.

1. **iCalendar.OCX**
2. **iCalendar.RTL**

NO OTHER FILES SUPPLIED WITH THE SOFTWARE MAY BE DISTRIBUTED WITH THE END USER APPLICATION.

THE SOFTWARE BY ITS NATURE ALSO REQUIRES SOME FILES WHICH ARE DISTRIBUTED BY MICROSOFT(R) TO BE PRESENT ON THE END USER'S COMPUTER FOR THE END USER APPLICATION TO FUNCTION. THIS LICENSE AGREEMENT DOES NOT IN ANY WAY PROVIDE FOR LICENSEE TO DISTRIBUTE THESE COMPONENTS WITH THE END USER AGREEMENT. THE LICENSEE MUST VERIFY THEIR RIGHTS TO DISTRIBUTE THESE COMPONENTS, SEPARATELY WITH MICROSOFT(R) BEFORE DISTRIBUTING THESE COMPONENTS WITH THE END USER APPLICATION

1. MFC42.DLL

2. OLEPRO32.DLL

3. REGSVR32.EXE

TECHNICAL SUPPORT

ONLY THOSE LICENSEES WHO HAVE PURCHASED A FULL PRODUCT KEY AND REGISTERED WITH IMAGINE IT BY FULLY COMPLETING AND RETURNING ALL THE INFORMATION ASKED FOR IN THE REGISTRATION FORM OR RELEVANT SECTION IN THE ON-LINE HELP DOCUMENTATION ARE QUALIFIED FOR RECEIVING TECHNICAL SUPPORT HELP ON THE SOFTWARE.

TECHNICAL SUPPORT IS RESTRICTED TO ANSWERING QUESTIONS ABOUT THE USE OF THE SOFTWARE OR ERRORS IN THE SOFTWARE IF ANY.

TECHNICAL SUPPORT DOES NOT INCLUDE DEBUGGING AND OTHER PROBLEM SOLVING TASKS ASSOCIATED WITH THE DEVELOPMENT OF ANY END USER APPLICATIONS.

USERS OF THE END USER APPLICATION ARE NOT ELIGIBLE FOR OBTAINING TECHNICAL SUPPORT FROM IMAGINE IT UNDER ANY CIRCUMSTANCES.

THIS SERVICE WILL BE PROVIDED ACCORDING TO IMAGINE IT'S TECHNICAL SUPPORT POLICIES FROM TIME TO TIME, AND IS LIMITED TO SUCH TIME AS IMAGINE IT SUPPORTS THE SOFTWARE OR OWNS THE SOFTWARE.

TECHNICAL SUPPORT MAY BE PROVIDED BY MEANS OF E-MAIL, FAX, BULLETIN BOARDS, ON-LINE SERVICES, WRITTEN CORRESPONDENCE OR TELEPHONE AT IMAGINE IT'S SOLE DISCRETION.

OWNERSHIP

IMAGINE IT SHALL REMAIN THE OWNERS OF ALL RIGHTS, TITLE AND INTEREST IN THE SOFTWARE

THIS LICENSE DOES NOT CONFER ANY OWNERSHIP RIGHTS TO THE LICENSEE

THE LICENSEE AGREES NOT TO USE IMAGINE IT'S NAME IN ANY WAY OR FORM

THE LICENSEE AGREES NOT TO IMPLY THAT ANY END USER APPLICATIONS IT DEVELOPS ARE EITHER APPROVED OR OTHERWISE SANCTIONED BY IMAGINE IT

THE LICENSEE AGREES TO CARRY A COPYRIGHT NOTICE IN THE ABOUT BOX OR HELP FILE OF EACH AND EVERY COPY OF ANY END USER APPLICATIONS IT DEVELOPS WITH THE HELP OF THIS SOFTWARE, WITH THE FOLLOWING WORDS :-

"PORTIONS OF THIS APPLICATION PROVIDED BY IMAGINE IT LIMITED, UNITED KINGDOM, COPYRIGHT (C) 1996-1997 IMAGINE IT LIMITED. ALL RIGHTS RESERVED"

TERMINATION

THIS AGREEMENT SHALL BE TERMINATED IN WRITING BY IMAGINE IT AT ANY TIME IF THE LICENSEE BREACHES ANY TERMS AND CONDITIONS CONTAINED IN THIS AGREEMENT OR CARRIES OUT ANY ACTIONS EXPRESSLY PROHIBITED IN THIS AGREEMENT, AND CONTINUES TO BE IN BREACH FOR 30 DAYS AFTER WRITTEN NOTICE IS GIVEN TO THE LICENSEE BY IMAGINE IT OR ITS REPRESENTATIVES.

UPON TERMINATION LICENSEE SHALL RETURN ALL COPIES OF THE SOFTWARE, ITS COMPONENTS AND DOCUMENTATION TO IMAGINE IT.

