

Inserting a Statement

`<statement>`

Any C/C++ statement.

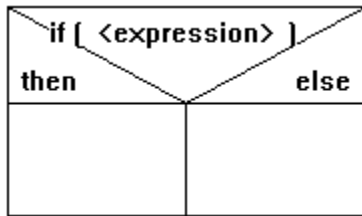
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting an *if*



Program branch.

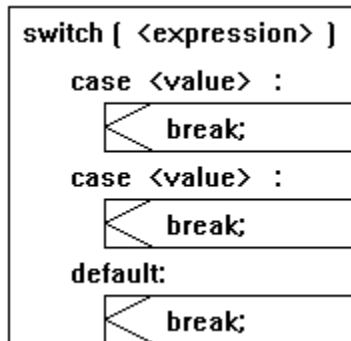
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *switch*



Case differentiation; further **case**-branches can be inserted.

See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a case Branch

```
case <value> :  
    break;
```

Additional **case** statement for **switch**.

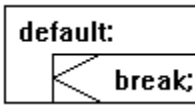
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *default* Branch



default branch for **switch**.

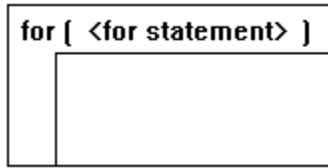
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *for*



Loop with fixed number of passes.

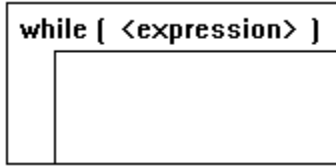
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *while*



Loop with entry condition.

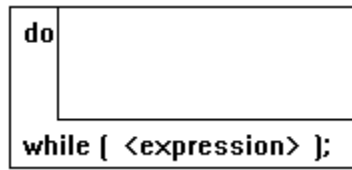
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *do while*



Loop with repeat condition.

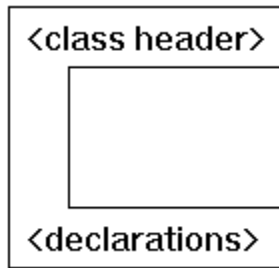
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *Class*



Class construct.

See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a private Branch into a *Class Construct*

private:

Private branch of the Class construct.

See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a public Branch into a *Class Construct*

public:

Public branch of the Class construct.

See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a protected Branch into a *Class* Construct

protected:

Protected branch of the Class construct.

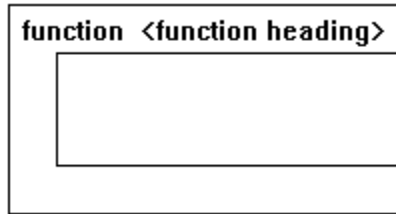
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a Function



C function.

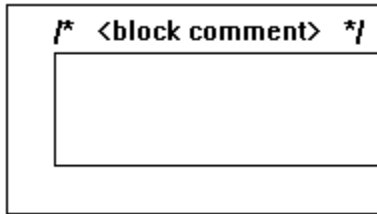
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a Block



C block.

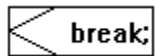
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *break*



Break statement.

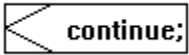
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *continue*



Continue statement.

See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Inserting a *return*

```
return <expression> ;
```

Return statement.

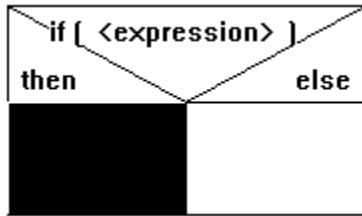
See also:

[Toolbox](#)

[Setting a Valid Insertion Point](#)

[Insert Menu](#)

Enclosing in an if - Selected Area as *then* Branch



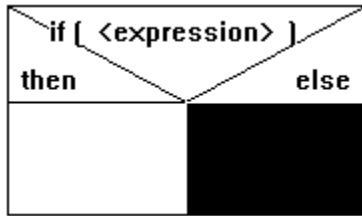
Use this command to enclose a selected area in an **if** construct. The selected area will become a **then** branch of the **if** construct.

See also:

[Selecting an Area](#)

[Enclosing in an if](#)

Enclosing in an if - Selected Area as *else* Branch



Use this command to enclose a selected area in an **if** construct. The selected area will become an **else** branch of the **if** construct.

See also:

[Selecting an Area](#)

[Enclosing in an if](#)

Enclosing in an if

To enclose an area in an **if** construct, you must first select an area which structurally complies with a **then** or **else** branch of the **if** construct.

After choosing the **if** construct in the **Enclose** command of the **Edit** menu, select one of the following branches:

then branch

else branch

See also:

Selecting an Area

Enclosing in Other Constructs

Enclosing in a *switch*

To enclose an area in the **switch** construct, you must first select an area which structurally complies with a **case** or **default** branch of the **switch** construct.

After choosing the **switch** construct in the **Enclose** command of the **Edit** menu, select one of the following branches:

case branch

default branch

See also:

Selecting an Area

Enclosing in Other Constructs

Enclosing in a *switch* - Selected Area as case Branch

```
switch [ <expression> ]
  case <value> :
    [ ]
    break;
  case <value> :
    [ ]
    break;
  default:
    [ ]
    break;
```

Use this command to enclose a selected area in a **switch** construct. The selected area will become a **case** branch of the **switch** construct.

See also:

[Selecting an Area](#)

[Enclosing in Other Branches of this Construct](#)

Enclosing in a *switch* - Selected Area as *default* Branch

```
switch ( <expression> )
  case <value> :
    break;
  case <value> :
    break;
  default:
    break;
```

Use this command to enclose a selected area in a **switch** construct. The selected area will become a **default** branch of the **switch** construct.

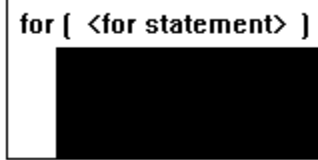
See also:

[Selecting an Area](#)

[Enclosing in Other Branches of this Construct](#)

Enclosing in a for

```
for [ <for statement> ]
```



Use this command to enclose a selected area in a **for** construct. The selected area will become the body of the **for** construct.

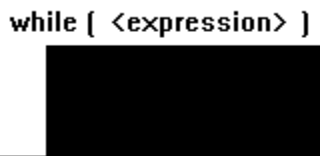
See also:

[Selecting an Area](#)

[Enclosing in Other Constructs](#)

Enclosing in a *while*

```
while [ <expression> ]
```



Use this command to enclose a selected area in a **while** construct. The selected area will become the body of the **while** construct.

See also:

[Selecting an Area](#)

[Enclosing in Other Constructs](#)

Enclosing in a *do-while*

```
do   
while [ <expression> ];
```

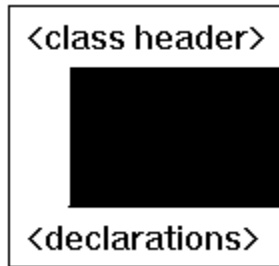
Use this command to enclose a selected area in a **do while** construct. The selected area will become the body of the **do while** construct.

See also:

[Selecting an Area](#)

[Enclosing in Other Constructs](#)

Enclosing in a *Class*



Use this command to enclose a selected area in a **Class** construct.

See also:

Selecting an Area

Enclosing in Other Constructs

Enclosing in a Function



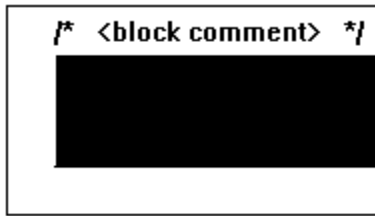
Use this command to enclose a selected area in a **Function** construct.

See also:

[Selecting an Area](#)

[Enclosing in Other Constructs](#)

Enclosing in a Block



Use this command to enclose a selected area in a **Block** construct.

See also:

[Selecting an Area](#)

[Enclosing in Other Constructs](#)

Edit Menu

The **Edit** menu provides commands for editing structure diagrams. Using these commands, you can edit text within constructs, rearrange structure diagrams and make the newly created diagrams accessible from outside EasyCODE(C++).

The following commands are available:

<u>Undo</u>	To undo the most recent <u>step</u>
<u>Edit text</u>	To edit text boxes
<u>Edit window</u>	To edit in an EasyCODE edit window
<u>External editor</u>	To load an external editor
<u>Split</u>	To split at caret
<u>Join</u>	To join into one construct
<u>Search/Replace</u>	To search and replace text in a structure diagram
<u>Continue search</u>	To continue a search
<u>Copy</u>	To copy an area
<u>Duplicate</u>	To duplicate an area
<u>Move</u>	To move an area
<u>Delete</u>	To delete an area
<u>Enclose</u>	To enclose an area
<u>Extract</u>	To extract an area
<u>Clipboard</u>	To cut, copy and paste an area
<u>Refine</u>	To refine a <u>statement</u>
<u>Push down</u>	To push down an area
<u>Pull up</u>	To pull up an area

See also:

Menus

Edit Text

With the **Edit text** command in the **Edit** menu, you can fill a [structure diagram](#) with text or edit existing text without opening a dialog window. If you have selected Edit text, an edit window surrounded by a hatched frame will appear at the position of the construct you want to edit.

To do this:

1. [Select a symbolic name](#), existing text or an empty box by [clicking](#) it once with the left mouse button or with the help of the keyboard.
2. Choose the **Edit text** command in the **Edit** menu or press the Shift+Enter keys.
3. .
EasyCODE changes into [Edit mode](#).

See also:

[Edit window](#)

[External Editors](#)

[Options Editors](#)

[Shortcuts](#)

[Speeding Up Text Editing](#)

[Hot Keys](#)

[Commands for Redesigning a Structure Diagram](#)

[Other Commands in the Edit Menu](#)

Edit window

With the **Edit window** command in the **Edit** menu you can open the EasyCODE Edit window and fill a structure diagram with text or edit an existing text.

To do this:

1. Select a symbolic name, existing text or an empty box by clicking it once with the left mouse button or with the help of the keyboard.
2. Choose the **Edit window** command in the **Edit** menu or press the Ctrl+Enter keys.

An edit window corresponding to the symbolic name will appear.

See also:

Edit text

External editor

Options Editors

Move Cursor

Selecting Text

Deleting Text

Replacing Text

Speeding Up Text Editing

Hot Keys

Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

External Editor

Use the **External editor** command in the **Edit** menu to fill a structure diagram with text or edit existing text with the help of an external editor defined with the Editors command in the **Options** menu.

To do this:

1. Select a symbolic name, existing text or an empty box by clicking it once with the left mouse button or with the help of the keyboard.
2. Choose the **External editor** command in the **Edit** menu or press the ALT + ENTER keys.

The editing window of the selected external editor will appear.

Note:

If it is possible to enter line and column numbers when you call the editor, you may view the required section as soon as the editor has been loaded (by pressing the ALT key and at the same time double-clicking the left mouse button) by entering the corresponding settings in the **Editors** command of the **Options** menu (%LINE%, %COLUMN%).

The Shortcut for External Editor changes to ENTER (without the ALT key), if you made the external editor the default editor in the Editors dialog window.

See also:

Edit window

Edit window

Option Editors

Selecting Text

Deleting Text

Replacing Text

Hot Keys

Other Commands in the Edit Menu

Split

Use the **Split** command in the **Edit** menu to split a statement into two statements. Place the caret at the position where you want to split the text and then enter the command **Split**.

See also:

[Hot Keys](#)

[Commands for Redesigning a Structure Diagram](#)

[Other Commands in the Edit Menu](#)

Join

Use the **Join** command in the **Edit** menu to join a selected number of statements to a construct. This command is available only if joinable constructs have been selected. The command can be undone with the command **Undo**.

See also:

[Hot Keys](#)

[Commands for Redesigning a Structure Diagram](#)

[Other Commands in the Edit Menu](#)

Search/Replace

Use the **Search/Replace** command in the **Edit** menu to view parts of structure diagrams containing a particular text and to replace this text with another text wherever you want.

To do this:

1. Select the area in which you want to start your search. If you want to search the entire structure diagram or segment, you do not need to select an area.
2. Choose the **Search/Replace** command in the **Edit** menu.
3. The Search/Replace dialog window appears. Enter the text you want to find and check the area you want to search.

See also:

Hot Keys

Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

Continue search

Use the command **Continue search** in the **Edit** menu to continue the last search for the selected criteria.

To do this:

1. Select the part of the structure diagram where or from where you want to search.
2. Choose the **Continue search** command in the **Edit** menu.

See also:

[Search/Replace](#)

[Hot Keys](#)


[Commands for Redesigning a Structure Diagram](#)

[Other Commands in the Edit Menu](#)

Copy

Use the **Copy** command in the **Edit** menu to copy an area to another position within the structure diagram.

To do this:

1. Select the area you want to copy.
2. Choose the **Copy** command in the **Edit** menu. The mouse pointer changes its shape to .
3. Insert the selected area at a valid insertion point.

Note:

To copy an area to a different segment, you can use the Clipboard or copy with the help of another segment window.

See also:

Hot Keys

Setting a Valid Insertion Point

Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

Duplicate

Use the **Duplicate** command in the **Edit** menu to copy and insert an area immediately after its current position. This speeds up your work particularly if you need similar areas arranged in sequence, because you only need to edit a few elements.

To duplicate an area:

1. Select the area you want to duplicate.
2. Choose the **Duplicate** command in the **Edit** menu.

This command immediately duplicates the selected area.

See also:

Hot Keys

Copy


Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

Move

Use the **Move** command in the **Edit** menu to copy an area to another position and delete the original position.

To move an area:

1. Select the area you want to move.
2. Choose the **Move** command in the **Edit** menu. The mouse pointer changes its shape to .
3. Insert the selected area at a valid insertion point.

Note:

To move an area to a different segment, use another segment window.

See also:

Hot Keys

Setting a Valid Insertion Point

Copy

Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

Delete

Use the **Delete** command in the **Edit** menu to delete an area.

To delete an area:

1. Select the area you want to delete.
2. Choose the **Delete** command in the **Edit** menu.

A selected area can either be deleted from the structure diagram, or replaced by the smallest possible element.

A construct, for example, is deleted, whereas the text for a symbolic name is replaced by the symbolic name.

See also:

Hot Keys

Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

Enclose

Use the **Enclose** command in the **Edit** menu to enclose an area in a construct.

To enclose an area:

1. Select the area you want to enclose.
2. Choose the **Enclose** command in the **Edit** menu. EasyCODE(C++) displays the constructs suitable for enclosing in a submenu.
3. Choose the desired construct.
4. If the chosen construct consists of more than one branch suitable for enclosing, these will be displayed in a submenu. Choose the desired branch.

See also:

Hot Keys

Commands for Redesigning a Structure Diagram

Other Commands in the Edit Menu

Constructs for Enclosing

if

switch

for

while

do while

Function

Block

Extract

Use the **Extract** command in the **Edit** menu to extract an area by deleting the enclosing construct.

To extract an area:

1. Select the area from which you want to delete the enclosing construct.
2. Choose the **Extract** command in the **Edit** menu.

The selected area takes the place of the former enclosing construct. The enclosing construct and the not selected constructs of the construct-chain in which the selection is, will be deleted without any warning.

Hints:

The **Extract** command is not available, if

- no area is selected,
- the whole section is selected,
- the section heading is selected,
- the selected area contains constructs, that would not be syntactically allowed after the extraction.

To undo the most recent **Extract** command use **Undo** from the **Edit** menu.

See also:

[Enclose Command in the Edit Menu](#)

[Other Commands in the Edit Menu](#)

Clipboard

Use the **Clipboard** command in the **Edit** menu to cut, copy and paste an area with the help of the Windows Clipboard.

You can use the clipboard to

- Copy and move areas within EasyCODE(C++),
- Copy and move areas from one EasyCODE(C++) level to another,
- Paste text from other applications into EasyCODE(C++),
- Paste areas from EasyCODE(C++) into other applications.

See also:

[Commands for Assembling](#)

[Object Linking and Embedding \(OLE\)](#)

[Other Commands in the Edit Menu](#)

Cut

Use the **Cut** command in the **Clipboard** submenu of the **Edit** menu to cut an area and put it into the Windows Clipboard. The selected area will be deleted from EasyCODE(C++).

To cut out an area:

1. Select an area.
2. Choose the **Clipboard** command in the **Edit** menu.
3. Choose the **Cut** command in the **Clipboard** submenu.

See also:

Hot Keys

Commands for Assembling

Other Commands in the Edit Menu

Copy

Use the **Copy** command in the **Clipboard** submenu of the **Edit** menu to copy an area into the Windows Clipboard without modifying the original data.

To copy an area:

1. Select an area, or undo any selection if you want to copy the entire segment including header to the clipboard.
2. Choose the **Clipboard** command in the **Edit** menu.
3. Choose the **Copy** command in the **Clipboard** submenu.

See also:

Hot Keys


Commands for Assembling

Other Commands in the Edit Menu

Paste

Use the **Paste** command in the **Clipboard** submenu of the **Edit** menu to paste data from the Windows Clipboard into the current structure diagram.

To do so:

1. Choose the **Clipboard** command in the **Edit** menu.
2. Choose the **Paste** command in the **Clipboard** submenu. The clipboard contents are represented by the following symbol:  .
3. Paste the clipboard contents at a valid insertion point.

See also:

Hot Keys

Commands for Assembling

Other Commands in the Edit Menu

Refine

Use the **Refine** command in the **Edit** menu to refine a statement by creating a segment.

To create a segment:

1. Select a statement.
2. Choose the **Refine** command in the **Edit** menu or press CTRL+F. EasyCODE(C++) switches to a lower hierarchical level and displays a worksheet containing only the text you have selected. This serves as a header for the segment you can create now.
3. Design a structure diagram corresponding to the function described in the segment header.
4. Choose Up in the View menu to return to the original structure diagram.

Note

The **Refine** command helps you to create a top-down design. The top level is a structure diagram with text e.g. written in natural language, giving a brief explanation of their function. Therefore, it is very clear and comprehensible, but it is not a complete program yet. By refining each statement, you create new segments which express the function of the statement more precisely. The statements contained in the segments can be either in natural language or in C/C++. If you continue refining until every segment consists only of C/C++ statements, you have created a structure diagram for a compilable C/C++ program and you have documented the design history of your program on multiple levels in a consistent and comprehensible way.

The source file created from the structure diagram contains the segment headers as comments.

See also:

Hot Keys

Hierarchical Structuring

Other Commands in the Edit Menu

Push Down

Choose the **Push down** command in the **Edit** menu to push down an area of the structure diagram onto a lower hierarchical level. This command automatically creates a segment which contains the selected area.

To do this:

1. Select an area.
2. Choose the **Push down** command in the **Edit** menu. The selected area will then be replaced by a gray box. This box represents the segment which has been placed on a lower hierarchical level and is currently not visible. Choose Down in the View menu to switch to this segment, if required. Within the gray box you will see the symbolic name <header>.
3. Edit the header.

Note

We recommend that you push down areas if a structure diagram has become too complex and therefore confusing. Wide structure diagrams are difficult to edit and the printouts must be prepared manually for documentation use. The technique of hierarchical structuring overcomes these problems easily.

See also:

Hot Keys

Other Commands in the Edit Menu

Pull Up

Choose the **Pull up** command in the **Edit** Menu to integrate a lower-level segment into the structure diagram on the current hierarchical level.

To do this:

1. Select a lower area. This may be a single lower segment or several lower segments arranged sequentially.
2. Choose the **Pull up** command in the **Edit** menu. The selected area is pulled up.

With this command, you can (re)integrate segments created automatically when you open or paste a source file as well as refinements and lower-level areas into the next hierarchical level.

Warning! The segment header for the higher-level segment is lost when this command is executed.

See also:

Hot Keys

Hierarchical Structuring

Other Commands in the Edit Menu

Undo

Use the **Undo** command in the **Edit** menu to undo the most recent edit. When you choose the **Edit** menu, the most recent edit that can be reversed is displayed to the right of **Undo**.

Note:

You can also reverse the last undo of an action. So you can switch between two states of a structure diagram as often as you like.

Use the **Undo Search/Replace** command to create the state of the structure diagram before you last chose **Replace** or **Replace all**.

See also:

[Hot Keys](#)

[Other Commands in the Edit Menu](#)

File Menu

The **File** menu provides commands for getting access to files and printers. You can use it for creating a file from a structure diagram, displaying files of various formats as structure diagrams, printing all or part of a structure diagram, executing programs and transferring files between your PC and a remote host.

The following commands are provided:

<u>New</u>	To create a new structure diagram
<u>Open</u>	To load an existing file
<u>Save</u>	To save a file
<u>Save as</u>	To save a file under a new name
<u>Print</u>	To print an <u>area</u> of a structure diagram
<u>Printer setup</u>	To install or select a printer
<u>Export as</u>	To export a structure diagram file
<u>Transfer</u>	To transfer files
<u>Execute</u>	To start a program
<u>Exit</u>	To quit EasyCODE(C++)
<u>Names of files opened last</u>	To open these structure diagram files

See also:

Menus

New

Use the **New** command in the **File** menu to quit editing a structure diagram and create a new one. This command is important if you want to scrap a structure diagram and start a new one or if you want to edit more than one structure diagram during a session. Keep in mind that by choosing the option "**Show each file in a separate window**" in the dialog window Options-Representation the new file is represented in addition to the file already open.

If the STDNEW.CPP file exists, the **New** command automatically places the structure diagram stored in this file as the first area on the worksheet. If the STDNEW.CPP file does not exist, the worksheet remains empty (only the symbolic name of the header will appear) .

When you start EasyCODE(C++), the **New** command will be executed automatically.

Note

Please note the difference between this command and the New command in the **Window** menu, which allows you to open a new segment window.

See also:

Show each file in a separate window

Assembling a Structure Diagram

Other Commands in the File Menu

Dialog Window Representation

Open

Use the **Open** command in the **File** menu to load a structure diagram file for editing it on the worksheet. Keep in mind that by choosing the option "**Show each file in a separate window**" in the dialog window Options-Representation the new file is represented in addition to the file already open.

EasyCODE(C++) can open 3 types of files:

Source files
SPX files
ETF files

To open a file:

1. Choose the **Open** command in the **File** menu.
2. The Open file dialog window will appear, in which you can make the appropriate entries.

See also:

Show each file in a separate window
General Notes on Opening and Inserting Source Files
Dialog Window Representation
Opening and Inserting Source Files
Opening and Inserting SPX Files
Opening and Inserting ETF Files
Using Files in a Network
Assembling
Hot Keys
Other Commands in the File Menu

Save

Use the **Save** command in the **File** menu to save a structure diagram in the source file you have opened or in a file that has already been named. In this case, the title bar shows the name of the current source file.

EasyCODE(C++) saves the structure diagram without opening a dialog window and automatically creates a backup copy of the original CPP file. When the structure diagram is saved, EasyCODE(C++) updates the line numbers entered into the constructs.

If you have activated the option **Copy without EasyCODE comments** in the **Source attributes** command of the **Options** menu, EasyCODE(C++) will create an additional copy of the source program file without EasyCODE comments, which has the filename extension CCP.

If you use the **Save** command for a new structure diagram, EasyCODE(C++) executes the Save as command. In this case, the title bar does not show the name of a source file.

See also:

[General Notes on Saving Source Files](#)

[Hot Keys](#)

[Other Commands in the File Menu](#)

Save As

Use the **Save as** command in the **File** menu to name and save a structure diagram as a source file. The title bar of the application window shows the current filename. EasyCODE(C++) automatically makes a backup copy of the original source file.

To name and save a structure diagram:

1. Choose the **Save as** command in the **File** menu.
2. The Save file as dialog window will appear, in which you can make the appropriate entries.

See also:

Save File

Hot Keys

Other Commands in the File Menu

Update

Use the **Update** command in the **File** menu to update a structure diagram existing as an embedded object in the target file. This command will replace the Save command when you use EasyCODE(C++) for embedding objects.

See also:

[Hot Keys](#)

[Save File](#)

[Other Commands in the File Menu](#)

Printing a Structure Diagram

Use the **Print** command in the **File** menu to print all or part of a structure diagram.

To print:

1. If you want to print an area of a structure diagram, select it.
2. Choose the **Print** command in the **File** Menu.
3. The Print dialog window will appear, in which you can choose the desired options.

While printing, EasyCODE(C++) informs you in the status bar about the segment and page number currently being printed.

During the printing process, a dialog window for canceling printing allows you to cancel printing, if required.

See also:

Hot Keys

Other Commands in the File Menu

Printer Setup

Use the **Printer setup** command in the **File** menu to choose the printer on which your structure diagram is to be printed. Here you can also specify the size and orientation of the print paper (portrait/landscape). When you start EasyCODE(C++), the active printer is the one selected in Windows or the one last selected. When you change the printer, the default setting is the default font for the selected printer.

To change the printer:

1. Choose the **Printer setup** command in the **File** menu.
2. The Printer setup dialog window will appear, in which you can choose the required settings.

See also:

Printer Font

Other Commands in the File Menu

Export As

Use the **Export as** command in the **File** menu to save a structure diagram file in EasyCODE text format (ETF file), so that you can use this file as an interface to other products.

To export a file:

1. Choose the **Export as** command in the **File** menu.
2. The Export as dialog window will appear, in which you can make the appropriate entries.

Note

The ETF format may be used as an interface to other tools such as text editors, so that you can, for instance, execute global changes or display EasyCODE structure diagrams in other forms of representation or vice versa. You may use the **Open** command in the **File** menu to import ETF files. For a more detailed description of the ETF file format please refer to the ETF.RTF file in the installation directory.

See also:

Open Command in the File Menu

Opening and Inserting ETF files

Hot Keys

Other Commands in the File Menu

Transfer

Use the **Transfer** command in the **File** menu to transfer one or several files from your personal computer into BS2000 and vice versa or load one or more files from BS2000 into your personal computer.

Above all, you can transfer the source program files created by EasyCODE(C++) to BS2000.

When you want to send a source file, please note that EasyCODE(C++) does not transfer the contents of the worksheet, but the file contents. Therefore you should save the current file before the transfer.

When you choose this command, the Transfer dialog window will appear.

See also:

Other Commands in the File Menu

Execute

Use the **Execute** command in the **File** menu to open a submenu listing the currently configured programs that can be executed. Choose a program to start it. If you have not yet configured an executable program, the **Execute** command is dimmed. Configure one or more executable programs with the [Program linking command in the Options menu](#).

After the selection of a program, it will be executed with the settings specified in the [Program settings dialog window](#). After execution of the program, EasyCODE(C++) will show the result file in the [Result file window](#) provided that you have selected this option in the Program settings dialog window and no errors have occurred in the program. Above all, you can [start a compiler directly from EasyCODE\(C++\)](#).

See also:

[Other Commands in the File Menu](#)

Start Program

Use this command to start the selected program with all parameters set in the **Options** menu.

See also:

[Execute Command in the File Menu](#)

[Other Commands in the File Menu](#)

Exit

To quit EasyCODE(C++), choose the **Exit** command in the **File** menu or the Close command in the control menu of the application window.

Note If you have changed any options concerning program linking, editors, saving, source attributes or file transfer, you will now be asked whether you want to keep these settings for later EasyCODE(C++) sessions.
Besides, the size and the position of the EasyCODE(C++) application window will be saved for the next EasyCODE(C++) session, no matter which option you choose.

See also:

Other Commands in the File Menu

Opening One of the Files Opened Last

At the bottom of the **File** menu, EasyCODE(C++) displays a list of the files opened last. In order to open a file from this list, select the filename or enter the number displayed to the left of the filename as a hot key. The file opened last will be at the top of the list, the file opened before will be listed next etc.

The maximum number of entries is 9, default value is 4. You can specify a value between 0 and 9 in the [Recent File List] section of the INI file (entry: **OpenFileListLength**).

See also:

Open

to open a structure diagram file

Insert Menu

The **Insert** menu provides constructs and areas of structure diagrams saved as files, which you need for designing a structure diagram. Some of them are dimmed to show that they cannot be used at present and are not available for the next editing action.

