

Chapter 4

The Custom Summary Window

Overview • Commands • Keywords

Overview

For a complete description of every Custom Summary command and keyword, see Appendix D.

This window executes a series of commands to display text, change the color of the window, and play sounds. It gets the text it displays from a file in the Lightning/2000 installation folder. The default name of the file is "Custom Summary Template.txt", but any text file can be used as the custom summary template.

The custom summary template is a text file that can be edited just like any other text file. On most systems, the Notepad application is the default application for editing text files. Word processors (such as Microsoft Word) will not produce a text file by default, however most can produce a plain text file as an option.

You may change the name of the text file used by the Custom Summary window by right-clicking in the window and choosing "**Custom Summary Manager...**" from the popup menu. See the manual section on the Custom Summary

Manager for details on how to change or edit the text file used as a template.

A template file is also used to generate the text displayed on the General Status Line. That template file is of the same format, and may use any of the keywords or commands used in the Custom Summary Window.

If you want to experiment with either the custom summary template or status line template, you should make a copy of it first so that any changes you make can be abandoned if they don't work out so well.

The custom summary window gives you unprecedented flexibility and control over what information is displayed, and (optionally) what information is automatically emailed or sent to your Twitter account. You might (for example) create a very brief custom summary that gets sent to as a text message to cell phones (assuming they can be reached via an email address), or a highly specific and descriptive custom summary that is to be emailed to a list of clients. It is entirely up to you.

If you have chosen to email the custom summary, the first line of text in the custom summary will be used as the title of the email. See the Email Options Dialog for more information.

The text file that generates the Custom Summary window consists of a series of statements. A statement may either be a conditional statement (one starting with **if**) or a command. The most frequently used commands are the ones used to write text into the window (the `#Write` and `#WriteLine` commands).

A comment line is any line beginning with a semicolon (;) character. Comment lines will not be included in the Custom Summary window. They are mainly for use in complex custom summary templates so that you can explain how you are using the various keywords in the template.

You may also end any line with a comment by typing a semicolon. Any text on the line after the semicolon will not be included in the custom summary window.

All keywords used in the Custom Summary start with the @ character. Commands start with the # character. There are 5 reserved words that you can use to control which commands are executed: **if**, **then**, **else**, **elseif**, and **endif**.

The structure of an **if** statement is:

```
if expression then
    One or more statements

elseif expression then
    One or more statements

elseif expression then
    One or more statements

else expression then
    One or more statements
endif
```

Only the **if**, **then**, and **endif** parts of the if statement are required. Both the **elseif** and **else** parts are optional. There may be any number of **elseif** parts, but there may only be one **else** part. The **else** part is executed only if neither the **if** part nor any of the **elseif** parts were executed.

A statement in the **if**, **else**, or **elseif** parts may be a command or another if statement.

An “expression” is basically any kind of mathematical expression, usually containing one or more of the keywords. There are several special symbols and words that can be used in an expression:

>	greater than
<	less than
=	equals
<>	not equals
<=	less than or equals to
>=	greater than or equals to
and	a logical and of two expressions

or a logical or of two expressions
xor a logical exclusive or of two expressions
+ the sum of two expressions or the concatenation of two quoted strings
- the difference of two expressions
***** the multiplication of two expressions
/ the division of two expressions
div the division of two expressions, truncated to a whole number

Here are a few examples of valid if statements:

```

if @StrokeCount( 5 ) > 1000 then
  #AlertStatusLineYellow
endif

if ( @CGFlashCount > 50 ) and ( @StrokeCount > 1000 )
then
  #AlertStatusLineYellow

  if @CloseStormCount > 0 then
    #Sound( 'beep5' )
    #WriteLine( 'A storm is nearby.' )
  endif
endif

if @HardwareType = 0 then
  #WriteLine( 'No lightning detector is installed.' )

elseif @HardwareType = 1 then
  #WriteLine( 'LD-250 installed' )

elseif @HardwareType = 4 then
  #WriteLine( 'LD-350 installed' )

else
  #WriteLine( 'StormTracker installed' )
endif

```

When using “and” in an expression, the part after the “and” is evaluated only if the part before the “and” is true. This is because the entire expression (before and after) can be true only if the before part is true and the after part is true. If the before part is false, then the entire expression is false whether or not the after part is true. Since it does not matter in this case if the after part is true or false, there is no sense in wasting time evaluating the after part.

When using “or” in an expression, the part after the “or” is evaluated only if the part before the “or” is false. This is because the entire expression (before or after) can be false only if the before part is false and the after part is false. If the before part is true, then the entire expression is true whether or not the after part is true. Since it does not matter in this case if the after part is true or false, there is no sense in wasting time bothering to evaluate the after part.

For a complete description of the format of the custom summary template, see Appendix C, “Custom Summary Language”.

A complete description of all Custom Summary commands and keywords may be found in Appendix D.

Some examples of Custom Summary templates can be found in Appendix E.

Commands

A command is any word preceded by a # sign. There are 3 commands that may be included in a template file that pertain to the general status line (the **#AlertStatusLine** commands) at the bottom of the main window, so they should only be used in templates for the general status line.

A command is typically issued as a result of one or more conditions being true. For example, the status line template may determine that a severe storm is nearby and issue the **#AlertStatusLineRed** command.

By using the custom summary commands, a wide variety of actions may be taken. Screen captures may be produced (in

addition to or instead of the normal automatic screen capture process). Text files may be produced, consisting of virtually any information, and then be sent to a server via FTP. Commands may be sent to a serial port to control external devices. Text may be written to the Custom Summary window, consisting of an almost unlimited variety of information produced by Lightning/2000. An automatically generated email may be sent.

The names of the commands are not case sensitive.

Keywords

Keywords that evaluate to a per minute stroke or flash count can be followed by a parameter. Then the count will evaluate to the number of strokes or flashes in the last N minutes, where N is the parameter.

The parameter "N" may be either a positive integer, or a positive real number. Real numbers less than 1 must start with a zero.

For example, "0.1" is a valid parameter, but ".1" is not.

For example, the **@CGFlashCount** keyword will evaluate to the number of CG flashes in the last minute, but if you specify **@CGFlashCount(2)**, then it will evaluate to the number of CG flashes in the last 2 minutes. **@CGFlashCount(1)** is identical to **@CGFlashCount**. There is a limit of 60 minutes for the various stroke counts.

By default, the **@Peak** keywords will return the peak count for the day.

Entering (N) (where N is a number of minutes) after the **@Peak** keyword will return the peak value in the last N minutes.

The **@Trend** keywords return a percentage difference between an earlier count and the current count.

Keywords are not case sensitive, so **@cgflashcount** is identical to **@CGFlashCount** or **@CGFLASHCOUNT**.

Appendix C

Custom Summary Language Definition

This is a formal description of the custom summary language, used in custom summary and status line template files.

The symbol to the left of the ::= sign is replaced by the symbols to the right of the ::= sign.

Symbols enclosed in curly brackets {} may be repeated any number of times, including zero times.

The | character represents a choice between two different sets of symbols.

The < and > signs enclose a symbolic term.