ALL RIGHTS GRANTED TO THE LICENSEE IN THIS AGREEMENT SHALL CEASE UPON TERMINATION OF THIS AGREEMENT EXCEPT FOR THOSE APPLYING TO ANY COPIES OF ANY END USER APPLICATIONS ALREADY PROPERLY DISTRIBUTED AND LICENSED WITHIN THE TERMS OF THIS AGREEMENT, PRIOR TO TERMINATION OF THIS AGREEMENT.

CONFIDENTIALITY

LICENSEE ACKNOWLEDGES THAT THE SOFTWARE AND ITS DOCUMENTATION AND DESIGN CONSTITUTE CONFIDENTIAL AND PROPRIETARY INFORMATION BELONGING TO IMAGINE IT.

LICENSEE AGREES NOT TO DISCLOSE ANY OF THIS INFORMATION TO OTHER THIRD PARTIES EXCEPT FOR ITS EMPLOYEES WHO ARE ENGAGED IN USING THE SOFTWARE AND ARE BOUND BY THESE SAME TERMS THROUGH A SEPARATE WRITTEN AGREEMENT WITH LICENSEE, OR EXCEPT WHERE REQUIRED TO DISCLOSE THE INFORMATION BY LAW.

WARRANTIES AND DISCLAIMERS

THE SOFTWARE IS PROVIDED "AS IS". IMAGINE IT DOES NOT WARRANT THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, OR THAT IT WILL MEET ALL OF THE LICENSEE'S REQUIREMENTS.

ALL WARRANTIES, EXPRESSED OR IMPLIED, ARE EXCLUDED FROM THIS AGREEMENT INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE BUT NOT EXCLUDING ANY STATUTARY RIGHTS PROVIDED FOR BY LAW.

IMAGINE IT SHALL NOT BE LIABLE FOR ANY DAMAGES INCLUDING BUT NOT LIMITED TO INCIDENTAL OR CONSEQUENTIAL LOSS, LOSS OF PROFIT, LOSS OF SALES, INJURY, DEATH, LOSS OF OR DAMAGE TO PROPERTY OR ANY OTHER LOSS ARISING DIRECTLY OR INDIRECTLY FROM THE USE OF THE SOFTWARE.

IMAGINE IT'S SOLE REMEDY TO THE LICENSEE, AT IMAGINE IT'S SOLE DISCRETION SHALL BE LIMITED TO EITHER THE REPAIR OR REPLACEMENT OF FAULTY SOFTWARE OR THE REPAYMENT OF THE LICENSE FEE PAID BY THE LICENSEE UPON RETURN OF ALL COPIES OF THE SOFTWARE. IMAGINE IT'S TOTAL LIABILITY SHALL NOT EXCEED THE AMOUNTS PAID BY LICENSEE FOR THE PURCHASE OF THIS LICENSE.

APPLICABLE LAW

THIS AGREEMENT SHALL BE GOVERNED BY TO THE LAWS OF ENGLAND.

END

In order to use **iCalendar** in your applications you need to distribute the OCX with your application. In addition you also need to ship some Microsoft® shared libraries which the OCX uses at run-time.

Before distributing any files which have been shipped with this product please read the license agreement and ensure that you comply with all its terms and conditions. Note that you do not have a license to distribute any Microsoft® files as part of this license and you should ensure that you have the appropriate licenses from Microsoft®. You should not distribute these third party components from the disks supplied with this product.

Shipping iCalendar

To ship **iCalendar** to your customers you MUST ship the following files - you may NOT ship any other files belonging to this product under any circumstances.

1. iCalendar.OCX
2. iCalendar.RTL

You should install these files on the target system in the Windows System Directory, for example :-

<WindowsDirectory> \ System \

After installing these files and the other Microsoft® files identified below, you should register iCalendar on the target system,. By running the following command :-

REGSVR32.EXE /s iCalendar.OCX

Shipping Microsoft® Components

The following files are required on the target system :-

MFC42.DLL

OLEPRO32.DLL

REGSVR32.EXE

These should be installed in the target system's Windows System Directory but only if these files are either not already installed or they are a later version than those already on the target system.

When you install and register a control, you should also register OLEPRO32.DLL. Using the following command :-

REGSCR32.EXE /s OLEPRO32.DLL

Perform this registration step only if you need to install OLEPRO32.DLL. If the DLL is installed already, you should assume that it has been registered.