To insert a construct or an area:

1. Choose the appropriate menu item in the **Insert** menu.
2. Select a valid insertion point for the construct or area.

See also:

[EasyCODE\(C++\) Constructs](#)

[Inserting a File](#)

[Selecting a Valid Insertion Point](#)

[Other Menus](#)

EasyCODE(C++) Constructs

Statement

if

switch

case branch

default branch

for

while

do while

Class

private

public

protected

Function

Block

break

continue

return

Inserting a File

Use the **File** command in the **Insert** menu to insert a [source file](#), an [SPX file](#) or an [ETF file](#) into your current [structure diagram](#).

The process of selecting the file to be inserted is similar to that described in the [Open](#) command in the **File** menu. The [Insert file dialog window](#) will appear, in which you can enter the filename.

After the **File** menu item, you see a list containing the [files inserted last](#). In order to insert a file from this list, select a filename or enter the number displayed to the left of the filename as a hot key.

After opening the file, you have to set a valid [insertion point](#).

See also:

[Open File](#)

[Setting a Valid Insertion Point](#)

[Assembling](#)

[Insert Menu](#)

Inserting One of the Files Inserted Last

EasyCODE(C++) displays a list of the files inserted last at the end of the **Insert** menu. In order to insert a file from this list, select a filename or enter the number displayed to the left of the filename as a hot key.

The file inserted last will be on top of the list, the file inserted before will be listed next etc.

The maximum number of entries is 9, default value is 4. You may specify a value between 0 and 9 in the [Recent File List] section of the INI file (entry: **InsertFileListLength**).

See also:

[Opening a File](#) to open a structure diagram file

Window Menu

The **Window** menu provides commands for editing structure diagrams in any number of document windows called segment windows.

To do this, use the following commands

New

Cascade

Tile

Arrange icons

Close window

and further options which allow you to activate the required segment window.

Each segment window displays a different segment of the same structure diagram. You can scroll and view independently within each segment window, and also view various areas at the same time.

This offers advantages in the following applications:

- Editing part of the control flow of a structure diagram while at the same time viewing the data definitions (and vice versa),
- Viewing and editing different segments at the same time,
- Copying and moving areas to other segments or over great distances within a segment.

See also:

Activating Windows

Other Menus

New Window

Choose the **New** command in the **Window** menu to open a new segment window. This is now the active window, and it displays the same segment (or the top hierarchical level) as the previously active segment window.

Note

Please note the difference between this command and the New command in the File menu, which allows you to create a new structure diagram.

See also:

Control Menu of the Segment Windows

Window Menu

Cascade

Use the **Cascade** command in the **Window** menu to arrange all open segment windows within the application window in such a way that they overlap like a stack of files. With this arrangement, the active segment window is displayed with a relatively large worksheet. Unless too many segment windows are open, you can see all the title bars and can activate any segment window by clicking it with the left mouse button.

Note

This command does not apply to reduced segment windows.

See also:

Control Menu of the Segment Windows
Window Menu

Tile

Choose the **Tile** command in the **Window** menu to arrange all open segment windows within the application window side by side, so that each window is fully displayed with a relatively small worksheet.

Note

This command does not apply to reduced segment windows.

See also:

Control Menu of the Segment Windows

Window Menu

Arrange Icons

Choose the **Arrange icons** command in the **Window** menu to arrange the icons for the open segment windows side by side.

See also:

Control Menu of the Segment Windows

Window Menu

Close Window

Choose the **Close window** command in the **Window** menu to close the active segment window. When you do this, you do not exit the structure diagram itself, but the active segment. The previously active window becomes the active window again.

If this command is used for the last but one open segment window, EasyCODE(C++) displays the last window in full size in the application window.

See also:

[Control Menu of the Segment Windows](#)
[Window Menu](#)

Activating a Segment Window

Each time a new segment window is opened, a line is added to the **Window** menu containing the segment number and the segment header of the segment displayed in the window. The entries are numbered consecutively, and the entry for the active segment window is checked.

To activate a segment window, you can either click it with the left mouse button, or select the corresponding entry in the **Window** menu. The second method is particularly useful if the required segment window is not visible and you cannot click it directly.

Notes:

1. The commands available in the EasyCODE(C++) menus depend on the active segment window. Each modification of the structure diagram in the active segment window automatically leads to a corresponding modification in all other segment windows.
2. After opening the tenth segment window, you will find the option **More windows** in the **Window** menu. Select this option to get a list of all open segment windows. Select the segment window you want to activate from this list.

See also:

[Control Menu of the Segment Windows](#)

[Window Menu](#)

More Than 10 Segment Windows

You are using so many segment windows that they cannot be displayed in the **Window** menu at the same time.

By choosing the **More windows** command, you will get a list of all segment windows. Use the scroll bar to the right of the list to scroll the list of segment windows until you find the required segment window. By doubleclicking its name, you can activate it.

See also:

Window Menu

Help Menu

This menu provides several subjects on which you can get information. To get access to this help information, choose the corresponding menu item. The required information will then be displayed.

The help menu contains the following commands:

[Help on selected text](#)

[Index](#)

[How to use](#)

[Procedures](#)

[Keyboard/mouse commands](#)

[Glossary](#)

[Help on Help](#)

[About EasyCODE\(C++\)...](#)

[System info...](#)

See also:

[Other Menus](#)

About EasyCODE(C++)

By choosing this menu item, you will get information about the EasyCODE(C++) manufacturer as well as the name, version number and copyright note of the product.

See also:

[Help Menu](#)

System Info

When you choose the **System info** command in the **Help** menu, the "Microsoft System Information" will appear.

Note: If the **MSINFO** application has not been installed in your system, the menu item **System info** will not be available.

Help on selected Text

Use this command to get information about the selected text. EasyCODE will search in a list of HLP files you have specified in the INI file in [HelpFileList]. If the list contains more than one file name, a menu with the alias names of the help files (as chosen in the Ini file; see description of the Ini file CUSTOM.DOC) opens, where you can choose any help file.

Define, for example, the help file(s) of your compiler or your software engineering environment (e.g. MFC.HLP for Microsoft Visual C++), to get the description appropriate for the function call.

You can also call this command by pressing Alt+F1.

See also:

Description of the INI entries in the CUSTOM.DOC file

[Help menu](#)

EasyCODE Add-In Command

You have called the help function for a command configured via the add-in interface, but there is not help text available.

You can write a help text yourself and assign the HLP file in the INI file, under "AddIn", entry "AddInCmd<n>".

Menu for choosing the HLP file

This menu appears if you have chosen **Help on selected text** and there are more than one help files available. The name of the help file is the one you have entered in the definition of the help file in the Ini file as alias name.

Choose one of the available names to activate the respective help file.

EasyCODE(C++) Online-Help

EasyCODE(C++) has an online help. With this you can get at any time get specific help for the current situation by pressing F1.

You have the following possibilities:

1. By calling a menu and pressing F1 you get help on the menu.
2. By calling a command in the menu and pressing F1 you get help on this command.
3. During a step you get help for this step by pressing F1.
4. If a report window appears, you can get help on this window by pressing F1.
5. If a dialog window appears, you can get help on this window by pressing F1.
6. If there is an active Windows help application you can get help on the use of the Windows help application by pressing F1.

See also:

[Index](#)

Proceed in the same way as when choosing a menu or command, but keep the left mouse button pressed while pressing F1.

Options Menu

With the commands in the **Options** menu, you can design the EasyCODE(C++) environment according to your requirements.

The **Options** menu provides the following commands:

[Repeat](#)

[Overview](#)

[Page preview](#)

[Structure list](#)

[Result file](#)

[Toolbox](#)

[Tool bar](#)

[Status bar](#)

[Page numbers](#)

[Representation](#)

[Font](#)

[Source attributes](#)

[Program linking](#)

[Editors](#)

[File transfer](#)

[Project](#)

[Save settings](#)

See also:

[Designing Your Working Environment](#)

[Other Menus](#)

Repeat

The **Repeat** command in the **Options** menu allows you to repeat the Insert menu commands and the Copy and Clipboard - Copy commands without selecting them each time.

Whenever the **Repeat** command is chosen, the mouse pointer representing an area remains unchanged until you choose a different command or press ESC. When you choose the **Repeat** command, a check mark appears on the menu item line in the **Options** menu. The **Repeat** command remains checked until you choose it again.

See also:
Options Menu

Representation

Use the **Representation** command in the **Options** menu to decide the colors to display the structure diagram on the screen, how to display the **if** construct, whether line numbers are to be displayed and whether segment numbers are to be displayed.

You can also choose whether or not to open a new EasyCODE window when opening or positioning.

First choose the **Layout** command in the **Options** menu. The Layout dialog window will appear, in which you can make the required entries.

See also:

Options Menu

Setting the Printer or Screen Font

Use the **Font** command in the **Options** menu to set the font in which the text in a structure diagram will be presented on the screen or on the printer.

See also:

Options Menu

Setting the Screen Font

Use the **Screen** command in the **Font** submenu of the **Options** menu to choose the font in which the text in a structure diagram will be displayed on the screen.

To choose a screen font:

1. Choose the **Font** command in the **Options** menu.
EasyCODE(C++) will then display a new menu.
2. Choose the **Screen** command in this menu.
The Screen font dialog window will appear.
3. Select the desired screen font.

See also:

Setting the Printer Font
Options Menu

Setting the Printer Font

Use the **Printer** command in the **Font** submenu of the **Options** menu to choose the font in which text in a structure diagram will be printed.

To choose a printer font:

1. Choose the **Font** command in the **Options** menu.
EasyCODE(C++) will then display a new menu.
2. Choose the **Printer** command in this menu.
The Printer font dialog window will appear.
3. Select the desired printer font.

See also:

Setting the Screen Font
Options Menu

Default Settings for File Transfer

Use the **File transfer** command in the **Options** menu to choose the transfer-system and to select the default settings for transferring files between your PC and BS2000. When you choose this command, the File transfer attributes dialog window will appear.

See also:

Transfer Command in the File Menu

Options Menu

Project

Use the **Project** command in the **Options** menu to choose the project for which you want to get information on data definitions and references. You enter the name of a project data base where the relations and the dependencies between all source files belonging to the project are stored. The creating of a project data base (specifying the source files and generating the relations) must have been done with a special program.

To do this:

1. Choose the **Project** command in the **Options** menu.
An Open project data file dialog window will appear.
2. Select the file of your choice in this dialog window.

See also:

Command View Definition in the View Menu

Command View Reference in the View Menu

Options Menu

Source Attributes

Use the **Source attributes** command in the **Options** menu to define the character set for the file to be saved as well as the indent, if "{" and "}" also should be indented and whether a copy of the source file, the CCP file, is to be created. When you choose this command, the Source attributes dialog window will appear.

See also:

Save Command in the File Menu

Options Menu

Opening/Closing the Overview Window

Use the **Overview** command in the **Options** menu to display the entire segment of the active segment window in a small separate window.

When you choose this command, the small window will appear. Choose the command again to close the window. You can also use the CTRL+V keys to choose this command.

See also:

[Viewing with the Overview Window](#)

[Control Menu of the Overview Window](#)

[Options Menu](#)

Switching the Toolbox On/Off

Use the **Toolbox** command in the **Options** menu to switch on the display of the toolbox; when you choose this command again, the display will be switched off. The toolbox contains one icon for each construct of the **Insert** menu.

See also:

[Description of the Toolbox](#)

[Options Menu](#)

Switching the Tool Bar On/Off

Use the **Tool bar** command in the **Options** menu to switch on the display of the tool bar; when you choose this command again, the display will be switched off.

See also:

[Description of the Tool Bar](#)

[Options Menu](#)

Switching the Status Bar On/Off

The Status bar informs you about the current status of the application and the keyboard. By choosing the **Status bar** command in the **Options** menu, you activate the status bar, by repeating the command you switch it off again.

When EasyCODE(C++) is started, the status bar will be switched on automatically if you did not switch it off in an earlier EasyCODE(C++)-session by using the Save settings command.

See also:

Options Menu

Switching Page Numbers On/Off

Use the **Page numbers** command in the **Options** menu to switch off the display of the number of pages required for printing the currently active segment shown in the status bar. Choose the **Page numbers** command again to switch the print page numbers on.

The numbers are displayed as follows:

W x H <number of pages for width> x <number of pages for height> ,

e.g. **W x H 2 x 3** for a structure diagram that is 2 print pages wide and 3 print pages high. In case of coordinate overflow or trouble with the printer (if EasyCODE(C++) is not able to calculate the number of pages required for printing), the display will show W x H ? x ?.

When you start EasyCODE(C++), the print page numbers are switched on automatically, provided that you did not switch them off during an earlier EasyCODE(C++) session using the Save settings command and the status bar is switched on.

See also:

Switching the Status Bar On/Off
Options Menu

Opening/Closing the Structure List Window

Use the **Structure list** command in the **Options** menu to open the Structure list window displaying a list of the segments, the classes or the functions of the structure diagram.

You can also use the CTRL+I keys for choosing this command.

See also:

Options Menu

Opening/Closing the Page Preview Window

Use the **Page preview** command in the **Options** menu to display the active segment in a small separate window.

When you choose this command, the window will appear. Choose the command again to close the window.

See also:

[Hot Keys](#)

[Use of the Page Preview Command](#)

[Control Menu of the Page Preview Window](#)

[Options Menu](#)

Save Settings

Use the **Save settings** command in the **Options** Menu to save all current options and settings created during the session for later EasyCODE(C++)-sessions.

When you choose this command, the Save settings dialog window will appear.

The settings will be saved in the [Settings] section of the EASY-C++.INI file.

When you start EasyCODE(C++) again, you will find all settings exactly the same as they were at the time of saving.

See also:

Options Menu

Opening/Closing the Result File Window

When you choose the **Result file** command in the **Options** menu, the current result file will be displayed in a small separate window.

When you choose the command, the window will appear. Choose the command again to close the window. You can also use the CTRL+E keys for choosing this command.

See also:

[Use of the Result File Window](#)

[Execute Command in the File Menu](#)

[Program Linking Command in the Options Menu](#)

[Program Settings Dialog Window](#)

[Options Menu](#)

Program Linking

Use the **Program linking** command in the **Options** menu to specify a name and various settings for a program which you can start with the **Execute** command in the **File** menu. When you choose the **Program linking** command, the Programs dialog window will appear.

See also:

Program Settings Dialog Window

Execute Command in the File Menu

Options Menu

Editors

Use the **Editors** command in the **Options** menu to define a text editor which you can start using the External editor command in the **Edit** menu.

Moreover, you define an editor for correcting errors and an editor for displaying the error file. When you choose this command, the Editors dialog window will appear.

See also:

Hot Keys

Options Menu

View Menu

Choose the commands of this menu to view a different segment or a different construct within the active segment window.

Note that - in case positioning has to be done in another segment - by choosing the option "**Show each file in a separate window**" in the dialog window Options-Representation the new segment is shown in addition to the segment already open.

If you want to edit several segments at the same time, we recommend that you use several segment windows (Window Menu) instead of having to switch from one to another.

To view a section of the same segment that is currently not visible, you can use the Overview window or scroll the structure diagram in the segment window.

See also:

Hierarchical Structuring

Other Menus

Dialog Window Representation

View Commands from EasyCODE(C++)

[View Down](#)

[View Up](#)

[View Top](#)

[View Line](#)

[View Definition](#)

[View References](#)

View Down

Choose the **Down** command in the **View** menu to switch to a lower hierarchical level within the active segment window to edit a segment.

To switch to a lower level:

1. Select the gray box symbolizing the segment you want to view. Make sure that you select not only the segment header, but the gray surface itself.
2. Choose the **Down** command in the **View** menu.

Note

You can also execute this command by doubleclicking the gray box with the left mouse button or by selecting the element and then pressing the ENTER key.

See also:

Hot Keys

Hierarchical Structuring

View Menu

Menu Positioning

Dialog Window Representation

View Up

Choose the **Up** command in the **View** menu to return from editing a segment to a higher hierarchical level within the active segment window.

After the command has been executed, the segment previously edited will be symbolized in the higher hierarchical level by a gray box containing the segment header of your choice.

You can also choose this command by doubleclicking the segment window outside the structure diagram with the left mouse button.

See also:

[Show Each Segment in a Separate Window](#)

[Hot Keys](#)

[Hierarchical Structuring](#)

[View Menu](#)

[Menu Positioning](#)

[Dialog Window Representation](#)

View Top

Choose the **Top** command in the **View** menu to return from editing a segment on a lower hierarchical level to the structure diagram on the top level in the active segment window.

See also:

Show Each Segment in a Separate Window

Hot Keys

Hierarchical Structuring

View Menu

Dialog Window Representation

View Line

Choose the **Line** command in the **View** menu to view the element which corresponds to a specific line in your structure diagram in the active segment window.

When you choose this command, the View dialog window will appear.

You can also use the key combination ALT+# to choose this command.

See also:

Show Each Segment in a Separate Window

View Menu

Dialog Window Representation

View Definition

Choose the **Definition** command in the **View** menu to view the corresponding **Definition** for a name.

The following definitions can be viewed:

- subroutines (functions),
- data,
- classes,
- types,
- macros,
- include files,
- library routines and
- subsystems

To do this:

1. Select a text area or move the mouse cursor to the word whose definition you want to view.
2. Choose the **Definition** command in the **View** menu.

If there is no unique definition, EasyCODE(C++) will show a selection window where you can decide which definition you want to view.

If the definition is in the same file - depending on the selection of "Show each segment in a separate file" - the definition is viewed and can be edited.

If the definition is in another file - depending on the selection of "Show each segment in a separate file" - this file is opened, the definition is viewed and can be edited.

If EasyCODE(C++) cannot find a suitable definition, the original view is kept.

See also:

[Show Each Segment in a Separate Window](#)

[Hot Keys](#)

[View Menu](#)

[Dialog Window Representation](#)

View References

Choose the **References** command in the **View** menu to view for a name all the occurrences of this element.

To do this:

1. Select a text area or move the mouse cursor to the word whose occurrences you want to view.
2. Choose the **References** command in the **View** menu.

If more than one references are found, EasyCODE(C++) will display a selection window where you can choose which reference you want to view.

If the (chosen) reference is in the same file - depending on the selection "Show each segment in a separate file" - this reference is viewed and you can edit it.

If the (chosen) reference is in another file - depending on the selection "Show each segment in a separate file" this file is opened, the reference is viewed and you can edit it.

If EasyCODE(C++) cannot find a suitable reference, the original view is kept.

See also:

[Show Each Segment in a Separate Window](#)

[Hot Keys](#)

[View Menu](#)

[Dialog Window Representation](#)

Menus

A menu is a list of commands or entries which you can choose to start an EasyCODE(C++) or Windows program function. You find the EasyCODE(C++) menus in the menu bar below the title bar.

The following menus are available:

<u>File</u>	To get access to files and printers
<u>Edit</u>	To edit a <u>structure diagram</u>
<u>Insert</u>	To insert <u>constructs</u> and <u>areas</u> of structure diagrams that have been saved
<u>Add-In</u>	Commands of an add-in program
<u>View</u>	To view <u>segments</u> and constructs
<u>Window</u>	To manage <u>segment windows</u>
<u>Options</u>	To choose EasyCODE(C++) settings
<u>Help</u>	To get information about EasyCODE(C++)

Most EasyCODE(C++) windows provide a control menu in the upper left corner containing commands for window manipulation. You can choose a menu by selecting it with the keyboard or by clicking it with the left mouse button. Within a selected menu, some commands or entries are dimmed to show that they are not available at present.

See also:

ProceduresIDX_PROCS

Add-In

This menu name is a symbolic name for a menu whose name and commands you can set yourself. With this add-in interface you can integrate a program into the EasyCODE interface and control it from there. This interface is designed primarily for the integration of debuggers or test tools, but you can combine any suitable program with EasyCODE.

For a detailed description of the use and the configuration of this interface, please contact the EasyCODE manufacturer.

<empty>

This command name is used as a symbolic name for a command of the Add-In whose function you can choose yourself. This command is passed via the add-in interface to the program configured there. You can define a name, a hot key, a toolbar bitmap, a status bar text and a help file for this menu point.

For a detailed description of the use and the configuration of this interface, please contact the EasyCODE manufacturer.

Control Menus

A control menu is symbolized by a small horizontal bar to the left of the title bar. It provides commands for controlling window size and position, closing the window and changing to other windows.

See also:

[Control Menu of the Application Window](#)

[Control Menu of the Segment Windows](#)

[Control Menu of the Structure List Window](#)

[Control Menu of the Overview Window](#)

[Control Menu of the Page Preview Window](#)

[Control Menu of the Result File Window](#)

[Control Menu of the Toolbox](#)

[Control Menu of the Dialog Windows](#)

[Control Menu of the Message Windows](#)

[Menus](#)

Control Menu of the Application Window

The control menu of the application window provides the following commands:

<u>Restore</u>	To restore the previous window size
<u>Move</u>	To move the window
<u>Size</u>	To change the window size
<u>Minimize</u>	To reduce window to symbol
<u>Maximize</u>	To enlarge the window to fill the entire screen
<u>Close</u>	To close the window and quit the application
<u>Switch to...</u>	To switch to the task list

See also:

Menus

Control Menu of the Segment Windows

The control menu of the segment windows provides the following commands:

<u>Restore</u>	To restore the previous window size
<u>Move</u>	To move the window
<u>Size</u>	To change the window size
<u>Minimize</u>	To reduce window to symbol
<u>Maximize</u>	To enlarge the window to fill the entire screen
<u>Close</u>	To close the window and quit the application
<u>Next</u>	To change to the next segment window

See also:

Window Menu

Menus

Control Menu of the Dialog Windows

The control menu of the dialog windows provides the following commands:

Move	To move the window
Size	To change the size of the editing window
Close	To close the window

See also:

[Menus](#)

Control Menu of the Message Windows

The control menu of the message windows provides the following commands:

Move	To move the window
Close	To close the window

It is, however, more comfortable to move the window by moving the mouse pointer to the title bar, pressing the left mouse button and dragging the window frame to the desired position.

See also:

[Menus](#)

Control Menu of the Overview Window

The control menu of the Overview window provides the following commands:

<u>M</u> ove	To move the window
<u>S</u> ize	To change the window size
<u>C</u> lose	To close the window
<u>S</u> witch to...	To switch to the task list

See also:

Menus

Control Menu of the Structure List Window

The control menu of the Structure list window provides the following commands:

<u>M</u> ove	To move the window
<u>S</u> ize	To change the window size
<u>C</u> lose	To close the window
<u>S</u> witch to...	To switch to the task list

See also:

Menus

Control Menu of the Page Preview Window

The control menu of the Page preview window provides the following commands:

<u>M</u> ove	To move the window
<u>S</u> ize	To change the window size
<u>C</u> lose	To close the window
<u>S</u> witch to...	To switch to the task list

See also:

Menus

Control Menu of the Result File Window

The control menu of the Result file window provides the following commands:

<u>M</u> ove	To move the window
<u>S</u> ize	To change the window size
<u>C</u> lose	To close the window
<u>S</u> witch to...	To switch to the task list

See also:

Menus

Control Menu of the Toolbox

The control menu of the toolbox provides the following commands:

<u>M</u> ove	To move the window
<u>C</u> lose	To close the window
<u>S</u> witch to...	To change to the task list

See also:

Menus

This Command is not Available in the Overview Window.

See also:

[Other Commands of this Control Menu](#)

[Use of the Overview Window](#)

This Command is not Available in the Structure List Window.

See also:

[Other Commands of this Control Menu](#)

[Use of the Structure List Window](#)

This Command is not Available in the Page Preview Window.

See also:

[Other Commands of this Control Menu](#)

[Use of the Page Preview Window](#)

This Command is not Available in the Result File Window.

See also:

[Other Commands of this Control Menu](#)

[Use of the Result File Window](#)

This Command is not Available in the Toolbox.

See also:

[Other Commands of this Control Menu](#)

[Description of the Toolbox](#)

Restore Command in the Control Menu of the Application Window

Choose this command to restore the original size of the application window from maximum size or from the symbol.

See also:

Control Menu of the Application Window

Control Menus

Menus

Restore Command in the Control Menu of the Segment Windows

Choose this command to restore the original size of the active segment window from maximum size or from the symbol.

See also:

Window Menu

Control Menu of the Segment Windows

Control Menus

Menus

Move Command in the Control Menu of the Application Window

Choose this command to move the application window on the screen without changing its size. Press the arrow keys until the window frame is in the desired position. Then click the left mouse button or press ENTER to move the window to its new position.

See also:

[Control Menu of the Application Window](#)

[Control Menus](#)

[Menus](#)

Move Command in the Control Menu of the Segment Windows

Choose this command to move the active segment window on the screen without changing its size. After having chosen this command, press the arrow keys until the window frame is at the desired position. Then click the left mouse button or press the ENTER key to move the window to its new position.

See also:

[Control Menu of the Segment Windows](#)

[Control Menus](#)

[Menus](#)

Move Command in the Control Menu of the Overview Window

Choose this command to move the Overview window on the screen without changing its size. To do this, move the gray window frame using the mouse or the arrow keys until it corresponds to the new window position. After having chosen this command, press the arrow keys until the window frame is in its new position. Then click the left mouse button or press the ENTER key to move the window to its new position.

See also:

Control Menu of the Overview Window

Control Menus

Menus

Move Command in the Control Menu of the Structure List Window

Choose this command to move the Structure list window on the screen without changing its size. To do this, move the gray window frame using the mouse or the arrow keys until it corresponds to the new window position. After having chosen this command, press the arrow keys until the window frame is in the desired position. Then click the left mouse button or press the ENTER key to move the window to its new position.

See also:

[Use of the Structure List Window](#)

[Control Menu of the Structure List Window](#)

[Control Menus](#)

[Menus](#)

Move Command in the Control Menu of the Page Preview Window

Choose this command to move the Page preview window on the screen without changing its size. To do this, move the gray window frame using the mouse or the arrow keys until it corresponds to the new window position. After having chosen this command, press the arrow keys until the window frame is in the desired position. Then click the left mouse button or press the ENTER key to move the window to its new position.

See also:

[Use of the Page Preview Window](#)

[Control Menu of the Page Preview Window](#)

[Control Menus](#)

[Menus](#)

Move Command in the Control Menu of the Result File Window

Choose this command to move the Result file window on the screen without changing its size. To do this, move the gray window frame using the mouse or the arrow keys until it corresponds to the new window position. After having chosen this command, press the arrow keys until the window frame is in the desired position. Then click the left mouse button or press the ENTER key to move the window to its new position.

See also:

[Use of the Result File Window](#)

[Control Menu of the Result File Menu](#)

[Control Menus](#)

[Menus](#)

Move Command in the Control Menu of the Toolbox

Choose this command to move the toolbox on the screen. To do this, move the gray window frame using the mouse or the arrow keys until it corresponds to the new window position. After having chosen this command, press the arrow keys until the window frame is in the desired position. Then click the left mouse button or press the ENTER key to move the window to its new position.

See also:

[Description of the Toolbox](#)

[Control Menu of the Toolbox](#)

[Control Menus](#)

[Menus](#)

Size Command in the Control Menu of the Application Window

Choose this command to change the size of the application window on the screen. To do this, use the mouse or the arrow keys to change the size of the gray window frame until it corresponds to the desired window size.

See also:

Control Menu of the Application Window

Control Menus

Menus

Size Command in the Control Menu of the Segment Windows

Choose this command to change the size of the segment window on the screen. To do this, use the mouse or the arrow keys to change the size of the gray window frame until it corresponds to the desired window size. You can enlarge a segment window only to the size of the application window.

See also:

Control Menu of the Segment Windows

Control Menus

Menus

Size Command in the Control Menu of the Overview Window

Choose this command to change the size of the Overview window on the screen. To do this, use the mouse or the arrow keys to change the size of the gray window frame until it corresponds to the desired window size.

See also:

[Use of the Overview Window](#)

[Control Menu of the Overview Window](#)

[Control Menus](#)

[Menus](#)

Size Command in the Control Menu of the Structure List Window

Choose this command to change the size of the Structure list window on the screen. To do this, use the mouse or the arrow keys to change the size of the gray window frame until it corresponds to the desired window size.