Anything in boldface is to be included exactly as it appears.

```

<custom summary template> ::= {<statement> | <comment>}
<comment> ::= ; {<alphanumeric character>}
<statement> ::= <command> | <if statement>
<command> ::= #<string> | #<string> ( <expression> {,
<expression>} )
<string> ::= <alphanumeric character> {<alphanumeric
character>}
<if statement> ::=
    if <expression> then {<statement>} {elseif
<expression> then {<statement>}} endif |
    if <expression> then {<statement>} {elseif
<expression> then {<statement>}} else {<statement>}
endif
<expression> ::= <simple expression> | <simple
expression> <relational operator> <simple expression>
<relational operator> ::= < | > | <= | >= | <> | =

```

```

<simple expression> ::= <sign> <term> {<adding
operator> <term>} | <term> {<adding operator> <term>}
<term> ::= <factor> {<multiplying operator> <factor>}
<sign> ::= + | -
<multiplying operator> ::= * | / | and | div
<adding operator> ::= + | - | or | xor
<factor> ::= <keyword> | <unsigned number> | (
<expression> ) | not <factor> | <quoted string>
<keyword> ::= @<string> | @<string> ( <expression> )
<unsigned number> ::= <unsigned integer> | <unsigned
real>
<unsigned integer> ::= <digit> {<digit>}
<unsigned real> ::= <unsigned integer> . <unsigned
integer> | <unsigned integer> . <unsigned integer> E
<scale factor> | <unsigned integer> E <scale factor>
<scale factor> ::= <unsigned integer> | <sign>
<unsigned integer>
<quoted string> ::= '<string>'

```

Appendix D

Custom Summary Commands and Keywords

*Commands • Count Keywords • Date/Time
Keywords • Miscellaneous Keywords • Peak
Count Keywords • Percent and Ratio
Keywords • Records Keywords • Total Count
Keywords • Trend Keywords*

Commands

A command is any word preceded by a # sign. There are 3 commands that may be included in a template file that pertain to the general status line at the bottom of the main window, so they should only be used in templates for the general status line.

A command is typically issued as a result of one or more conditions being true. For example, the status line template may determine that a severe storm is nearby and issue the **#AlertStatusLineRed** command.

By using the custom summary commands, a wide variety of actions may be taken. Screen captures may be produced (in addition to or instead of the normal automatic screen capture process). Text files may be produced, consisting of virtually any information, and then be sent to a server via FTP. Commands may be sent to a serial port to control external devices. Text may be written to the Custom Summary window, consisting of an almost unlimited variety of information produced by Lightning/2000. An automatically generated email may be sent.

The names of the commands are not case sensitive.

#AddToUploadQueue('filename')

Adds the specified file to the upload queue so that it will be uploaded to a server the next time an FTP is done. Here is an example of this command:

```
#AddToUploadQueue( @L2Kfolder + 'mystats.txt' )
```

Any errors encountered during the execution of the #AddToUploadQueue command will be written to the Custom Summary window.

#AlertStatusLineGreen

Cancels any general status yellow, and cancels a general status red alert unless the red alert has been in effect for less than 5 minutes.

#AlertStatusLineYellow

Sets the alert level of the general status line to yellow, unless a red alert is already in effect for the general status line.

#AlertStatusLineRed

Sets the alert level of the general status line to red. A general status line red alert always remains in effect for at least 5 minutes.

#AppendFile('filename', expression)**#AppendFile('filename', expression, expression, ..., expression)**

Writes a line of text to the end of a file. The previous contents of the file (if any) are preserved. The name of the file is specified in the 'filename' parameter. The expression may be either a quoted string, a mathematical expression (including keywords), or a combination. Multiple elements in the same #AppendFile command may be concatenated with plus signs, or they may be separated by commas. Examples of #AppendFile commands:

```
#AppendFile( @L2kfolder + 'ftplog.txt', 'FTP started at  
' + @LocalTime )
```

```
#AppendFile( 'C:\test.txt', 'just testing' )
```

Any errors encountered during the execution of the #AppendFile command will be written to the Custom Summary window.

#BackgroundColor(R,G,B)

Sets the background color of the Custom Summary window. The background color is specified as an RGB (red, green, blue) value. Each of the numbers for the red, green, and blue values may range from 0 to 255.

For example, the RGB value 255,0,0 is the brightest pure red that can be displayed; 0,255,0 equates to pure green; 0,0,255 produces a pure blue.

Example: the command `#BackgroundColor(255,255,0)` turns the Custom Summary window a brilliant yellow.

If you use this command to signify an alert of some kind, be sure and return the window to its normal color once the alert condition is no longer in effect.

#ConfigureSerialOutput('parameter', value)

With this command you may configure several parameters pertaining to a serial port. This command is normally used in conjunction with the `#SerialOutput` command. Here are the permissible parameters for this command:

#ConfigureSerialOutput('BaudRate', value)

The default baud rate is 9600.

#ConfigureSerialOutput('DataBits', value)

The default number of data bits is 8. Permissible values are 4, 5, 6, 7, and 8.

#ConfigureSerialOutput('Parity', 'value')

The default parity is 'none'. Permissible values are 'None', 'Odd', 'Even', 'Space', and 'Mark'.

#ConfigureSerialOutput('StopBits', value)

The default value is 1. Permissible values are 1, 1.5, and 2.

#ConfigureSerialOutput('ComPort', 'value')

The default value is 'COM2'.

Any errors encountered during the execution of the #ConfigureSerialOutput command will be written to the Custom Summary window.

#CustomSummaryUpdateInterval(Minutes)

Sets the number of minutes between updates of the Custom Summary window. The normal update interval is 1 minute. Permissible values for this command are from 1/60 to 60.

Any errors encountered during the execution of the #CustomSummaryUpdateInterval command will be written to the Custom Summary window.

#PerformFTP

Causes an immediate FTP transfer of all new screen captures (since the last FTP) to be started. For an example of how this command is used, see the #ProduceScreenCapture command.

In order to send e-mails or text messages, you will need to fill out the server information in the E-Mail Options dialog (**Options | Internet | E-Mail...** from the main menu).

An example of how to use this command can be found in Appendix E.

#ProduceScreenCapture

This command causes a screen capture of the specified window to be produced immediately. By using this command, screen captures of all windows may be produced during the execution of the custom summary rather than the normal process of the Screen Captures dialog.

Here are the permissible forms for the command:

```
#ProduceScreenCapture(
  'Real-time Lightning',
  'filename',
  width,
  height )

#ProduceScreenCapture(
  'Real-time Lightning',
  'filename',
  width, height,
  'place name', radius )

#ProduceScreenCapture(
  'Raw Lightning Data',
  'filename',
  width, height )

#ProduceScreenCapture(
  'Graph',
  'filename',
  width, height )

#ProduceScreenCapture(
  'Analysis',
  'filename',
  width, height )

#ProduceScreenCapture(
  'Threat Assessment',
  'filename',
  width, height )

#ProduceScreenCapture(
  'Rates',
  'filename',
  percentage )

#ProduceScreenCapture(
  'Totals',
  'filename',
  percentage )

#ProduceScreenCapture( 'Nowcast', 'filename' )
#ProduceScreenCapture( 'Summary', 'filename' )
#ProduceScreenCapture( 'Custom Summary', 'filename' )
```

'Filename' is the name of the file into which the screen capture is written. Width and height are the dimensions of the screen capture image. Percentage is the relative size (from 1 to 100) of the full-sized image.

The second form of the Real-time Lightning screen capture allows you to produce a screen capture image centered on a specific place. The radius is the distance from the center of the image to the edge, in miles or kilometers (depending on the setting selected by the “Options | Metric units” menu command).

