



## Getting Started with the CIUU Control

### Related Topics

This topic introduces UUEncode/Decode standard and its implementation with the Crescent Internet ToolPak UUEncode/Decode Control (CIUU). You can access the following topics:

- [What UUEncode/Decode is](#)
- [What the CIUU control is](#)
- [How to use the CIUU control](#)

The last topic, How to use the CIUU control, identify tasks that CIUU supports; it lists the properties to set and the methods to call, then it illustrates the task with a simple code sample.



## What is UUEncode/Decode?

### Related Topics

UUEncode/Decode is a standard that specifies the format of Internet message bodies. It defines how you can include multiple objects (of different types) in a single message. The different types of objects that you can include in a message include: images, audio, and other material that might be in a character set that is not US-ASCII. The standard supports the exchange of messages so that no data is lost during the message exchange.



## What is the CIUU Control?

### Related Topics

The Crescent Internet ToolPak UUEncode/Decode (CIUU) can encode or decode files or mail attachments in stand-alone mode or when combined with the CISMTP or CIPOP controls. When used with the CISMTP control, the encoded text is fed to the CISMTP.Message property just as any message would be sent. When used with the CIPOP control, the encoded text is fed to the CIUU control via the CIPOP.Message property. If you use CIUU to encode, the source is a non-text file. If you use CIUU to decode, the source is a mail message.

CIUU has a simple programming interface in which you set property values and call methods to perform the desired tasks. The methods (which generally do not take arguments) take the necessary data directly from the property sheet. When the method completes, the control updates properties and fires events as necessary. The control, by default will fire the ongoing status of the encoding or decoding process. Note that this enhanced status firing does negatively affect performance.

CIUU has a simple programming interface in which you set property values and call methods to perform the desired tasks. The methods (which generally do not take arguments) take the necessary data directly from the property sheet. When the method completes, the control updates properties and fires events as necessary.



## Programming Tasks

### Related Topics

The following topics describe how to program some common tasks with the CIUU control.

- [How to encode a file](#)
- [How to decode a file](#)

Each topic lists the properties that you must set, and the methods that you must call to perform a specific task, and identifies the events that are fired. A code fragment that illustrates the task follows the task list.



## Encoding a File

### Related Topics

CIUU can encode data whose source is a file, and it can deliver the encoded data as a stream.. This task describes how CIUU reads a file, encodes the data, and writes the encoded data to a file.

Follow these steps to encode data using CIUU:

1. Set these properties:  
    SourceFileName  
    DestinationFileName
2. Call the UUEncode method.
3. When the data is encoded, the EncodingFinished event fires.

Heres what the code looks like:

```
Form_Load
    CIUU1.SourceFileName = C:\Windows\CarvedStone.bmp
    CIUU1.DestinationFileName = C:\temp\cs.bmp
    CIUU1.UUEncode

CIUU1_EncodingFinished()
```

*The encoded data is available in the DestinationFileName.*

Note that if you do not supply a SourceFileName or a DestinationFileName, CIUU displays a dialog that prompts you for this information.



## Decoding a File

### Related Topics

CIUU can decode data whose source is a file or a data stream, and it can deliver the decoded data as a stream or in a file. This task describes how CIUU reads a file, decodes the data, and writes the decoded data to a file.

Follow these steps to decode a file encoded using CIUU:

1. Set these properties:  
    SourceFileName  
    DestinationFileName
2. Call the UUDecode method.
3. The data is decoded when the DecodingFinished event fires.

Heres what the code looks like:

```
Form_Load
```

```
    CIUU1.SourceFileName = C:\temp\cs.bmp  
    . CIUU1.DestinationFileName = C:\Windows\CarvedStone.bmp  
    . CIUU1.UUDecode
```

```
CIUU_DecodingFinished()
```

*The decoded data is available in the file identified by the DestinationFileName property.*

Note that if you do not supply a SourceFileName or a DestinationFileName, CIUU displays a dialog that prompts you for this information.

Getting Started with the CIUU Control

What is UUEncode/Decode?