See also:

[Use of the Structure List Window](#)

[Control Menu of the Structure List Window](#)

[Control Menus](#)

[Menus](#)

Size Command in the Control Menu of the Page Preview Window

Choose this command to change the size of the Page preview window on the screen. To do this, use the mouse or the arrow keys to change the size of the gray window frame until it corresponds to the desired window size.

See also:

[Use of the Page Preview Window](#)

[Control Menu of the Page Preview Window](#)

[Control Menus](#)

[Menus](#)

Size Command in the Control Menu of the Result File Window

Choose this command to change the size of the Result file window on the screen. To do this, use the mouse or the arrow keys to change the size of the gray window frame until it corresponds to the desired window size.

See also:

[Use of the Result File Window](#)

[Control Menu of the Result File Window](#)

[Control Menus](#)

[Menus](#)

Minimize Command in the Control Menu of the Application Window

Choose this command to reduce the application window to a symbol.

See also:

Control Menu of the Application Window

Control Menus

Menus

Minimize Command in the Control Menu of the Segment Windows

Choose this command to reduce the active segment window to a symbol.

See also:

Window Menu

Control Menu of the Segment Windows

Control Menus

Menus

Maximize Command in the Control Menu of the Application Window

Choose this command to enlarge the application window to maximum size, so that it fills the entire screen.

See also:

Control Menu of the Application Window

Control Menus

Menus

Maximize Command in the Control Menu of the Segment Windows

Choose this command to enlarge the active segment window to its maximum size within the application window. No other segment window of the application will be visible.

See also:

Window Menu

Control Menu of the Segment Windows

Control Menus

Menus

Close Command in the Control Menu of the Application Window

Choose this command to close the application window and quit EasyCODE(C++). You can also use the hot keys ALT+F4 to choose this command.

See also:

[Exit in the File Menu](#)

[Control Menu of the Application Window](#)

[Control Menus](#)

[Menus](#)

Close Command in the Control Menu of the Segment Windows

Choose this command to close the active segment window. You can also use the hot keys CTRL+F4 to choose this command. Since at least one segment window is always open, this command is available only if at least two segment windows are open when the command is chosen.

See also:

[Close Window in the Window Menu](#)

[Window Menu](#)

[Control Menu of the Segment Windows](#)

[Control Menus](#)

[Menus](#)

Close Command in the Control Menu of the Overview Window

Choose this command to close the Overview window. You can also use the hot keys ALT+F4 or CTRL+V.

See also:

Overview Window in the Options Menu

Control Menu of the Overview Window

Control Menus

Menus

Close Command in the Control Menu of the Structure List Window

Choose this command to close the Structure list window. You can also use the hot keys ALT+F4 or CTRL+I.

See also:

[Use of the Structure List Window](#)

[Control Menu of the Structure List Window](#)

[Control Menus](#)

[Menus](#)

Close Command in the Control Menu of the Page Preview Window

Choose this command to close the Page preview window. You can also use the hot keys ALT+F4 or CTRL+A.

See also:

[Use of the Page Preview Window](#)

[Control Menu of the Page Preview Window](#)

[Control Menus](#)

[Menus](#)

Close in the Control Menu of the Result File Window

Choose this command to close the Result file window. You can also use the hot keys ALT+F4 or CTRL+S.

See also:

[Use of the Result File Window](#)

[Control Menu of the Result File Window](#)

[Control Menus](#)

[Menus](#)

Close Command in the Control Menu of the Toolbox

Choose this command to close the toolbox. You can also use the hot keys ALT+F4 or CTRL+L.

See also:

[Description of the Toolbox](#)

[Control Menu of the Toolbox](#)

[Control Menus](#)

[Menus](#)

***Switch to...* Command in the Control Menu of the Application Window**

When you choose this command, the task list is displayed. With the help of the task list, you can comfortably change to other applications or close them. You can also use the hot keys CTRL+ESC to choose this command.

See also:

[Control Menu of the Application Window](#)

[Control Menus](#)

[Menus](#)

Next Command in the Control Menu of the Segment Windows

Choose this command to change to the next segment window. If only one segment window is available, this command has no effects.

See also:

[Control Menu of the Segment Windows](#)

[Control Menus](#)

[Menus](#)

***Switch to...* Command in the Control Menu of the Overview Window**

When you choose this command, the task list is displayed. With the help of the task list, you can comfortably change to other applications or close them.

See also:

[Control Menu of the Overview Window](#)

[Control Menus](#)

[Menus](#)

***Switch to...* Command in the Control Menu of the Structure List Window**

When you choose this command, the task list is displayed. With the help of the task list, you can comfortably change to other applications or close them.

See also:

[Use of the Structure List Window](#)

[Control Menu of the Structure List Window](#)

[Control Menus](#)

[Menus](#)

Switch to Command in the Control Menu of the Page Preview Window

When you choose this command, the task list is displayed. With the help of the task list, you can comfortably change to other applications or close them.

See also:

[Use of the Page Preview Window](#)

[Control Menu of the Page Preview Window](#)

[Control Menus](#)

[Menus](#)

***Switch to* in the Control Menu of the Result File Window**

When you choose this command, the task list is displayed. With the help of the task list, you can comfortably change to other applications or close them.

See also:

[Use of the Result File Window](#)

[Control Menu of the Result File Window](#)

[Control Menus](#)

[Menus](#)

***Switch to* in the Control Menu of the Toolbox**

When you choose this command, the task list is displayed. With the help of the task list, you can comfortably change to other applications or close them.

See also:

[Description of the Toolbox](#)

[Control Menu of the Toolbox](#)

[Control Menus](#)

[Menus](#)

Open Dialog Window

This dialog window will appear when you choose the Open command in the File menu. It provides the following dialog boxes:

File Name

Enter here the filename of the file to be opened. You can also enter the desired extension or select one of those provided in the **List Files of Type** box. Then click **Ok** to get a list of the files with this extension displayed. Select a name of this list.

Directories

Select here the directory in which the desired file has been stored.

List Files of Type

Choose one of the following filename extensions:

source files	(* .CPP) (* .CXX) (* .C) (* .H)	will display in the File Name dialog box all files with this extension (<u>source files</u>) that are stored in the current directory.
--------------	--	---

If you have specified other extensions than those mentioned above using the Source attributes or the Save settings commands in the **Options** menu, these extensions will also be offered here.

SPX files	(* .SPX)	will display in the File Name dialog box all files with this extension (<u>SPX files</u>) that are stored in the current directory.
ETF files	(* .ETF)	will display in the File Name dialog box all files with this extension (<u>ETF files</u>) that are stored in the current directory.
All files	(* .*)	will display in the File Name dialog box all files that are stored in the current directory.

You may add any filename extension to this list by entering the extension into the **File Name** field and opening a file with this extension. Use the **Save settings** command in the **Options** menu to save this extension for other EasyCODE sessions.

Drives

Select here the drive on which the file you want to open is stored.

Short info

When you choose a file, this dialog box will show the short information text you entered when you saved this file. If the text is too long to be fully displayed in this dialog box, you can use the scroll bar to the right to scroll to the desired position.

Last modification

This entry shows when the selected file was last saved.

write-protected

see Using Files in a Network

Click **Ok** to open the desired file. You can also open the file by doubleclicking the filename displayed in the **File Name** list with the left mouse button.

Click **Cancel** to escape.

Dialog Window Representation

Save As Dialog Window

This dialog window will appear when you choose the Save as command in the **File** menu or when you choose the Save command in the **File** menu for a file that has not yet been named or that has been opened with write-protection. It provides the following dialog boxes:

File Name

Enter here the filename of the file to be saved. You can also enter the required filename extension or select one of the extensions provided in the **List Files of Type** dialog box. Then click **Ok** to get a list of the files with this extension displayed. Select a name from this list.

Directories

Choose here the directory in which the file is to be saved. The default setting is the current drive and directory.

List Files of Type

Choose one of the following filename extensions:

source files	(* .CPP)	will display in the File Name dialog box all files with this extension (<u>source</u>
	(* .CXX)	<u>files</u>) that are stored in the current directory.
	(* .C)	
	(* .H)	

In addition to that, the source file extensions you specified using the Source attributes command or the Save settings command.

All files (*.*)	will display in the File Name dialog box all files that are stored in the current directory.
-----------------	---

Drives

Choose the drive on which you want to save the file.

Short info

If necessary, enter here explanatory information concerning the file.

Click **Ok** to name and save the file in the selected directory.

Click **Cancel** to escape.

Print Dialog Window

This dialog window will appear when you choose the Print command in the **File** menu. The following dialog boxes are available:

Area

Choose one of the options **Current segment**, **Selected area**, **Entire structure diagram** to select the area to be printed.

Default setting is **Entire structure diagram**.

Including lower segments

Choose this option if lower segments in the selected area are to be printed, too.

Maximum segment depth

Here you can specify how many lower levels of the structure diagram are to be printed.

If you have chosen "Current segment" or "Selected area" in the **Area** box, EasyCODE(C++) begins counting the levels from the current level, if you have chosen "entire structure diagram", the counting begins at the top level.

By moving the selection cursor into a different dialog box (e.g. by pressing the TAB key or by clicking the list box "Width x Height") in the Preview list, you will see the changes in the number of segments.

Page header

Choose this option if the print pages are to have a **header**. The header contains the EasyCODE component name, the name of the printed file, the segment number of the print page, the current date, the current time and consecutive page numbering.

Page numbers start at:

Here you can enter the first page number manually.

Fit into page

Choose this option if you want EasyCODE(C++) to choose an appropriate printer font size so that every segment which (with the current font size) will exceed one print page will fit exactly into one print page. If EasyCODE(C++) is not able to find an appropriate font, it will choose the smallest size of the current font still within the limits of the "Minimum font size" settings.

You can see the change in the segment size resulting from the font size modification by moving the enter cursor to a different entry box (e.g. by pressing the TAB key or by clicking the list box "Width x Height") in the Preview list.

Note: You should only use "TrueType fonts" and "Vector fonts" for this option, as they can be scaled smoothly, and the font size will fit exactly.

Minimum font size

Here you can specify the font size in points (pt) which is to be the minimum value for printing.

Default setting: 6 pt

Save space

Choose this option if EasyCODE(C++) shall not start a new page if another segment including the segment identification fits on the current page.

Preview >>

Click **Preview >>** to get information about the pages that will be required for printing.

When you choose this field, the Print window will be enlarged to the right.

With the use of the single-line *Segments* list field you determine whether only those segments that are too wide, only those that are too high, or both of them are to be listed.

For the selected printing area, a list of all segments requiring more than one print page - corresponding to the selected setting - will be displayed. Each entry shows the number of print pages

required in width (W) and height (H) as well as the segment number and the segment name. Every modification of a parameter relevant to the page size (different printer, page size, font size, header,...) will have immediate consequences on the number of pages required for printing, which will be updated at once. In case of coordinate overflow or trouble with the printer (if EasyCODE(C++) cannot calculate the number of pages required for printing), ? x ? will be displayed.

When you doubleclick a list entry with the left mouse button or select a list entry and then click the **View** button, EasyCODE(C++) will display this segment in the main window.

The Print window will then be closed, the modified settings will, however, be stored.

Click **Ok** to print the structure diagram with the selected options.

Click **Cancel** to close the dialog window without printing the structure diagram.

See also:

[Page Preview](#)

[Printer Font](#)

[Printer Setup in the File Menu](#)

Print Structure List

By choosing the option **Structure list** in the **Print** dialog window of the **File** menu you can print the list of the construct type entered in the structure list.

The print is done with the font entered in the **Options-Font-Printer** printer font on the entered printer. Each print page contains a header containing the application name, the product version, the file name, date, time as well as page number.

Under the header there is the header structure list and the construct type.

During the print the process can be canceled from a dialog window.

You can also print the structure list without opening EasyCODE. Enter into the command line the option P(rint)S(tructure) and a parameter for the construct type whose structure list is to be printed.

Parameter for the construct type

- 0 Segments
- 1 Classes
- 2 Functions

See also:

[Dialog Window Print](#)

[Page View](#)

[Print Font](#)

[Printer Setup in the File Menu](#)

Printer Setup Dialog Window

This dialog window will appear when you choose the Printer setup command in the **File** menu. It provides the following dialog boxes:

Printer

Specify whether you want to use the default printer or a special printer.

Click the arrow to the right of the one-line list box to get a list of all available printers displayed. The Special printers list contains all printers installed in Windows. You may install additional printers using the Windows Control Panel application.

Orientation

Specify whether you want to print in portrait or landscape format.

Paper

Click the arrow to the right of the one-line list box indicating size or feeding type to get a list of all available options displayed. Select the required option.

Click the **Options** button to modify the default settings of the selected printer.

Click **OK** to accept the printer and the selected printer options.

Click **Cancel** to ignore the changes.

Note

When you start EasyCODE(C++), the active printer is the one selected in Windows or the one last selected.

When you change to another printer, the default setting is the default font of the selected printer.

Export as Dialog Window

This dialog window will appear when you choose the Export as command in the **File** menu. The following dialog box are provided:

File Name

Enter here the filename you want to assign to the exported file. You can also enter the required filename extension or choose one of the extensions provided in the **List Files of Type** dialog box. Then click **Ok** to get a list of files with this extension displayed. Select a name from this list.

Directories

Choose here the directory into which the file is to be exported. The default setting is the current directory and drive.

List Files of Type

Choose one of the following filename extensions:

ETF files	(*.*TF)	will display in the File Name dialog box all files with this extension (<u>ETF files</u>) that are stored in the current directory.
All files	(*.*)	will display in the File Name dialog box all files stored in the current directory.

Drives

Choose here the drive on which the file is to be exported.

Click **Ok** to name and export the file into the selected directory.
Click **Cancel** to escape.

<File> is sent to <printer> on <port>

This dialog window will appear as long as the structure diagram is being prepared for printing and has not been consigned to the Windows Print Manager.

Click **Cancel** to cancel printing.

Insert Dialog Window

This dialog window will appear when you choose the Insert File command. It provides the following dialog boxes:

File Name

Enter here the filename of the file to be inserted. You can also enter the required filename extension, or select one of the extensions provided in the **List Files to Type** dialog box. Then click **Ok** to get a list of the files with this extension displayed. Select a name from this list.

Directories

Choose here the directory in which the required file is stored.

List Files of Type

Choose one of the following filename extensions:

source files	(* .CPP) (* .CXX) (* .C) (* .H)	will display in the File Name dialog box all files with this extension (<u>source files</u>) that are stored in the current directory.
SPX files	(* .SPX)	will display in the File Name dialog box all files with this extension (<u>SPX files</u>) that are stored in the current directory.
ETF files	(* .ETF)	will display in the File Name dialog box all files with this extension (<u>ETF files</u>) that are stored in the current directory.
All files	(* .*)	will display in the File Name dialog box all files stored in the current directory.

Drives

Choose the drive on which the file to be inserted is stored.

Short info

If a file is selected, this dialog box will show the short information text you entered when you saved this file. If the text is too long to be fully displayed in this dialog box, you can use the scroll bar to the right to scroll to the desired position.

Last modification

This entry shows when the selected file was last saved.

Click **Ok** to insert the required file. You can also insert the file by doubleclicking the filename in the **File Name** list with the left mouse button.

Click **Cancel** to escape

Multi-Line Editing Window

This dialog window will appear when you edit one of the following symbolic names, existing text or an empty box:

- <block comment>
- <class header>
- <declarations>
- <expression>
- <for expression>
- <function header>
- <header>
- <statement>
- <value>

The title bar shows the symbolic name corresponding to the text you are editing.

The dialog window provides the following dialog elements:

Multi-line text field

Enter here text corresponding to the symbolic name (not more than about 30,000 characters). When you edit existing text, the cursor is placed on the character you clicked to select the text in the structure diagram.

When you press the CTRL+TAB key combination, the cursor moves to the next tab position. The tab positions are predefined as follows:

9, 17, 25,(+8)

Functions of the ENTER key in a text box:

ENTER key	moves cursor to next line with same indent as line before
CTRL+ENTER	moves cursor to first column of next line
ALT+ENTER	same function as Ok button

Line

This dialog element shows the line number in relation to the editing window in which the cursor is placed.

Column

This dialog box displays the column position of the cursor in relation to the editing window.

Source line

This dialog element displays the line number in relation to the source file already generated in which the cursor is placed.

Overwrite

Activate this option, if EasyCODE(C++) shall overwrite the existing text as you enter the new text.

Click **Ok** to accept the edited text and transfer it to the structure diagram.

Click **Cancel** to ignore the changes.

You can move the multi-line editing window on the worksheet and change its height. The position and size of the last multi-line editing window are stored for the next call of a multi-line editing window. If you want the size and position of the editing window to be stored for the next EasyCODE(C++) session, you have

to choose the **Editing window** option in the [Save settings dialog window](#).

See also:

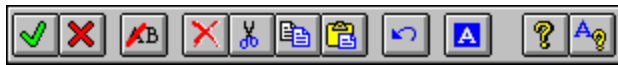
[Text Editing](#)

Editing with EasyCODE

With EasyCODE you can fill an area with text or edit an existing text directly in the structure diagram without opening a dialog window. If you have selected editing, a frameless edit field will appear at the position of the construct you want to edit.

During the editing a hatched frame will enclose the field to display the edit mode optically.

The standard symbol bar is replaced by the edit symbol bar.



Click a symbol to get further information on it.

Invoking the Edit process (for selected structure diagram area):

- by selecting the menu item **Edit-Edit text**

- by pressing **Shift+Return Key** or just the return key (as standard editor),

- by doubleclicking the editable area with the mouse,

- by pressing the  button, and

- by just typing.

The insertion label is placed at the position where it was set while selecting.

Confirming the Edit process (and taking over the changes):

- by pressing the  button in the edit toolbar,

- by pressing Alt+Return

- by choosing the corresponding menu point in the edit context menu,

- by mouseclick outside the edited area in the EasyCODE window,

- by choosing an EasyCODE command

Canceling the Edit process:

- by pressing the  button in the edit toolbar,

- by pressing ESC

- by choosing the corresponding menu point in the edit context menu,

Special key functions:

Tab	adds spaces up to the next tab position.
-----	--

Strg+Tab	changes the MDI window.
----------	-------------------------

Return	inserting a line change while keeping the current indentation.
--------	--

Strg+Return	inserting a line change and ignoring the current indentation.
-------------	---

Alt+Return	ends the editing
------------	------------------

Status bar

During the editing the line and column information is displayed in the status bar:

line: nnn, column: nnn, source line: nnn, source column: nnn.

W x H nn x nn size information in pages for width (W) and height (H),

OV

Overwrite mode; activated/deactivated by the **Ins** key, the **Overwrite** button or the corresponding command in the context menu.

Note:

As an alternative to direct editing within the structure diagram you can also use the EasyCODE-[Edit Window](#) or an [External Editor](#).

See also:

[Edit Command](#)

[Options Editors](#)

[Hot Keys](#)

File Transfer Settings Dialog Window

This dialog window will appear when you choose the File transfer command in the Options menu. It provides the following dialog boxes:

Transfer system

Processor name

Remote file

Remote Host

Transfer direction

File type

Follow-up processing

overwrite existing file

compressed transfer

Additional processing

Directory for FT files

Click **Ok** to accept the settings.

Click **Cancel** to escape.

See also:

Symbols in Transfer Jobs

File Transfer Command in the Options Menu

Transfer Command in the File Menu

Transfer Dialog Window

This dialog window will appear when you choose the Transfer command in the File menu. It provides the following dialog boxes:

Transfer list

The upper left section of the window displays a transfer list to which you can add any number of transfer jobs by using the buttons displayed to the right.

Default transfer list

Click this button to create a transfer list with which you can send the current CPP file and maybe also an ETF file, if existing, to the remote host.

Add

Click this button to add one or more transfer jobs to the transfer list. The Add dialog window will appear.

Modify

Click this button to alter a transfer job. The Modify dialog window will appear.

Change transfer direction

Click this button to reverse the transfer direction for an existing transfer job (from "send" to "receive" and the other way round).

Delete

Select one or more transfer jobs in the transfer list, then click this button to delete the selected transfer jobs.

Open transfer list

Click this button to open an FT file in order to edit a saved transfer list and start it, if required. The Open transfer list dialog window will appear.

Save transfer list

Click this button to save a transfer list as an FT file. The Save transfer list dialog window will appear.

Default logon password (or default user profile) of remote host

Depending on whether you have chosen user ID or user profile as your default logon authorization, you can enter a default password or a default user profile for the remote host. This default password or default user profile applies to all transfer jobs for which you do not enter a specific password or user profile.

Additional processing

Choose the **Start transfer list** button to carry out all transfer jobs on the transfer list.

Choose the **Start selected transfer s** button to carry out only the selected transfer jobs on the transfer list.

Choose the **Close** button to scrap the transfer list and quit transferring.

See also:

File Transfer Command in the Options Menu

Transfer Command in the File Menu

Add Dialog Window

This dialog window will appear when you click **Add** in the [Transfer dialog window](#). It contains the settings you selected for the [File transfer command in the Options menu](#) and provides the following dialog boxes:

[Local file](#)

[Remote file](#)

[Remote Host](#)

[Transfer direction](#)

[File type](#)

[Follow-up processing](#)

[overwrite existing file](#)

[compressed transfer](#)

Click **Ok** to add one or more [transfer jobs](#) to the [transfer list](#) using the entered filename of the local file.

Click **Cancel** to escape.

See also:

[Symbols in Transfer Jobs](#)

[File Transfer Command in the Options Menu](#)

[Transfer Command in the File Menu](#)

Modify Dialog Window

This dialog window will appear when you click **Modify** in the [Transfer dialog window](#). It displays the settings concerning the [transfer job](#) to be altered. Modify one or more of the following dialog boxes:

[Local file](#)

[Remote file](#)

[Remote Host](#)

[Transfer direction](#)

[File type](#)

[Follow-up processing](#)

[overwrite existing file](#)

[compressed transfer](#)

Click **Ok** to save your modifications.

Click **Cancel** to keep the original settings of the transfer job.

See also:

[Symbols in Transfer Jobs](#)

[File Transfer Command in the Options Menu](#)

[Transfer Command in the File Menu](#)

Local File Dialog Window

This dialog window will appear when you click **Browse** in the [Add dialog window](#) or in the [Modify dialog window](#). It provides the following dialog boxes:

File Name

Enter a filename which may contain the wildcards * or ?. You can also enter the required filename extension or select one of the extensions provided in the **List Files of Type** dialog box. Then click **Ok** so that EasyCODE(C++) will display the corresponding list of files. Choose a name from this list.

Directories

Choose here the directory in which the required file is stored.

List Files of Type

Choose one of the available filename extensions.

Drives

Choose the drive on which you want to search for the file.

Click **Ok** if you want the selected file to become the local file of the current [transfer job](#).

Click **Cancel** to escape.

See also:

[File Transfer Command in the Options Menu](#)

[Transfer Command in the File Menu](#)

Open Transfer List Dialog Window

This dialog window will appear when you click **Open transfer list** in the [Transfer dialog window](#). It provides the following dialog boxes:

File Name

Enter the name of an [FT file](#) which may contain the wildcards * and ?. You can also enter the required filename extension or select one of the extensions offered in the **List Files of Type** dialog box. Then click **Ok**, so that EasyCODE(C++) will either display the corresponding list of files or immediately open the [transfer list](#) if a definite filename has been selected.

Directories

Choose the directory in which the required FT file is stored.

List Files of Type

Choose one of the filename extensions provided here.

Drives

Choose the drive on which the required file is stored.

Click **Cancel** to escape anytime.

See also:

[File Transfer Command in the Options Menu](#)

[Transfer Command in the File Menu](#)

Save Transfer List Dialog Window

This dialog window will appear when you click **Save transfer list** in the [Transfer dialog window](#). It provides the following dialog boxes:

File Name

Enter the name of an [FT file](#) that may contain the wildcards * and ?. You can also enter the required filename extension or select one of the extensions provided in the **List Files of Type** dialog box. Then click **Ok**, so that EasyCODE(C++) will either display the corresponding list of files or immediately save the [transfer list](#), if a definite filename has been selected.

Directories

Choose the directory in which the required FT file is to be saved.

List Files of Type

Choose one of the available filename extensions:

Drives

Choose the drive on which you want to save the file.

Click **Cancel** to escape anytime.

See also:

[File Transfer Command in the Options Menu](#)

[Transfer Command in the File Menu](#)

Follow-up Processing Dialog Window

This dialog window will appear when you click **Follow-up processing** in the File transfer settings dialog window, the Add dialog window or the Modify dialog window.

Enter a command for sending and/or receiving a file to/from the local and/or the remote host both in case of a successful or failed file transfer.

In every follow-up processing command you may use symbols as often as you like.

Wildcard Syntax in the Local Filename

You may use the wildcards * and ? in the local filename.
These mean:

- * ...Any string of characters,
- ? ...No character or exactly one character.

Thus, you can select several local files, each of which will then create a transfer job.

Symbols in Transfer Jobs

The following symbols are available:

%LOCNAME%	<u>Local filename</u> (without path or extension),
%LOCEXT%	Extension of the local filename,
%LOCFILE%	Local filename without path
%LOCFULLNAME%	Full local filename
%LOCDRIVE%	Drive of local file
%LOCPATH%	Path of local file
%REMFIL%	<u>Remote filename</u> .

You may use %LOCNAME%, %LOCEXT% and %LOCFILE% in the remote filename as well as in the commands for follow-up processing, while %LOCFULLNAME%, %LOCDRIVE% and %LOCPATH% may be used only in the commands for local, %REMFIL% only in the commands for remote follow-up processing.

For reasons of compatibility, the symbols used in the previous version are still valid in this product version:

%1 ...	<u>Local filename</u> (without path or extension),
%2 ...	Extension of the local filename,
%3 ...	<u>Remote filename</u> .

You may use the symbols %1 and %2 in the remote filename and in the commands for follow-up processing, while the symbol %3 can be used in the commands for follow-up processing only.

When a transfer job is started, EasyCODE(C++) replaces the symbols with the names currently valid for the transfer job.

Transfer System

Choose the transfer system that is installed on your PC in the [File transfer settings dialog window](#).

EasyCODE(C++) supports the following transfer systems:

- FT-PCD (SIEMENS)
- DFT (INTRA-SYS)
- WFT-BAM (INTRA-SYS)
- WFT TCP/IP (INTRA-SYS)
- SIECON-WIN 975x (CO&KOM)
- SIAM-FT-DOS (COMPUTER-KONTOR)
- RDAC (CONWARE)
- LAN (Schneider&Koch)
- LAN (EXCELAN)
- FTOS (CCOM)

Processor Name

In case you have selected one of the transfer systems **WFT BAM (INTRA-SYS)** or **WFT TCP/IP (INTRA-SYS)**, enter the processor name. For all other file transfer systems, this field is dimmed.

The processor name is the name of the sending processor which has been entered into the network description book by the administrator (ask your administrator).

Local File

Enter a DOS filename into the **File Name** box of the Add dialog window or the Modify dialog window or click **Browse**. If you click **Browse**, the Local file dialog window will appear.

In the *Add dialog window* you can enter a filename using wildcard syntax to create several transfer jobs.

Remote File

Enter the filename of the remote file and, if necessary, the password of the remote file into the File transfer settings dialog window, the Add dialog window or the Modify dialog window.

In the filename of a remote file, you may

- Use any number of symbols, and, thus, define the name on the basis of the name of the local file.
- Define a pattern for **all** remote files (e.g. with pre- and suffix: abc%LOCFILE%xyz)

Remote Host

Enter the **host name** into the [File transfer settings dialog window](#), the [Add dialog window](#) or the [Modify dialog window](#).

Enter a **user ID** and, if necessary, an **account number** as your logon authorization for the remote host. If you have selected FT-PCD or SIAM as your transfer system, you may use a **user profile** instead of a user ID and account number as your logon authorization. In the BS2000, the product FTAC is available for this purpose.