Any errors encountered during the execution of the #ProduceScreenCapture command will be written to the Custom Summary window.

An example of how to use this command is in Appendix E.

#SendEmail

Causes an email to be sent. The contents of the email are defined in the Email Options dialog. The email will be sent even if automatically generated emails are not enabled in the Email Options dialog. However no email will be sent if the required information (server name, recipients, etc.) has not been entered into the Email Options dialog.

You will need to select the “Enabled” option in the Email Options dialog in order to enter the required information. Click the “Enabled” checkbox again before closing the dialog if you do not wish emails to be automatically generated. The information entered in the dialog will be saved whether or not the “Enabled” button is checked.

Here is an example of how this command is typically used:

```
; if there have been some strong/nearby flashes in the  
past 5 minutes  
; send an email
```

```
if ( @StrongFlashCount( 5 ) > 0 ) and
```

```
( @MinutesSinceEmailSent >= 30 ) then
#SendEmail
```

Other permissible formats of this command are:

```
#SendEmail( 'recipients' )
#SendEmail( 'recipients', 'email title' )
#SendEmail( 'recipients', 'email title', 'attachment',
..., 'attachment' )
```

This form of the command is used to send an email to a specific recipient, rather than the recipients listed in the Email Options dialog.

Recipients is a string specifying one or more email addresses. If multiple addresses are listed, they should be separated by commas. For example:

```
#SendEmail( 'joe@Pittsburgh.com,fred@nyc.com' ).
```

If no title is specified, a title will be generated telling how many flashes per minute are currently being detected.

In the expanded form of this command, an attachment may be one of three things: (1) a string representing one of the standard email body text types ('Custom Summary', 'Status Line', 'Nowcast', or 'Summary'), (2) the name of an image file to attach to the email or, (3) the name of a text file (one whose extension is '.txt').

If one or more text files are specified as attachments, the contents of those files will be used as the body of the email and not actually "attached" to the email. For example, you could create a single-line text file to send as a text message to one or more recipients.

#SerialOutput('string')

Causes 'string' to be sent to a serial port. The name of the serial port may be specified by using the #ConfigureSerialOutput command.

Any errors encountered during the execution of the #SerialOutput command will be written to the Custom Summary window.

#Sound('filename')

Causes a sound to be played. If only a filename is specified (such as 'beep3'), the file must exist in either the Lightning/2000 installation folder or the "Sounds" sub-folder of the installation folder.

Any errors encountered during the execution of the #Sound command will be written to the Custom Summary window.

#Squelch(SquelchLevel)

Changes the squelch level of the lightning detector. Legal squelch levels can be from 0 to 15. A squelch level of 0 results in the highest sensitivity level for the detector.

Any errors encountered during the execution of the #Squelch command will be written to the Custom Summary window.

#Write(expression)**#Write(expression, expression, ..., expression)**

Writes text to the custom summary window. Text written following anything written with this command will appear on the same line. The expression may be either a quoted string, a mathematical expression (including keywords), or a combination. Multiple elements in the same #Write command may be concatenated with plus signs, or they may be separated by commas.

Examples of #Write commands:

```
#write( 'A thunderstorm is nearby' )
#write( 'The record was set on ' +
@RecordTotalStrokesDate )
#write( 'a' + 'b' + 'c' + 1 ) ; writes "abc1"
#write( 'a', 'b', 'c', 1 ) ; also writes "abc1"
```

#WriteFile('filename', expression)**#WriteFile('filename', expression, expression, ..., expression)**

Writes a line of text to a file, erasing the previous contents (if any) of the file. The name of the file is specified in the 'filename' parameter. The expression may be either a quoted string, a mathematical expression (including keywords), or a combination. Multiple elements in the same #WriteFile command may be concatenated with plus signs, or they may be separated by commas. Examples of #WriteFile commands:

```
#writeFile( @L2kfolder + 'ftplog.txt', 'FTP started at
' + @LocalTime )
```

```
#writeFile( 'C:\test.txt', 'just testing' )
```

Any errors encountered during the execution of the #WriteFile command will be written to the Custom Summary window.

#WriteLine(expression)

#WriteLine(expression, expression, ..., expression)

Writes text to the custom summary window. Text written following anything written with this command will appear on a new line. The expression may be a quoted string, a mathematical expression (including keywords), or a combination. Multiple elements in the same #WriteLine command may be concatenated with plus signs, or they may be separated by commas.

Examples of #WriteLine commands:

```
#writeLine( 'this is a line of just text' )
#writeLine( 'There are ' + @StrokeCount + ' strokes
per minute' )
#writeLine( 4 )
#writeLine( 'There are ', @StrokeCount, ' strokes
per minute' )
```

Count Keywords

Keywords that evaluate to a per minute stroke or flash count may be followed by a parameter. A parameter is a number or expression in parentheses. Then the count will evaluate to the number of strokes or flashes in the last N minutes, where N is the parameter.

The parameter "N" may be either a positive integer, or a positive real number. Real numbers whose absolute value is less than 1 must start with a zero.

For example, "0.1" is a valid parameter, but ".1" is not.

For example, the **@CGFlashCount** keyword will evaluate to the number of CG flashes in the last minute, but if you specify **@CGFlashCount(2)**, then it will evaluate to the number of CG flashes in the last 2 minutes. **@CGFlashCount(1)** is identical to **@CGFlashCount**, and is also the same as **@CGFlashCount(0)**.

A value of N that is greater than 60 could return an incorrect count.

Keywords are not case sensitive, so **@cgflashcount** is identical to **@CGFlashCount** or **@CGFLASHCOUNT**.

@CGFlashCount(N)

The number of CG flashes in the last N minutes.

@CGNegativeFlashCount(N)

The number of -CG flashes in the last N minutes.

@CGNegativeStrokeCount(N)

The number of -CG strokes in the last N minutes.

@CGPositiveFlashCount(N)

The number of +CG flashes in the last N minutes.

@CGPositiveStrokeCount(N)

The number of +CG strokes in the last N minutes.

@CGStrokeCount(N)

The number of CG strokes in the last N minutes.

@CIDStrokeCount(N)

The number of CID strokes in the last N minutes.

@CloseSevereThunderstormCount

The number of storms (as defined in the analysis window) determined to be severe thunderstorms that are at least partly inside the red alert range ring.

@CloseStormCount

The number of storms (as identified in the Analysis window) that are at least partly inside the red alert range ring.

@CloseStrongThunderstormCount

The number of storms (as defined in the analysis window) determined to be strong thunderstorms that are at least partly inside the red alert range ring.

@CloseThundershowerCount

The number of storms (as defined in the analysis window) determined to be thundershowers that are at least partly inside the red alert range ring.

@CloseThunderstormCount

The number of storms (as defined in the analysis window) determined to be thunderstorms that are at least partly inside the red alert range ring.

@DetectionCount(N)

The number of strokes plus the number of noises in the last N minutes. This is a count of the total number of detections.

@DistantSevereThunderstormCount

The number of storms (as defined in the analysis window) determined to be severe thunderstorms that are completely outside the yellow alert range ring.

@DistantStormCount

The number of storms (as identified in the Analysis window) that are completely outside the yellow alert range ring.

@DistantStrongThunderstormCount

The number of storms (as defined in the analysis window) determined to be strong thunderstorms that are completely outside the yellow alert range ring.