What is the CIUU Control?

Programming Tasks

How to encode a file

How to decode a file

[What is UUEncode/Decode?](#)

[What is the CIUU Control?](#)

[Programming Tasks](#)

[How to encode a file](#)

[How to decode a file](#)

[The Crescent Internet ToolPak CIUU Control Reference](#)



[Getting Started with the CIUU Control](#)

[What is the CIUU Control?](#)

[Programming Tasks](#)

[How to encode a file](#)

[How to decode a file](#)

[The Crescent Internet ToolPak CIUU Control Reference](#)

[Getting Started with the CIUU Control](#)

[What is UUEncode/Decode?](#)

[Programming Tasks](#)

[How to encode a file](#)

[How to decode a file](#)

[The Crescent Internet ToolPak CIUU Control Reference](#)

[Getting Started with the CIUU Control](#)

[What is UUEncode/Decode?](#)

[What is the CIUU Control?](#)

[How to encode a file](#)

[How to decode a file](#)

[The Crescent Internet ToolPak CIUU Control Reference](#)

[Getting Started with the CIUU Control](#)

[What is UUEncode/Decode?](#)

[What is the CIUU Control?](#)

[Programming Tasks](#)

[How to decode a file](#)

[The Crescent Internet ToolPak CIUU Control Reference](#)

[Getting Started with the CIUU Control](#)

[What is UUEncode/Decode?](#)

[What is the CIUU Control?](#)

[Programming Tasks](#)

[How to encode a file](#)

[The Crescent Internet ToolPak CIUU Control Reference](#)



# The Crescent Internet ToolPak UUEncode/Decode Control

## Related Topics

### Control File

CIUU.OCX

### Object Type

CIUU

### Purpose

The Crescent Internet ToolPak UUEncode/Decode control can be used to provide UUEncode/Decode capability to mail. The decoded data can be sent to the Crescent CISMTP control. In addition, the data can originate from the Crescent CIPOP control.

The Crescent Internet ToolPak UUEncode/Decode control can be used to:

- Encode files (to 7-bit ASCII standard) that you intend to mail.
- Unencode files that have been encoded using the UUEncode standard.

### Properties

AboutBox	Container	<u>DestinationFileName</u> c
<u>DestinationFileSize</u> c	DragIcon	DragMode
<u>FireStatus</u> c	Height	Index
Left	Name	Object
Parent	ShowWhatsThis	<u>SourceFileName</u> c
<u>SourceFileSize</u> c	Tag	Top
Visible	WhatsThisHelpID	Width

### Events

<u>Decoded</u> c	<u>DecodingFinished</u> c	<u>DecodingStarted</u> c
DragDrop	DragOver	<u>Encoded</u> c
<u>EncodingFinished</u> c	<u>EncodingStarted</u> c	

### Methods

<u>UUDecode</u> c	<u>UUEncode</u> c
-------------------	-------------------

c A custom or modified property, method, or event.

## DestinationFileName Property

### Applies To

CIUU

### Purpose

The DestinationFileName property sets or returns the fully qualified pathname of the file where the data that has been encoded or decoded data finally resides.

### Syntax

```
[Form.] CIUU.DestinationFileName[ = string$]
```

### Data Type

String

### Usage

Read/Write at design time and runtime.

### Comments

You must set the DestinationFileName property before calling the UUDecode or UUEncode methods. If you do not set the DestinationFileName property, CIUU displays a dialog that prompts you for this information.

### See Also

[DecodingFinished](#), [EncodingFinished](#), [UUDecode](#), [UUEncode](#)

## DestinationFileSize Property

### Applies To

CIUU

### Purpose

The DestinationFileSize property returns the size (in bytes) of the file (DestinationFileName property) where the data that has been encoded or decoded finally resides.

### Syntax

```
[Form.] CIUU.DestinationFileSize[ = long&]
```

### Data Type

Long

### Usage

Read only at runtime.

### Comments

The DestinationFileSize property is updated at the end of the encode/decode process.