You should also register MFC40.DLL. Unlike OLEPRO32.DLL, you should always register this DLL, even if it is already installed. To register this DLL run the following command :-

REGSVR32.EXE /s MFC40.DLL

To UnInstall **iCalendar** from your development system please follow these instructions :-

1. Un-Register the OCX by clicking on the ‘‘un-register iCalendar’’ icon or running the following command

REGSVR32.EXE /u iCalendar.OCX

2. Run the Add / Remove programs applet from the Control Panel and select the **iCalendar** component to be removed.

iCalendar supports the following Stock Properties:

BackColor	Background Color
ForeColor	Foreground Color
Font	Font name, type and size

iCalendar supports the following Custom Properties:

CenturyCutOff	The year that is used to distinguish the century when using 2 digits for the year.
DateFormat	The default format for dates.
DateStr	The current iCalendar date formatted as a string using the current DateFormat
Day	The Day part of the current iCalendar Date
WeekDay	The week day of the current iCalendar Date
DaysPerWeek	Number of days per week
DisplayMode	Sets iCalendar to one of four display modes
EDays	Elapsed days – this read only property contains the number of elapsed days as calculated by the DateDiff method
EHours	Elapsed hours – this read only property contains the number of elapsed hours as calculated by the TimeDiff method
EMinutes	Elapsed minutes – this read only property contains the number of elapsed minutes as calculated by the TimeDiff method
EMonths	Elapsed months – this read only property contains the number of elapsed months as calculated by the DateDiff method
ESeconds	Elapsed seconds – this read only property contains the number of elapsed seconds as calculated by the TimeDiff method
EYears	Elapsed years – this read only property contains the number of elapsed years as calculated by the DateDiff method
Hour	The Hour part of the current iCalendar Time
LastError	The last error that occurred in iCalendar.
LastErrorString	The error message corresponding to LastError
LongDayName	The current long day name formatted as a string
LongMonthName	The current long month name formatted as a string
Minute	The Minute part of the current iCalendar Time
Month	The Month part of the current iCalendar Date

RealTime	Sets iCalendar to real time mode to continually track the system time.
Second	The Second part of the current iCalendar Time
ShortDayName	The current short say name formatted as a string
ShortMonthName	The current short month name formatted as a string
TimeFormat	The default format for time.
TimeQualifier	The current am / pm qualifier for 12 hour time formats
TimeStr	The current iCalendar time formatted as a string using the current TimeFormat
Year	The Year part of the current iCalendar Date

See Also

Methods

Examples

Use this property to set or get the default date format.

Syntax

controlname.**DateFormat** [= *DateFormat*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

By default iCalendar uses the system date format (see control panel / regional settings) to display dates. The date format is a specification string that allows you to select how dates will be displayed. (e.g. dd/mm/yy is a format string that will display dates like 12/03/97). Override the default format by setting this property.

A date format string consists of any number of format specifiers and literals:

Valid date format specifiers are:

d - date displayed as one or two digits (without leading zero)

dd - date displayed as two digits (with a leading zero if required)

ddd - day displayed as short day name – eg Mon

dddd - day displayed as long day name – eg Monday

M - month displayed as one or two digits (without leading zero)

MM - month displayed as two digits (with a leading zero if required)

MMM - month displayed as a short month name – eg Jan

MMMM - month displayed as a full month name – eg January

y - year displayed without century as one or two digits (without leading zero)

yy - year displayed without century as two digits (with leading zero if required)

yyyy - year displayed with century as four digits

Literals:

Any other character is considered a literal character and will be displayed as is. If you need to display a literal character that is also a date format specifier then enclose it in single quotes.

Example Date Formats:

Format	Example Date Display
d/M/y	1/1/97
dd-MM-yy	01-01-97

dddd dd MMM yyyy	Monday 01 Jan 97
MMMM dd, yy	January 01, 97
‘Today is ‘ dd MMMM yyyy	Today is 01 January 1997

Use this property to specify the default time format.

Syntax

controlname.**TimeFormat** [= *TimeFormat*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

By default iCalendar uses the system time format (see control panel / regional settings) to display time. The time format is a specification string that allows you to select how time will be displayed. (e.g. hh:mm:ss is a format string that will display time like 11:50:30). Override the default format by setting this property.

A time format string consists of any number of format specifiers and literals:

Valid time format specifiers are:

h - hour displayed as one or two digits (without leading zero)

hh - hour displayed as two digits (with a leading zero if required)

H - hour (24 hour clock) displayed as one or two digits (without leading zero)

HH - hour (24 hour clock) displayed as two digits (with a leading zero if required)

m - minute displayed as one or two digits (without leading zero)

mm - minute displayed as two digits (with a leading zero if required)

s - second displayed as one or two digits (without leading zero)

ss - second displayed as two digits (with leading zero if required)

tt - AM / PM specifier for 12 hour clock

Literals:

Any other character is considered a literal character and will be displayed as is. If you need to display a literal character that is also a time format specifier then enclose it in single quotes.

Example time Formats:

Format	Example Time Display
h m s	9 1 29
HH:mm:ss	21:01:29
hh mm tt	09 01 PM
‘The time now is ‘ HH:mm ‘Hours’	The time now is 21:09 Hours

Use this property to get or set the day part of the current iCalendar date.

Syntax

controlname.**Day** [= *Day*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar is always set to a date known as the current date. When iCalendar is initialised this is set to the system date.

The current day element can be set or retrieved using this property.

e.g. if the current date is 12 Mar 1997, the day property will be 12.

Use this property to get or set the month part of the current iCalendar date.