@DistantThundershowerCount

The number of storms (as defined in the analysis window) determined to be thundershowers that are completely outside the yellow alert range ring.

@DistantThunderstormCount

The number of storms (as defined in the analysis window) determined to be thunderstorms that are completely outside the yellow alert range ring.

@FlashCount(N)

The total number of flashes in the last N minutes.

@ICFlashCount(N)

The number of IC flashes in the last N minutes.

@ICNegativeFlashCount(N)

The number of -IC flashes in the last N minutes.

@ICNegativeStrokeCount(N)

The number of -IC strokes in the last N minutes.

@ICPositiveFlashCount(N)

The number of +IC flashes in the last N minutes.

@ICPositiveStrokeCount(N)

The number of +IC strokes in the last N minutes.

@ICStrokeCount(N)

The number of IC strokes in the last N minutes.

@NoiseCount(N)

The number of noises in the last N minutes.

@RegionalSevereThunderstormCount

The number of storms (as defined in the analysis window) determined to be severe thunderstorms that are outside the red alert range ring but inside the yellow alert range ring.

@RegionalStormCount

The number of storms (as identified in the Analysis window) that are closer than the yellow alert range, but further than the red alert range.

@RegionalStrongThunderstormCount

The number of storms (as defined in the analysis window) determined to be strong thunderstorms that are outside the red alert range ring but inside the yellow alert range ring.

@RegionalThundershowerCount

The number of storms (as defined in the analysis window) determined to be thundershowers that are outside the red alert range ring but inside the yellow alert range ring.

@RegionalThunderstormCount

The number of storms (as defined in the analysis window) determined to be thunderstorms that are outside the red alert range ring but inside the yellow alert range ring.

@StrokeCount(N)

The total number of strokes in the last N minutes.

@StrongFlashCount(N)

The number of strong (potentially nearby) flashes in the last N minutes.

@TotalStormCount

The current total number of storms, as identified in the analysis window.

Date/Time Keywords

Keywords are not case sensitive, so **@localdate** is identical to **@LocalDate** or **@LOCALDATE**.

@DateFilename

Returns a string suitable for use in a filename that includes the date. The format is always `yyyymmdd`. For example, if the date is December 25, 2010 then the returned string will be '20101225'.

@DateTimeFilename

Returns a string suitable for use in a filename that includes the date and time to the nearest second. The format is always `yyyymmddhhmmss`. For example, if the time is 11:25:30 PM on July 2, 2010, then '20100702232530' will be returned.

@DayOfWeek

Returns a number from 1 to 7 representing the day of the week. Sunday is 1, Monday is 2, etc.

@HourOfDay

Returns a number from 0 to 23 representing the hour of the day.

@LocalDate

The current date, in the format defined on your computer.

@LocalTime

The local time, in the format defined on your computer.

@MinutesSinceMidnight

The number of minutes, to within a tenth of a minute, that have elapsed since midnight local time.

@PeakCGFlashCountTime

The time of today's peak CG flash count.

@PeakCGNegativeFlashCountTime

The time of today's peak -CG flash count.

@PeakCGNegativeStrokeCountTime

The time of today's peak -CG stroke count.

@PeakCGPositiveFlashCountTime

The time of today's peak +CG flash count.

@PeakCGPositiveStrokeCountTime

The time of today's peak +CG stroke count.

@PeakCGStrokeCountTime

The time of today's peak CG stroke count.

@PeakCIDCountTime

The time of today's peak CID count.

@PeakEnergyCountTime

The time of today's peak energy count.

@PeakEnergyRatioTime

The time of today's peak energy ratio.

@PeakFlashCountTime

The time of today's peak total flash count.

@PeakICFlashCountTime

The time of today's peak IC flash count.

@PeakICNegativeFlashCountTime

The time of today's peak -IC flash count.

@PeakICNegativeStrokeCountTime

The time of today's peak -IC stroke count.

@PeakICPositiveFlashCountTime

The time of today's peak +IC flash count.

@PeakICPositiveStrokeCountTime

The time of today's peak +IC stroke count.

@PeakICStrokeCountTime

The time of today's peak IC stroke count.

@PeakNoiseCountTime

The time of today's peak noise count.

@PeakStrokeCountTime

The time of today's peak stroke count.

@PeakStrongFlashCountTime

The time of today's peak strong (potentially nearby) flash count.

@RecordCGFlashCountDate

The date of the all-time record CG flash count.

@RecordCGFlashCountTime

The time of the all-time record CG flash count.

@RecordCGNegativeFlashCountDate

The date of the all-time record -CG flash count.

@RecordCGNegativeFlashCountTime

The time of the all-time record -CG flash count.

@RecordCGNegativeStrokeCountDate

The date of the all-time record -CG stroke count.

@RecordCGNegativeStrokeCountTime

The time of the all-time record -CG stroke count.

@RecordCGPositiveFlashCountDate

The date of the all-time record +CG flash count.

@RecordCGPositiveFlashCountTime

The time of the all-time record +CG flash count.

@RecordCGPositiveStrokeCountDate

The date of the all-time record +CG stroke count.

@RecordCGPositiveStrokeCountTime

The time of the all-time record +CG stroke count.

@RecordCGStrokeCountDate

The date of the all-time record CG stroke count.

@RecordCGStrokeCountTime

The time of the all-time record CG stroke count.

@RecordCIDCountDate

The date of the all-time record CID count.

@RecordCIDCountTime

The time of the all-time record CID count.

@RecordEnergyRatioDate

The date of the all-time record energy ratio.

@RecordEnergyRatioTime

The time of the all-time record energy ratio.

@RecordFlashCountDate

The date of the all-time record flash count.

@RecordFlashCountTime

The time of the all-time record flash count.

@RecordICFlashCountDate

The date of the all-time record IC flash count.

@RecordICFlashCountTime

The time of the all-time record IC flash count.

@RecordICNegativeFlashCountDate

The date of the all-time record -IC flash count.

@RecordICNegativeFlashCountTime

The time of the all-time record -IC flash count.

@RecordICNegativeStrokeCountDate

The date of the all-time record -IC stroke count.

@RecordICNegativeStrokeCountTime

The time of the all-time record -IC stroke count.

@RecordICPositiveFlashCountDate

The date of the all-time record +IC flash count.

@RecordICPositiveFlashCountTime

The time of the all-time record +IC flash count.

@RecordICPositiveStrokeCountDate

The date of the all-time record +IC stroke count.

@RecordICPositiveStrokeCountTime

The time of the all-time record +IC stroke count.

@RecordICStrokeCountDate

The date of the all-time record IC stroke count.

@RecordICStrokeCountTime

The time of the all-time record IC stroke count.

@RecordNoiseCountDate

The date of the all-time record noise count.

@RecordNoiseCountTime

The time of the all-time record noise count.

@RecordStrokeCountDate

The date of the all-time record stroke count.

@RecordStrokeCountTime

The time of the all-time record stroke count.

@RecordStrongFlashCountDate

The date of the all-time record strong (potentially nearby) flash count.

@RecordStrongFlashCountTime

The time of the all-time record strong (potentially nearby) flash count.

@RecordTotalCGFlashes

The record daily total of CG flashes.

@RecordTotalCGFlashesDate

The date of the record daily CG flash total.

