



NetCode Control (TM) Version 2.0

[Properties](#)

[Events](#)

[Error Messages](#)

[Licensing Information](#)

Description

The NetCode Control can be used to encode or decode files or strings. **UUEncode** as well as MIME's **Base64** and **Quoted-Printable** formats are currently supported.

File Name(s)

NETC20.VBX, NETCOD16.OCX, NETCOD32.OCX

Object Type

NetCode

Remarks

NetCode's operation is controlled by assigning a value to the [Action](#) property. The encoding format is specified by the [Format](#) property. The binary data, or the name of the file containing binary data, is specified by the [DecodedData](#) property and the encoded data or filename is specified by the [EncodedData](#) property. The [FileName](#) property may be used to override the default file names or to specify a directory during uuencoding. After uudecoding, the [FileName](#) property contains the fully specified file name of the decoded file.

Understanding Encoding/Decoding

Most of mail systems use only 7 bits to transmit messages. A binary file such as an archive or a non-plain text formatted file produced from a text processor should be first encoded in 7 bit code before email transfer.

Decoding is the inverse process: creation of the original file from the encoded data. Encoded data are often split over several files because of the size limit placed on the email message. Each message is preceded by information about splitting and also the mail header. NetCode supports this case in both directions:

- during *encoding* setting the [MaxFileSize](#) property to the maximum size of a message body instructs NetCode to split the encoded data over several files. Multiple filenames can be specified by using question marks "?".

- during *decoding*, the body messages can be saved in separate files named namexxx.ext where xxx is a numeral starting at 000 and ext is ".uue", ".b16", or ".q_p" according to [Format](#). This multiple filename should be given to [EncodedData](#). While uudecoding, if the [IntelliCode](#) property is set to True, NetCode tries to filter the data that don't belong to the pure uuencoded data (also the mail header).

Action Property

Description

Controls the operation of NetCode.

Usage

[*form.*][*netcodecontrol.*]**Action**[= *value%*]

Default Value

0

Remarks

The following are the possible values for the **Action** property and the corresponding descriptions:

0 (Idle)	Default.
1 (DecodeToFile)	<p>Decode with the format given in <u>Format</u>. Data is read from the <u>EncodedData</u> file(s) and written in the filename specified in the encoded data. Multiple filenames must exist as name000.ext, name001.ext etc. and specified as name??? in <u>EncodedData</u> (the number of digits/question marks is optional). The value of the extension <i>ext</i> is one of the following: <i>uue</i> for UUDecoding, <i>b64</i> for Base64 Decoding, and <i>q_p</i> for Quoted Printable Decoding, depending on the value of <u>Format</u>.</p> <p>When UUDecoding the created filename will by default be the one specified in the uuencoded data. <u>DecodedData</u> or, if <u>DecodedData</u> is empty, <u>FileName</u> can be used to override this value.</p> <p>If the string terminates with a backslash "\" it is interpreted as a directory in which to create decoded files. <u>FileName</u> contains the filename for the created file.</p>
2 (EncodeToFile)	<p>Encode with the format given in <u>Format</u>. Data is read from the file <u>DecodedData</u> and is encoded in the file(s) <u>EncodedData</u>. When uuencoding, the uuencoded data contain as filename the value given in <u>FileName</u> or, if empty, the value of <u>DecodedData</u>.</p> <p>While UUEncoding it is recommended to give the full file path in <u>EncodedData</u> and the filename in <u>FileName</u>.</p> <p>The current version of NetCode creates no message headers. If a MIME standard encoding is used such as Base64 Encoding or Quoted Printable Encoding the user should fill the header values appropriately.</p> <p>If <u>MaxFileSize</u> is set and the encoded data take more space than <u>MaxFileSize</u>, the data is split over several files. In this case, the user can specify more than one filename by ending the filename with question marks "?". NetCode expands them into numerals starting from 000 (as many digits as question marks specified) and appends the extension ".uue", ".b16", or ".q_p" (depending on the value of <u>Format</u>) to the filename.</p>

- 3 (DecodeToString) Decode according to Format: The string EncodedData is encoded into DecodedData. FileName contains the filename contained in the uuencoded data.
- 4 (EncodeToString) Encode according to Format: The string DecodedData (which may also contain binary data) is encoded into EncodedData. When uuencoding the filename is taken from the FileName property.

Data Type

Integer

DecodedData Property

Description

Filename of the decoded data or the decoded data itself.

Usage

```
[form.][netcodecontrol.]DecodedData[ = value$]
```

Default Value

""

Remarks

When uuencoding, the value of FileName is the filename that will be written in EncodedData. If FileName is empty, the value of the **DecodedData** property is taken. It is recommended that EncodedData contain the full path and FileName only the filename, so that no problems occur while uudecoding in a foreign system.