### See Also

DecodingFinished, DestinationFileName, EncodingFinished, UUDecode, UUEncode



## FireStatus Property

### Applies To

CIUU

### Purpose

The FireStatus property determines whether the Encoded or Decoded events fire as a file is being encoded or decoded.

### Syntax

```
[Form.] CIUU.FireStatus[ = boolean]
```

### Data Type

Boolean

### Usage

Read/Write at design time and runtime.

### Comments

The FireStatus property lets you determine whether you want status information to be available to a Visual Basic application. When set to True, the Encoded and Decoded events fire as a file is being encoded or decoded. When set to False, the Encoded and Decoded events fire after the decode or encode process is complete. Setting this property to True can cause the encode or decode operation to take longer because of the overhead associated with firing the Encoded and Decoded events.

### See Also

Decoded, Encoded

## SourceFileName Property

### Applies To

CIUU

### Purpose

The SourceFileName property sets or returns the fully qualified pathname of the file where the data to be encoded or decoded originates.

### Syntax

```
[Form.] CIUU.SourceFileName[ = string$]
```

### Data Type

String

### Usage

Read/Write at design time and runtime.

### Comments

You must set the SourceFileName property before calling the UUDecode or UUEncode methods. If you do not set the SourceFileName property, CIUU displays a dialogbox that prompts you for this information.

### See Also

[DecodingFinished](#), [DestinationFileName](#), [EncodingFinished](#), [UUDecode](#), [UUEncode](#)

## SourceFileSize Property

### Applies To

CIUU

### Purpose

The SourceFileSize property returns the size (in bytes) of the source file where the encoded or unencoded data originates.

### Syntax

```
[long&] = [Form.] CIUU.SourceFileSize
```

### Data Type

Long

### Usage

Read only at runtime.

### Comments

The size is available once the CIUU control has opened the file to read either encoded or decoded data from it.

### See Also

[DecodingFinished](#), [DestinationFileName](#), [EncodingFinished](#), [UUDecode](#), [UUEncode](#),  
[SourceFileName](#)

## Decoded Event

### Applies To

CIUU

### Purpose

The Decoded event gives the Visual Basic application statistics about the current decoding operation. It fires each time the control has encoded a line and written the encoded line to the destination file.

### Syntax

```
Sub CIUU_Decoded(BytesRead as Long, BytesWritten as Long, LinesDecoded as Long)
```

### Comments

This event is useful in determining the status of a decode operation. You can determine how much data has been processed by checking the values of *BytesRead*&, *BytesWritten*&, and *LinesDecoded*&.

### See Also

DestinationFileName, DestinationFileSize, Encoded, SourceFileName, UUDecode

## DecodingFinished Event

### Applies To

CIUU

### Purpose

The DecodingFinished event notifies the Visual Basic application that the decoding operation has finished and that the DestinationFileSize property can now be read.

### Syntax

```
Sub CIUU1_DecodingFinished()
```

### Comments

The DecodingFinished event fires when the control closes the source and destination files.

You initiate a decode operation by calling the UUDecode method which fires the DecodingStarted event. If you set the FireStatus property to true, you can gather data about the progress of the decode operation through the Decoded event.

### See Also

Decoded, DecodingStarted, DestinationFile, SourceFile, UUDecode

# DecodingStarted Event

## Applies To

CIUU

## Purpose

The DecodingStarted event notifies the Visual Basic application that the decoding operation has started and that the SourceFileSize property can now be read.

## Syntax

```
Sub CIUU1_DecodingStarted()
```

## Comments

The DecodingStarted event fires when the control has opened the source (SourceFileName property) and destination files (DestinationFileName property) and the decoding operation has started.

You initiate a decode operation by calling the UUDecode method. When all of the data is decoded, the DecodingFinished event fires. If you set the FireStatus property to true, you can gather data about the progress of the decode operation through the Decoded event.

## See Also

Decoded, DecodingFinished, DestinationFile, SourceFile, UUDecode

# Encoded Event

## Applies To

CIUU

## Purpose

The Encoded event provides statistics about the current encoding operation to the Visual Basic application. It fires each time the control has encoded a line and written the encoded line to the destination file.