@RecordTotalCGNegativeFlashes

The record daily total of -CG flashes.

@RecordTotalCGNegativeFlashesDate

The date of the record daily -CG flash total.

@RecordTotalCGNegativeStrokes

The record daily total of -CG strokes.

@RecordTotalCGNegativeStrokesDate

The date of the record daily -CG stroke total.

@RecordTotalCGPositiveFlashes

The record daily total of +CG flashes.

@RecordTotalCGPositiveFlashesDate

The date of the record daily +CG flash total.

@RecordTotalCGPositiveStrokes

The record daily total of +CG strokes.

@RecordTotalCGPositiveStrokesDate

The date of the record daily +CG stroke total.

@RecordTotalCGStrokes

The record daily total of CG strokes.

@RecordTotalCGStrokesDate

The date of the record daily CG stroke total.

@RecordTotalCID

The record daily total of CIDs.

@RecordTotalCIDDate

The date of the record daily CID total.

@RecordTotalEnergy

The record daily total of energy.

@RecordTotalEnergyDate

The date of the record daily energy total.

@RecordTotalFlashes

The record daily total of flashes.

@RecordTotalFlashesDate

The date of the record daily flash total.

@RecordTotalICStrokes

The record daily total of IC strokes.

@RecordTotalICStrokesDate

The date of the record daily IC stroke total.

@RecordTotalICFlashes

The record daily total of IC flashes.

@RecordTotalICFlashesDate

The date of the record daily IC flash total.

@RecordTotalICNegativeFlashes

The record daily total of -IC flashes.

@RecordTotalICNegativeFlashesDate

The date of the record daily -IC flash total.

@RecordTotalICNegativeStrokes

The record daily total of -IC strokes.

@RecordTotalICNegativeStrokesDate

The date of the record daily -IC stroke total.

@RecordTotalICPositiveFlashes

The record daily total of +IC flashes.

@RecordTotalICPositiveFlashesDate

The date of the record daily +IC flash total.

@RecordTotalICPositiveStrokes

The record daily total of +IC strokes.

@RecordTotalICPositiveStrokesDate

The date of the record daily +IC stroke total.

@RecordTotalNoises

The record daily total of noises.

@RecordTotalNoisesDate

The date of the record daily noise total.

@RecordTotalStrokes

The record daily total of strokes.

@RecordTotalStrokesDate

The date of the record daily stroke total.

@RecordTotalStrongFlashes

The record daily total of strong (potentially nearby) flashes.

@RecordTotalStrongFlashesDate

The date of the record daily total of strong (potentially nearby) flashes.

@TimeZone

The time zone abbreviation for the current time zone on your computer. For example, if you are on Central Standard Time, this keyword will return CST.

@UTDate

The current date in Universal Time (GMT+0).

@UTTime

The current Universal Time (GMT+0)

Miscellaneous Keywords

Keywords are not case sensitive, so **@localdate** is identical to **@LocalDate** or **@LOCALDATE**.

@AntennaType

Returns 0 for the original Boltek antenna, 1 for the original prototype, and 2 for the final prototype.

@CloseSevereThunderstormCount

The number of storms (as defined in the analysis window) determined to be severe thunderstorms that are at least partly inside the red alert range ring.

@CloseStormCount

The number of storms (as identified in the Analysis window) that are inside the red alert range ring.

@CloseStrongThunderstormCount

The number of storms (as defined in the analysis window) determined to be strong thunderstorms that are at least partly inside the red alert range ring.

@CloseThundershowerCount

The number of storms (as defined in the analysis window) determined to be thundershowers that are at least partly inside the red alert range ring.

@CloseThunderstormCount

The number of storms (as defined in the analysis window) determined to be thunderstorms that are at least partly inside the red alert range ring.

@CommandErrorCode

Returns a value indicating whether or not the most recently executed command was processed successfully. If the command was processed with no problems, 0 is returned by this keyword. If a problem was encountered before the command could be executed, 1 is returned. If a problem was encountered during the execution of the command, -1 is returned.

@DateFilename

Returns a string suitable for use in a filename that includes the date. The format is always `yyyymmdd`. For example, if the date is December 25, 2010 then the returned string will be '20101225'.

@DateTimeFilename

Returns a string suitable for use in a filename that includes the date and time to the nearest second. The format is always `yyyymmddhhmmss`. For example, if the time is 11:25:30 PM on July 2, 2010, then '20100702232530' will be returned.

@DayOfWeek

Returns a number from 1 to 7 representing the day of the week. Sunday is 1, Monday is 2, etc.

@DistantSevereThunderstormCount

The number of storms (as defined in the analysis window) determined to be severe thunderstorms that are completely outside the yellow alert range ring.

@DistantStormCount

The number of storms (as identified in the Analysis window) that are outside the yellow alert range ring.

@DistantStrongThunderstormCount

The number of storms (as defined in the analysis window) determined to be strong thunderstorms that are completely outside the yellow alert range ring.

@DistantThundershowerCount

The number of storms (as defined in the analysis window) determined to be thundershowers that are completely outside the yellow alert range ring.

@DistantThunderstormCount

The number of storms (as defined in the analysis window) determined to be thunderstorms that are completely outside the yellow alert range ring.

@FormatInteger(N)

Formats the parameter N so that thousands separators are inserted between groups of digits. For example, @FormatInteger(2345678) would return the string 2,345,678 if the thousands separator character is a comma, and 2.345.678 if the thousands separator is a period.

@HardwareType

Returns a number indicating what type of lightning detection hardware is enabled. Valid values are: 0 (no hardware enabled), 1 (Boltek LD-250), 2 (Boltek PCI card), 3 (Boltek ISA card), and 4 (Boltek LD-350).

@HourOfDay

Returns a number from 0 to 23 representing the hour of the day.

@InitialMemoryUsagePhysical

The amount of memory in use by the computer in megabytes just before starting Lightning/2000.

@InitialMemoryUsageVirtual

The amount of virtual memory in use by the computer in megabytes just before starting Lightning/2000.

@L2KFolder

Returns the name of the folder where the Lightning/2000 executable file (lightning.exe) is located. The backslash at the end of the folder name is included.

@LightningDetectedIn('place')

Returns true if a lightning flash is detected in the past 10 minutes in a rectangular area that bounds 'place'. You may specify the name of a city, town, place, or region that has been added to the background map. For maps containing U.S. counties, you may specify the name of a county, for example, 'Davidson County,

TN' or 'Davidson, TN'. Place names are not case sensitive.

This keyword can be used to insert special text in the custom summary if lightning is detected in the specified place. It can also be used to send an email to someone in the specified place if lightning has been detected in their area. For example:

```
If @MinutesSinceEmailSent( 'sue@abc.com' ) >= 30 then
  If @LightningDetectedIn( 'Nashville' ) then
    #ProduceScreenCapture( 'Real-time Lightning',
    'x.png', 'Nashville', 50 )
    #SendEmail( 'sue@abc.com', 'Lightning in
    Nashville area', 'custom summary', 'x.png' )
  endif
endif
```

@LocalDate

The current date, in the format defined on your computer.

@LocalTime

The local time, in the format defined on your computer.

@MaxCloseIntensity

The maximum intensity of any storm that lies at least partly inside the red alert range ring. Possible return values are: 0 - no storm is present, 1 - a thundershower, 2 - a thunderstorm, 3 - a strong thunderstorm, 4 - a severe thunderstorm.