Transfer Direction

Choose **Send** or **Receive** as your transfer direction in the File transfer settings dialog window, the Add dialog window or the Modify dialog window, depending on whether you want to transfer files from your PC to the remote host or vice versa.

File Type

Choose **Text** or **Binary** in the File transfer settings dialog window, the Add dialog window or the Modify dialog window, depending on the file type of the files to be transferred.

Source files are **text** files, SPX files **binary** files.

Overwrite Existing File

Check this option in the File transfer settings dialog window, the Add dialog window or the Modify dialog window, so that, in case a file with the same filename exists in the target computer, it will be replaced with the transferred file.

Compressed Transfer

Check this option in the [File transfer settings dialog window](#), the [Add dialog window](#) or the [Modify dialog window](#) so that strings of identical characters existing in the file to be transferred will be transferred more quickly.

Additional Processing

Both in the File transfer settings dialog window and the Transfer dialog window, you can enter into the **Additional processing** dialog element MSDOS commands for **Preprocessing** and **Postprocessing**, which are to be executed before or after the file transfer commands. The commands entered here are stored in the transfer list and executed at the start of the transfer.

Directory for FT Files

Enter a path for saving the transfer list into the **Directory for FT files** dialog box of the File transfer settings dialog window. The path you specify here will be entered into the "Open transfer list" and "Save transfer list" dialog windows and stored in the INI file using the **FtFileDirectory** entry. If you do not know the exact path name, you may click the "... " button to the right of the dialog box; the Default window for FT files dialog window will appear, in which you may specify the path.

Default Directory for FT Files

This dialog window will appear when you click the "..." button to the right of the one-line editing window *Directory for FT files* in the File transfer settings dialog windows. It provides the following dialog boxes:

Directories

Select here the required directory.

Drive

Select the drive on which the required directory is stored.

Click **Ok** if you want EasyCODE(C++) to accept the specified working directory for the *Directory for FT files* editing box in the File transfer settings dialog window.

Click **Cancel** to escape.

If you are using **Windows for Workgroups**, the dialog window will also provide the **Network...** button to create links within the network.

Representation Dialog Window

This dialog window will appear when you choose the Representation command in the Options menu. It provides the following dialog boxes:

if

Choose one of the options **horizontal** or **vertical** for the layout of the **if** in the structure diagram. Use the **vertical** layout form if your structure diagram has become too wide.

Colors

Choose in the drop-down list field the structure diagram element, for which you want to set a color. In the box to the right, the current color for this element is displayed. Select this box to get the Windows Dialog Window for Color Selection. Choose then any color from this field.

Display

Choose the **Segment numbers** option to display segment numbers in the structure diagram.

Choose the **Line numbers** option if you want line numbers to be displayed in the structure diagram.

Display each file in a separate window

Choose this option if EasyCODE shall display for explicit opening (**New** or **Open** commands in the **File** menu) or for implicit opening (Opening after program run) of a file this file in a separate instance. In this case, after the execution of a command both the file already open and the new file are displayed on the screen.

Display each segment in a separate window

Choose this option if EasyCODE shall represent for viewing an open file (explicitly with all view commands or implicitly by doubleclicking in the structure list or result file windows or **Edit-Refine**) the found area in a separate instance. With this, both the previously viewed and the newly viewed segments are displayed on the screen after executing the command.

For these two options, the following holds: If an action concerning this option (New, Open, View, Doubleclick) is called while the Ctrl key is pressed, a new EasyCODE window will be opened in any case. If, on the other hand, you keep the Shift key pressed, no new window will be opened.

Click **Standard** to set the options to their standard values.

Click **Ok** to accept the layout settings.

Click **Cancel** to escape.

Standard Values for Options

By clicking **Standard** the structure diagram elements are set as follows:

if

Standard value **quer**

Colors

Element	RGB value	Color
structure diagram background	(255, 255, 255)	white
construct lines	(0, 0, 0)	black
conditions frame	(0, 0, 192)	blue
nestings	(192, 192, 192)	light gray
line/construct numbers (incl. frame)	(0, 0, 0)	black
segment numbers (incl. frame)	(0, 0, 0)	black
fixed texts	(0, 0, 255)	blue
jump label	(128, 0, 128)	magenta
variable texts	(0, 0, 0)	black
texts of symbolic names	(0, 128, 0)	green
frame of the inline editor window	(192, 192, 192)	light gray
segment in the overview window	(255, 0, 0)	red
page number in page view	(0, 255, 0)	light green
critical constructs in page view	(255, 0, 0)	red
special lines (e.g. break points)	(255, 0, 0)	red

The other options of the dialog window are **off**.

Note

"Special lines" are designed for the use with an active add-in interface and are only available in this connection. The name of the "special lines" in the dialog window representation can only be set with an INI entry (e.g. "break point").

Screen Font Dialog Window

This dialog window will appear when you choose the Screen command in the Font submenu of the **Options** menu. It provides the following dialog boxes:

Fonts

Choose one of the fonts listed here.

Font style

This dialog box shows all styles available for the font selected in the **Fonts** dialog element. Select one of the font styles listed here.

Size

This dialog box shows all sizes available for the font selected in the **Fonts** dialog element. Select one of the font sizes listed here.

Sample

Displays a text sample showing the current settings of the selected screen font.

Click **Ok** to accept the settings. The structure diagram will then be displayed in the selected screen font. Click **Cancel** to keep the current font settings.

Printer Font Dialog Window

This dialog window will appear if you choose the Printer command in the Font submenu of the **Options** menu. It provides the following dialog boxes:

Fonts

Choose one of the fonts listed here.

Font style

This dialog window shows all styles available for the font selected in the **Fonts** dialog box. Select one of the font styles listed here.

Size

This dialog element shows all sizes available for the font selected in the **Fonts** dialog box. Select one of the font sizes listed here.

Sample

Displays a text sample showing the current settings of the selected screen font.

Click **Ok** to accept the settings. The structure diagram will be printed in the selected printer font.

Click **Cancel** to keep the current font settings.

Click **Test print** to get a test print of the current printer font.

Save Settings Dialog Window

This dialog window will appear when you choose the Save settings command in the Options menu. It provides the following dialog boxes:

Options

Check this box to save the settings you have selected with the help of the **Options** menu commands.

These settings include:

- the Repeat option
- the display of the Overview window
- the display of the Page Preview window
- the display of the Structure List window
- the display of the Result File window
- the display of the toolbox
- the display of the tool bar
- the display of the status bar
- the display of the page numbers in the status bar
- the layout format of the if
- the display of segment numbers
- the display of line numbers
- the selected screen font
- the selected printer font
- all options for program linking
- the editor settings
- the default settings for file transfer
- the setting Show each file in a separate window for New and Open
- the setting Show each segment in a separate window for View
- the source attributes

Directories/extensions

Source files

Check this box to save the current path for source files or select a different path by entering it into the one-line editing box. You can also select the path by clicking the "..." button to the right of the editing box and using the following dialog window.

The file extension is predefined with the EasyCODE default value but you can change it as you like.

ETF files

Check this box to save the current path for export files or select a different path by entering it into the one-line editing box. You can also select the path by clicking the "..." button to the right of the editing box and using the following dialog window.

The file extension is predefined with the EasyCODE default value but you can change it as you like.

Default for Open file

Check this box to save the current path for opening files or select a different path by choosing it in the list box. The paths shown in the list are those you defined in the above dialog elements.

Insert files

Check this box to save the current path for insert files or select a different path by entering it into the one-line editing box. You can also select the path by clicking the "..." button to the right of the editing box and using the following dialog window.

The file extension is predefined with the EasyCODE default value but you can change it as you like.

Editing windows

Check this box to save size and position of the multi-line EasyCODE editing window you opened last for the next EasyCODE(C++) session.

The Uppercase option are also stored in the INI file.

Click **Ok** if you want EasyCODE(C++) to save the selected settings.

Click **Cancel** to escape.

Source Attributes Dialog Window

This dialog window will appear when you choose the Source attributes command in the Options menu. It provides the following dialog boxes:

File names

Default extension

Choose the default filename extension to be used for EasyCODE(C++) source files. Select one of the filename extensions offered here or enter your own filename extension into the editing box.

Character set

Choose one of the options **OEM** or **ANSI**, according to which character set the source file that you want to open or save is coded or should be coded.

C programs for Windows should be ANSI coded, C programs for DOS or other systems should be OEM coded.

Save options

Line break coding

Choose one of the options **DOS** or **UNIX**, depending on the conventions with which the source file you want to create is to be coded. (For DOS: new line \r\n, for UNIX: new line \n).

Indent

Defines the indent of nested structures. The indicated value determines the number of columns nested structures are indented. The generated file will be arranged in a way that allows you to understand nested structures more easily.

Default setting: 3.

{ } indented

Choose this option if you want "{" and "}" also to be indented.

Default: indented.

Copy without EasyCODE comments

Choose this option if you want to create a copy of the source file without EasyCODE comments. The copy will have the name of the source file, but the filename extension CC (**C-Copy**). The following EasyCODE comments can, if desired, be converted into C++ comments in the CC file:

Last modification date

Choose this option if EasyCODE(C++) is to write the date of the saving of the source file as C++ comment at the beginning of the CCP file.

Short info

Choose this option if EasyCODE(C++) is to write the contents of the dialog element "short info" as C comment at the beginning of the CC file (after the change date, if existing). If no short information exists, no comment is created.

Segment headers

Choose this option if EasyCODE(C++) is to convert the contents of the segment headers into C comments. For empty segment headers no empty comments `/* */` are created.

Block comments

Choose this option if EasyCODE(C++) is to convert the contents of the block comments into C comments. For empty block comments, no empty comments `/* */` are created.

Click **Ok** to confirm the options set.

Click **Cancel** to discard your changes.

See also:

[Save As Command in the File Menu](#)

Program Settings Dialog Window

This dialog window will appear when you click **Add** or **Edit** in the [Programs dialog window](#). It provides the following dialog boxes:

Name

Name of the program as displayed in the **Execute** submenu of the **File** menu and in the program list of the Programs dialog window. If a character of the program name is to be underlined (hot key), you have to enter an "&" before this character.

Type

Specifies the type of the program to be executed. Select DOS for a DOS program or Windows for a Windows program.

Command line

Enter here the filename of the program to be executed (EXE, PIF, COM or BAT file) including possible program parameters, or click the "..." button to the right of the one-line editing box to select the appropriate settings in the following [Execute program dialog window](#).

Working directory

Enter here the directory you want to use as working directory for the program to be executed, or click the "..." button to the right of the one-line editing window to select the appropriate settings in the following [Working directory dialog window](#).

Save before Program Call

Choose this option if you want the structure diagram to be saved before the program is started.

Open after Program Call

Choose this option if the specified file has to be opened after the program is started.

Execute as symbol

Choose this option if the program is to be reduced to a symbol immediately after its start.

Message on termination

Choose this option if you want EasyCODE(C++) to display a message indicating that the program has been terminated.

Export before Program Call

Choose this option if you want the structure diagram to be exported as an [ETF file](#) before the program is started.

Redirect STDIN to

Choose this option if you want to redirect the standard input for the program to be executed. Enter the name of the file from which the input data are to be read into the edit control to the right.

Redirect STDOUT to

Choose this option if you want to redirect the standard output for the program to be executed. Enter the name of the file to which the output data is to be written into the edit control to the right.

Redirect STDERR to

Choose this option if you want to redirect the standard error output for the program to be executed. Enter the name of the file to which the error output data is to be written into the edit control to the right.

Redirection PIF file

Enter the name of the PIF file which will start EASY-RE.EXE, a program for redirecting STDIN, STDOUT and STDERR that is provided by EasyCODE. You can use the PIF file EASY-RE.PIF that is supplied with EasyCODE(C++) or a copy of this PIF file in which you have changed the default values. The redirection PIF file is required because DOS does not allow STDERR to be redirected using the command line. This redirection is especially important if you want to link EasyCODE(C++) to a compiler.

Result file

Enter the name of a file that is to be displayed in the Result file window. You can also use all symbols with the exception of %TEMP%.

Choose one of the options **OEM** or **ANSI**, according to which character set the result file is coded. Result files of Windows applications usually are ANSI coded, result files of DOS programs or other systems usually are OEM coded.

Display on termination

Choose this option if you want the Result file window to be opened automatically after the program has been terminated.

display only if non-empty

Choose this option if you have chosen the button above, but only non-empty result files are to be displayed.

Click **Ok** to confirm the selected options.

Click **Cancel** to escape.

See also:

[Symbols for Program Linking](#)

[Program Linking Command in the Options Menu](#)

[Result File Command in the Options Menu](#)

[Execute Command in the File Menu](#)

[Programs Dialog Window](#)

Execute Program Dialog Window

This dialog window will appear when you click the "..." button to the right of the one-line editing box *Command line* in the *Program settings* dialog window. It provides the following dialog boxes:

File Name

Enter here the name of a program. You can also enter the required filename extension or select "Programs" in the **List Files of Type** dialog box. Then click **Ok** to get a list of files with this extension displayed. Choose a file of this list.

Directories

Choose here the directory in which the required program is stored.

List Files of Type

Here you find the predefined entry **Programs**.

Drives

Choose the drive on which the required program is stored.

Click **Ok** so that EasyCODE(C++) will display the selected program in the *Command line* editing box. Click **Cancel** to escape.

Working Directory Dialog Window

This dialog window will appear when you click the button to the right of the one-line editing box *Working directory* in the *Program settings* dialog window. It provides the following dialog boxes:

Directories

Choose here the required directory.

Drives

Choose the drive on which the required directory is stored.

Click **Ok** so that EasyCODE(C++) will display the selected working directory in the *Working directory* edit control.

Click **Cancel** to escape.

Editors Dialog Window

This dialog window will appear when you choose the **Editors** command in the **Options** menu. In this dialog window you may define

- the standard editor for the **Edit text** command in the **Edit** menu
- an editor for the **External editor** command in the **Edit** menu
- an editor for correcting errors in source/ ETF files
- an editor for displaying the list of errors (ERR file).

The editor defined as **Standard** is used for mouse commands and hot keys calling an editor. The default setting is directed editing in the structure diagram.

Enter the name of an editor including path into the corresponding edit control or click the icon to the right of the line to select an editor in a separate dialog window.

The following applies to the **External editor** edit control:

The %TEMPFILE% symbol must, the %LINE% and %COLUMN% symbols may be entered into the command line.

They mean: %TEMPFILE% Filename of a temporary file
%LINE% Line number in selected text
%COLUMN% Column number in selected text

external Editor as default editor

Choose this checkbox to make the external editor the default editor. In this case the shortcuts for the commands **Edit text** and **External editor** are changed.

The following symbols may be used in the edit controls **Editor to be used for correction of invalid sources or ETF files** and **Editor to be used for displaying error files**:

The %SOURCEFILE% and %ERRFILE% symbols must, the %LINE% symbol may be entered into the command line.

%SOURCEFILE% For the name of the analyzed file
%ERRFILE% For the name of the error list
%LINE% For the line number of the first error

Click **Ok** so that EasyCODE(C++) will accept the settings.
Click **Cancel** to escape.

Notes

External editor: When you call the external editor, the selected text will be stored in a temporary file. The name of this temporary file is passed on to the editor by means of the %TEMPFILE% symbol.

If your editor allows line and column specification, you may specify the current values (mouse click) by means of the %LINE% and %COLUMN% symbols. If line or column specification is not possible, the value 1 will be used.

File analyzing:

If you enter the name of an editor in the second edit control and answer **Yes** to the question "Call editor for correcting errors?", EasyCODE(C++) will open the file in the specified editor.

If you enter the name of an editor in the third edit control and answer **Yes** to the question "Display error list?", EasyCODE(C++) will open the file in the specified editor.

If you enter the name of an editor in the second and third edit controls and answer **Yes** to the question "Display error list and call editor for correcting errors?", EasyCODE(C++) will open both files in the specified editors.

See also:

[Open](#)

[Insert File](#)

[Open and Insert ETF files](#)

Editor Dialog Window

This dialog window will appear when you click the "..." button to the right of the one-line editing boxes *External editor*, *Editor to be used for correction of invalid sources or ETF files* or *Editor to be used for displaying error files* in the *External editor* dialog window. The following dialog boxes are provided:

File Name

Enter here the name of an editor. You can also enter the required filename extension or select "Programs" in the **List Files of Type** dialog box. Then click **Ok** to get a list of files with this extension displayed. Choose a file of this list.

Directories

Choose here the directory in which the required editor is stored.

List Files of Type

Here you find the predefined entry **Programs**.

Drives

Choose the drive on which the required editor is stored.

Click **Ok** so that EasyCODE(C++) will display the selected editor in the edit control.
Click **Cancel** to escape.

Click **Ok** to have EasyCODE(C++) take over the editor chosen into the edit field.
Click **Cancel** to cancel this.

Standard Directory for Open Dialog Window

This dialog window will appear when you click the "..." button to the right of the one-line editing box *Open file* in the *Save settings* dialog window. The following dialog boxes are provided:

Directories

Choose here the required directory.

Drives

Choose the drive on which the required directory is stored.

Click **Ok** so that EasyCODE(C++) will save the specified directory.
Click **Cancel** to escape.

Standard Directory for Insert Dialog Window

This dialog window will appear when you click the "..." button to the right of the one-line editing box *Insert file* in the *Save settings* dialog window. The following dialog boxes are provided:

Directories

Choose here the required directory.

Drives

Choose the drive on which the required directory is stored.

Click **Ok** so that EasyCODE(C++) will save the specified directory.
Click **Cancel** to escape.

Standard Directory for Export Dialog Window

This dialog window will appear when you click the "." button to the right of the one-line editing box *Export file* in the *Save settings* dialog window. The following dialog boxes are provided:

Directories

Choose here the required directory.

Drives

Choose the drive on which the required directory is stored.

Click **Ok** so that EasyCODE(C++) will save the specified directory.
Click **Cancel** to escape.

Symbols for Program Linking

In **all** edit controls (with the exception of *Name*) of the Program settings dialog window, you may use the following symbols:

%DIR%	Directory in which the current file is stored
%DRIVE%	Drive on which the current file is stored
%EXT%	Filename extension of the current file
%FILE%	Full pathname of the current file
%FILEPATH%	Drive and directory of the file you opened last using the File command in the Insert menu
%NAME%	Filename of the current file (without drive, pathname or filename extension)
%OPENPATH%	Drive and directory of the current file
%EXPDIR%	Directory in which the exported file is stored
%EXPDRIVE%	Drive on which the exported file is stored
%EXPEXT%	Filename extension of the exported file
%EXPFILE%	Full pathname of the exported file
%EXPNAME%	Filename of the exported file (without drive, pathname or filename extension)
%EXPPATH%	Drive and directory of the exported file
%PROMPT:<text>%	<u>Prompting</u> for program data
%PROMPT<n>:<text>%	<u>Prompting</u> for program data; n is a number to save the promptstring in the also numbered variable %PROMPTVAL<n>%. (n= 1...9).
%PROMPTVAL%	Value given in %PROMPT:<text>%
%PROMPTVAL<n>%	Value given in %PROMPT<n>:<text>% (n= 1...9).

Instead of a combination of fixed text and the symbols mentioned above, you may also use the following symbols in the *STDIN* and *STDOUT* edit controls:

%TEMP%	Temporary file created by EasyCODE(C++) and stored in the TEMP directory.
--------	---

Instead of a combination of fixed text and the symbols mentioned above, you may also use the following symbols in the *Result file* entry box:

%STDOUT%	File specified in the <i>STDOUT</i> edit control
%STDERR%	File specified in the <i>STDERR</i> edit control

Note

Depending on whether you enter the symbols in uppercase or lowercase letters, the text to be replaced will be modified. If the symbol is written in lowercase letters only, the text will be changed to lowercase letters; and if the symbol is written in uppercase letters only, the text will be changed to uppercase letters. If the symbol is written in a combination of uppercase and lowercase letters, the text will not be modified.

Prompting for Program linking

In **all** edit fields (except Name) of the Program Settings Dialog window the symbolic name **%PROMPT:<text>%** can be used, to make the entries in the fields not before the program is executed. In that case a dialog window will be opened. The text you write after the symbolic name will appear in the window. The length of the valid text is only limited by the maximum length of the input field in the dialog window. During the program run, however, text that is too long for the following dialog window is cut on the right.

If you want to use the Prompting function more than once the symbolic names can be numbered ("%PROMPT1:.....% up to %PROMPT9:.....%"). The values given in "%PROMPT:<text>%" resp. "PROMPT<n>:<text>%" are stored in the variable %PROMPTVAL% (resp. %PROMPTVAL1% up to %PROMPTVAL9%) and may be used as symbolic names in the edit fields of the "Program settings" dialog window.

See also:

[Symbols for program linking](#)

[Program linking command in the options menu](#)

[Result file command in the options menu](#)

[Execute command in the files men](#)

[Programs dialog window](#)

Prompting Dialog Window

This dialog window will appear when the symbolic name %PROMPT:<text>% resp. %PROMPT<n>:<text>% is used in one or more edit fields of the [Program Settings Dialog Window](#). Every usage of the symbolic name causes an own dialog window.

The values given in "%PROMPT:<text>%" resp. "PROMPT<n>:<text>%" are stored in the variable %PROMPTVAL% (resp. %PROMPTVAL1% up to %PROMPTVAL9%) and may be used as symbolic names in the edit fields of the "Program settings" dialog window.

Enter the required data and click the **Ok** button to resume the program.

Click the **Cancel** button to truncate the program.

See also:

[Prompting for Program Linking](#)

[Symbols for Program linking](#)

[Program linking command in the options menu](#)

[Result file command in the options menu](#)

[Execute command in the files menu](#)

[Programs dialog window](#)

Programs Dialog Window

This dialog window will appear when you choose the Program linking command in the Options menu. It provides the following dialog boxes:

Program list

Displays the programs currently configured in alphabetic order.

Add

Click this button to add a new program to the program list. The Program settings dialog window will appear. The button is dimmed when the maximum number of programs (32) has been entered.

Edit

Select an entry in the program list and click the **Edit** button to change the program settings of a program already configured. The Program settings dialog window will appear.

Remove

Select one or more entries in the program list and click the **Remove** button to remove the selected program(s) from the program list.

Click **Ok** to accept the changes.

Click **Cancel** to escape.

Note

The changes apply only to the current session. If you want to save the program linking options for further sessions, use the **Save settings** command in the **Options** menu.

See also:

Program Linking Command in the Options Menu

Execute Command in the File Menu

Use of the Result File Window

Project Dialog Window

This dialog window will appear if you have chosen the Project command in the Options menu. It contains the following dialog elements:

File name

Enter the file name of the project you want to open. You can also enter the desired file name extension or choose one of the extensions offered in the dialog element File format. Then click **Ok** to view in the list field the files with this extension. Choose a name from this list.

Directories

Choose the directory in which the file is stored.

File format

Choose one of the file name extensions offered:

Drive

Choose the drive where the project you want to open is stored.

Click **Ok** to open the file chosen. You can also open this file by doubleclicking the file name in the **File name** list with the left mousebutton.

Click **Cancel** to cancel this process.

See also:

[View Definition](#)

[View Reference](#)

Color Dialog Window

This dialog window will appear if you want to change the color for an area and have chosen the color option in the Representation Dialog Window. It contains the following dialog elements:

Basic colors

If you want to use one of the primary colors, choose it by mouseclick.

Custom colors

If you want to use one of the colors you have designed yourself, also choose it by mouseclick.

Choose **Define Custom colors** to mix a color. For this, the dialog window is enlarged to the right, where you can mix the colors. Choose then **Add to Custom color** to add this color to the dialog element **Custom colors**.

Click **Ok** to add the chosen color to the Dialog Window Representation.

Click **Cancel** to cancel this process.

About EasyCODE(C++)

This dialog window will appear when you choose the About EasyCODE(C++) command in the Help menu. It provides information about name and version number of the product, the name of the manufacturer, a copyright note and information about the memory available.

Click **Ok** to close this dialog window.

View Dialog Window

This dialog window will appear when you choose the Line command in the View menu.

Enter a number between 1 and 65535 into the Line number dialog box.

After the viewing the corresponding construct is selected automatically and offered for editing - with correct cursor positioning - if this is possible for the construct type.

Click **Ok** to scroll to the corresponding line.

Click **Cancel** to escape.

Definition Dialog Window

This dialog window will appear if you have chosen the Definition command in the View menu and more than one fitting definition has been found.

In this window all required definitions are listed.

Select the desired definition and click **Ok** to view this position.

Click **Cancel** if you want to cancel this.

Reference Dialog Window

This dialog window will appear if you have chosen the Reference command in the View menu and more than one fitting references have been found.

In this window, all references found are listed.

Select the desired reference and click **Ok** to view this reference.

Click **Cancel** to cancel this.

ProVista Definition Dialog Window

This dialog window will appear if you have a symbol that is not unique while viewing a **Reference**. In this case, the ProVista Browser must be informed of the respective definition.

Select the symbol and click **Ok** to view this reference.

Click **Cancel** to cancel this.

Search/Replace Dialog Window

This dialog window will appear when you choose the Search/Replace command in the **Edit** menu. You will find the following dialog boxes:

Search

Enter the text you want to find. If you have selected an area directly before calling **Search/Replace** the selected text is offered automatically in the edit field. If no text has been selected and Search/Replace has already been used in this EasyCODE session, the text used last is offered.

Match upper/lowercase

Select this option, when upper-/lowercase is important in the search text.

Replace with

Select this option when the search text should be replaced with the text entered in this field.

Capitalize

Select this option, when the text entered in the replacement field should be capitalized automatically.

Area

Select one of the following areas

Current segment,

Selected area,

Entire structure diagram,

From the selected area to the end of the current segment or

From the selected area to the end of the entire structure diagram

as your range for the **Search/Replace** process.

Default setting: **Entire structure diagram**.

Including lower segments

Select this option, when the **Search/Replace** process should be executed also in the lower segments of the selected search range.

Click the **Ok** button to begin the **Search/Replace** process with the specified options.

Click the **Cancel** button to escape.

Each time the search text is encountered, the Search dialog window or the Replace dialog window will appear, in which you decide about how to proceed.

Replace Dialog Window

This dialog window will appear each time the search process encounters the search text which you want to replace with the text entered in the replacement field of the Search/Replace window.

Select the **Replace** button to replace the search text with the replacement text.

Select the **Do not replace** button to leave the original text and to continue the search.

Select the **Replace all** button to replace the search text with the replacement text automatically from this position in the structure diagram.

Select the **Cancel** button to quit the **Search/Replace** process. Changes made up to this point will remain.

Search Dialog Window

This dialog window will appear each time the search process encounters the search text which you entered in the search field of the Search/Replace window without selecting the '**Replace with**' box.

Select the **Continue** button to continue the search.

Select the **Cancel** button to quit the search process.

Open and Insert Source Files

You can generate a [source file](#) with EasyCODE(C++), with EasyCODE(C) or with a text editor. When you open or insert a source file that has not been generated with EasyCODE(C++), it will automatically be structured into [segments](#), so that each function will become a segment of its own.

The source file should contain a complete C/C++ program that complies with the structure conventions. If EasyCODE(C++) detects structural errors during file analysis, an [ERR file](#) containing warnings and error messages will be created. If the analysis produces only warnings, the source file will be displayed as a structure diagram. If the file contains errors, the structure diagram cannot be displayed. In this case EasyCODE(C++) will ask you if you want to open the source file in order to correct the errors or if you want to see the ERR file, depending on the settings specified in the [Editors dialog window](#).

Note: You can use the "Keyword<n>" INI entry to define keywords. The keywords defined in this entry will be displayed as statement constructs, allowing for language extensions which might be interpreted incorrectly to be displayed in correct form (e.g. Structured Exception Handling, MFC extensions, ...). See also the CUSTOM.DOC file.

See also:

[Open](#)

[Save](#)

[Insert File](#)

[Insert Clipboard](#)

Open and Insert ETF Files (Import)

An ETF file may be created or edited with EasyCODE(C++) or with a text editor.

