

# StdCoder

DEFINITION StdCoder;

```
IMPORT TextModels, Views, Dialog;
```

```
TYPE
```

```
  ParList = RECORD
```

```
    list: Dialog.Selection;
```

```
    storeAs: Dialog.String
```

```
  END;
```

```
VAR par: ParList;
```

```
PROCEDURE CloseDialog;
```

```
PROCEDURE Decode;
```

```
PROCEDURE DecodeAllFromText (text: TextModels.Model; beg: INTEGER; ask: BOOLEAN);
```

```
PROCEDURE EncodeDocument;
```

```
PROCEDURE EncodeFile;
```

```
PROCEDURE EncodeFileList;
```

```
PROCEDURE EncodeFocus;
```

```
PROCEDURE EncodeSelection;
```

```
PROCEDURE EncodedInText (text: TextModels.Model; beg: INTEGER): TextModels.Model;
```

```
PROCEDURE EncodedView (v: Views.View): TextModels.Model;
```

```
PROCEDURE ListEncodedMaterial;
```

```
PROCEDURE Select (op, from, to: INTEGER);
```

```
PROCEDURE StoreAll;
```

```
PROCEDURE StoreSelection;
```

```
PROCEDURE StoreSelectionGuard (VAR p: Dialog.Par);
```

```
PROCEDURE StoreSingle;
```

```
PROCEDURE StoreSingleGuard (VAR p: Dialog.Par);
```

```
END StdCoder.
```

*StdCoder* (or *Coder* for short) can be used to encode a document, a view, a text stretch, or files into a textual form. The encoding uses characters that are expected not to be changed by any mail system. White space characters (blanks, tabs, new lines, etc.) may be added or removed arbitrarily, because they are ignored upon decoding.

Upon decoding a document or a text stretch a new window will be opened. Files will be stored on disk.

Suggested menu items, as they are standard in menu *Tools*:

"Encode Document"	""	"StdCoder.EncodeDocument"	"StdCmds.WindowGuard"
"Encode Selection"	""	"StdCoder.EncodeSelection"	"TextCmds.SelectionGuard"
"Encode File..."	""	"StdCoder.EncodeFile"	""
"Encode File List"	""	"StdCoder.EncodeFileList"	"TextCmds.SelectionGuard"
"Decode..."	""	"StdCoder.Decode"	"TextCmds.FocusGuard"
"About Encoded Material"	""	"StdCoder.ListEncodedMaterial"	"TextCmds.FocusGuard"

#### PROCEDURE **EncodeDocument**

Guard: StdCmds.WindowGuard

Encodes the document in the current front window and opens a new window with the generated code.

#### PROCEDURE **EncodeSelection**

Guard: TextCmds.SelectionGuard

Encodes the current text selection, and opens a new window with the generated code.

#### PROCEDURE **EncodeFile**

Encodes one file determined through the standard file opening dialog and opens a new window with the generated code.

#### PROCEDURE **EncodeFileList**

Guard: TextCmds.SelectionGuard

Encodes several files and opens a new window with the generated code.

A list of valid file names must be selected. File names must be specified with their complete path name relative to the BlackBox directory. Use the slash character ("/") to separate the individual parts of the path name. (Example: *Std/Code/Coder .*)

The path name is stored along with the encoding to allow for easy decoding of entire file packages.

However, sometimes a file is kept in a different place than where it should be installed later. In such a case, a destination path name can be specified. The path name to be used during decoding can be given after an "arrow" ("=>").

Example: *NewStd/Code/NewCoder => Std/Code/Coder* will lead to encoding of the file *NewCoder* in the directory *NewStd/Code*, but for decoding the name *Std/Code/Coder* will be used.

#### PROCEDURE **Decode**

Guard: TextCmds.FocusGuard

Decodes the information in the encoded text contained by the front window. The command scans for "StdCoder.Decode" as a tag that marks the begin of a code sequence. This allows e-mail headers and other text to precede the actual code. If a text selection is active in the front window, scanning will start at the begin of that selection.

Depending on the kind of encoded data, that is, on the command used for encoding, one of the following actions will be taken:

- if a document was encoded (EncodeDocument, EncodeSelection), it is opened in a new window.
- if a single file was encoded (EncodeFile), the standard file store dialog is opened allowing the user to store the file.
- if a list of files was encoded (EncodeFileList), a special dialog is opened allowing the user to select files for decoding. The path names included during the encoding are shown in a list. One or several files can be selected from that list. After pressing a command button the selected files will be decoded. If only one file is selected, the path name can be changed or the standard file store dialog can be used to browse the directory hierarchy. Via another command button all files can be decoded and stored under the listed names, regardless of the selection.

**Note:** The dialog should always be closed using the cancel button to allow StdCoder to release resources allocated temporarily to manage the dialog.

#### PROCEDURE **ListEncodedMaterial**

Guard: TextCmds.FocusGuard

Opens a text informing about what is encoded in the text displayed by the focus window. Like Decode, the command scans for a tag that marks the beginning of a code sequence.

In the programming interface, two further procedures are available:

#### PROCEDURE **DecodeAllFromText** (text: TextModels.Model; beg: INTEGER; ask: BOOLEAN);

Works like *Decode*, but uses the passed text as source instead of the focus. At text position *beg* a search for the tag is started and the text following the tag is processed. Like with *Decode*, encoded documents are opened in windows and single files are stored via the standard store dialog. If the code was generated through *EncodeFileList*, the behavior depends on the value of the parameter *ask*. If it equals to *TRUE*, a dialog is displayed as done by *Decode* and the user is put into control. If it equals to *FALSE*, no dialog is displayed, but all files are decoded and installed under the name stored with them. This behavior

corresponds to pressing the "Decode All" button in the dialog.

PROCEDURE **EncodedView** (v: Views.View): TextModels.Model;  
Generates a text containing the encoding of the passed view.

PROCEDURE **EncodedInText** (text: TextModels.Model; beg: INTEGER): TextModels.Model;  
Generates a text containing information about what is encoded in "text".