@MaxDistantIntensity

The maximum intensity of any storm that lies at least outside the yellow alert range ring. Possible return values are: 0 - no storm is present, 1 - a

thundershower, 2 - a thunderstorm, 3 - a strong thunderstorm, 4 - a severe thunderstorm.

@MaxRegionalIntensity

The maximum intensity of any storm that lies outside the red alert range ring, but at least partly inside the yellow alert range ring. Possible return values are: 0 - no storm is present, 1 - a thundershower, 2 - a thunderstorm, 3 - a strong thunderstorm, 4 - a severe thunderstorm.

@MemoryUsagePhysical

The amount of physical memory in use by the computer in megabytes.

@MemoryUsageVirtual

The amount of virtual memory in use by the computer in megabytes.

@MinutesSinceEmailSent

@MinutesSinceEmailSent('recipient')

The number of minutes since Lightning/2000 last sent out an automatically generated email.

The second form of the keyword returns the number of minutes since an email was sent to the specified recipient by using the #SendEMail command.

@MinutesSinceFTP

The number of minutes since Lightning/2000 performed an automatic FTP to a server.

@MinutesSinceMidnight

The number of minutes, to within a tenth of a minute, that have elapsed since midnight local time.

@MinutesSinceTweet

The number of minutes since Lightning/2000 last sent a status update to your Twitter account.

Twitter status updates will be performed no more often than every five minutes.

@NearbyStormRange

The range in miles (or km if using metric units) to the center of the special “nearby” storm depicted in the Threat Assessment window.

@NumRedAlerts

The current number of red alerts in effect.

@NumYellowAlerts

The current number of yellow alerts in effect.

@RegionalSevereThunderstormCount

The number of storms (as defined in the analysis window) determined to be severe thunderstorms that are outside the red alert range ring, but at least partly inside the yellow alert range ring.

@RegionalStormCount

The number of storms (as identified in the Analysis window) that are closer than the yellow alert range, but further than the red alert range.

@RegionalStrongThunderstormCount

The number of storms (as defined in the analysis window) determined to be strong thunderstorms that are outside the red alert range ring, but at least partly inside the yellow alert range ring.

@RegionalThundershowerCount

The number of storms (as defined in the analysis window) determined to be thundershowers that are outside the red alert range ring, but at least partly inside the yellow alert range ring.

@RegionalThunderstormCount

The number of storms (as defined in the analysis window) determined to be thunderstorms that are outside the red alert range ring, but at least partly inside the yellow alert range ring.

@TimeZone

The time zone abbreviation.

@TotalStormCount

The current number of storms, as identified in the analysis window.

@UTDate

The current date in Universal Time (GMT+0).

@UTTime

The current Universal Time (GMT+0)

@VersionNumber

A string representing the version number of Lightning/2000.

Peak Count Keywords

The **@PeakXXXCount** keywords return either the peak count for the day (since midnight local time), or the peak count within the last N minutes.

The parameter "N" may be either a positive integer, or a positive real number. Real numbers less than 1 must start with a zero.

For example, "0.1" is a valid parameter, but ".1" is not.

The **@PeakXXXCountTime** keywords return the time of the peak count for the day.

Keywords are not case sensitive, so **@peakcgflashcount** is identical to **@PeakCGFlashCount** or **@PEAKCGFLASHCOUNT**.

@PeakCGFlashCount(N)

Today's peak CG flash count, or the peak CG flash count in the last N minutes.

@PeakCGFlashCountTime

The time of today's peak CG flash count.

@PeakCGNegativeFlashCount(N)

Today's peak -CG flash count, or the peak -CG flash count in the last N minutes.

@PeakCGNegativeFlashCountTime

The time of today's peak -CG flash count.

@PeakCGNegativeStrokeCount(N)

Today's peak -CG stroke count, or the peak -CG stroke count in the last N minutes.

@PeakCGNegativeStrokeCountTime

The time of today's peak -CG stroke count.

@PeakCGPositiveFlashCount(N)

Today's peak +CG flash count, or the peak +CG flash count in the last N minutes.

@PeakCGPositiveFlashCountTime

The time of today's peak +CG flash count.

@PeakCGPositiveStrokeCount(N)

Today's peak +CG stroke count, or the peak +CG stroke count in the last N minutes.

@PeakCGPositiveStrokeCountTime

The time of today's peak +CG stroke count.

@PeakCGStrokeCount(N)

Today's peak CG stroke count, or the peak CG stroke count in the last N minutes.

@PeakCGStrokeCountTime

The time of today's peak CG stroke count.

@PeakCIDCount(N)

Today's peak CID count, or the peak CID count in the last N minutes.

@PeakCIDCountTime

The time of today's peak CID count.

@PeakEnergyCount

Today's peak energy count.

@PeakEnergyCountTime

The time of today's peak energy count.

@PeakEnergyRatio(N)

Today's peak energy ratio, or the peak energy ratio in the past N minutes.

@PeakEnergyRatioTime

The time of today's peak energy ratio.

@PeakFlashCount(N)

Today's peak total flash count, or the peak flash count in the last N minutes.

@PeakFlashCountTime

The time of today's peak total flash count.

@PeakICFlashCount(N)

Today's peak IC flash count, or the peak IC flash count in the last N minutes.

@PeakICFlashCountTime

The time of today's peak IC flash count.

@PeakICNegativeFlashCount(N)

Today's peak -IC flash count, or the peak -IC flash count in the last N minutes.

@PeakICNegativeFlashCountTime

The time of today's peak -IC flash count.

@PeakICNegativeStrokeCount(N)

Today's peak -IC stroke count, or the peak -IC stroke count in the last N minutes.

@PeakICNegativeStrokeCountTime

The time of today's peak -IC stroke count.

@PeakICPositiveFlashCount(N)

Today's peak +IC flash count, or the peak +IC flash count in the last N minutes.

@PeakICPositiveFlashCountTime

The time of today's peak +IC flash count.

@PeakICPositiveStrokeCount(N)

Today's peak +IC stroke count, or the peak +IC stroke count in the last N minutes.

@PeakICPositiveStrokeCountTime

The time of today's peak +IC stroke count.

@PeakICStrokeCount(N)

Today's peak IC stroke count, or the peak IC stroke count in the last N minutes.

@PeakICStrokeCountTime

The time of today's peak IC stroke count.

@PeakNoiseCount(N)

Today's peak noise count, or the peak noise count in the last N minutes.

@PeakNoiseCountTime

The time of today's peak noise count.

@PeakStrokeCount(N)

Today's peak stroke count, or the peak stroke count in the last N minutes.

@PeakStrokeCountTime

The time of today's peak stroke count.

@PeakStrongFlashCount(N)

Today's peak strong (potentially nearby) flash count, or the peak strong flash count in the last N minutes.

@PeakStrongFlashCountTime

The time of today's peak strong (potentially nearby) flash count.

Percent and Ratio Keywords

Each of these keywords may be used with a (N) parameter. If the (N) parameter is not used, the keyword returns the percentage or ratio in the last minute.

The parameter "N" may be either a positive integer, or a positive real number. Real numbers less than 1 must start with a zero.

For example, "0.1" is a valid parameter, but ".1" is not.

Legal values for the parameter N range from 0 to 60. Values for N greater than 60 will not return the correct value.