The ETF file should contain a complete structure diagram description complying with the EasyCODE structure conventions (for the description of the ETF Format please refer to the ETF.DOC file).

If EasyCODE(C++) detects structural errors during the file importing process, an ERR file containing warnings and error messages will be created. If the importing process produces only warnings, the ETF file will be displayed as a structure diagram. If the file contains errors, the structure diagram cannot be displayed. In this case EasyCODE(C++) will ask you if you want to open the ETF file in order to correct the errors or if you want to see the ERR file, depending on the settings specified in the Editors dialog window.

See also:

Open

Save

Insert File

Insert Clipboard

Open and Insert SPX Files

EasyCODE(C++) can open SPX files or insert SPX files directly or from the Clipboard. Elements are automatically converted from EasyCODE(SPX) format into EasyCODE(C++) format according to predefined conversion rules.

Conversion Rules from EasyCODE(SPX) to EasyCODE(C++) Structure Diagrams

Structure diagrams which have been created with the language-independent editor *EasyCODE(SPX)* and are stored as SPX files, will be converted to C++ structure diagrams according to the following rules:

EasyCODE(SPX) Construct	EasyCODE(C++) Construct
Statement	Statement
IF-THEN-ELSE	if
SWITCH	nested ifs
WHEN	then branches of the if constructs
WHEN NONE	innermost else branch
CASE	switch
OF branch	case branch with break at the end
OTHERWISE	default with break at the end
FOR	for
WHILE	while
REPEAT	do while with negated condition as expression
LOOP	for with two semicolons as for expression
EXIT	if with break in the then branch
Function	Function
Procedure	Function
Procedure Call	Statement
Block	Block
Frame	Block
AND, OR, NOT operators	Expression

Object Linking and Embedding of EasyCODE(C++) Structure Diagrams (OLE)

Object Embedding

Embedding allows you to paste an EasyCODE(C++) structure diagram into the file of another application program. By embedding, a copy of the source file is transferred to the target file. In order to modify the structure diagram in the source file, you only have to doubleclick the structure diagram in the target file to start EasyCODE(C++) and have the structure diagram displayed on the worksheet. After having finished your work, you can update the structure diagram in the target file by using the **Update** command in the **File** menu. However, if you modify the structure diagram directly in the target file, the source file remains unchanged.

Object Linking

When you link an EasyCODE(C++) structure diagram to the file of another application program, you create a reference from this file to the structure diagram file. In order to modify the structure diagram in the source file, you only have to doubleclick the structure diagram in the target file to start EasyCODE(C++) and have the structure diagram displayed on the worksheet. When you modify a structure diagram linked to another application, you modify the source file itself at the same time.

You can link several target files to one source file. In this case you only have to modify the source file to have all target documents changed.

Since object linking means nothing but creating a reference to a structure diagram file, you can create links only with files that have been saved.

There are two categories of application programs that support object linking and embedding. Application programs that can supply objects to be embedded into or linked to other documents are called *server applications*. Applications that can accept embedded or linked objects are called *client applications*. EasyCODE(C++) can supply structure diagrams to be used in other applications, therefore it functions as a **server application**.

For more detailed information concerning object linking and embedding, please refer to the Microsoft Windows User Manual or the manual of the client application concerned.

See also:

[Example: Linking a Structure Diagram to a Word-for-Windows File](#)

[Example: Embedding a Structure Diagram into a Word-for-Windows File](#)

Linking a Structure Diagram to a Word-for-Windows File

To link an existing structure diagram to a Word for Windows file:

1. Possibility

1. Start EasyCODE(C++) and open the required structure diagram.
2. Select the entire segment or undo existing selection if you want to link the entire segment including the header.
3. Choose the **Clipboard - Copy** command in the **Edit** menu.
The segment is copied into the Clipboard.
4. Switch to Word for Windows and open the file to which you want to link the structure diagram.
5. Place the cursor at the position where you want the upper left corner of the structure diagram to appear.
6. Choose the **Paste Special** command in the **Edit** menu.
7. In the dialog window that will appear you select either **Graphics** or **Bitmap**, then click the **Insert link** button.

2. Possibility

1. Start Word for Windows.
2. Open or create the target document.
3. Choose the **Object** command in the **Insert** menu.
4. Choose in the **Object** dialog window in the **Generate from file** register menu the EasyCODE(~EASYKOMP~) file whose first segment you want to link, mark the control box **link with file** and then click **Ok**.

Note: The second possibility can only be used if you want to link a complete file. If you want to link only selected areas, use the first possibility.

Embedding a Structure Diagram into a Word-for-Windows File

To create and embed a structure diagram:

1. Start Word for Windows.
2. Open or create the target document.
3. Choose the **Object** command in the **Insert** menu.
4. Choose EasyCODE(C++) in the **Object** dialog window.
5. Create the structure diagram in EasyCODE(C++).
6. Choose the **Exit and return to Microsoft Word** command in the **File** menu of EasyCODE(C++).

To embed an existing structure diagram:

1. Possibility

1. Start EasyCODE(C++) and open the required structure diagram.
2. Select the structure diagram or the required element.
3. Choose the **Clipboard - Copy** command in the **Edit** menu.
The selected area is copied into the Clipboard.
4. Switch to Word for Windows and open the file in which the structure diagram is to be embedded.
5. Place the cursor at the position where you want to embed the structure diagram.
6. Choose the **Paste Special** command in the **Edit** menu.
7. A dialog window will appear, in which you select **EasyCODE(C++) Object** and then click the **Insert** button.

2. Possibility

1. Start Word for Windows.
2. Open or create the target document.
3. Choose the **Object** command in the **Insert** menu.
4. Choose in the **Object** dialog window in the **Generate from file** register menu the EasyCODE("EASYKOMP") file whose first segment you want to link, mark the control box **link with file** and then click **Ok**.

Glossary

Application Window

Backup Copy CCP File

Clicking

Clipboard Construct Cursor

Doubleclicking

Drag EASY-CPP.INI

EasyCODE(C++) Session

ERR File

ETF File

File

FT File

Hierarchical Level

Insertion Point

Line Numbers

Mouse Pointer

Move

Overview Window

Result File

Segment

Segment Window

Segment Number

Segment Header

Selection SPX File

Sequence

Source Document Source File

Statement

Step STDNEU.CPP

Structure Diagram Area

Structure Diagram

Structure Diagram Element

Structure List

Symbolic Name

Target Document

Tool Bar

Toolbox

Transfer Job

Transfer List

Segment

Area on the top level or on a lower hierarchical level. A segment on a lower hierarchical level is displayed on the current hierarchical level as a gray box with a header and an optional hierarchical segment number. A segment is created automatically by opening a source file that has not been created with EasyCODE(C++), by refining a statement, or by pushing down an area.

Statement

Construct which may contain any C/C++ statements in the form of text.

Structure Diagram File

Any structure diagram which is available as a source or SPX file or as an ETF file.

EASY-CPP.INI

A text file stored in the Windows installation directory and structured like the WIN.INI file, which contains settings important for executing EasyCODE(C++).

Source File

A text file containing a C/C++ source program. You use the **Source attributes** command in the **Options** menu to specify the default extension.

Source Document

A document which makes data, graphic representations or other information (e.g. structure diagrams) available to other documents by way of object linking. If you change any information in the source document, you can update it in the linked document, the target document.

Target Document

A document containing data, graphic representations or other information (e.g. structure diagrams) coming from another document. With the help of object linking, you can change any information contained in the target document by changing it in the source.

CC File

Copy of an EasyCODE(C) file where the EasyCODE comments have been removed or some EasyCODE information - selected by you - has been converted to C comment.

CCP File

Copy of an EasyCODE(C++) file in which the EasyCODE comments have been removed or in which some EasyCODE information - selected by you - has been converted to C++ comments.

ETF File

File used for storing structure diagrams in a standardized text format. The default extension is ETF, which can be changed by the user.

Result File

File in which the results (e.g. an error list) of a program you started using the **Execute** command in the **File** menu are stored.

ERR File

A file containing warnings and error messages. An ERR file is created, when EasyCODE(C++) detects structural errors when opening or inserting a source or an ETF file. The extension is ERR.

SPX File

File in an internal format defined by EasyCODE(SPX) to store a structure diagram. Default extension is SPX, it can, however, be changed by the user.

Insertion Point

Element before, after or instead of which an area may be inserted or where an area may be copied or moved to.

Construct

Rectangular box for designing a structure diagram. Please refer to the **Insert** Menu for supported constructs.

Hierarchical Level

Segments on the same level. The hierarchical level of a segment can be determined by the number of points in its segment number.

Selection

Inverted representation of an area.

Symbolic Name

A reference enclosed in angular brackets in a structure diagram in which text is to be inserted.

Structure Diagram

Graphic representation of a C/C++ program.

A structure diagram is created by arranging constructs sequentially or nested, and by filling them with text. The text and the graphic representation describe the function of the construct.

Structure Diagram Area

A sequence of structure diagram elements which can be selected.

Structure Diagram Element

Part of a structure diagram which can be selected by a single mouse click.

Elements can be:

- Constructs,
- Symbolic names,
- Text,
- Empty boxes and
- Segments.

Sequence

A line-up of all elements that have the same width.

Tool Bar

The tool bar provides push buttons which you may click with the mouse to execute commands.

Toolbox

The toolbox provides push buttons which you may click with the mouse to insert constructs and files.

Line Number

Line numbers are created when you save a source file. The line numbers assigned to the constructs correspond to the line numbers in the source file. The same line numbers remain assigned to the constructs when you edit the structure diagram. EasyCODE(C++) updates the line numbers only when you save the structure diagram. If you create a copy without EasyCODE comments while saving a source file, the line numbers in the structure diagram will refer to the source file, not to the copy.

Segment Number

Hierarchical reference number for a segment in decimal notation.

Cursor

Blinking vertical line to indicate the position where entered characters will be inserted.

Drag

Moving the mouse while pressing a button.

Move

Moving the mouse without pressing a button.

Mouse Pointer

Symbol indicating the mouse position on the screen.

Click

Pressing a mouse button and releasing it immediately.

DoubleClick

Pressing and releasing a mouse button twice within a short interval.

STDNEW.CPP

Any CPP file. It is automatically placed on the worksheet as a structure diagram when EasyCODE(C++) is started, or when the **New** command in the **File** menu is chosen. EasyCODE(C++) searches for the STDNEW.CPP file first in the directory selected for inserting files and then in the installation directory.

Backup Copy

Second to last version of a CPP file. The backup copy has the same name as the CPP file, but the extension ~ followed by the first two letters of the original extension.

Application Window

Window in which EasyCODE(C++) is running. It shows the menu bar and contains at least one segment window.

Segment Window

Window in which a segment of the structure diagram is displayed.

Overview Window

Window in which an overview of the active segment window is displayed.

Segment Window at Maximum Size

Segment window which fills the entire application window.

Application Window at Maximum Size

Application window which fills the entire screen.

Segment Window Icon

The application window and each segment window can be minimized to a symbol.

Active Segment Window

A segment window is considered active when all inputs by the user are directed to this window. The active window can be recognized by a title bar and a window frame displayed in a different color.

EasyCODE(C++) Session

Time between starting and quitting your work with EasyCODE(C++).

Step

Executing a command which changes the state of a structure diagram.

Clipboard

A temporary storage area in Windows with the help of which data can be transferred from one application program to another. The clipboard contents remain stored until you delete them, overwrite them or exit Windows. For further information please refer to the Microsoft Windows User Manual.

Segment Header

Text for the symbolic name <header> in a segment. It appears in the segment itself above the structure diagram and in the title bar, in the next higher hierarchical level in a gray box, in the **Window** menu and in the Structure list, and below the minimized segment or application window's icon.

Structure List

Window displaying a list containing the segments, classes or functions of the structure diagram.

Transfer Job

Instruction to send or retrieve a file to or from a remote host. A transfer job is represented in the transfer list by the transfer direction and the name of the DOS file.

Transfer List

List of transfer jobs.

FT File

File in an internal format defined by EasyCODE(C++) for storing a transfer list. Default extension is FT, it can, however, be changed by the user.

EasyCODE Comment

Special comment in a C/C++ file containing information about the structure diagram which exceeds C/C++ structures. The most important EasyCODE comments for reconstructing the structure diagram from a source file concerns information about segment structure.

Keyboard/Mouse Commands

Key Functions

Mouse Functions

Key Functions

Copying Text

Deleting Text

Doubleclicking

ENTER Key

ESC Key

Executing Programs

Getting Help

Hot Keys

Moving the Cursor

Replacing Text

Scrolling the Application Window

Scrolling the Structure Diagram

Selecting Constructs

Selecting Text

Setting an Insertion Point

Mouse Functions

[Context-Sensitive Pop-Up Menus](#)

[Doubleclicking](#)

[Drag&Drop](#)

[Dragging Icons](#)

[Getting Help](#)

[Moving the Cursor](#)

[Selecting an Area](#)

[Selecting an Element](#)

[Selecting Several Elements](#)

[Selecting Text](#)

[Setting an Insertion Point](#)

Context-Sensitive Pop-Up Menus

Depending on the position of the mouse pointer and the selected area, a pop-up menu containing the commands currently available from the **Insert**, **View** and **Options** menus will be opened when you press the right mouse button.

Note: You can also open this menu by pressing the Shift+F10 keys.

Click the right mouse button to

status bar, tool bar, Overview window, Page preview window, Structure list window, result file, toolbox

worksheet of an application window

edit mode

to open

menu containing commands from the **Options** menu for switching on/off the bars or opening/closing the windows.

menu containing commands from the **Edit** and **View** menus which may be applied to the current selection.

menu with the commands of the edit symbol bar

The context menu can also be called by pressing Shift+F10.

Doubleclicking

Doubleclick the left mouse button to speed up some frequently used actions.

Doubleclick	to
EasyCODE(C++) file in the MS-DOS window or the Windows File Manager	start EasyCODE(C++) + open file
list item in a dialog window	choose list item and select default button
segment in the <u>structure list window</u>	view this segment
list item in the programs list	select this list item and choose the Edit button in the Programs dialog window (Program Linking command in the Options menu)
line number in the Result file window	view and edit the corresponding element
list element in the Print window	view the selected segment
construct in Page preview window	view and select this construct
list item in transfer list	modify the transfer job
function in the Structure list window	view and select this function
gray segment box	<u>Down</u> command in the View menu
empty space on the worksheet	<u>Up</u> command in the View menu
CTRL+doubleclick empty space of the worksheet	<u>TOP</u> command in the View menu

Note:

The shortcuts for **Edit text** and **External editor** are changed if you made the external editor the default editor. (Checkbox "external editor as default editor" in the **Editors** command of the **Options** menu).

Doubleclick Functions during the Editing

Doubleclicking a selected construct invokes the edit form you have set as standard in the Options Editors dialog window.

If you keep the Shift key pressed at the same time, this means direct editing in the structure diagram. If you keep the Ctrl key pressed during the doubleclicking, the edit window is called. Alt key + doubleclick invokes the external editor. The setting of the standard editor is not taken into consideration while pressing these keys.

See also:

Other Key Functions

Hot Keys

With the help of hot keys, you can speed up the selection of most commands. The hot keys corresponding to a command are listed to the right of the command name in the EasyCODE menus.

The following **hot keys** are available:

<u>Press key (combination)</u>	<u>to choose command</u>	<u>in menu</u>
CTRL+N	<u>New</u>	<u>File</u>
CTRL+O	<u>Open...</u>	<u>File</u>
CTRL+S	<u>Save</u> or <u>Update</u>	<u>File</u>
CTRL+A	<u>Save as...</u>	<u>File</u>
CTRL+P	<u>Print</u>	<u>File</u>
CTRL+X	<u>Export as</u>	<u>File</u>
ALT+BKSP	<u>Undo</u>	<u>Edit</u>
CTRL+T	<u>Edit text...</u>	<u>Edit</u>
ALT+CTRL+T	<u>External editor...</u>	<u>Edit</u>
CTRL+H	<u>Search/Replace...</u>	<u>Edit</u>
CTRL+C	<u>Copy</u>	<u>Edit</u>
CTRL+L	<u>Duplicate</u>	<u>Edit</u>
CTRL+M	<u>Move</u>	<u>Edit</u>
DEL	<u>Delete</u>	<u>Edit</u>
SHIFT+DEL	<u>Cut</u>	<u>Edit - Clipboard</u>
CTRL+INS	<u>Copy</u>	<u>Edit - Clipboard</u>
SHIFT+INS	<u>Paste</u>	<u>Edit - Clipboard</u>
CTRL+F	<u>Refine</u>	<u>Edit</u>
CTRL+D	<u>Push down</u>	<u>Edit</u>
CTRL+U	<u>Pull up</u>	<u>Edit</u>
Alt+PGDN	<u>Down</u>	<u>View</u>
Alt+PGUP	<u>Up</u>	<u>View</u>
Alt+HOME	<u>Top</u>	<u>View</u>
Alt+#	<u>Line...</u>	<u>View</u>
Alt+!	<u>Definition</u>	<u>View</u>
Alt+*	<u>Reference</u>	<u>View</u>
CTRL+R	<u>Repeat</u>	<u>Options</u>

CTRL+V	<u>Overview</u>	<u>Options</u>
CTRL+I	<u>Structure list</u>	<u>Options</u>
CTRL+A	<u>Page preview</u>	<u>Options</u>
CTRL+E	<u>Result file</u>	<u>Options</u>

Notes:

When you use these keys, you do not need to select a menu. If you have selected a menu and then decide to use hot keys, press the ALT key to cancel the menu first. Then press the hot keys.

The shortcuts for **Edit text** and **External editor** are changed if you made the external editor the default editor. (Checkbox "external editor as default editor" in the **Editors** command of the **Options** menu).

See also:

[Other Key Functions](#)

Getting Help

Press

the F1 key

the left or right mouse button + F1

to get context-sensitive help

on the active window

depending on shape and position of the mouse cursor

See also:

[Other Key Functions](#)

Moving the Cursor with Keys

Press the key(s)	<u>to move the cursor</u>
LEFT ARROW	left one character
RIGHT ARROW	right one character
UP ARROW	up one line
DOWN ARROW	down one line
TAB LEFT	to the previous entry box
TAB RIGHT	to the next entry box
PGUP	to the previous page
PGDN	to the next page
ENTER	to a new line having the same indent
CTRL + ENTER	to the beginning of a new line
HOME	to the beginning of a line
END	to the end of a line
CTRL+RIGHT ARROW	to the next word
CTRL+LEFT ARROW	to the previous word
CTRL+HOME	to the beginning of the text
CTRL+END	to the end of the text
CTRL+TAB	to the next TAB position in the <u>multi-line editing window</u> and insert blanks

See also:
[Other Key Functions](#)

Scrolling the Structure Diagram with Keys

With the following keys, you can scroll a segment in a segment window provided the Scroll Lock function key is activated.

<u>Press the key(s)</u>	<u>to scroll the segment</u>
LEFT ARROW	one column to the left
RIGHT ARROW	one column to the right
UP ARROW	up one line
DOWN ARROW	down one line
PGUP	up one screen
PGDN	down one screen
CTRL+PGUP	left one screen
CTRL+PGDN	right one screen
HOME	to the top of the segment window
END	to the bottom of the segment window
CTRL+HOME	to the left end of the segment window
CTRL+END	to the right end of the segment window

You can also use the scroll bar thumb for scrolling.

See also:

[Other Key Functions](#)

Executing Programs

If you have linked programs to EasyCODE(C++) using the **Program linking** command in the **Options** menu, you can start them not only via the menu, but also using the following key combinations:

<u>Press the keys</u>	<u>to start the program</u>
CTRL + 1..0	number 1 to 10
CTRL + Shift + 1..0	number 11 to 20
CTRL + Alt + 1..0	number 21 to 30
CTRL + Shift + Alt + 1..0	number 31 to 40

Notes:

The names of the programs are listed alphabetically.

Although the key combinations shown above allow 40 entries, it is not possible to use more than 32, since the maximum number of programs that can be linked to EasyCODE(C++) is restricted to 32.

Setting an Insertion Point with Keys

Before you insert a construct, you have to select the element (see [Selecting Elements](#) and [Move Selection](#)) before or after which the construct is to be inserted.

<u>Press the key(s)</u>	<u>to insert a construct</u>
ENTER	after the last selected element or instead of the selected element
SHIFT+ENTER	before the last selected element or instead of the selected element

Note: If the insertion point is restricted (i.e. if a construct can be inserted only before or only after a selected element), the insertion attempt is rejected; you remain, however, in the insertion mode so that you can correct the insertion direction (before < - > after).
If the insertion point is not valid, the insertion attempt is also rejected, you remain in the insertion mode even after moving the selection.

See also:
[Other Key Functions](#)

Scrolling the Application Window with Keys

Prerequisite: SCROLL LOCK function key must be activated.

<u>Press the key(s)</u>	<u>to scroll the application window</u>
SHIFT+LEFT ARROW	one column to the left
SHIFT+RIGHT ARROW	one column to the right
SHIFT+UP ARROW	up one line
SHIFT+DOWN ARROW	down one line
SHIFT+PGUP	up one screen
SHIFT+PGDN	down one screen
SHIFT+CTRL+PGUP	left one screen
SHIFT+CTRL+PGDN	right one screen
SHIFT+HOME	to the top
SHIFT+END	to the bottom
SHIFT+CTRL+HOME	to the left end
SHIFT+CTRL+END	to the right end

See also:
[Other Key Functions](#)

Selecting Constructs with Keys

[Extending/Reducing the Selection](#)

[Moving the Selection](#)

[Selecting Elements](#)

Note: The SCROLL-LOCK function must be deactivated.

See also:

[Other Key Functions](#)

Selecting Elements

Prerequisite: The SCROLL-LOCK function must be deactivated.

Press the key(s)	to select
HOME	the first construct within the <u>sequence</u> of an existing selection
END	the last construct within the <u>sequence</u> of an existing selection
CTRL+HOME	the first construct in the segment of the top <u>sequence</u>
CTRL+END	the last construct in the segment of the top <u>sequence</u>
Ctrl+right arrow	the first construct within an existing selection which can syntactically contain a <u>sequence</u>
Ctrl+Shift+right arrow	the next <u>sequence</u> inside an existing selection
Ctrl+Shift+left arrow	whole <u>sequence</u> in which the construct enclosing the selection lies
Ctrl+Shift+up arrow	the sequence predecesing the <u>sequence</u> in which the selection lies
Ctrl+Shift+arrow down	the sequence following the <u>sequence</u> in which the selection lies
SPACE BAR	a visible element which is situated as near to the top as possible

Note: If the selection cannot be made, existing selections remain unchanged.

See also:

[Selecting Functions](#)

[Other Key Functions](#)

Moving the Selection

Prerequisite: The SCROLL-LOCK function must be deactivated.

<u>Press the key(s)</u>	to move the selection
UP ARROW	to the previous construct within the <u>sequence</u> or to the header
DOWN ARROW	to the following construct within the <u>sequence</u>
LEFT ARROW	"up" one element
RIGHT ARROW	"down" one element
TAB	to the next element that can be edited
SHIFT+TAB	to the previous element that can be edited
CTRL+LEFT ARROW	to the enclosing construct
CTRL+RIGHT ARROW	to the first construct of the first branch in the first selected construct
CTRL+DOWN ARROW	to the first construct of the next branch in the enclosing construct
CTRL+UP ARROW	to the last construct of the previous branch in the enclosing construct

Note: If the selection cannot be moved, existing selections remain unchanged.

See also:

[Selecting Functions](#)

[Other Key Functions](#)

Extending/Reducing the Selection

Prerequisite: The SCROLL-LOCK function must be deactivated.

Press the key(s)	to
SHIFT+UP ARROW	extend the selection by one element at the top or reduce the selection by one element from the bottom within the <u>sequence</u> - depending on the direction of the existing selection
SHIFT+DOWN ARROW	to extend the selection by one element at the bottom or reduce the selection by one element from the top within the <u>sequence</u> - depending on the direction of the existing selection
SHIFT+HOME	to extend the selection from the beginning of the existing selection to the first construct within the <u>sequence</u>
SHIFT+END	to extend the selection from the beginning of the existing selection to the last construct within the <u>sequence</u>
CTRL+SHIFT+HOME	to extend the selection to the topmost enclosing construct within the segment and to the beginning of the segment
CTRL+SHIFT+END	to extend the selection to the topmost enclosing construct within the segment and to the end of the segment

Note: If the selection cannot be extended, existing selections remain unchanged.

See also:

[Selecting Functions](#)

[Other Key Functions](#)

ESC Key

Press the ESC key when

the menu is pulled down

you select an insertion point

an area is selected

a mouse key is pressed in the Overview
window

to

close the menu

escape

undo the selection

keep the visible segment

Enter Key

Pressing the return key at a selected area invokes the standard edit mode.

The additional keys Shift, Ctrl and Alt in addition to the return key cause - no matter which editor is the current standard editor - the following:

SHIFT+ENTER	editing
CTRL+ENTER	edit window
ALT+ENTER	external editor

Press the ENTER key

after selecting an editable element of a structure diagram

after selecting a number in the Result file window

after selecting a gray segment box

after selecting a segment in the Structure list window

to

choose the **Edit text** command in the **Edit** menu

have the corresponding source line displayed in the structure diagram +
choose the **Edit text** command in the **Edit** menu

choose the **Down** command in the **View** menu

have this segment displayed in the active segment window

Selecting Text with Keys

Whether a selection is extended or reduced, depends on the direction in which an existing selection has been made.

<u>Press the keys</u>	<u>to extend or reduce text selection</u>
SHIFT+LEFT ARROW	by one character to the left
SHIFT+RIGHT ARROW	by one character to the right
SHIFT+UP ARROW	up one line
SHIFT+DOWN ARROW	down one line
SHIFT+HOME	to the beginning of the line
SHIFT+END	to the end of the line
SHIFT+PGUP	up one screen
SHIFT+PGDN	down one screen
SHIFT+CTRL+LEFT ARROW	to the previous word
SHIFT+CTRL+RIGHT ARROW	to the next word
SHIFT+CTRL+HOME	to the beginning of the text
SHIFT+CTRL+END	to the end of the text

See also:

[Selecting Text with the Mouse](#)

[Other Key Functions](#)

Copying Text to Clipboard with Keys

Press the keys

CTRL+INS

to copy

selected text into the Windows Clipboard

See also:

Other Key Functions

Deleting Text

<u>Press the key(s)</u>	<u>to delete</u>
DEL	the character to the right of the <u>cursor</u> or selected text
CTRL+DEL	text from the cursor to the end of the line
BACKSPACE	the character to the left of the cursor
SHIFT+DEL	<u>selected</u> text and place it into the Windows <u>Clipboard</u>

See also:
[Other Key Functions](#)

Replacing Text

Press the key(s)

SHIFT+INS

any text

See also:

Other Key Functions

to replace

selected text with the Windows Clipboard contents or to insert the Clipboard contents at cursor position

selected text

Dragging Icons

Drag

a file from the File Manager to active
EasyCODE(C++)

a file from the File Manager to active
Print Manager

to

open this file

print this file

Index

General:

[How to Use...](#)

[How to Start EasyCODE\(C++\)](#)

[New Features Compared With EasyCODE Version 5.1](#)

Methods:

[Procedures](#)

[Menus](#)

[Tools](#)

References:

[Glossary](#)

[Keyboard/Mouse Commands](#)

[Messages](#)

[Troubleshooting](#)

[Additional Documentation](#)

How to Start EasyCODE(C++)

When you install EasyCODE(C++) using the SETUP program, EasyCODE(C++) asks you if you want to include the application into a Program Manager group, from which you can start EasyCODE(SD) by doubleclicking the program icon. You can modify the command line for starting EasyCODE(C++) according to your requirements by adding starting options as follows:

easy-cpp.exe [option] [filename]

option: /p	Print		
/ps={0 1 2 3}	Print Structure		
/o=<pfad><filter>	OpenFileDialog		
/ro	read only		
/l=<line_number>	Line		
/prj	Project data base	/s=<filename>	Save
/e=<filename>	Export		
/i=<filename>	Infile		
/Initialize	Initialize		

The following options are available:

Print

Use this option to print the specified file. The main application window will not be opened, only the window for canceling the printing process will appear. After printing, the application will be closed. The default print settings will be used. You may, however, also add an entry to the INI file so that the "Print" dialog window will be opened before printing and you can modify the printing settings.