Syntax

controlname.**Month** [= *Month*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar is always set to a date known as the current date. When iCalendar is initialised this is set to the system date.

The current month element can be set or retrieved using this property.

e.g. if the current date is 12 Mar 1997, the month property will be 3.

Use this property to get or set the year part of the current iCalendar date.

Syntax

controlname.**Year** [= *Year*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar is always set to a date known as the current date. When iCalendar is initialised this is set to the system date.

The current year element can be set or retrieved using this property.

e.g. if the current date is 12 Mar 1997, the year property will be 97 or 1997 depending on the current DateFormat property setting.

Use this property to get or set the hour part of the current iCalendar time.

Syntax

controlname.**Hour** [= *Hour*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar is always set to a time known as the current time. When iCalendar is initialised this is set to the system time.

The current hour element can be set or retrieved using this property.

e.g. if the current time is 23:30:00, the hour property will be 23.

Use this property to get or set the minute part of the current iCalendar time.

Syntax

controlname.**Minute** [= *Minute*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar is always set to a time known as the current time. When iCalendar is initialised this is set to the system time.

The current minute element can be set or retrieved using this property.

e.g. if the current time is 23:30:00, the minute property will be 30.

Use this property to get or set the second part of the current iCalendar time.

Syntax

controlname.**Second** [= *Second*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar is always set to a time known as the current time. When iCalendar is initialised this is set to the system time.

The current second element can be set or retrieved using this property.

e.g. if the current time is 23:30:00, the second property will be 0.

Use this property to specify the year which will act as a cut off to determine the century when using a 2 digit year.

Syntax

controlname.**CenturyCutOff** [= *CenturyCutOff*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

When using 2 digit years the century is implied. However this can cause ambiguity. e.g. is 1-Jan-10 referring to the year 1910 or 2010 ? This is particularly important with the year 2000 approaching !

To overcome this ambiguity the CenturyCutOff property allows you to specify what the century part should be. The CenturyCutOff is specified as a full four digit year e.g. 1980.

A 2 digit year that is greater than or equal to the CenturyCutOff year part is treated as being the same century as the CenturyCutOff.

A 2 digit year that is less than the CenturyCutOff year part is treated as being the next century after the CenturyCutOff.

Examples:

CenturyCutOff	2 digit Year	Actual Year
1980	94	1994
1980	10	2010
1756	56	1756
1756	55	1855

Use this property to specify the number of days in a week depending on the type of calendar being used.

Syntax

controlname.**DaysPerWeek** [*CalendarType*] [= *DaysPerWeek*]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

This property is reserved for future use. Currently it is always set to 7.

Use this property to get the last error that occurred in iCalendar.

Syntax

controlname.**LastError**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

This read-only property is useful to check what type of error occurred. Every method sets this property to either 0 (no error) or an error code. Error codes are defined in the iCalendarConstants file.

Use this property to get the error description of the last error that occurred in iCalendar.

Syntax

controlname.**LastErrorString**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

This read-only property is useful for getting a sensible error message for the last error that occurred.

Use this property to get the current weekday

Syntax

controlname.**WeekDay**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

Use this property to get the day of the week corresponding to the current date as a numeric value.

Current Day	WeekDay
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6
Sunday	7

Use this property to get the number of elapsed days after using the DateDiff or TimeDiff methods

Syntax

controlname.**EDays**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Long

Remarks

This read-only property is set after using the DateDiff or TimeDiff methods.

See Also

DateDiff method

TimeDiff method

Use this property to get the number of elapsed months after using the DateDiff or TimeDiff methods

Syntax

controlname.**EMonths**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Long

Remarks

This read-only property is set after using the DateDiff or TimeDiff methods.

See Also

DateDiff method

TimeDiff method

Use this property to get the number of elapsed years after using the DateDiff or TimeDiff methods

Syntax

controlname.**EYears**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Long

Remarks

This read-only property is set after using the DateDiff or TimeDiff methods.

See Also

DateDiff method

TimeDiff method

Use this property to get the number of elapsed hours after using the DateDiff or TimeDiff methods

Syntax

controlname.**EHours**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Long

Remarks

This read-only property is set after using the DateDiff or TimeDiff methods.

See Also

DateDiff method

TimeDiff method

Use this property to get the number of elapsed minutes after using the DateDiff or TimeDiff methods

Syntax

controlname.**EMinutes**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Long

Remarks

This read-only property is set after using the DateDiff or TimeDiff methods.

See Also

DateDiff method

TimeDiff method

Use this property to get the number of elapsed seconds after using the DateDiff or TimeDiff methods

Syntax

controlname.**ESeconds**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Long

Remarks

This read-only property is set after using the DateDiff or TimeDiff methods.

See Also

DateDiff method

TimeDiff method

Use this property to set iCalendar to real time mode

Syntax

controlname.**RealTime** *[= True / False]*

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Boolean

Remarks

This property is useful if you want to track the real 9system) time continually. By setting this property to True iCalendar continually sets the current date and time in sync with the ssystem clock. This allows you to easily get the current system date and time as well as display the current system date and time.