Keywords are not case sensitive, so **@PercentageICFlashes** is identical to **@percentageicflashes** or **@PERCENTAGEICFLASHES**.

@PercentageCGPositiveFlashes(N)

The percentage of CG flashes that are +CG flashes in the last N minutes. The percentage is a number between 0 and 100.

@PercentageCIDFlashes(N)

The percentage of total flashes that are CID flashes in the last N minutes. The percentage is a number between 0 and 100.

@PercentageICFlashes(N)

The percentage of total flashes that are IC flashes in the last N minutes. The percentage is a number between 0 and 100.

@PercentageICStrokes(N)

The percentage of total strokes that are IC strokes in the last N minutes. The percentage is a number between 0 and 100.

@PercentageStrongFlashes(N)

The percentage of total flashes that are strong (potentially nearby) flashes in the last N minutes. The percentage is a number between 0 and 100.

@RatioCGPositiveToCGNegativeFlashes(N)

The ratio of the +CG flash count to the -CG flash count over the last N minutes.

@RatioCGPositiveToCGNegativeStrokes(N)

The ratio of the +CG stroke count to the -CG stroke count over the last N minutes.

@RatioCGStrokesPerFlash(N)

The number of cloud-to-ground strokes per cloud-to-ground flash. If the parameter N is omitted, the keyword returns the number of CG strokes per CG flash in the last minute. Values for N greater than 60 minutes will not produce the correct output. This value typically increases as a storm approaches.

@RatioICPositiveToICNegativeFlashes(N)

The ratio of the +IC flash count to the -IC flash count over the last N minutes.

@RatioICPositiveToICNegativeStrokes(N)

The ratio of the +IC stroke count to the -IC stroke count over the last N minutes.

@RatioICStrokesPerFlash(N)

The number of intercloud strokes per intercloud flash. If the parameter N is omitted, the keyword returns the number of IC strokes per IC flash in the last minute. Values for N greater than 60 minutes will not produce the correct output. This value typically increases as a storm approaches.

@RatioICToCGFlashes(N)

The ratio of the IC flash count to the CG flash count over the last N minutes.

@RatioICToCGStrokes(N)

The ratio of the IC stroke count to the CG stroke count over the last N minutes.

@RatioStrokesPerFlash(N)

The number of strokes per flash. If the parameter N is omitted, the keyword returns the number of strokes per flash in the last minute. Values for N greater than 60 minutes will not produce the correct output. This value typically increases as a storm approaches.

Records Keywords

The **@Record** keywords return information about record values for the various counts that have been set on your computer.

You can use the **@Record** keywords to display either the record value, the date of the record value, or the time of day that the record value was set.

Keywords are not case sensitive, so **@recordicflashcount** is identical to **@RecordICFlashCount** or **@RECORDICFLASHCOUNT**.

@RecordCGFlashCount

The all-time record CG flash count.

@RecordCGFlashCountDate

The date of the all-time record CG flash count.

@RecordCGFlashCountTime

The time of the all-time record CG flash count.

@RecordCGNegativeFlashCount

The all-time record -CG flash count.

@RecordCGNegativeFlashCountDate

The date of the all-time record -CG flash count.

@RecordCGNegativeFlashCountTime

The time of the all-time record -CG flash count.

@RecordCGNegativeStrokeCount

The all-time record -CG stroke count.

@RecordCGNegativeStrokeCountDate

The date of the all-time record -CG stroke count.

@RecordCGNegativeStrokeCountTime

The time of the all-time record -CG stroke count.

@RecordCGPositiveFlashCount

The all-time record +CG flash count.

@RecordCGPositiveFlashCountDate

The date of the all-time record +CG flash count.

@RecordCGPositiveFlashCountTime

The time of the all-time record +CG flash count.

@RecordCGPositiveStrokeCount

The all-time record +CG stroke count.

@RecordCGPositiveStrokeCountDate

The date of the all-time record +CG stroke count.

@RecordCGPositiveStrokeCountTime

The time of the all-time record +CG stroke count.

@RecordCGStrokeCount

The all-time record CG stroke count.

@RecordCGStrokeCountDate

The date of the all-time record CG stroke count.

@RecordCGStrokeCountTime

The time of the all-time record CG stroke count.

@RecordCIDCount

The all-time record CID count.

@RecordCIDCountDate

The date of the all-time record CID count.

@RecordCIDCountTime

The time of the all-time record CID count.

@RecordEnergyRatio

The all-time record energy ratio.

@RecordEnergyRatioDate

The date of the all-time record energy ratio.

@RecordEnergyRatioTime

The time of the all-time record energy ratio.

@RecordFlashCount

The all-time record total flash count.

@RecordFlashCountDate

The date of the all-time record flash count.

@RecordFlashCountTime

The time of the all-time record flash count.

@RecordICFlashCount

The all-time record IC flash count.

@RecordICFlashCountDate

The date of the all-time record IC flash count.

@RecordICFlashCountTime

The time of the all-time record IC flash count.

@RecordICNegativeFlashCount

The all-time record -IC flash count.

@RecordICNegativeFlashCountDate

The date of the all-time record -IC flash count.

@RecordICNegativeFlashCountTime

The time of the all-time record -IC flash count.

@RecordICNegativeStrokeCount

The all-time record -IC stroke count.

@RecordICNegativeStrokeCountDate

The date of the all-time record -IC stroke count.

@RecordICNegativeStrokeCountTime

The time of the all-time record -IC stroke count.

@RecordICPositiveFlashCount

The all-time record +IC flash count.

@RecordICPositiveFlashCountDate

The date of the all-time record +IC flash count.

@RecordICPositiveFlashCountTime

The time of the all-time record +IC flash count.

@RecordICPositiveStrokeCount

The all-time record +IC stroke count.

@RecordICPositiveStrokeCountDate

The date of the all-time record +IC stroke count.

@RecordICPositiveStrokeCountTime

The time of the all-time record +IC stroke count.

@RecordICStrokeCount

The all-time record IC stroke count.

@RecordICStrokeCountDate

The date of the all-time record IC stroke count.

@RecordICStrokeCountTime

The time of the all-time record IC stroke count.

@RecordNoiseCount

The all-time record noise count.

@RecordNoiseCountDate

The date of the all-time record noise count.

@RecordNoiseCountTime

The time of the all-time record noise count.

@RecordStrokeCount

The all-time record stroke count.

@RecordStrokeCountDate

The date of the all-time record stroke count.

@RecordStrokeCountTime

The time of the all-time record stroke count.

@RecordStrongFlashCount

The all-time record strong (potentially nearby) flash count.

@RecordStrongFlashCountDate

The date of the all-time record strong (potentially nearby) flash count.

@RecordStrongFlashCountTime

The time of the all-time record strong (potentially nearby) flash count.

@RecordTotalCGFlashes

The record daily total of CG flashes.

@RecordTotalCGFlashesDate

The date of the record daily CG flash total.

@RecordTotalCGNegativeFlashes

The record daily total of -CG flashes.

@RecordTotalCGNegativeFlashesDate

The date of the record daily -CG flash total.

@RecordTotalCGNegativeStrokes

The record daily total of -CG strokes.

@RecordTotalCGNegativeStrokesDate

The date of the record daily -CG stroke total.

@RecordTotalCGPositiveFlashes

The record daily total of +CG flashes.

@RecordTotalCGPositiveFlashesDate

The date of the record daily +CG flash total.