Syntax: {-|/}P[rint]
Example: EASY-CPP.EXE /P ABC.CPP

Print Structure

With this option the contents of the structure list window are printed on the selected printer with the selected print font. The number denotes the structure type in the sequence in which it can be chosen in the structure list window (starting with 0 for segments). See also [Print structure list](#).

Syntax: {-|/}P[rint]S[tructure][={0|1|2|3}]
Example: EASY-CPP.EXE /PrintStructure=0

OpenFileDialog

With this option the Open file dialog appears immediately after starting the application. If existing, the path specified in the option and the filter for initializing the dialog window (directory, filter) is used.

Syntax: {-|/}O[penFileDialog][=<path>][<filter>]
Example: EASY-CPP.EXE /o=c:\work*.clp

Project

With this option the project data base specified in the command line is opened. The project data base entered in the INI file is not used for this.

Syntax: {-|/}project/prj
Example: EASY-CPP.EXE /prj ccpreloc.pvx

ReadOnly

With this option the file specified in the command line is opened in read-only mode. This option is only useful for network installation, as single-user installation does not support the read-only mode for opening files.

Syntax: {-|/}R[ead]O[nly]
Example: EASY-CPP.EXE /ro abc.spx

Line

Use this option if you want EasyCODE(C++) to try and find the corresponding line number in the specified file after it has been opened. If the line number does not exist, EasyCODE(C++) will display the proceeding line number. If you start the application without specifying a filename or if you specify a file not containing line numbers, this option will have no effect.

Syntax: {-|/}L[ine]=<line_number>
Example: EASY-CPP.EXE /L=200 ABC.CPP

Save

Use this option to save the specified file. If no path is specified the file will be loaded from the working directory. You can also enter a name for the file to be saved. If no path is specified the file will be stored in the current working directory. (e.g. /s=xyz.spx). The main window of the application will not appear. The application terminates after the saving of the files.

Syntax: {-|/}S[ave]=<filename>
Example: EASY-CPP.EXE /S=ABC.CPP

Export

Use this option to export the specified file. If no path is specified the file will be loaded from the working directory. You can also enter a name for the file to be exported. If no path is specified the file will be stored in the current working directory. (e.g. /e=xyz.etf).
The application terminates after the saving of the files.

Syntax: {-|/}E[xport]=<filename>
Example: EASY-CPP.EXE /E=ABC.CPP

Inifile

Use this option if you do not want to use the default INI file(EASY-CPP.INI in the Windows directory), but the file specified in the option as your INI file. This allows for example different users to use different INI files. This applies to all operations which change the INI file or read from it (e.g. initialization at the start of a program, saving the settings etc.).

Syntax: {-|/}I[nifile]=<filename>
Example: EASY-CPP.EXE /I=CPP-NEW.INI

Initialize

Use this option if you want EasyCODE(C++) to redesign all entries within the WIN.INI file and in the registration database and overwrite existing entries. The EasyCODE(C++) application window will not be opened, and the application will be closed immediately afterwards.

Syntax: {-|/}Initialize
Example: EASY-CPP.EXE /INITIALIZE

List of symbols: { | } alternatives, [] optional entries

Tools

EasyCODE(C++) provides some utilities which you may start by choosing the **Execute** command in the **File** menu via program linking.

<u>CPP2SPX</u>	to convert an EasyCODE(C++) structure diagram into an EasyCODE(SPX) structure diagram.
<u>CPPREDOC</u>	to remove C++ statements from a C++ structure diagram, while retaining the structure for documentation purposes.
<u>EASY-RE</u>	to redirect STDIN, STDOUT and STERR into a file during program linking.
<u>WINNOP</u>	to display any text file in the Result file window.

CPP2SPX

Function

This DOS tool helps you to convert an EasyCODE(C++) structure diagram into an EasyCODE(SPX) structure diagram via ETF file format.

Command Line for Program Call

```
[installationpath]CPP2SPX.EXE <inputfile> <outputfile> [/l=<int>] [/c]
```

<inputfile> input file; ETF file created by EasyCODE(C++).
<outputfile> output file; ETF file which can be read by EasyCODE(SPX).
/l=<int> Level=<0-99>, segment level up to which the structure diagram is to be converted.
/c only comments are to be transferred from statements.

Error Messages

Syntax errors in the program call will be written to STDERR.

Conversion errors will be written to STDOUT.

Example of Program Linking

The following example demonstrates how to integrate the CPP2SPX tool into EasyCODE(C++) to convert an EasyCODE(C++) source file into an EasyCODE(SPX) structure diagram by choosing the **CPP2SPX** subcommand in the **Execute** command of the **File** menu.

Make the following entries into the **Program settings** dialog window (**Options** menu - **Program linking** command - **Add** button):

Name: CPP2SPX
Type: DOS
Command line: C:\EASY\CPP2SPX.EXE %expfile% %exppath%%expname%.REV
 The filename extension of the output file is REV ("Reverse Engineering"), you may, however, change it according to your requirements.

"Execute as symbol" option on

"Message on termination" option on

"Export before Program Call" option on

CPPREDOC

Function

This DOS tool helps you to create from an EasyCODE(C++) source file a file from which all C++ statements will be removed and which will contain only the structure and C++ comments, so that you can easily display the program structure for documentation purposes.

Command Line for Program Call

```
[installationpath]CPPREDOC.EXE <inputfile> <outputfile>
```

<inputfile> input file; ETF file created by EasyCODE(C++).
<outputfile> output file; ETF file that can be read by EasyCODE(C++).

Error Messages

Syntax errors in the program call will be written to STDERR.

Conversion errors will be written to STDOUT.

Example of Program Linking

The following example demonstrates how to integrate the CPPREDOC tool into EasyCODE(C++) to convert a source file by choosing the **CPPREDOC** subcommand in the **Execute** command of the **File** menu.

Make the following entries into the **Program settings** dialog window (**Options** menu - **Program linking** command - **Add** button):

Name: CPPREDOC
Type: DOS
Command line: C:\EASY\CPPREDOC.EXE %expfile% %exppath%%expname%.RDC
 The filename extension of the output file is RDC ("Retrieve Documentation"), you may, however, change it according to your requirements.

"Execute as symbol" option on

"Message on termination" option on

"Export before Program Call" option on

EASY-RE

Function

This DOS tool helps you to redirect the standard assignments STDIN, STDOUT and STDERR to a file during program linking (**Execute** command in the **File** menu).

Command Line for Program Call

[installationpath]EASY-RE.EXE

The program will be called automatically by EasyCODE.

Error Messages

None.

Example of Program Linking

The following example demonstrates how to integrate the EASY-RE.EXE tool into EasyCODE to redirect STDOUT.

Make the following entries into the **Program settings** dialog window (**Options** menu - **Program linking** command):

"Redirect STDOUT to" option: %TEMP%

Redirection PIF file: EASY-RE.PIF

Result file: %STDOUT%

WINNOP

Function

This Windows tool helps you to display any text file in the Result File window, so that you can easily open a source file or an ETF file directly in EasyCODE.

Command Line for Program Call

[installationpath]WINNOP.EXE

Error Messages

None.

Example of Program Linking

The following example demonstrates how to integrate the WINNOP tool into EasyCODE(C++) to display a source file in the Result file window by choosing the **Show source** subcommand in the **Execute** command of the **File** menu.

Make the following entries into the **Program settings** dialog window (**Options** menu - **Program linking** command - **Add** button):

Name: Show Source
Option button : "Windows"
Command line: C:\EASY\WINNOP.EXE
"Execute as symbol" option on
"Save before Program Call" option on
Result file: %FILE%
Option button : "OEM"
"Display on termination" option on

Additional Documentation

If you want to use EasyCODE for special purposes, there are various documentations available in the form of DOC files. You may use any word-processing program which is able to process DOC files to view or print these documentation files, e.g. Word for Windows.

The following files are supplied with EasyCODE

CUSTOM.DOC

This file describes EasyCODE configuration procedures which cannot be carried out with the help of the **Options** menu.

They include:

- additional entries into the INI file, which you may use to change EasyCODE default settings (e.g. of dialog messages),
- entries into SETUP.INF, which you may use to accelerate the installation process (especially the installation of network licences),
- command line options, which you may use to activate certain actions (e.g. printing) when EasyCODE is started, but especially to assign other files than the default INI file to an application.

EASYPARS.DOC

This file describes the interfaces between EasyCODE and a parser DLL used for reengineering source files. In compliance with these interfaces you can write parser DLLs for programming languages, for which at present there is no parser DLL available.

DEBUG.DOC

This file describes the debugger interfaces of EasyCODE. EasyCODE is able to communicate with a debugger using special unequivocal messages, which are registered by EasyCODE and the debugger via RegisterWindowMessage(). The file DEBUG.DOC is not automatically part of the EasyCODE package, but can be ordered from your EasyCODE vendor.

ETF.DOC

This file describes the composition of a structure diagram file stored in the ETF format. ETF files are created with the **Export** command in the **File** menu.

You need to know the ETF format, if you want to

- redesign an ETF file exported by EasyCODE manually or with other tools,
- write a structure diagram generator for other products, so that you can edit these structure diagrams with EasyCODE,
- write converter programs to convert structure diagrams from one EasyCODE component into another according to the instructions specified by you.

SETUP.DOC

This file describes the possible ways of configuring the EasyCODE setup. The information contained in this file is to be used only in special cases (e.g. for automatic software distribution). We recommend that you do not modify the default configuration. In particular copy right notes must by no means be modified or removed.

New Features Compared With EasyCODE(C++) V5.1

Direct Editing in the Structure Diagram

Structure Diagram Representation in User-Defined Colors

Better Search/Replace Function

Continue Search Function

Split Command

Join Command

Help on Selected Text

Add-In Interface

MDI Simulation for New, Open and View

Drag & Drop

Paper-Saving Printing

Print Structure List

EasyCODE(C++) Programmers Tool Kit

Messages from EasyCODE(C++)

All messages that can occur during an EasyCODE(C++) session appear in a dialog window that is characterized by a specific symbol and contains push buttons.

The following types of messages may appear:



Click one of the icons displayed above to get a list of possible messages.

To most EasyCODE(C++) message windows you can call the specific help function by pressing the function key F1. Messages which are not listed here come from another active application, e.g. from Windows or from the Print Manager. Please refer to the relevant user manual for details. The title bar of the message window tells you which program is responsible for the message.

When a source file or an ETF file is opened or inserted, there can be messages concerning their analysis. These messages are stored in an ERR file and can be read with an editor.



These dialog messages inform about the result of an action.

EasyCODE comment replaced <n> times.

"/*" or "*/" in block comments removed <n> times.

Procedure could not be found.

Program <program> terminated.

Search text not found.

Search text was replaced <n> times.

Other Messages



These dialog messages inform about the unsuccessful result of an action or remind you of the consequences before executing a command.

Analysis of <file>: <n> errors and <m> warnings.

Analysis of <file>: <n> errors and <m> warnings. Error list in <error file>.

Analysis of <file>: <n> errors and <m> warnings. Unable to create error file <file>.ERR. Data in <supplementary information> does not have EasyCODE(C++/SPX) format.

Default printer not found.

DLL <DLL> cannot be used by more than one application at the same time.

Dynamic Link Library EASY-FTL.DLL has a wrong version. File transfer not available.

Dynamic Link Library <name>.dll not found or incorrect.

Error during registration of OLE server

Error in file transfer.

Error in network. Please quit EasyCODE(C++) and start again.

Error when opening file <file>.

Error while executing program <program>. Unable to create temporary file.

Error while exporting <file>. Unable to display error file, because a program creating a result file is running. Error while printing - out of disk space.

Error while printing - out of memory.

Error while printing - printing aborted.

Error while printing (unidentifiable).

Error while writing to <INI file>. Extension ERR not allowed.

External editor not configured and cannot be used as default editor.

File <file> has FT format <version>. This format is not supported.

File <file> does not have FT format.

File <file> in use.

File <file> in use. File was opened with write-protection.

File <file> in use - unable to open file.

File <file> in use - unable to write to file. File <file> is not a ProVista project file.

File <file> is write-protected.

File <file> lost to other user after opening/saving. File remains open with write protection.

File <file> not found.

File <file> opened with write-protection. Please save under a different name.

File <file> will be converted from EasyCODE(SPX) into EasyCODE(C++) format.

File <file> will be converted from EasyCODE(SPX) into EasyCODE(C++) format. File is not protected from access by other users.

File <file> will be converted from ETF into EasyCODE(C++) format.

File <file> will be converted from ETF into EasyCODE(C++) format. File will not be protected from access by other users.

File <file> will be converted from source into EasyCODE(C++) format.

File <file> will be converted from source into EasyCODE(C++) format. File will not be protected from access by other users.

Filename for redirecting STDIN/STDOUT/STDERR is missing.

Filename for redirection program is missing.

Invalid prompt in program settings:<prompt>.

Internal stack overflow. Invalid name for PIF file.

Maximum number of users reached. EasyCODE(C++) will not be started.

Maximum text length for Result file window exceeded.

No definition for <selected text> found

No line numbers assigned.

No printer installed. No references for <selected text> configured
No transfer system selected. Unable to start file transfer.

Only one file can be edited. The first selected file will be opened.

Out of disk space (drive <drive>). Job cancelled.

Out of memory. Job cancelled.

Program name is not clear.

Replacement string <replacement string> missing.

Resulting text too long - unable to replace text.

Short info is too long and will be cut off during generation.

Structure diagram in <supplementary information> contains a structural error.

Structure diagram in <supplementary information> has EasyCODE(C++/SPX) format <version>. This format is not supported.

Structure diagram <segment information> exceeds worksheet area.

Structure diagram <segment information> exceeds worksheet area when printing.

This application is not a network version! Usage in the network may cause problems.

Unable to activate other file transfer, as long as execution file <execution file> exists.

Unable to create file. Unable to execute FTD.EXE.

Unable to execute program <program>.

Unable to execute WFT.EXE.

Unable to load Help Manager.

Unable to load printer information.

Unable to print file.

Unable to save file.

Unable to write to <INI file>.

Other Messages



These dialog messages require a basic decision before executing a command.

Analysis of <file>: <n> errors and <m> warnings. Call editor for correcting errors?

Analysis of <file>: <n> errors and <m> warnings. Display error list?

Analysis of <file>: <n> errors and <m> warnings. Display error list and call editor for correcting errors? Delete transfer job(s) from transfer list?

Error in file transfer. Continue with next transfer job? File <file> does not exist. Create file? File

<file> has been changed by another application. Overwrite anyway?

File <file> has been changed. Discard current changes and reload file?

File <file> has been changed. Reload file?

File <file> in use. Open file with write-protection?

File <file> must be opened with write-protection either in EasyCODE or in the editor. Write-protection in EasyCODE?

<Generate/export/save> changes?

Line <line number> not found. View preceding line?

Remove program(s) from program list?

Save changes to <file>?

Save Clipboard contents?

Save settings?

Save transfer list?

The command you have chosen will break off the link between the open embedded object and <program>. Update embedded object before proceeding?

Unable to create backup copy <file>. Save without backup copy?

Update object embedded in <program> before closing?

Other Messages



These dialog messages indicate that a system error has to be corrected before executing a command.

At the moment, no error messages of this kind are documented.

Other Messages



EasyCODE comment replaced <n> times.

You tried to save a structure diagram containing an EasyCODE(C++) comment in a construct.

EasyCODE(C++) replaced the text not allowed in a structure diagram by a replace text <n> times (blank replaced by underline).
<n> times (blank replaced by underline).



Search text not found.

The text you wanted to search for with the Search/Replace command was not found.



Search text was replaced <n> times.

The text you wanted to replace was replaced as often as indicated.



'/*' or '*/' in block comments, headers and short info removed <n> times.

You saved a structure diagram containing the symbols '/*' or '*/' in block comments, headers or in the short info.

EasyCODE(C++) has removed these comment symbols not allowed in headers, block comments or the short info <n> times.



Program <program> terminated.

The program started with the Execute command in the File menu has been terminated.

This message will appear only if you have chosen the "Message on termination" option in the Program settings dialog window for the program in question.



Procedure not found.

You tried to view a procedure with the View Procedure command. EasyCODE(C++) was not able to locate this procedure.

Repeat the command with the correct procedure name.



Resulting text too long - unable to replace text.

You tried to use the Replace All command resulting in the text becoming too long. The current maximum text length is approximately 30,000 characters.

To prevent this, EasyCODE(C++) will cancel the **Replace all** command and allows you to replace only one text at a time.

The text items before the text that is too long were correctly replaced.

For further unrestricted use of the **Replace all** function, you should use the editing option *Split at cursor* to divide the text(if it is a statement) .

See also:

Undo



File <file> lost to other user after opening/saving. File remains open with write protection.

When you open or save a file, you lose control of the file (lock) for a moment. If during this time another application takes control of the file, your application can no longer lock it. You have read-only access.

Try again to open or save the file, or save the file under a different filename.



**File <file> has FT format <version>.
This format is not supported.**

You tried to open an FT file that was created with an incompatible version of EasyCODE(C++).

Every EasyCODE(C++) version can open FT files created with the previous version of EasyCODE(C++). Older versions than the previous one are not always supported. You can update your FT files by editing them with the intermediate versions of EasyCODE(C++).

private:

File <file> does not have FT format.

You tried to open an FT file. EasyCODE(C++) identified (partial) damage to the file.

Create the FT file again.

private:

File <file> in use.

You tried to save a file which is currently used by another user.

Save the file under a different name.

private:

File <file> in use - unable to write to file.

You tried to save a file which is currently read or written to by another application.

As soon as the other application has finished reading or writing, you can save the file.

private:

File <file> in use - unable to open file.

You tried to open a file which is currently read or written to by another application.

As soon as the other application has finished reading or writing, you can open the file.

private:

File <file> in use. File is write-protected.

You tried to open a file which is currently used by another application.

If you want to change the file, you must save the file under a different name.

private:

File <file> is not a ProVista project.

You tried to define a file which cannot be identified as a ProVista project file as this.

Check if the filename really denotes a ProVista project file.

private:

File <file> is write-protected.

You tried to open a file which is either explicitly write-protected or currently used by another user.

If you want to change the file, you must save the file under a different name.

private:

Unable to save file <file>.

You tried to save a file. The internal TMP file necessary for this action could not be found or is damaged.

Use the backup copy, modify the file again, if necessary, then save the file.

You tried to save a structure diagram containing a structural error.

This is an internal structural error which should never occur. Please contact your EasyCODE vendor.

private:

Unable to create file <file>.

You tried to save a structure diagram under a filename that is not accepted by DOS.

Please remember that

- a filename may contain only certain special characters,
- a filename must not contain more than 8 characters,
- a filename extension must not contain more than 3 characters,
- the specified path must exist,
- you must have write access to the specified path.

You tried to save a structure diagram on a write-protected network drive.

Save the structure diagram on a different drive, or have the network administrator change the access authorization (if possible).

You tried to save a write-protected file under the same filename.

Save the structure diagram under a different filename, or remove the write-protection.

private:

File <file> not found.

You tried to open a file which EasyCODE(C++) could not find in the specified directory or on the disk.

Make sure that

- the file is stored in the selected directory,
- the correct path was entered.

private:

File <file> will be converted from ETF into EasyCODE(C++) format.

You tried to open or insert an ETF file.

EasyCODE(C++) has to convert the ETF file into a CPP file.

Click **Ok** if you want the file to be converted automatically.

Click **Cancel** if you do not want the file to be converted.

private:

File <file> will be converted from ETF into EasyCODE(C++) format. File will not be protected from access by other users.

You tried to open or insert an ETF file.

EasyCODE(C++) has to convert the ETF file into a CPP file. After conversion, the file will not be protected from access by other network users.

Click **Ok** if you want the file to be converted automatically.

Click **Cancel** if you do not want the file to be converted.

private:

File <file> will be converted from source into EasyCODE(C++) format.

You tried to open or insert a source file.

EasyCODE(C++) has to convert the source file into a CPP file.

Click **Ok** if you want the file to be converted automatically.

Click **Cancel** if you do not want the file to be converted.

private:

File <file> will be converted from source into EasyCODE(C++) format. File will not be protected from access by other users.

You tried to open or insert a source file.

EasyCODE(C++) has to convert the source file into a CPP file. After conversion, the file will not be protected from access by other network users.

Click **Ok** if you want the file to be converted automatically.

Click **Cancel** if you do not want the file to be converted.

private:

File <file> was opened with write-protection. Please save under a different name.

You tried to save a file opened with write-protection using the same name.

Save the file under a different name.

private:

Filename for redirecting STDIN/STDOUT/STDERR is missing.

You selected one of the options Redirect STDIN to, Redirect STDOUT to or Redirect STDERR to for a DOS program, but you did not enter a filename into the corresponding dialog box.

Enter a filename into the corresponding edit control. You can also use symbols for the redirection.

private:

Name for Redirection PIF file is missing.

You selected one of the options Redirect STDIN to, Redirect STDOUT to or Redirect STDERR to for a DOS program, but you did not enter a filename into the Redirection PIF file dialog box.

Enter the name of the PIF file which will start the redirection program. You can use the PIF file provided on the setup disk or a copy of this file modified according to your requirements.

private:

Data in <supplementary information> does not have EasyCODE(C++/SPX) format.

You tried to open a file, insert a file or the Clipboard contents or edit an embedded object, but the data cannot be identified as areas from an EasyCODE(C++) or EasyCODE(SPX) structure diagram.

Use only data that have been created with EasyCODE(C++) or EasyCODE(SPX) .

You tried to view a lower hierarchical level in order to edit a segment or you tried to save or update a structure diagram, but EasyCODE(C++) could not identify the internal TMP file as valid.

Open the CPP file again to retrieve the status of the structure diagram before you last saved it and modify it again, if necessary.

private:

Analysis of <file>: <n> errors and <m> warnings.

You tried to open or insert an ETF file or a source file, in which EasyCODE(C++) detected structural errors.

During file analysis, EasyCODE(C++) detected errors which are not displayed in an error list.

private:

Analysis of <file>: <n> errors and <m> warnings. Unable to create error file <file>.ERR.

You tried to open or insert an ETF file or a source file, in which EasyCODE(C++) detected structural errors.

EasyCODE(C++) was unable to create the error file.

Possible reasons:

- existing ERR file is write-protected
- existing ERR file is currently being used
- lack of memory
- etc.

private:

Analysis of <file>: <n> errors and <m> warnings. Error list in <error file>.

You tried to open or insert an ETF file or a source file, in which EasyCODE(C++) detected structural errors.

During file analysis, EasyCODE(C++) detected errors which are listed in the file <error file>.

private:

Short Info exceeds permitted length and will be truncated when generated.

The text you entered into the "Short info" box including the line breads added by EasyCODE(C++) exceeds the maximum number of characters for short infos.

Click **Ok** if you want EasyCODE to continue the generation.

private:

Dynamic Link Library <parser_dll> cannot be used by more than one application at the same time.

You started an action requiring a parser DLL that is currently used by another application or another EasyCODE level, or an error occurred in the DLL.

Make sure that the parser DLL is used by only one application at a time. If the error still occurs, quit EasyCODE(C++) and Windows and start again.

private:

This application is not a network version! Usage in the network may cause problems.

You tried to start EasyCODE(C++) in a network, although the version you are using is not a network version or setup errors have been identified. EasyCODE(C++) will not run as a network version.

As to the problems that may arise in this connection, please contact your EasyCODE vendor.

private:

Unable to load printer information.

You started an action requiring some information from the printer driver (Page preview, Page numbers, Preview button in the "Print" window, inserting an area that has been put into the Clipboard using the Copy or Cut commands,), but EasyCODE(C++) could not load the printer driver.

Make sure that

- the printer driver is installed properly,
- the Print Manager can be loaded (line *spooler=yes* in the WIN.INI file),
- there is enough memory for the temporary Windows files. For details see the Microsoft Windows User Manual.

private:

Dynamic Link Library <name>.DLL not found or incorrect

You tried to start EasyCODE(C++), but the EASY-INI.DLL file required by EasyCODE(C++) could not be found or was incorrect.

The EASY-INI.DLL file is stored on the Setup disk. Install EasyCODE(C++) again or copy the EASY-INI.DLL file to the directory that contains EasyCODE(C++) with the help of the Windows "EXPAND" utility (e.g. EXPAND A:\EASY-INI.DL_ C:\EASY).

You tried to start EasyCODE(C++), but the EASY-FTL.DLL file required by EasyCODE(C++) could not be found or was incorrect.

The EASY-FTL.DLL file is stored on the Setup disk. Install EasyCODE(C++) again or copy the EASY-FTL.DLL file to the directory that contains EasyCODE(C++) with the help of the Windows "EXPAND" utility (e.g. EXPAND A:\EASY-FTL.DL_ C:\EASY).

You tried to import an ETF file, but EasyCODE(C++) could not find the EASYIMP.DLL file required for this action

Make sure that the required Dynamic Link Library from the setup disk is installed in a directory on the path, e.g. in the EasyCODE(C++) installation directory, if possible.

Returncodes:

- 1 DLL does not fulfill the EasyCODE requirements, i.e. the found DLL has an invalid version or is not an EasyCODE DLL.
- 0 Too little memory available or DLL damaged.
- 2 DLL not found.
- 3 Directory not found.
- 5 DLL does not have DLL but EXE format or a network share conflict occurred.
- 6 DLL with multiple data segments (not possible in Windows 3.1x).
- 8 Too little memory available.
- 10 DLL for invalid Windows version.
- 11 Invalid DLL format (either not a Windows application or format error).
- 12 DLL for another operating system.
- 13 DLL especially for DOS 4.0.
- 14 Unknown EXE/DLL format.
- 15 Real Mode DLL was loaded in Protected Mode.
- 16 Second instance of a DLLs with multiple data segments.
- 19 Compressed file - must be decompressed first.
- 20 DLL invalid or further DLLs required by the DLL could not be loaded.
- 21 32-bit-DLL.

private:

Dynamic Link Library EASY-FTL.DLL has a wrong version. File transfer not available.

You tried to start EasyCODE(C++), but EasyCODE(C++) found that the version of the current Dynamic Link Library EASY-FTL.DLL required for file transfer is not compatible with EasyCODE(C++).

The EASY-FTL.DLL file is stored on the Setup disk. Install EasyCODE(C++) again or copy the EASY-FTL.DLL file to the directory containing EasyCODE(C++) with the help of the Windows utility "EXPAND" (e.g. EXPAND A:\EASY-FTL.DL_ C:\EASY).

private:

Replacement string <replacement string> missing.

You tried to assign an external editor, an editor for correcting the source, or an editor for displaying the error list to EasyCODE(C++) using the Editors command in the Options menu without entering the symbol %TEMPFILE% or %SOURCEFILE% and %ERRFILE%.

Enter %TEMPFILE% or %SOURCEFILE% and %ERRFILE% into the corresponding edit controls of the **Editors** dialog window where the editor expects the filename.

private:

Extension <extension> not allowed.

You tried to open, insert, save, or export a file with the filename extension ERR, EXE or DLL.

The filename extension ERR is reserved for the error file that may be created by EasyCODE(C++) when a source or ETF file is opened or imported, while the extensions EXE and DLL are reserved for executable files and Dynamic Link Libraries respectively.

Rename the file and choose any extension except ERR, EXE or DLL.

private:

Unable to print file.

You tried to print a structure diagram on a printer which was not installed properly.

Make sure that

- the printer is switched on,
- the printer is on-line,
- the paper is loaded properly,
- the printer is connected properly,
- the printer driver is installed properly,
- the printer is selected in the Printer setup menu,
- the Print Manager can be loaded if the WIN.INI file contains the line *spooler=yes*,
- there is enough memory for the temporary Windows files. For details please refer to the Microsoft Windows User Manual.

private:

Only one file can be edited! The first selected file will be opened.