When this property is set you should not set any date or time properties or use any methods for date / time calculations.

See Also

Use this property to change the way iCalendar is displayed

Syntax

controlname.**DisplayMode** *[= displaymode]*

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

iCalendar can be programmed to display itself in one of four modes:

Mode	Setting	Description
Date	1 – DATE_MODE	Only display the Date edit box
Time	2 – TIME_MODE	Only display the time edit box
Date and Time	3 – DATE_TIME_MODE	Display both the date and time edit boxes
Calendar	4 – CALENDAR_MODE	Only display the monthly calendar

The settings are defined as constants in the iCalendarConstants file.

You should not change this property at run-time since the display appearance may effect they way your user interface looks.

See Also

Use this property to get the current month as a string

Syntax

controlname.**LongMonthName**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

This read-only property provides the current long month name as a string. e.g. January.

See Also

Use this property to get the current month as a string

Syntax

controlname.**ShortMonthName**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

This read-only property provides the current short month name as a string. e.g. Jan.

See Also

Use this property to get the current day name as a string

Syntax

controlname.**LongDayName**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

This read-only property provides the current long day name as a string. e.g. Monday.

See Also

Use this property to get the current day name as a string

Syntax

controlname.**ShortDayName**

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

This read-only property provides the current short day name as a string. e.g. Mon.

See Also

Use this property to get or set the current iCalendar date.as a string

Syntax

controlname.**DateStr** [= date]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

iCalendar is always set to a date known as the current date. When iCalendar is initialised this is set to the system date. Use this property to get or set the current date as a string.

The date string must be formatted according to the DateFormat property

Example

DateStr = "12/01/92"

Use this property to get or set the current iCalendar time.as a string

Syntax

controlname.**TimeStr** [= time]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

String

Remarks

iCalendar is always set to a time known as the current time. When iCalendar is initialised this is set to the system time. Use this property to get or set the current time as a string.

The time string must be formatted according to the TimeFormat property

Example

TimeStr = ‘‘12:00:00’’

Use this property to get or set the time qualifier - AM or PM for 12 hour clock formats

Syntax

controlname.**TimeQualifier** [= date]

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Type

Short

Remarks

Use this property to specify whether a 12 hour time is AM or PM or Unspecified

Setting	Value
AM	1 - TIME_QUALIFIER_AM_INT
PM	2 - TIME_QUALIFIER_PM_INT
Default	3 - TIME_QUALIFIER_DEFAULT_INT

The values are defined as constants in the iCalendarConstants file

Example

TimeQualifier = TIME_QUALIFIER_AM_INT

iCalendar provides a comprehensive range of methods for date and time manipulation and calculations.

iCalendar methods include:

CompareDatesEx	Compares any two dates to check whether one is equal to, less than or greater than the other
CompareDates	Compares a date with the current date to check whether it is equal to, less than or greater than the current date
CompareTimesEx	Compares any two times to check whether one is equal to, less than or greater than the other
CompareTimes	Compares a time with the current time to check whether it is equal to, less than or greater than the current time
DateDiffEx	Determines the number of years, months and days between two specified dates
DateDiff	Determines the number of years, months and days between the specified date and the current date
DeltaDateEx	Adds or subtracts specified number of years, months and days to or from a specified date
DeltaDate	Adds or subtracts specified number of years, months and days to or from the current date
DeltaTimeEx	Adds or subtracts specified number of hours, minutes and seconds to or from a specified time
DeltaTime	Adds or subtracts specified number of hours, minutes and seconds to or from the current time
GetDate	Get the current date as a formatted string.
GetDayOfWeekEx	Gets the day of the week for a specified date
GetLongDayNameEx	Gets the day name for a specified date
GetLongMonthNameEx	Gets the month name for a specified date
GetShortDayNameEx	Gets the day name for a specified date
GetShortMonthNameEx	Gets the month name for a specified date
GetTime	Get the current time as a formatted string.
SetDate	Set the current date using a formatted string
SetTime	Set the current time using a formatted string
TimeDiffEx	Determines the number of hours, minutes and seconds between two specified times
TimeDiff	Determines the number of hours, minutes and seconds between the specified time and the current time

See Also

Properties

Examples

Gets the current iCalendar date as a formatted string

Syntax

controlname.**GetDate** (String FormatSpec)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

FormatSpec	The preferred format specification of the date.

Remarks

Formats the current iCalendar date according to the FormatSpec parameter and returns the date as a string. If FormatSpec is empty then the date is formatted according to the DateFormat property.

For a full description of Format Specification strings see the DateFormat property

Example

GetDate (“dd/mmmm/yyyy”)	will return “01/January/1997”
GetDate (“dddd dd mmm yy”)	will return “Wednesday 01 Jan 97”

See Also

SetDate

Sets the current iCalendar Date using a specified format.