When decoding, the control tries to generate the name for the created file in the following order: **DecodedData**, FileName, or when the format is UUEncode, the specified filename in the uuencoded data itself. If **DecodedData** or FileName end with a backslash "\" they are interpreted as directories and the control tries to create the given filename in this directory. If this is the case, FileName shows the name of the created file.

Data Type

String

EncodedData Property

Description

Filename of the encoded data or the encoded data itself.

Usage

```
[form.][netcodecontrol.]EncodedData[ = value$]
```

Default Value

""

Remarks

If the encoded data are spread over several files, the filenames should exist as *path\file000.ext* and passed to **EncodedData** in the form *name???*. Question marks "?" are expanded to numerals starting with 000 (the number of zeros "0" equals that of question marks "?"). The control appends the extension ".uue", ".b16", or ".q_p" depending on Format immediately after the generated numbers. FileCnt contains the number of encoded files .

See also the Format, IntelliCode, and MaxFileSize properties when working with multiple files.

Data Type

String

FileCnt Property

Description

Shows the number of encoded files the control has read from or written into.

Usage

*[form.]**[netcodecontrol.]***FileCnt**

Default Value

0

Remarks

If the user specifies one or more questions marks "?" in EncodedData they will be expanded to 000 to **FileCnt** - 1 (the number of question marks "?" specifies the number of figures).

Please refer to EncodedData for filename conventions.

This property is read-only.

Data Type

Integer

FileName Property

Description

The destination filename for the encoded data.

Usage

[*form.*][*netcodecontrol.*]**FileName**[= *value*\$]

Default Value

""

Remarks

When uuencoding, the **FileName** property contains the filename that is specified in the uuencoded data.

When decoding, if the value of the property is not empty, it shows the control where to write the decoded data. It can be either the filename or the directory where the file should be written. If a directory, it should end with a backslash "\".

After a decode Action, the **FileName** property contains the filename for the created file. If the file couldn't be created because of an illegal filename, examining **FileName** might give a hint about the reason of the failure.

FileName must be set to "" (empty string) after each decode operation since it contains the full specification of the file.

An error occurs if **FileName** is empty and the UUEncode Format and the EncodeToString Action are selected.

If you want to first check the uuencoded filename before creating it on disk, you may assign an illegal directory name to **FileName** (always use a closing backslash "\" to denote it as directory), then trap the produced error and check the filename appended to **FileName**.

Data Type

String

Format Property

Description

Shows the type of encoding to be used.

Usage

`[form.][netcodecontrol.]Format[= value%]`

Default Value

0

Remarks

The following are the possible values for the **Format** property, and the corresponding descriptions:

0 (UUEncode)	The most popular standard. 3 Bytes are encoded into 4 readable characters. If multiple filenames are specified, then the extension ".uue" is used/expected.
1 (Base64)	Encoding format of MIME. Much like UUEncode but another subset of printable characters is used. If multiple filenames are specified, then the extension ".b64" is used/expected.
2 (Quoted Printable)	Another MIME format coding only special characters. Mostly used if the text contains special accented characters. If multiple filenames are specified, then the extension ".q_p" is used/expected.

Data Type

Integer

IntelliCode Property

Description

Controls whether NetCode should try to interpret several concatenated messages while uudecoding.

Usage

`[form.][netcodecontrol.]IntelliCode[= {True | False}]`

Default Value

True

Remarks

If a file is uuencoded and split over several mail messages, NetCode will normally attempt to filter the opening and closing lines such as message headings and/or decode scripts which don't belong to the encoded data. You should set **IntelliCode** to False only in the improbable case that the data is decoded improperly.

IMPORTANT: The uuencoded messages should appear in the proper order. The control handles only the filtering of packing information and does no sorting.

The **IntelliCode** property applies only to the 'uuencoded' Format.

Data Type

Integer (Boolean)

MaxFileSize Property

Description

When decoding, controls whether encoded data should be split into several files and gives the maximum allowed size for these files.

Usage

[*form.*][*netcodecontrol.*]**MaxFileSize**[= *value*&]

Default Value

0

Remarks

Some mailing systems require that the message size shouldn't exceed a certain size. Assigning a nonzero value to **MaxFileSize** makes the control split the encoded data into several files. FileCnt will contain the number of files created. Please refer to EncodedData for filename conventions.

IMPORTANT: if a non-zero value is assigned to **MaxFileSize** then a multiple filename should also be assigned to EncodedData (end with question marks "?") so that the control can expand the filenames.

Please refer to the Error Messages section for a complete list of possible errors.