You selected several files in the File Manager and tried to open them by moving them to EasyCODE(C++).

As EasyCODE(C++) is able to open only one file at a time, you may decide whether the first selected file is to be opened.

Click **Ok** to open the first selected file.

Click **Cancel** if you do not want to open the file.

private:

External Editor cannot be used as standard editor if not configured.

You have defined the external editor as standard editor in the Options - Editors dialog window but not entered an editor.

Enter the name of an editor in the Options - Editor dialog window in the edit field External editor.

private:

Error <key> accessing BS2000 file.

You tried to access a BS2000 file or a BS2000 library element during which an error in the BS2000 has occurred.

Remove the reported error in the BS2000 and restart the action.

private:

Error initializing SDF-DOORS.

You tried to start EasyCODE(C++) or edit an imbedded of linked EasyCODE(C++) structure diagram for which the cooperation is configured with SDF-DOORS.

Check that both EasyCODE and SDF-DOORS are installed correctly.

Contact your EasyCODE vendor.

private:

Error during registration of OLE server

You tried to start EasyCODE(C++) or to edit a structure diagram embedded in or linked to EasyCODE(C++), but the OLE server could not be started.

Make sure that enough memory is available. If necessary, delete files that are no longer needed.

Make sure that the registration database is not damaged. If the registration database should be destroyed, load EasyCODE(C++) using the /INITIALIZE option and add the EasyCODE(C++) entries again to the registration database.

private:

Unable to load Help Manager.

You tried to get help from EasyCODE(C++), but the Windows Help Manager could not be loaded.

Make sure that the Help Manager WINHELP.EXE and the Help file is installed and not damaged.

private:

Unable to execute <program>.

You tried to start a program using the Execute command in the File menu, but an error occurred that could not be identified.

or

You tried to open the error file created by EasyCODE(C++), but the editor required for this action could not be loaded.

or

You tried to start an editor using the External editor command in the Edit menu

or

the editor required for displaying the analyzed file or the error file could not be started.

or

You tried to get system information using the System info command in the Help menu.

Make sure that

- the program linking settings are correct,
- the settings specified in the Editors dialog window are correct,
- enough memory is available,
- the program is installed and accessible.

private:

Error while executing <program>. Unable to create temporary file.

You tried to start a program using the Execute command in the File menu, but the temporary file required by this program could not be created.

You tried to start an editor using the External editor command in the Edit menu, but the temporary file required for this action could not be created.

Make sure that

- enough memory is available,
- the program has write access to the "TEMP" directory.

private:

Error while executing FTD.EXE.

You tried to start a file transfer using the FTOS (CCOM) transfer system, but failed.

Possible reasons:

- the TOOLHELP.DLL file is not available,
- error in the registration of the checking routine,
- the environment variable FTC97 containing the path of FTD.EXE has not been set,
- the FtDirectory entry (if existing, see CUSTOM.DOC file) in the INI file is incorrect,
- lack of memory, program not found or damaged etc.

Try to find out what has caused the error by means of this list and then try to eliminate it.

private:

Error while executing WFT.EXE.

You tried to start a file transfer using the WFT-BAM transfer system, but failed.

Possible reasons:

- the TOOLHELP.DLL file is not available,
- error in the registration of the checking routine,
- the environment variable WFTDIR containing the path of WFT.EXE has not been set,
- the FtDirectory entry (if existing, see CUSTOM.DOC file) in the INI file is incorrect,
- lack of memory, program not found or damaged etc.

Try to find out what has caused the error by means of this list and then try to eliminate it.

private:

Error while printing - (unidentifiable).

An unidentifiable error occurred during the test print.

Make sure that

- the printer is switched on,
- the printer is on-line,
- the printer is connected properly,
- the printer driver is installed properly,
- the correct printer is selected,
- the Print Manager can be loaded, if the WIN.INI contains the line *spooler=yes*,
- there is enough memory for the temporary Windows files.

For details see the Microsoft Windows User Manual.

private:

Error while printing - printing aborted.

You terminated the test print, although the Print Manager has not yet received all pages.

Start the test print again, if necessary.

private:

Error while printing - out of memory.

There is not enough memory available for the test print.

To make memory available, close other active applications.

If this error message occurs frequently, take one of the following steps:

- Expand the main memory of your personal computer.
- Remove resident programs and drivers from the system's main memory.

private:

Error while printing - out of disk space.

There is not enough disk space to create the temporary files required for the test print.

Try to make memory available for the temporary Windows files. For details see the Microsoft Windows User Manual.

private:

Error while exporting <file>.

Error file cannot be displayed because a program that creates a result file is running.

Errors occurred during the export of a file. Since both the export error file and the result of the program started with the Execute command in the File menu are to be written into the result file, the result file cannot be displayed, as long as the program is running.

Repeat the exporting process as soon as the program currently running is terminated.

private:

Error while opening <file>.

You tried to open a file required by EasyCODE(C++), but failed for unknown reasons.

Try to find the cause of the error and eliminate it (you may e.g. check the PATH statement, use a backup copy of a deleted file,...).

Note: This message will also appear when EasyCODE(C++) or an external editor started by EasyCODE(C++) is not able to open the internal TMP file required for this action (~XXXXXXX.TMP).

You tried to create a JOB file, but failed, because the drive or the file is write-protected.

Remove the write-protection, if possible, and start the generating process again.

private:

Unable to write to *EASY-CPP.INI*.

You tried to install a printer or choose the Save settings command in the Options menu, but EasyCODE(C++) found that the INI file is write-protected.

Reset the read-only attribute using the File Managers or the DOS command
attrib -r <Ini-Datei>

private:

Error in file transfer. Return symbol=1.

You tried to start a transfer job using the FTOS(CCOM) file transfer system, but failed.

Try to locate the error with the help of the FTOS file transfer description and the return code and eliminate the error.

private:

Error in network. Please quit EasyCODE(C++) and start again.

During one of the regular checks of the network server concerning your user authorization, EasyCODE(C++) found a change in the license you are using.

Quit EasyCODE(C++) and start again.

private:

Internal stack overflow. Job cancelled.

During the analysis of a file you wanted to open or insert, an internal stack overflow occurred.

Possible remedies:

- Reducing the file.
- Expanding the memory.

private:

No printer installed.

You tried to install a printer, but no printer is available.

You can add a printer using the **Add printer** command or the **Ports** command in the Windows **Control Panel** application.

private:

No transfer system selected. Unable to start file transfer.

You loaded a transfer list without having selected a transfer system.

Choose the **File transfer** command in the **Options** menu to select a transfer system in the "File transfer settings" dialog window.

private:

Unable to activate file transfer as long as execution file <execution file> exists.

You tried to start a file transfer, but a file transfer is already in progress.

Wait until the active file transfer will be terminated and EasyCODE(C++) will automatically delete the execution file. Then restart the new file transfer.

You tried to start a file transfer, although a previous file transfer has been aborted and therefore EasyCODE(C++) was not able to delete the execution file automatically.

Delete the execution file and then restart the new file transfer.

private:

No definition found for <selected text>.

You tried to view with the help of a project database the definition of an elements you have selected in the structure diagram, but no fitting definition could be found.

Check if you have selected the right passage and if so, change the selection.

private:

No references found for <selected text> .

You tried to view with the help of a project database the references of an element, but not fitting reference could be found.

Check if you have selected the right text passage and if not so, change the selection. If the selection is correct, check whether you actually use the selected element.

private:

Maximum user number reached. EasyCODE(C++) will not be started.

You tried to start EasyCODE(C++). During a check concerning your user authorization, EasyCODE(C++) found that all licences are currently in use.

Wait until another user finishes working with EasyCODE(C++),

or

install an additional license package to increase the number of potential users.

private:

Maximum size of result file exceeded. Unable to display entire file <file>..

You tried to display in the result file window a file exceeding the maximum size of a result file.

EasyCODE(C++) will display only a part of the file (about 30,000 characters) in the Result file window.

private:

Program name is not clear.

You entered a program name into the Name dialog box of the Program settings dialog window which already exists in the program list.

Enter a different name for the program.

private:

Default printer not found.

You tried to execute an action for which Windows requires a default printer.

Make sure that a valid printer is listed as the default printer in the WIN.INI file.

private:

No line numbers assigned.

You tried to view a line with a certain line number, but no line numbers are currently assigned.

Save the structure diagram to create line numbers.

private:

Structure diagram <segment information> exceeds worksheet area.

You tried to open or design a structure diagram, but the diagram or the indicated segment exceeds the coordinates for screen output.

Windows restricts the screen coordinates to **65536*65536** pixels. The higher the resolution of the screen, the more restricted the display of very large structure diagrams.

Although EasyCODE(C++) was not able to display the structure diagram, the complete structure diagram is correctly stored in EasyCODE(C++).

Try

- choosing a smaller font,
- using the alternative layout of the **IF-THEN-ELSE** construct,
- choosing the option "fit into page",
- breaking the structure diagram down into segments.

The latter is perhaps the best, even if the most complicated solution.

The following situations may occur:

The structure diagram cannot be displayed on the screen since the last designing step.

Undo the last designing step, then break down the structure diagram into segments. Now you can carry out the last designing step again.

The structure diagram cannot be displayed on the screen when you open it.

Choose one of the following options:

- use a lower-resolution screen and break the structure diagram down into segments,
- contact your EasyCODE supplier.

private:

Structure diagram <segment information> exceeds worksheet area when printing.

You tried to print a structure diagram or activated the Preview list in the Print dialog window, but the size of the entire structure diagram exceeds the coordinates for printer output

or

You opened the Page preview window and/or the display of page numbers in the status bar, but the size of the selected segment exceeds the coordinates for printer output.

Windows restricts the coordinates to 65536*65536 pixels. The higher the resolution of the printer, the more restricted the printing of very large structure diagrams.

If it is only a certain segment that is too large, only this segment will not be printed. The other parts of the structure diagram will be printed correctly.

Although EasyCODE(C++) was not able to print the segment, it is correctly stored in EasyCODE(C++).

Try

- choosing a smaller font,
- using the alternative layout of the **IF-THEN-ELSE** construct,
- breaking the structure diagram down into segments.

The latter is perhaps the best, even if the most complicated solution.

private:

Structure diagram in <supplementary information> is EasyCODE(C++/SPX) format <version>. This format is not supported.

You tried to open a file, to insert a file or the Clipboard contents or to edit an embedded object, but the data were created with an older version of EasyCODE(C++) or EasyCODE(SPX) .

Every EasyCODE(C++) version can edit data created with the previous version of EasyCODE(C++) or EasyCODE(SPX) . Older versions are not supported. You can update your data by editing them with the intermediate versions of EasyCODE(C++) or EasyCODE(SPX) .

You tried to edit your data with an older EasyCODE(C++) version.

Use the adequate EasyCODE(C++) version.

private:

Structure diagram in <supplementary information> contains a structural error.

You tried to open a file, to insert a file or the Clipboard contents or to edit an embedded object, but EasyCODE(C++) found that the data are partially damaged.

Try to reconstruct the structure diagram from your backup copy.

You tried to view a lower hierarchical level to edit a segment or to print, save or update a structure diagram, but EasyCODE(C++) could not identify the internal TMP file as valid and therefore was unable to load it.

Open the CPP file again to use the backup copy and carry out the last changes again, if required.

private:

Structure diagram in <supplementary information> will be converted from EasyCODE(SPX) into EasyCODE(C++) format.

You tried to open or insert an SPX file, insert it from the Clipboard or edit it as an embedded object.

EasyCODE(C++) has to convert all EasyCODE(SPX) constructs into EasyCODE(C++) constructs.

Click **Ok** to accept automatic conversion.

Click **Cancel** to escape.

See also:

[Conversion Rules from SPX into C++ Format](#)

private:

Structure diagram in <file> will be converted from EasyCODE(SPX) into EasyCODE(C++) format. File is not protected from access by other users.

You tried to open or insert an SPX file.

EasyCODE(C++) has to convert the EasyCODE(SPX) constructs into EasyCODE(C++) constructs. The converted file will not be protected from access by other users in the network.

Click **OK** to accept automatic conversion.
Click **Cancel** to escape.

See also:

[Conversion Rules from SPX into C++ Format](#)

private:

Invalid name for PIF file.

You entered an invalid name for the redirection program into the Program settings dialog window.

Enter a valid filename including the path, if required, but without filename extension.

private:

Invalid prompt string in Program Settings:<prompt>.

You have used a syntactically incorrect symbolic name for the user input in the Program Settings Window.

Enter for the query **%PROMPT:<text>%** or **%PROMPT<n>:<text>%** and for the use of the values entered in subsequent text field **%PROMPTVAL%** and **%PROMPTVAL<n>%** respectively. <n> must be a value between 1 and 9.

private:

Out of memory. Job cancelled.

You tried to execute an action which requires more than the available main memory.

To make main memory available, try one of the following measures:

- View the structure diagram on top level and save it.
- Close other applications.
- If this message occurs frequently, you may try one of the following steps:
 - Break down the structure diagram into segments.
 - Expand the main memory of your personal computer.
 - Remove resident programs and drivers from the system's main memory.

private:

Out of disk space (drive <drive>). Job cancelled.

You tried to save or generate a file, but there is not enough memory available on the selected drive.

Delete files that are no longer needed.

private:

File <file> does not exist. Create file?

You tried to start EasyCODE(C++) using as a parameter a non-existent file.

Click **Yes** if you want EasyCODE(C++) to create the file.

Click **No** to escape.

private:

File <file> in use. Open file with write-protection?

You tried to open a file which is currently used by another application.

You may open the file with write-protection. If you modify the file, you must save it under a different name.

private:

File <file> cannot be opened without write-protection both in EasyCODE(C++) and in the editor. Write-protection in EasyCODE?

You tried to correct an error detected by EasyCODE(C++) while reading the source file with the help of the "editor to be used for correction of invalid sources or ETF files", but the file cannot be opened without write-protection both in EasyCODE(C++) and in the editor.

Click **Yes** if you want EasyCODE(C++) to open the file with write-protection, so that you are able to correct the file in the editor.

Click **No** if you want the file to be opened with write-protection in the editor and without write-protection in EasyCODE(C++).

Click **Cancel** to open the file without calling the editor.

Note: You may predefine a default value by entering "LockSourceOnWarnings" in the INI file (see CUSTOM.DOC file), so that this message will not appear.

private:

File <filename> has been changed. Reload file?

EasyCODE(C++) found that the current file has been changed by another application. Since you have not changed your file, you may reload it.

Click **Yes** to load the actual version of the file.

Click **No**, to resume your work on the file.

private:

File <filename> has been changed. Discard current changes and reload file?

EasyCODE(C++) found that the current file has been changed by another application.

Click **Yes** to get the actual version of the file. The changes you made since the last saving of the file will be lost.

Click **No**, to resume your work on the file.

private:

File <filename> has been changed by another application. Overwrite anyway?

When you tried to save a file EasyCODE(C++) found that the file has been changed by another application since the last saving or opening.

Click **Yes** to get the actual version of the file. The changes you made since the last saving of the file will be lost.

Click **No** an, to save the file with another filename.

Click **Cancel**, if you do not want to save the file at the moment.

private:

The command you have chosen will break off the link between the open embedded object and <document>. Update embedded object before proceeding?

You were editing an embedded structure diagram with EasyCODE(C++) and then chose one of the commands "New", "Open" or "Save as".

Click **Yes** to update the embedded object in the target document and then break off the link to this object.

Click **No** to break off the link without updating the embedded object.

Click **Cancel** to escape.

private:

Analysis of <file>: <n> errors and <m> warnings. Call editor for correcting errors?

You tried to open or insert an ETF file or a source file in which EasyCODE(C++) detected structural errors.

Click **Yes** if you want EasyCODE(C++) to display the file in the editor you specified using the **Editors** command in the **Options** menu, so that you can correct the errors.

Click **No** if you want to escape.

private:

Analysis of <file>: <n> errors and <m> warnings. Display error list <error file>?

You tried to open or insert an ETF file or a source file, in which EasyCODE(C++) detected structural errors.

Click **Yes** if you want EasyCODE(C++) to display the error file in the editor you specified using the **Editors** command in the **Options** menu.

Click **No** to escape.

private:

Analysis of <file>: <n> errors and <m> warnings. Display error list <error file> and call editor for correcting errors?

You tried to open or insert an ETF file or a source file, in which EasyCODE(C++) detected structural errors.

Click **Yes** if you want EasyCODE(C++) to display the error file in the editor you specified using the **Editors** command in the **Options** menu, so that you can correct the source or ETF file.
Click **No** to escape.

private:

Incorrect file transfer. Return code=1. Start next transfer job?

You tried to start a transfer job using the file transfer system FTOS(CCOM), but failed.

Click **Yes** to start the next transfer job.

Click **No** to escape.

private:

Unable to create backup copy <file>. Save without backup copy?

You tried to save a file, but EasyCODE(C++) was not able to create the corresponding backup copy.

Possible reasons:

- A backup copy has already been created with the read-only attribute.
- No access authorization for backup copy or original file.

Click **Yes** to save the file without backup copy.

Click **No** to escape without saving.

private:

Update object embedded in <program> before closing?

You were editing an embedded object in EasyCODE(C++) and tried to quit EasyCODE(C++).

Click **Yes** to update the embedded object in the target document and quit EasyCODE(C++).

Click **No** to quit EasyCODE(C++) without updating the embedded object.

Click **Cancel** to escape.

private:

Remove program(s) from program list?

You selected a program name in the Programs dialog window and then clicked the Remove button.

Click **Ok** to remove the program from the program list.
Click **Cancel** to keep the program in the program list.

private:

Save the Clipboard contents?

You copied a large amount of data (more than can be displayed on one screen) to the Clipboard and then quit EasyCODE(C++).

Click **Yes** if you want EasyCODE(C++) to store the Clipboard contents until you copy to the Clipboard again or exit Windows.

Click **No** to delete the Clipboard contents.

private:

Close current BS2000 connection?

You tried to open a BS2000 connection with the Open BS2000 connection command in the Options menu, although you have already opened a BS2000 connection.

Click **Yes** and EasyCODE(C++) will close the current BS2000 connection.
Click **No** to cancel this action.

private:

Save transfer list?

You have modified the transfer list since last saving it. Now you want to execute a command which would cancel the transfer list.

Click **Yes** to save the transfer list.

Click **No** to scrap the changes in the transfer list.

Click **Cancel** to escape. Your modifications to the transfer list will not be lost.

private:

<Update> changes?

You tried to transfer a file, but EasyCODE(C++) found that the current file has been changed since last saving or exporting it.

Click **Yes** if you want the current file to be updated.

Click **No** to start the file transfer immediately (without updating the file).

Click **Cancel** to escape.

private:

Save changes in <file>?

You have modified the current structure diagram or options stored in the structure diagram (layout, font) since last saving it, and now you tried to execute a command which would result in losing the modifications.

This message will appear when you choose one of the following commands:

New

Open

Exit

Close in the control menu of the application window.

Click **Yes** to save the structure diagram.

Click **No** if you do not want to save the changes.

Click **Cancel** to escape. The changes will not be lost.

private:

Save settings?

You changed the source attributes, program linking or file transfer options. When you quit EasyCODE(C++) or Windows, these settings will be lost.

Click **Yes** to keep the settings for later EasyCODE(C++) sessions.

Click **No** to scrap the settings.

Click **Cancel** to escape.

private:

Delete transfer job(s) from transfer list?

You selected the name of (a) transfer job(s) in the Transfer dialog window and then clicked the Delete button.

Click **Ok** to delete the transfer job(s) from the transfer list.
Click **Cancel** to keep the transfer job(s) in the transfer list.

private:

Line <number> not found. View preceding line?

You tried to view a line with a number that does not exist in the structure diagram.

Click **Yes** to view the preceding line number. If no lower line number is available, EasyCODE(C++) will display the top hierarchical level and select the first element in the structure diagram.

Click **No** to keep your original position.



Printing is not available in Demo Version.

You can acquire the complete version of EasyCODE(C++) from your EasyCODE vendor.



Demo version cannot cooperate with other applications.

You can acquire the complete version of EasyCODE(C++) from your EasyCODE vendor.



Executing this command is only available for small structure diagrams in demo version.

You can acquire the complete version of EasyCODE(C++) from your EasyCODE vendor.

Introduction

EasyCODE(C++) is a product from the EasyCODE family. It is a program for easy, computer-aided software engineering, using the techniques of structured programming. With EasyCODE(C++), you can create, edit and print C/C++ programs in the form of structure diagrams.

EasyCODE(C++) is ideal both for designing new C/C++ programs, and for maintaining, documenting, and developing existing C/C++ programs.

Designing, Maintaining, Developing and Transferring C/C++ programs

Starting Your Work

Editing Structure Diagrams

Editing Text

Saving Structure Diagrams

Transferring Files

Starting Programs directly from EasyCODE(C++)

(compilers in particular)

Using EasyCODE(C++) in a Network

Documenting Your Work

Changing Printers

Choosing Printer Fonts

Printing Structure Diagrams

Object Linking and Embedding (OLE)

See also:

Procedures

Menus

Starting Your Work

You can
create new structure diagrams,
edit existing structure diagrams,
start your work from language-independent program designs
or
start from existing C/C++ programs.

See also:
Procedures

Editing Structure Diagrams

You can

[insert constructs](#)

[enclose constructs](#)

[edit structure diagrams without the mouse](#)

[insert files](#)

[copy, cut and paste using the Windows Clipboard,](#)

[redesign your structure diagram](#)

See also:

[Procedures](#)

[Commands](#)

Editing Text

You can
fill symbolic names and empty boxes with text,
modify existing text,
search and replace text.

See also:

Procedures

Commands

Saving Structure Diagrams

During an EasyCODE session, we recommend that you save your structure diagram at regular intervals to prevent data loss in the event of system failure or other interruptions to your work.

After having finished your structure diagram, you should save it in order to be able to retrieve it later.

For this purpose, you use the Save and Save as commands in the **File** menu.

The structure diagram is stored in a source file.

See also:

Procedures

Commands

Transferring Files

You can
[choose a transfer system and specify default settings](#)
[create, modify and start transfer jobs](#)

See also:
[Procedures](#)

EasyCODE(C++) Network Version

Using EasyCODE(C++) in a Network

Functions of EasyCODE(C++) in a Network

Using Files in a Network

Using EasyCODE(C++) in a Network

Basically, the functions of EasyCODE(C++) installed in a network are the same as of EasyCODE(C++) installed on the harddisk of a personal computer. In a network, you can make files accessible to other users and allow them to modify your file, or you can protect your file from changes.

You can use the network server to save files and exchange them between different users.

Moreover, all users participating in the network can use a printer connected to the network server.

See also:

[Functions of EasyCODE\(C++\) in a Network](#)

[Using Files in a Network](#)

Functions of EasyCODE(C++) in a Network

[Starting EasyCODE\(C++\) in a Network](#)

[Working with EasyCODE\(C++\) in a Network](#)

[Quitting EasyCODE\(C++\) in a Network](#)

See also:

[Using EasyCODE\(C++\) in a Network](#)

[Using Files in a Network](#)

Using Files in a Network

In a network, a file can be used by several users. Therefore it is necessary to have some kind of protective mechanism to prevent the file from undesired changes. Choose one of the following methods to protect files used by more than one user from undesired changes:

- Protective mechanism on system level (see system description)
- Protective mechanism on network level (see network description)
- Protective mechanism in EasyCODE(C++)

Methods of protecting files in EasyCODE(C++)

General notes

In general, files in a network are protected from write access by other applications. As a consequence, not even the user himself is able to change his files using other applications, if they are open in EasyCODE. In case you want to edit a file in several applications at the same time (e.g. in EasyCODE and Visual C++), you may switch off the protective mechanism for files by entering `UnlockDrives=<drive>` into the INI file (description see CUSTOM.DOC info file). You must, however, remember that the file may then be changed by any other network user, too.

Open

You can protect a CPP file from undesired changes by selecting the **Read only** option in the "Open" dialog window. When the file is open, the title bar displays the words "Read only" after the filename. The file can no longer be saved under this filename, no matter whether the file is opened by other users or not.

If another user has already opened the file without write-protection, the file will automatically be read-only.

Only one user at a time can open a file without write-protection. Other users can, however, open the file with write-protection and read it. If you open a CPP file without write-protection, only you can modify it, until you open another file, start creating a new structure diagram, or quit EasyCODE(C++).

When opening a source or an SPX file, a message tells you that the file has to be converted into the internal format. During the conversion the file is not protected against access by other users and the checkbox "Read only" is not available.

Save

You can save files on diskettes, on the harddisk or of your own workstation, or on the network server. Please note that

- it is not possible to save a file under the name of a file that is currently used by another user.
- if you save a file that has been opened with write-protection, the Save as dialog window will appear. After the file has been saved under a different name, it will no longer be read-only.

See also:

[Using EasyCODE\(C++\) in a Network](#)

[Functions of EasyCODE\(C++\) in a Network](#)

Starting EasyCODE(C++) in a Network

Before you can start EasyCODE(C++) from the workstation of a network, it must be installed both on the network server and the workstation using the SETUP program supplied with EasyCODE(C++) (see EasyCODE User Manual).

When you start EasyCODE(C++) from a workstation, EasyCODE(C++) will check the number of user licences. If the maximum number of licences is already reached, an error message will appear and EasyCODE(C++) will not be started.

If there should be other errors (e.g. incorrect server or workstation setup), EasyCODE(C++) will be started, but not as a network version (no file protection). When this occurs, the following message will be displayed: "This application is not a network version. Usage in a network may cause problems."

Working with EasyCODE(C++) in a Network

Every EasyCODE(C++) application that has been started from a workstation of the network server will regularly update its "time mark" in a special file on the server. User licences that have not updated their time marks for a certain time will be made available to other users.

Note: If the time mark has not been updated although you have not quit EasyCODE(C++) (e.g. when you have opened a DOS shell), another user may obtain your licence. When you return to EasyCODE(C++), the following message will appear: "Error in network. Please quit EasyCODE(C++) and start again."

Quitting EasyCODE(C++) in a Network

When you quit EasyCODE(C++), your user licence will be made available to other users.

Using the Windows Clipboard

EasyCODE(C++) is able to copy an area to the Windows Clipboard in two different formats:

1. as a Bitmap
2. as a picture based on the selected printer font.

You can look at the contents of the Windows Clipboard in both formats with the help of the Windows application **Clipboard**. When using the contents of the Windows Clipboard in another application, you have to select the appropriate format. *Word for Windows* for example uses the picture format.

For details concerning the Windows Clipboard please refer to the Microsoft Windows User Manual.

See also:

[Cut, Copy and Paste Areas](#)

How to Use the Windows Clipboard

During your work, you may encounter the following cases:

[Copying Areas During an EasyCODE\(C++\) session](#)

[Copying Areas into Different Structure Diagrams](#)

[Copying Areas from EasyCODE\(SPX\)](#)

See also:

[Cut, Copy and Paste Areas](#)

General Notes on Editing Dialog Windows

A **dialog window** consists of several **dialog boxes**, each of which is determined for certain entries.

Before you enter anything into a dialog box, you have to position the cursor to the appropriate edit control. You can use one of the following methods:

- Click the desired dialog box with the left mouse button.
- Push the ALT key and at the same time the character underlined in the name of the dialog box.
- Use the TAB key several times. The input cursor will jump from one edit control to the next until it returns to the first box. Pressing the TAB key and the SHIFT key at the same time will move the cursor backwards.

When you press the CTRL+TAB keys in the multi-line editing window, the cursor moves to fixed positions.

Status Bar

The status bar is located above the lower edge of the application window. You will see text to the left and a display to the right.

The **text** informs you about the current status of EasyCODE(C++), gives you tips on how to proceed, or recommends using **Help** for more detailed information.