Syntax

controlname. **SetDate** (String FormatSpec, String Date)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

None

Parameters

FormatSpec	A valid date format
Date	A date string

Remarks

This sets the current iCalendar date to Date parameter and updates the DateStr, Year, Month and Day properties

The FormatSpec parameter is used to decode the Date parameter. You can pass an empty string as the FormatSpec in which case the current date format will be used to decode the Date parameter. If the Date parameter does not match the specified FormatSpec the LastError property is set.

For a full description of Format Specification strings see the DateFormat property

Examples

DateStr = ‘‘12/1/97’’ set the current date

DateFormat = ‘‘dd/mm/yy’’ set the current date format

SetDate (‘‘dd/mmm/yyyy’’, ‘‘12/Mar/1992’’) will set DateStr to ‘‘12/03/92’’

SetDate (‘‘dd/mmm/yyyy’’, ‘‘12/12/92’’) will cause an error

SetDate (‘‘’’, ‘‘1/11/99’’) will set DateStr to ‘‘1/11/99’’

See Also

DateFormat

GetDate

Gets the current iCalendar time as a formatted string

Syntax

controlname.**GetTime** (String FormatSpec)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

FormatSpec	The preferred format specification of the time.

Remarks

Formats the current iCalendar time according to the FormatSpec parameter and returns the time as a string. If FormatSpec is empty then the time is formatted according to the TimeFormat property.

For a full description of Format Specification strings see the TimeFormat property

Example

GetTime (“hh:mm:ss”) will return “08:22:45”

GetTime (“hh:mm:ss tt”) will return “08:22:45 AM”

See Also

SetTime

Sets the current iCalendar Time using a specified format.

Syntax

controlname. **SetTime** (String FormatSpec, String Time)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

None

Parameters

FormatSpec	A valid time format
Time	A time string

Remarks

This sets the current iCalendar time to Time parameter and updates the TimeStr, Hour, Minute and Second properties. If a time qualifier (AM, PM) is not mentioned, iCalendar will automatically set the time as AM. This does not apply when the Time parameter uses the 24 hour format.

The FormatSpec parameter is used to decode the Time parameter. You can pass an empty string as the FormatSpec in which case the current time format will be used to decode the Time parameter. If the Time parameter does not match the specified FormatSpec the LastError property is set.

For a full description of Format Specification strings see the TimeFormat property

Examples

TimeStr = "13:01:01" set the current time

TimeFormat = "HH:mm:ss" set the current time format

SetTime ("HH:mm", "12:12") will set TimeStr to "12:12:01"

SetTime ("HH:mm:ss", "12:12") will cause an error

SetTime ("", "12:12:12") will set TimeStr to "12:12:12"

See Also

TimeFormat property

GetTime

Compares a date with the current iCalendar date.

Syntax

controlname.**CompareDates** (String Date1)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

Parameters

Date1	A date string in any of the specified date formats

Remarks

Effectively subtracts Date1 from the current date

Comparison is performed starting with the year field, then the month field and finally the day field. If there is a difference in one of the fields, the result is returned immediately.

Returns 0 if current date is equal to Date1

Returns -1 if current date is greater than Date1

Returns 1 if current date is less than Date1

Example

SetDate (“1/1/97”)

CompareDates (“1/1/97”) will return 0

CompareDates (“1/1/96”) will return -1

CompareDates (“1/2/97”) will return 1

See Also

CompareDatesEx

CompareTimes

method_comparetimesex

Compares a time with the current iCalendar time.

Syntax

controlname.**CompareTimes** (String TimeSrc)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

Parameters

Time1	A time string in any of the specified time formats

Remarks

Effectively subtracts Time1 from the current date

Comparison is performed starting with the hour field, then the minute field and finally the second field. If there is a difference in one of the fields, the result is returned immediately.

Returns 0 if current time is equal to TimeSrc

Returns -1 if current time is greater than TimeSrc

Returns 1 if current time is less than TimeSrc

Example

TimeStr = (“12:00:00 PM”

CompareTimes (“12:00:00 PM”) will return 0

CompareTimes (“08:45:00”) will return -1

CompareTimes (“18:00:00”) will return 1

See Also

CompareTimesEx

Compares two date strings

Syntax

controlname.**CompareDatesEx** (String Date1, String Date2)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

Parameters

Date1	A date string in any of the specified date formats
Date2	A date string in any of the specified date formats

Remarks

Comparison is performed starting with the year field, then the month field and finally the day field. If there is a difference in one of the fields, the result is returned immediately. Effectively subtracts Date2 from Date1

Returns 0 if Date1 is equal to Date2

Returns -1 if Date1 is less than Date2

Returns 1 if Date1 is greater than Date2

Example

CompareDatesEx (“1/1/97”, “1/1/97”) will return 0

CompareDatesEx (“1/1/97”, “1/2/97”) will return -1

CompareDatesEx (“1/1/97”, “1/1/96”) will return 1

See Also

CompareDates method

Compares two time strings

Syntax

controlname.**CompareTimesEx** (String Time1, String Time2)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