@RecordTotalCGPositiveStrokes

The record daily total of +CG strokes.

@RecordTotalCGPositiveStrokesDate

The date of the record daily +CG stroke total.

@RecordTotalCGStrokes

The record daily total of CG strokes.

@RecordTotalCGStrokesDate

The date of the record daily CG stroke total.

@RecordTotalCID

The record daily total of CIDs.

@RecordTotalCIDDate

The date of the record daily CID total.

@RecordTotalEnergy

The record daily total of energy.

@RecordTotalEnergyDate

The date of the record daily energy total.

@RecordTotalFlashes

The record daily total of flashes.

@RecordTotalFlashesDate

The date of the record daily flash total.

@RecordTotalICStrokes

The record daily total of IC strokes.

@RecordTotalICStrokesDate

The date of the record daily IC stroke total.

@RecordTotalICFlashes

The record daily total of IC flashes.

@RecordTotalICFlashesDate

The date of the record daily IC flash total.

@RecordTotalICNegativeFlashes

The record daily total of -IC flashes.

@RecordTotalICNegativeFlashesDate

The date of the record daily -IC flash total.

@RecordTotalICNegativeStrokes

The record daily total of -IC strokes.

@RecordTotalICNegativeStrokesDate

The date of the record daily -IC stroke total.

@RecordTotalICPositiveFlashes

The record daily total of +IC flashes.

@RecordTotalICPositiveFlashesDate

The date of the record daily +IC flash total.

@RecordTotalICPositiveStrokes

The record daily total of +IC strokes.

@RecordTotalICPositiveStrokesDate

The date of the record daily +IC stroke total.

@RecordTotalNoises

The record daily total of noises.

@RecordTotalNoisesDate

The date of the record daily noise total.

@RecordTotalStrokes

The record daily total of strokes.

@RecordTotalStrokesDate

The date of the record daily stroke total.

@RecordTotalStrongFlashes

The record daily total of strong (potentially nearby) flashes.

@RecordTotalStrongFlashesDate

The date of the record daily strong (potentially nearby) flash total.

Total Count Keywords

Each of the **@Total** keywords will return the total number of strokes or flashes of various types since midnight local time.

Keywords are not case sensitive, so **@totalflashcount** is identical to **@TotalFlashCount** or **@TOTALFLASHCOUNT**.

@TotalCGFlashCount

The total number of CG flashes today.

@TotalCGNegativeFlashCount

The total number of -CG flashes today.

@TotalCGNegativeStrokeCount

The total number of -CG strokes today.

@TotalCGPositiveFlashCount

The total number of +CG flashes today.

@TotalCGPositiveStrokeCount

The total number of +CG strokes today.

@TotalCGStrokeCount

The total number of CG strokes today.

@TotalCIDCount

The total number of CIDs today.

@TotalEnergyCount

The total energy today.

@TotalFlashCount

The total number of flashes today.

@TotalICFlashCount

The total number of IC flashes today.

@TotalICNegativeFlashCount

The total number of -IC flashes today.

@TotalICNegativeStrokeCount

The total number of -IC strokes today.

@TotalICPositiveFlashCount

The total number of +IC flashes today.

@TotalICPositiveStrokeCount

The total number of +IC strokes today.

@TotalICStrokeCount

The total number of IC strokes today.

@TotalNoiseCount

The total number of noises today.

@TotalStormCount

The current number of storms, as identified in the analysis window.

@TotalStrokeCount

The total number of strokes today.

@TotalStrongFlashCount

The total number of strong (potentially nearby) flashes today.

Trend Keywords

The **@Trend** keywords return the percentage change in one of the stroke or flash rates. If the N parameter is not specified, the keyword returns the percent change in the rate between one minute ago and the current time. Values for N that are less than 1 are not allowed, and will return the same value as when no parameter is specified.

The parameter "N" may be either a positive integer, or a positive real number.

Legal values for N are from 1 to 1440.

The percent change can range from -100 to infinity, though for our purposes we limit the maximum percentage increase to 1000000.

Here are some examples:

If the stroke rate one minute ago was 100 per minute, and the stroke rate now is 80 per minute, the stroke rate has decreased by 20% in the last minute. **@TrendStrokeRate** will return -20; **@TrendStrokeRate(1)** will also return a value of -20.

If the stroke rate 5 minutes ago was 80 per minute, and the stroke rate now is 100, the stroke rate has increased by 25% in the last 5 minutes. **@TrendStrokeRate(5)** will return a value of 25.

If the stroke rate 10 minutes ago was 8 per minute, and the stroke rate now is 0, the stroke rate has decreased by 100% in the last 8 minutes. **@TrendStrokeRate(8)** will return -100.

If the stroke rate a minute ago was 0, and the stroke rate now is 1 per minute, the stroke rate has increased by a factor of infinity in the last minute. **@TrendStrokeRate** will return 1000000.

Keywords are not case sensitive, so **@trendcgflashrate** is identical to **@TrendCGFlashRate** or **@TRENDCGFLASHRATE**.

@TrendCGFlashRate(N)

The percentage change in the CG flash rate from N minutes ago to the current time.

@TrendCGNegativeFlashRate(N)

The percentage change in the -CG flash rate from N minutes ago to the current time.

@TrendCGPositiveFlashRate(N)

The percentage change in the +CG flash rate from N minutes ago to the current time.

@TrendCGStrokeRate(N)

The percentage change in the CG stroke rate from N minutes ago to the current time.

@TrendCGNegativeStrokeRate(N)

The percentage change in the -CG stroke rate from N minutes ago to the current time.

@TrendCGPositiveStrokeRate(N)

The percentage change in the +CG stroke rate from N minutes ago to the current time.

@TrendCIDRate(N)

The percentage change in the CID rate from N minutes ago to the current time.

@TrendDetectionRate(N)

The percentage change in the total detection rate from N minutes ago to the current time. Detections are the sum of the number of strokes per minute and the number of noises per minute. As it turns out, most noises (especially when the stroke rate is high and a storm is nearby) are actually strokes that are too garbled to be deciphered, so the sum of the stroke rate and the noise rate is a reasonably accurate measure of the total detection rate.

@TrendFlashRate(N)

The percentage change in the total flash rate from N minutes ago to the current time.

@TrendICFlashesPercentage(N)

The percentage change in the percentage of IC flashes from N minutes ago to the current time.

@TrendICFlashRate(N)

The percentage change in the IC flash rate from N minutes ago to the current time.

@TrendICNegativeFlashRate(N)

The percentage change in the -IC flash rate from N minutes ago to the current time.

@TrendICPositiveFlashRate(N)

The percentage change in the +IC flash rate from N minutes ago to the current time.

@TrendICStrokeRate(N)

The percentage change in the IC stroke rate from N minutes ago to the current time.

@TrendICNegativeStrokeRate(N)

The percentage change in the -IC stroke rate from N minutes ago to the current time.

@TrendICPositiveStrokeRate(N)

The percentage change in the +IC stroke rate from N minutes ago to the current time.

@TrendICStrokesPercentage(N)

The percentage change in the percentage of IC strokes from N minutes ago to the current time.

@TrendStrokeRate(N)

The percentage change in the total stroke rate from N minutes ago to the current time.

@TrendStrongFlashRate(N)

The percentage change in the strong (potentially nearby) flash rate from N minutes ago to the current time.