The **display** shows the number of pages required for printing and the settings of the keyboard lock functions. From left to right these are the following:

W x H	<number>x<number>	number of pages required for printing (W idth x H eight)
NUM		the numeric keypad is on
SHIFT		SHIFT is locked
SCROLL		scrolling is activated
OV		Overwrite mode

See also:

[Switching on/off the Status Bar](#)

Linking a Compiler to EasyCODE(C++)

You can start a compiler directly from EasyCODE(C++) to compile the source program created from a structure diagram. First configurate the compiler you want to link to EasyCODE(C++) using the Program linking command in the Options menu. The error file created during compilation will then be displayed in the result file. Then start the compiler by choosing the appropriate entry in the window that will appear when you choose the Execute command in the File menu. After compilation you can use the Result file window to view the elements to be corrected with the help of the line numbers shown in the result file.

General Notes on Opening and Inserting Source Files

EasyCODE(C++) can open and insert source files, no matter whether they have been created by EasyCODE(C++) or not.

A source file not created by EasyCODE(C++) or a source file created by EasyCODE(C++) which has been modified with the help of a text editor, should comply with the following structure conventions:

- All C/C++ language constructs must be formulated in source code syntactically correct as regards syntactical units such as braces or brackets. It is particularly important that the source file does not contain any preprocessor statements or macros which are used for redefining syntactical units or for making them invisible or which have to be resolved so that C constructs will be syntactically correct,
- Preprocessor statements must not be placed between a keyword and the following brace,
- Comments and string constants must be terminated correctly,
- The EasyCODE comments created by EasyCODE(C++) in the source file must remain unchanged.

When you open or insert a source file that has not been created by EasyCODE(C++), it will be displayed as a structure diagram or as an area according to the following conventions:

1. Statement lines which are arranged sequentially and which do not contain any control statements (if, while, ...) will be combined and displayed as one construct.
2. The structure diagram will be split up into segments according to the following rules:
 - Every function will become a separate segment; the function name will become its header.
 - Every class will become a separate segment; the class name will become its header.
3. Preprocessor statements will not be interpreted, but displayed as statements.
4. Tabulators in the source code will be replaced by blanks, proceeding from a tabulator space of 8 characters.

General Notes on Saving Source Files

When you save a structure diagram, a source file will be created according to the following rules:

- The constructs within the structure diagram will be represented by language constructs valid in C/C++, with EasyCODE(C++) automatically creating the necessary syntactical additions which include:
 - all necessary braces,
 - a semicolon after each **do while**, **break**, **continue** and **return**,
 - a colon after the value of each **case**,
 - brackets surrounding the expressions named in the **if**, **while**, **do while** and **switch** constructs as well as the for expression formulated in the **for** construct,
 - comment symbols `/*` and `*/` surrounding block comments and headers,
 - It is therefore not necessary to enter the syntactical additions mentioned above, when you edit a structure diagram. EasyCODE(C++) does, however, not create a semicolon at the end of a statement!
- The generated source file contains additional information important for EasyCODE(C++) formulated as command lines in the following form:

```
/* EasyCODE x <text> */.
```

x stands for one of the following characters, which mean:

E	end of the class footer
F	beginning of the function
O	file-specific options
V	source file was created by the EasyCODE(C++) version mentioned below at the date mentioned below and contains the short information mentioned below.
-	separates statements
=	empty statement
:	beginning of block
(beginning of segment
)	end of segment
<	beginning of text not complying with the structure conventions
>	end of text not complying with the structure conventions

<text> is a version number, a sequence of file-specific options, a block comment, a header combined with administrative information required by EasyCODE(C++) or empty.

Note: The two comments surrounding text not complying with the structure conventions are of utmost importance for EasyCODE(C++) to be able to reconstruct a structure diagram still containing parts that are incomplete, not yet adequately refined or syntactically incorrect from the stored source file. Please do by no means remove these comments from the source file.

Text for Symbolic Names

Text used for symbolic names may contain any characters except the following:

- Block comments, headers and short info for source files may not contain the comment brackets `/*` and `*/`. This is necessary for avoiding nested comments within the source file when it is saved.
- Text may not contain the sequence `/* EasyCODE`. This is necessary for an exact identification of this additional information especially reserved for EasyCODE(C++) when a source file is opened.

EasyCODE(C++) identifies such errors when a structure diagram is saved. EasyCODE(C++) will then remove `/*` and `*/` from the short info of the source file, from block comments and from headers, and will replace `/* EasyCODE` with `/*_EasyCODE_`.

These restrictions in text design ensure that source files that are incomplete, insufficiently refined, or contain syntax errors and are not compilable, can be saved and opened again.

Design the text for symbolic names as follows:

- `<statement>` any sequence of C statements and/or comments including all necessary semicolons.
- `<expression>` C expression without enclosing brackets.
- `<declarations>` names and class declarations including terminating semicolon.
- `<function header>` type, name, and formal parameters of a function including parameter declaration
- `<class header>` class header including class name, class type (class|struct|union) and deduction
- `<for expression>` up to three C expressions separated by semicolons without enclosing brackets.
- `<value>` constant expression without terminating colon.

Please make sure that comments and string constants do not extend to more than one statement.

If you want to convert a program element into a comment instead of deleting it - as is common in program design - please do not use comments in EasyCODE(C++). Instead, add the following statement

#if 0

before the area in question and the following statement

#endif

after the area in question.

Tool Bar

Tool bars contain buttons which you may click to execute commands. You may, for instance, open a new document by clicking the New button in the default tool bar.

When you click a button of the tool bar, the status bar will display corresponding help text. This help text will be deleted when you release the mouse button.

The icons following below have been assigned to menu commands. Click an icon to get more information about it.



See also:

[Switching on/off the Tool Bar](#)

[Menus](#)

New File

Click this icon to execute the **New** command in the **File** menu.

Open File

Click this icon to execute the Open command in the **File** menu.

Save File

Click this icon to execute the **Save** command in the **File** menu.

Note: The button is dimmed if the structure diagram has not been modified since the last saving action, so that it is not necessary to save it again.

Print File

Click this icon to execute the **Print** command in the **File** menu.

Edit Text

Click this icon to execute the Edit text command in the Edit menu.

Split

Click this icon to execute the **Split** command of the **Edit** menu.

Search

Click this icon to execute the Search command in the Edit menu.

Continue Search

Click this icon to execute the **Continue search** of the **Edit** menu.

Delete

Click this icon to execute the Delete command in the Edit menu.

Cut

Click this icon to execute the **Cut** command in the **Clipboard** submenu of the **Edit** menu.

Copy

Click this icon to execute the **C**opy command in the **Clipboard** submenu of the **Edit** menu.

Paste

Click this icon to execute the **Paste** command in the **Clipboard** submenu of the **Edit** menu.

Undo

Click this icon to execute the Undo command in the **Edit** menu.

Refine

Click this icon to execute the **Refine** command in the **Edit** menu.

Push Down

Click this icon to execute the **Push down** command in the **Edit** menu.

Pull Up

Click this icon to execute the **Pull up** command in the **Edit** menu.

View Down

Click this icon to execute the Down command in the **View** menu.

View Up

Click this icon to execute the Up command in the **View** menu.

View Top

Click this icon to execute the Top command in the **View** menu.

View Line

Click this icon to execute the Line command in the **View** menu.

Help

Click this icon to execute the [Index](#) command in the **Help** menu.

Ok

Click this icon to take over the edited text into the structure diagram.

Cancel

Click this icon to cancel the editing.

Overwrite

Click this icon to overwrite existing text.

Note:

The state (on/off) of this option is remembered and used for the next editing.

Delete

Click this symbol to delete the selected text.

Cut

Click this icon to delete the selected text and put it in the Windows Clipboard.

Copy

Click this icon to put the selected text in the Windows Clipboard without changing the original data.

Insert

Click this icon to insert a text from the Windows Clipboard at the current text cursor position.

Undo

Click this icon to undo the last action in the edit mode.

Note: In this case undo refers only to the editing and is independent of the **Edit-Undo** command.

Select the Whole Text

Click this icon to select the whole text of the currently edited area.

Help

Click this icon to view the help chapter [Editing](#).

Help on Selected Text

Click this icon to call the Help function for the selected text.

Moving The Cursor

You can move the cursor within the visible section of the text by moving the mouse pointer to the desired position and clicking the left mouse button.

The cursor keys on the keyboard offer further possibilities to position the cursor.

See also:

Other Procedures for Editing Text

Editing Text

EasyCODE(C++) allows you to edit text more quickly and more comfortably by using keyboard and mouse.

You can

[move the cursor](#)

[select text](#)

[delete text](#)

[replace text](#)

[copy text](#)

See also:

[Other procedures](#)

Selecting Text with the Mouse

You can delete or replace larger text sections by selecting the text you want to delete or replace.

To select text:

1. Point to the first character you want to select and press the left mouse button.
2. Drag the mouse pointer to the last character you want to select while keeping the left mouse button pressed.
3. Release the mouse button.

Another way to select text:

1. Point to the first character you want to select and click the left mouse button.
2. Move the cursor to the last character you want to select, press the SHIFT key and click the left mouse button again.

See also:

[Selecting Text with the Keyboard](#)

[Other Procedures for Editing Text](#)

Selecting an Area

You can select one element of the structure diagram or select an area of the structure diagram consisting of more than one element.

A selected area of the structure diagram is indicated by inverted display. You can also select the entire diagram or areas which cannot be fully displayed on the screen.

See also:

Other Procedures

Selecting an Element

Click the selected element with the left mouse button.

If you press the Ctrl key and the left mouse button at the same time, the whole construct sequence in which the selected construct lies is selected.

Since some elements contain further elements (e.g. some constructs contain symbolic names), you may accidentally select an undesired element. In this case, simply click the desired element again outside the wrongly selected element.

See also:

[Hot Keys](#)

[Selecting Text with the Keyboard](#)

[Selecting Several Elements](#)

[Other Procedures](#)

Selecting Several Elements

To select an area consisting of more than one element:

1. Move the mouse pointer to the first or last element in the area you want to select.
2. Press the left mouse button.
3. Drag the mouse pointer over the area you want to select. EasyCODE(C++) will automatically invert the selected elements.
4. If the area you want to select is not fully displayed on the screen, drag the mouse pointer to the lower or upper edge of the window. The structure diagram will be scrolled automatically.
5. If you have selected too many elements, move the mouse pointer in the opposite direction to undo the selection.
6. Release the mouse button to fix the selection.

Another way to select several elements:

1. Point to the first element you want to select and click the left mouse button.
2. Move the pointer to the last element you want to select, press the SHIFT key and click the left mouse button again.
If the area you want to select is not fully displayed on the screen, scroll the structure diagram to the desired position using the scroll bars or the arrow keys (provided that the SCROLL LOCK function has been activated).

If you click a construct within existing selection while pressing the SHIFT key, the selection will be removed from the end of the selection up to the selected construct (depending on the direction of the original selection).

Note

In order to select an area of the structure diagram, which is not fully shown on the screen, you can scroll the structure diagram up and down with the arrow keys, while pressing the left mouse button (provided that the SCROLL LOCK function has been activated). EasyCODE(C++) will select the area starting from the first element of the structure diagram that has been selected to the element in which the mouse pointer is found after scrolling.

See also:

[Selecting Text with the Keyboard](#)

[Selecting an Element](#)

[Other Procedures](#)

Setting an Insertion Point

The mouse pointer symbolizes the area you want to insert, copy, or move.

To set an insertion point:

1. Scroll the structure diagram until the desired insertion point is visible.
2. Press a mouse button and drag the mouse pointer across the structure diagram.

Every inverted element is a valid insertion point.

Depending on whether you press the left or the right mouse button, an arrow at the edge of the mouse pointer will indicate whether the area will be inserted before or after the inverted element. To reverse the direction of the arrow, press the other mouse button and then release the original button. If no arrow is visible, the inverted element will be replaced.

3. Release the mouse button at a valid insertion point.

The area that you want to insert, copy, or move will be displayed in the desired position.

See also:

[Insert Menu](#)

[Copy Command in the Edit Menu](#)

[Move Command in the Edit Menu](#)

[Other Procedures](#)

Copying Text to Clipboard

You can use the [Windows Clipboard](#) to copy text.

To copy text:

1. Select the text you want to copy.
2. Press CTRL+INS to move the selected text to the Windows Clipboard.
3. Move the cursor to the desired position within your text.
4. Press SHIFT+INS to insert the text from the Windows Clipboard.

See also:

[Other Procedures for Editing Text](#)

Procedures

[Selecting an Area](#)

[Setting an Insertion Point](#)

[Redesigning a Structure Diagram](#)

[Hierarchical Structuring](#)

[Assembling](#)

[Editing Text](#)

[Designing Your Working Environment](#)

[Documenting Your Work](#)

[Designing More Comfortably](#)

[Starting a Compiler Directly from EasyCODE\(C++\)](#)

See also:

[Troubleshooting](#)

[Index](#)

Copying an Area During an EasyCODE(C++) Session

To do this:

1. Copy the selected area to the Windows Clipboard using the Copy Clipboard command in the Edit menu.
2. Continue to edit your structure diagram without changing the Windows Clipboard.
3. Choose the Paste Clipboard command in the Edit menu to insert the Clipboard contents at a valid insertion point.

Note:

To copy an area within a segment, you can also use the Copy command in the **Edit** menu.

See also:

Copying with the Help of Segment Windows
Other Clipboard Functions

Copying an Area to a Different Structure Diagram

To do this:

1. Start EasyCODE(C++) a second time.
2. Use the Copy Clipboard command in the Edit menu to copy the selected area into the Windows Clipboard.
3. Use the Paste Clipboard command in the Edit menu to insert the desired area into the other application at a valid insertion point.

See also:

Other Clipboard Functions

Copying an Area from an EasyCODE(SPX) Structure Diagram

To do this;

1. Start EasyCODE(SPX) in addition to EasyCODE(C++).
2. Use the Copy Clipboard command in the Edit menu of EasyCODE(SPX) to copy the selected area into the Windows Clipboard.
3. Use the Paste Clipboard command in the Edit menu of EasyCODE(C++) to insert the desired area at a valid insertion point.

See also:

Conversion Rules from EasyCODE(SPX) to EasyCODE(C++)

Other Clipboard Functions

Copying an Area to Another Segment Window

You can copy an area to another segment of the structure diagram with the help of two segment windows.

To do this:

1. Choose the New command in the Window menu to open a new segment window, if required, and view the segment in which you want to insert the area to be copied.
2. Select the area you want to copy.
3. Choose the Copy command in the **Edit** menu.
4. Select an insertion point for the area in the other segment window.

See also:

Window Menu

Copying with Clipboard

Moving Areas in Two Segment Windows

Moving Areas in Two Segment Windows

You can move an area to another segment of the structure diagram with the help of two segment windows.

To do this:

1. Choose the New command in the Window menu to open a new segment window, if required, and view the segment in which you want to insert the area to be moved.
2. Select the area you want to move.
3. Choose the Move command in the **Edit** menu.
4. Select an insertion point for the area in the other segment window.

See also:

[Copying an Area to Another Segment Window Window Menu](#)

Scrolling a Structure Diagram

To scroll the structure diagram in the segment window, you can click the vertical or horizontal scroll bars at the desired position, or drag the scroll bar thumb to the desired position. You can also scroll a structure diagram using the arrow keys.

Note

To scroll the entire area of the application window including the segment windows, use the vertical or horizontal scroll bars of the application window.

See also:

Other Commands for Designing More Comfortably

Searching for an Insertion Point While Pressing the Mouse Key

In order to insert the selected construct into the structure diagram, you must first select a valid insertion point. When **Help** was started, the cursor was not at a valid insertion point.

Note

By calling the **Help** function, you left the insertion mode. In order to insert the selected construct, you have to choose it again in the Insert menu.

See also:

Selecting a Valid Insertion Point

Inserting a Construct into an Element

In order to insert the selected construct into the structure diagram, you must first select a valid insertion point.

To leave the insertion mode, press the ESC key.

Note

By calling the **Help** function, you left the insertion mode. In order to insert the selected construct, you have to choose it again in the Insert menu.

Inserting a Construct Before an Inverted Element

You have already selected a valid insertion point for the selected construct. By releasing the right mouse button, you insert the construct before the inverted area.

Note

By calling the **Help** function, you left the insertion mode. In order to insert the selected construct, you have to choose it again in the Insert menu and select this insertion point again.

See also:

Selecting a Valid Insertion Point

Inserting a Construct After an Inverted Element

You have already selected a valid insertion point for the selected construct. By releasing the left mouse button, you insert the construct after the inverted area.

Note

By calling the **Help** function, you left the insertion mode. In order to insert the selected construct, you have to choose it again in the Insert menu and select this insertion point again.

See also:

Selecting a Valid Insertion Point

Inserting a Construct Instead of an Inverted Element

You have already selected a valid insertion point for the selected construct. By releasing the mouse button, you insert the construct instead of the inverted area.

Note

By calling the **Help** function, you left the insertion mode. In order to insert the selected construct, you have to choose it again in the Insert menu and select this insertion point again.

See also:

Selecting a Valid Insertion Point

Cancelling the Insertion Mode

Click this push button, if you have already selected an icon but want to quit the insertion mode.

- Note:** You can also quit the insertion mode by
- Selecting a menu,
 - Clicking an invalid insertion point with the mouse,
 - Pressing the ESC key.

Redesigning a Structure Diagram

EasyCODE(C++) offers several methods of modifying a structure diagram.

To redesign a structure diagram, use the following commands:

Edit text

Copy an area

Duplicate an area

Move an area

Delete an area

Enclose an area.

See also:

Clipboard Command in the Edit Menu

Other Procedures

Hierarchical Structuring

To create a top-down design, you use stepwise refinement. You can push down or pull up areas of a structure diagram to maintain structural clarity.

To do this, use the following commands:

Refine in the Edit menu

Up in the View menu

Down in the View menu

Top in the View menu

Push down in the Edit menu

Pull up in the Edit menu

These commands allow you to divide a structure diagram into segments.

Segments can be created as follows:

1. by refining a statement,
2. by pushing down an area,
3. automatically when a source file is opened which was not created with EasyCODE(C++).

See also:

Other Procedures

Assembling

EasyCODE(C++) offers several methods of assembling a structure diagram with the help of existing files and areas.

To assemble a structure diagram, use the following commands:

New in the File menu

Open in the File menu

File in the Insert menu

Clipboard in the Edit menu

A common feature of the **Open**, **Clipboard** and **Insert File** commands is the ability to integrate structure diagrams and areas that were created as SPX files using the language-independent EasyCODE component EasyCODE(SPX).

See also:

Other Procedures

Designing Your Working Environment

Use the commands in the Options menu and the Printer setup command in the **File** menu to design the EasyCODE(C++) environment according to your requirements.

See also:

Other Procedures

Documenting Your Work

EasyCODE(C++) offers several tools which allow you to use a structure diagram in applications other than EasyCODE(C++).

To do this, use the following commands:

Printer setup

Print

Clipboard

See also:

Other Procedures

Designing More Comfortably

EasyCODE(C++) offers several tools which allow you to design a structure diagram faster and more easily.

You may

use keyboard functions

use hot keys

use the tool bar and the toolbox

search and replace text

view lines

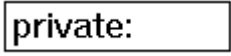
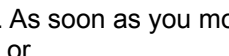
See also:

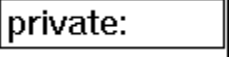
Other Procedures

Drag & Drop

Use Drag & Drop to move or copy a selected area within an opened structure diagram.

To do this:

1. Select the area you want to move or copy.
2. If you want to copy the selected area, click the area with the left mouse button.
If you want to move the selected area, click the area with the left mouse button and keep it pressed.
3. Move or copy the area to a new location. As soon as you move the mouse, the mouse pointer will change its shape to  or 

 respectively. Valid insertion points will appear inverted.

Note:

The distinction between copying and moving must take place at the beginning of the action and cannot be changed subsequently, as EasyCODE makes syntactic checks of the insertion points, and this check has different results for copying and moving.

When moving or copying into another segment window, the window is activated automatically and if you move the mouse pointer near the window border, the window is scrolled automatically.

Viewing with the Overview Window

Press the left mouse button in the Overview window and move the mouse pointer until the desired section of the structure diagram is displayed within the frame. When you release the mouse button, the chosen section is automatically displayed in the active window.

If you press the ESC key before releasing the mouse button, the displayed section remains unchanged.

See also:

[Opening and Closing the Overview Window](#)

[Functions of the Overview Window](#)

Functions of the Overview Window

The Overview window shows the active segment of the structure diagram in a reduced format. The section of the structure diagram displayed within the active segment window is indicated by a frame.

You can also use the Overview window for [moving the visible section](#).

By [clicking](#) outside the Overview window and inside the [application window](#) you can deactivate the Overview window and continue to work in your application as usual.

See also:

[Opening and Closing the Overview Window](#)

[Control Menu of the Overview Window](#)

Structure List Window

The Structure list window displays a list of the segments or classes or functions of a structure diagram. You can determine the settings for this list by selecting the corresponding entries in the list box displayed in the upper part of the window.

In the segment list all segments with segment numbers and headers are displayed. The active segment is selected. Doubleclick the desired segment with the left mouse button or select the required segment in the list and press the ENTER key to scroll the active segment window to this segment.

The lower part of the structure list window displays a list of all function names or class names. When you select an area that is part of one of the structures displayed, EasyCODE(C++) will select the corresponding structure in the list. Doubleclick a structure in the list with the left mouse button or select it in the list and press the ENTER key to scroll the active segment window to this structure and select it.

See also:

[Key Functions - ENTER key](#)

[Opening and Closing the Structure List Window](#)

[Control Menu of the Structure List Window](#)

Functions of the Page Preview Window

The Page preview window displays the current segment of your structure diagram in reduced format. A (green) grid within the window shows how the segment will be divided into print pages, one grid element corresponding to one page printed on the current printer.

Changes concerning printer settings, printer font, paper size and page header option will have immediate consequences on the size of the grid elements.

Constructs influencing the structure diagram width are displayed as red rectangles.

You can doubleclick a construct in the Page preview window with the left mouse button to select this construct and have it displayed in the middle of the main window. As in the application window, you may use the keyboard to select constructs also in the Page preview window, so that you can quickly change the width of a construct.

See also:

[Opening/Closing the Page Preview Window](#)

[Control Menu of the Page Preview Window](#)

Result File Window

After having executed a program, EasyCODE(C++) displays the contents of the result file defined in the Program settings dialog window in a small separate window, if you have chosen the "*Message on termination*" option for the program concerned.

You can also open or close the Result file window using the Result file command in the Options menu, regardless of the settings you chose. When the window is opened, the last result file will be loaded and displayed in this window.

In this window, you can scroll the result file, place the cursor and copy text to the Windows Clipboard, if required. The file displayed may not contain more than about 30,000 characters. If the file contains more characters, the remaining part of the file will not be displayed.

To close the Result File Window, you may use the **Result file** command in the **Options** menu or the control menu box of the window. To save the current status of the Result file window (on/off), use the **Save settings** command in the **Options** menu.

See also:

[Control Menu of the Result File Window](#)

[Starting a Compiler Directly from EasyCODE\(C++\)](#)

[Searching for Errors Using the Result File Window](#)

Searching for Errors Using the Result File Window

When you doubleclick a number or place the cursor on a number and then press the ENTER key or select a number and then press the ENTER key in the [Result file window](#), EasyCODE(C++) will display the corresponding line in the current structure diagram and gives you the possibility to edit the structure diagram element. This is especially useful when you want to see the error list of a program compilation in the Result file window. By doubleclicking a line number in the Result file window you can easily find the incorrect element and edit it immediately.

See also:
[Result File Window](#)




















Toolbox

When you choose the **Toolbox** command in the **Options** menu, a window will be opened in which all constructs of the **Insert** menu are represented by icons. You may move this window anywhere on the screen. With the help of the toolbox, you are able to select a construct and insert it at a valid insertion point by clicking the corresponding icon with the mouse.

In the upper left corner of the toolbox, there is an additional button displaying an arrow. Click this button to quit the insertion mode without inserting the selected construct. If you did not select a construct, this button is active.

After having clicked a push button, you can see the corresponding help information in the status bar. If you want to see help information without inserting a construct, move the mouse pointer away from the push button before releasing the mouse button.

The following table shows you which push buttons are provided in the toolbox and explains which construct will be inserted.

	<u>cancel insertion mode</u>		<u>continue</u>
	<u>Statement</u>		<u>return</u>
	<u>if</u>		<u>Class</u>
	<u>switch</u>		<u>private</u>
	<u>case</u>		<u>public</u>
	<u>default</u>		<u>protected</u>
	<u>for</u>		<u>function</u>
	<u>while</u>		<u>block</u>
	<u>do while</u>		<u>file</u>
	<u>break</u>		

See also:

[Switching on/off the Toolbox](#)

[Insert Menu](#)

Troubleshooting Info

What to do if

[the printer does not print](#)

[the DOS message "No memory available in the environment" appears during the execution of a BAT file](#)

[the right print margin is exceeded while printing a structure diagram](#)

[the performance of EasyCODE\(C++\) is unusually low](#)

[EASY-CPP.EXE is not loaded when you doubleclick a CPP file](#)

[non-English text appears in dialog and message windows](#)

[a structure diagram inserted into a target document by object linking or embedding has been damaged](#)

See also:

[Procedures](#)

[Index](#)

The printer does not print.

You have routed the printer output to a file.

The Printer setup command in the **File** menu informs you about the default printer settings and the default port. If a filename is selected in the ports list, this file contains the previously printed structure diagram. You can print this file on a suitable printer using the DOS command PRINT. Choose **Printer setup** in the **File** menu to select an installed printer.

The printer is not online, or the printer is out of paper.

Switch the printer to online, or load paper. The structure diagram will now be printed. Sometimes you have to activate the printer queue in the Windows Print Manager. See your *Microsoft Windows* Manual for details.

The right print margin is exceeded while printing a structure diagram.

The structure diagram is too wide and will not be printed on a single page.

To avoid having to change the structure diagram manually, you can:

1. Choose the **fit into page** button in the Print dialog window.
2. Choose a smaller printer font.
3. Choose a vertical layout for the if construct.
4. Break the structure diagram down into segments.

The Performance of EasyCODE(C++) is unusually low.

The structure diagram has become too large for the main memory.

Break the structure diagram down into segments, save it, then load it again.

EASY-CPP.EXE is not loaded when you doubleclick a CPP file.

The corresponding entry for the associated application is missing in the Registration Database or the WIN.INI file.

Use the File Manager to link the required extension to EasyCODE(CPP) or add the following line to the [extensions] section of the WIN.INI file:

```
<extension>=EASY-CPP.EXE ^.<extension>
```

<extension> is the file extension required for the structure diagram file to be opened when starting EasyCODE(C++).

The directory containing EASY-CPP.EXE is not listed in your PATH statement.

Add drive and directory on which EASY-CPP.EXE is stored to your PATH statement.

Non-English text appears in dialog and message windows.

Some dialog and message windows are not coming from EasyCODE, but from Windows. If you, for instance, have installed a German version of Windows, some windows will contain German text instead of English text.

You can convert this text into English by replacing the German COMMDLG.DLL file in your Windows setup directory by an English version of this file. This file is supplied with the EasyCODE setup disk(s).

During the execution of a BAT file the DOS message "No memory available in the environment" appears.

This message appears because before execution of your BAT file, the redirection program EASY-RE.EXE had to be run and, because of a DOS restriction, this program can pass on to your BAT file only the DOS default value or the value concerning the environment size it requires itself. When you have installed Windows 3.1, the value defined in the SYSTEM.INI file using CommandEnvSize is neglected in your BAT file.

When you start your BAT file (command line in the Program settings dialog window), use the following line:

```
COMMAND.COM /E:<bytes_for_environment> /C <file>.BAT
```

A document inserted into a target document by object linking or embedding has been damaged.

You have inserted a relatively wide structure diagram having image format into the target document. The printer font you selected in EasyCODE(C++) for printing the picture is too large.

Choose a smaller printer font so that the space available in the client application will not be exceeded.