Data Type

Long

Overwrite Property

Description

Controls whether created file(s) should overwrite already existing file(s).

Usage

```
[form.][netcodecontrol.]Overwrite[ = {True | False}]
```

Default Value

False

Remarks

None.

Data Type

Integer (Boolean)

ProgressStep Property

Description

Controls the granularity at which the Progress event is fired. Values 0-100.

Usage

[form.][netcodecontrol.]**ProgressStep**[= value%]

Default Value

1

Remarks

The Progress event will be fired when 0%, $n * \mathbf{ProgressStep}$ and 100% of input data is read if **ProgressStep** is set. The Progress event is fired exactly $(100 + \mathbf{ProgressStep} - 1) \setminus \mathbf{ProgressStep} + 1$ times.

Setting **ProgressStep** to 0 disables the Progress event.

Data Type

Integer

Progress Event

Description

Occurs when PercentDone of the input is read.

Syntax

Sub *netcodecontrol_Progress*(*PercentDone As Integer*)

Remarks

The ProgressStep property determines how often the event is fired.

The **Progress** event is fired at 0%, 100% and at multiples of ProgressStep. The event will be fired exactly $(100 + \text{ProgressStep} - 1) \setminus \text{ProgressStep} + 1$ times if ProgressStep is not zero.

NetCode Error Messages

NetCode Errors

- 20001 The starting 'begin' was not found. Is it uuencoded data?
- 20002 Short uuencoded file.
- 20003 The closing `end' was not found (uudecoded file may be too short).
- 20004 Can't create the file for write (illegal name or disk is write-protected).
- 20005 Can't open the file for read (doesn't exist?).
- 20006 Can't read from file.
- 20007 Can't write to file (disk full?).
- 20008 Please supply a filename for uuencoded data.
- 20009 File exists and Overwrite was set to False.
- 20010 There's not enough space in output string.
- 20011 The given filenames are insufficient to split all the encoded data.
- 20012 Please supply a filename for the encoded data.
- 20013 Encoded data can't be written in more than 100 files.
- 20014 Couldn't write to file. Permission denied.
- 20015 Couldn't write to file. Disk full.
- 20016 Too many open files. No more file handles are available, so no more files can be opened.
- 20017 The given filename doesn't exist.
- 20018 There are no more files where to read from or write to.

Properties

* <u>Action</u>	Left
* <u>DecodedData</u>	* <u>MaxFileSize</u>
* <u>EncodedData</u>	Name
* <u>FileCnt</u>	* <u>Overwrite</u>
* <u>FileName</u>	* <u>ProgressStep</u>
* <u>Format</u>	Top
Index	
* <u>IntelliCode</u>	

Events

Progress

Licensing Information



This software is Copyright (c) 1995, devSoft Inc. - All Rights Reserved.

To obtain a license please contact:

devSoft Inc.

P.O. Box 13821, Research Triangle Park, NC 27709, USA

email: devsoft_sales@devlink.com

www: <http://www.devlink.com/devsoft/>

tel/fax: (919) 493-5805

The use of this product constitutes your acceptance of the following terms and conditions and your Agreement to abide by them.

This is a legal Agreement between you, the end user, and **devSoft Inc.** By using this package, you are agreeing to be bound by the terms of this Agreement.

You, the single developer, are granted a nonexclusive license to use this software under the terms stated in this Agreement. This license allows you to use this software in your design environment as well as distribute executables that use it as a runtime component. There are no royalties, or runtime fees for such distribution, nor any other limitations on the distribution of your applications. We do however require that runtime components developed by devSoft Inc. be distributed in their original format, without any changes or modifications of any kind.

You may only install one registered copy of this software in a single workstation at any time. Use of a registered copy in more than one workstation is against the terms of this licensing Agreement. In particular, you are specifically prohibited from distributing a registered version of this software except as a runtime component of one of your applications.

You may not rent or lease the software, but you may transfer it on a permanent basis if the person receiving it agrees to the terms of the Agreement. If the software is an update, any transfer must include the update and all prior versions.

THE SOFTWARE AND ACCOMPANYING DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU.

devSoft Inc. will not be liable for any special, incidental, consequential, indirect or similar damages due to the loss of data, loss of business profits, business interruption or any other reason arising from the use of this software. In no event shall devSoft Inc.'s liability for any damages ever exceed the price paid for the license to use the software, regardless of the form of the claim. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

If any provision of this Agreement is found void, invalid or unenforceable, it will not affect the validity of the balance of this Agreement, which shall remain valid and enforceable according to its terms.

Should you have any questions concerning this Agreement please contact devSoft Inc., P. O. Box 13821, RTP, NC 27709. or via e-mail at: devsoft_sales@devlink.com