Parameters

Time1	A time string in any of the specified time formats
Time2	A time string in any of the specified time formats

Remarks

Comparison is performed starting with the hour field, then the minute field and finally the second field. If there is a difference in one of the fields, the result is returned immediately. Effectively subtracts Time2 from Time1

Returns 0 if Time1 is equal to Time2

Returns -1 if Time1 is less than Time2

Returns 1 if Time1 is greater than Time2

Example

CompareDatesEx (“6:00:00 PM”, “18:00:00”) will return 0

CompareDatesEx (“08:00:00 AM”, “08:00:00 PM”) will return -1

CompareDatesEx (“18:00:00”, “14:00:00”) will return 1

See Also

CompareTimes

Performs date addition or subtraction on the current iCalendar date.

Syntax

controlname.**DeltaDate** (Short iCalType, Long Days, Long Months, Long Years)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

iCalType	Reserved – this should always be 0
Days	The number of days to be added to or deducted from the current date
Months	The number of months to be added to or deducted from the current date
Years	The number of years to be added to or deducted from the current date

Remarks

The specified amount of years, months and days are added to or subtracted from the current iCalendar Date.

This will update the DateStr, Day, Month and the Year properties and will return the modified Date formatted according to the DateFormat property

*Example*s

SetDate("25 December 1987")

DateFormat("dd/MM/yyyy")

DeltaDate(0,7,0,0) will return 01/01/1988

SetDate("1/1/2000")

DateFormat("dd/MM/yyyy")

DeltaDate(0,-1,-1,-1) will return 31/11/1998

See Also

DeltaDateEx

Performs date addition or subtraction.

Syntax

controlname.**DeltaDateEx** (String Date, Short iCalType, Long Days Long Months, Long Years)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

Date	A date string in any of the specified date formats
iCalType	Reserved – this should always be 0
Days	The number of days to be added to or deducted from the Date parameter
Months	The number of months to be added to or deducted from the Date parameter
Years	The number of years to be added to or deducted from the Date parameter

Remarks

The specified number of days, months and years are added to or subtracted from the specified date. This will return a date string formatted according to the DateFormat property. The current iCalendar settings are not effected. See the DeltaDate method for examples.

See Also

DeltaDate

Performs addition or subtraction on the current iCalensar time.

Syntax

controlname.**DeltaTime** (Short iCalType, Long Hours, Long Minutes, Long Seconds)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

iCalType	Reserved – this should always be 0
Hours	The number of hours to be added to or deducted from the current time
Minutes	The number of minutes to be added to or deducted from the current time
Seconds	The number of seconds to be added to or deducted from the current time

Remarks

The specified number of hours, minutes and seconds are added or subtracted. This will update the Hour, Minute, and Second properties and will return the modified Time formatted according to the TimeFormat property. The DateStr property (and the corresponding Day, Month and Year properties) may also be affected if the time crosses the day boundary (see Example 3)

Example 1

TimeStr = "12:00:00 PM"

TimeFormat = "hh:mm:ss tt"

DeltaTime(0,2,30,10) will return "02:30:10 PM"

Example 2

SetTime("13:15:20")

TimeFormat("H:m:s")

DeltaTime(0,-4,-5,-71) will return "9:9:9"

Example 3

TimeStr = "12:00:01 AM"

DateStr = "1/1/2000"

TimeFormat = "hh:mm:ss"

DeltaTime(0,0,0,-2) will return "11:59:59 PM" and will set the date to "31/12/1999"

See Also

DeltaTimeEx

Performs time addition or subtraction.

Syntax

controlname.**DeltaTimeEx** (String Time, Short iCalType, Long Hours, Long Minutes, Long Seconds)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

Time	A time string in any of the specified time formats
iCalType	Reserved – this should always be 0
Hours	The number of hours to be added to or deducted from the current time
Minutes	The number of minutes to be added to or deducted from the current time
Seconds	The number of seconds to be added to or deducted from the current time

Remarks

The specified amount of hours, minutes and seconds are added to or subtracted from the specified Time..

This will return the result of the calculation as a string formatted according to the TimeFormat property.

Example

DeltaTimeEx("09:10:05", 0, 2, 5, 20) will return 11:15:25

DeltaTimeEx("03:45:59", 0, -1, -5, -58) will return 02:40:01

See Also

DeltaTime

Determines the time span between the current iCalendar date and the date parameter.

Syntax

controlname.**DateDiff** (String Date, Short iCalType)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

This value indicates 1 for success and 0 for failure

Parameters

Date	A date string in any of the specified date formats
iCalType	Reserved – This should always be 0

Remarks

Finds the number of days, months and years difference between the current iCalendar date and the user specified date and updates the EDays, EMonths and EYears properties accordingly. The elapsed values are always positive and are normalised to give the whole number of years, months and days – i.e. EMonths will never be greater that 11 and EDays will never be greater than 30. Intervening leap years are taken into account.