## Syntax

```
Sub CIUU1_Encoded(BytesRead as Long, BytesWritten as Long, LinesEncoded as Long
```

## Comments

You initiate an encode operation by calling the UUEncode method. Once the encode operation starts, the EncodingStarted event fires.

This event is useful in determining the status of an encode operation. You can determine how much data has been processed by checking the value of *BytesRead*&, *BytesWritten*&, and *LinesEncoded*&.

## See Also

Decoded, DestinationFileName, DestinationFileSize, SourceFileName, UUEncode,

# EncodingFinished Event

## Applies To

CIUU

## Purpose

The EncodingFinished event notifies the Visual Basic application that the encoding operation has finished and that the DestinationFileSize property can now be read.

## Syntax

```
Sub CIUU1_EncodingFinished()
```

## Comments

The EncodingFinished event fires when CIUU closes the source (SourceFileName property) and the destination (DestinationFile property) property.

You initiate an encode operation by calling the UUEncode method which fires the EncodingStarted event. When you call the UUEncode method, the EncodingStarted event fires. You can gather data about the progress of the encode operation through the Encoded event.

## See Also

Encoded, EncodingStarted, DestinationFile, SourceFile, UUDecode



# EncodingStarted Event

## Applies To

CIUU

## Purpose

The EncodingStarted event notifies the Visual Basic application that the encoding operation has started and that the SourceFileSize property can now be read.

## Syntax

```
Sub CIUU1_EncodingStarted()
```

## Comments

The EncodingStarted event fires when the CIUU control has opened the source file (SourceFileName property) and the destination file (DestinationFileName property) and the encoding operation has started.

You initiate an encode operation by calling the UUEncode method. When all of the data is encoded, the EncodingFinished event fires. You can gather data about the progress of the encode operation through the Encoded event.

## See Also

[DestinationFile](#), [Encoded](#), [EncodingFinished](#), [SourceFile](#), [UUDecode](#)

# UUDecode Method

## Applies To

CIUU

## Purpose

The UUDecode method initiates the decode operation on the file identified by the SourceFileName property. The result of the decode operation resides in the file identified by the DestinationFileName property.

## Syntax

```
Boolean = CIUU.UUDecode
```

## Data Type

Boolean Integer

## Comments

When the return code (Boolean) is True, UUDecode succeeded; when False, it failed.

You must set the SourceFileName and DestinationFileName properties before calling the UUDecode method. The UUDecode method updates the SourceFileSize and DestinationFile Size properties.

UUDecode initiates the decoding operation which causes the DecodingStarted event to fire. When the decoding operation completes, the DecodingFinished event fires. If you set the FireStatus property to True, the Decoded event fires during the decoding operation (depending on the file size).

## See Also

Decoded, DecodingFinished, DecodingStarted, DestinationFileName, SourceFileName, UUEncode,

# UUEncode Method

## Applies To

CIUU

## Purpose

The UUEncode method initiates the encode operation on the file identified by the SourceFileName property. The result of the encode operation resides in the file identified by the DestinationFileName property.

## Syntax

```
Boolean% = CIUU.Encode
```

## Data Type

Boolean Integer

## Comments

When the return value (Boolean%) is True, the Encode operation succeeded; when False, it failed.

You must set the SourceFileName and DestinationFileName properties before calling the UUEncode method. If you do not provide this information, CIUU displays a dialogbox that prompts you for it. The UUEncode method updates the SourceFileSize and DestinationFile Size properties.

UUEncode initiates the encoding operation which causes the EncodingStarted event to fire. When the encoding operation completes, the EncodingFinished event fires. If you set the FireStatus property to True, the Encoded event fires during the decoding operation (depending on the file size).

## See Also

DestinationFileName, Encoded, EncodingFinished, EncodingStarted, SourceFileName, UUDecode



## General Tab

### Applies To

CIUU

This custom tab lets you set these custom properties:

SourceFileName

DestinationFileName

FireStatus