Example

SetDate("12 March 1992")

DateDiff("12/8/96",0) returns SUCCESS and updates the above properties as follows:

EDays = 0

EMonths = 5

EYears = 4

See Also

DateDiffEx

Determines the time span between the current iCalendar time and a specified time.

Syntax

controlname.**TimeDiff** (String Time, Short iCalType)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

This value indicates 1 for success and 0 for failure

Parameters

Time	A time string in any of the specified time formats
iCalType	Reserved – This should always be 0

Remarks

Finds the number of hours, minutes and seconds difference between the current iCalendar time and the user specified time and updates the EHours, EMinutes and ESeconds properties accordingly. The elapsed values are always positive and are normalised to give the whole number of hours, minutes and seconds – i.e. EMinutes and ESeconds will never be greater than 59.

Example

SetTime("11:00:15")

TimeDiff("09:10:10",0) returns SUCCESS and updates the above properties as follows:

EHours = 1

EMinutes = 50

ESeconds = 10

See Also

TimeDiffEx

Returns the day of the week for a given date

Syntax

controlname.**GetWeekDayEx** (String DateStr)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

Parameters

DateStr	A date string in any of the specified date formats

Remarks

Gets the week day for a given date as an integer value from 1 to 7 where 1 is Monday through to 7 is Sunday.

Example

GetDayOfWeekEx(“1/1/96”) returns the value 1

See Also

Determines the time span between 2 user defined date parameters.

Syntax

controlname.**DateDiffEx** (String Date1, String Date2, Short iCalType)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

This value indicates 1 for success and 0 for failure

Parameters

Date1	A date string in any of the specified date formats
Date2	A date string in any of the specified date formats
iCalType	Reserved – This should always be 0

Remarks

Finds the number of days, months and years between the 2 given dates and updates the EDays, EMonths and EYears properties accordingly. The elapsed values are always positive and are normalised to give the whole number of years, months and days – i.e. EMonths will never be greater that 11 and EDays will never be greater than 30. Intervening leap years are taken into account.

Example

DateDiffEx("12/7/96", "2/2/97", 0) returns SUCCESS and updates the above properties as follows:

EDays = 21

EMonths = 6

EYears = 0

See Also

DateDiff

Determines the time span between 2.user defined time parameters

Syntax

controlname.**TimeDiffEx** (String Time1, String Time2, Short iCalType)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

Short

This value indicates 1 for success and 0 for failure

Parameters

Time1	A time string in any of the specified time formats
Time2	A time string in any of the specified time formats
iCalType	

Remarks

Finds the number of hours, minutes and seconds between the 2 times and updates the EHours, EMinutes and ESeconds properties accordingly. The elapsed values are always positive and are normalised to give the whole number of hours, minutes and seconds – i.e. EMinutes and ESeconds will never be greater than 59.

Example

TimeDiffEx(“09:10:10”, “12:12:12”, 0) returns SUCCESS and updates the above properties as follows:

EHours = 3

EMinutes = 2

ESeconds = 2

See Also

TimeDiff

Gets the long day name for a specified date

Syntax

controlname.**GetLongDayNameEx** (String Date)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

Date	A date string in any of the specified time formats

Remarks

Gets the day name for any specified date

Example

GetLongDayNameEx (“4-Apr-1997”) will return “Friday”

See Also

GetShortDayNameEx method

Gets the long month name for a specified date

Syntax

controlname.**GetLongMonthNameEx** (String Date)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

Date	A date string in any of the specified time formats

Remarks

Gets the month name for any specified date

Example

GetLongMonthNameEx (‘‘31-04-1997’’) will return ‘‘April’’

See Also

GetShortMonthNameEx method

Gets the short day name for a specified date

Syntax

controlname.**GetShortDayNameEx** (String Date)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

Date	A date string in any of the specified time formats

Remarks

Gets the day name for any specified date

Example

GetShortDayNameEx (“04-Apr-1997”) will return “Fri”

See Also

GetLongDayNameEx method

Gets the short month name for a specified date

Syntax

controlname.**GetShortMonthNameEx** (String Date)

controlname is the name of the **iCalendar** Control object, for example, iCalendar1.

Return Value

String

Parameters

Date	A date string in any of the specified time formats

Remarks

Gets the month name for any specified date

Example

GetShortMonthNameEx (“31-04-1997”) will return “Apr”

See Also

GetLongMonthNameEx method

In order to help you quickly get started with using **iCalendar** we have included a Visual Basic sample application with this product.

The sample application is installed in the directory where you chose to install iCalendar as follows :-

<InstallDirectory> \ SAMPLES \ VB \

We have also included files containing the constant definitions which you may find useful for writing your applications. These include files are installed as follows :-

<InstallDirectory> \ SAMPLES \ MFC \ iCalendarConstants.h for C / C++

<InstallDirectory> \ SAMPLES \ VB \ iCalendarConstants.bas for Visual Basic