

# EasyCODE 9.3 for C/C++

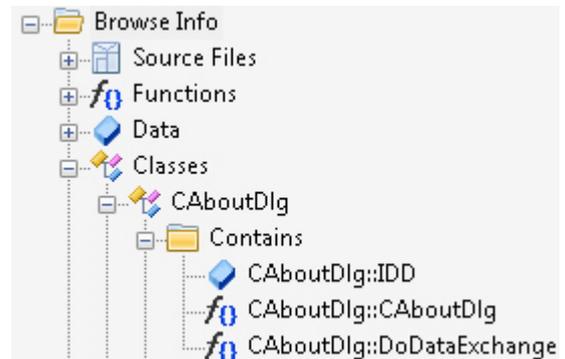
## Realize challenging projects quickly and reliably

EasyCODE provides all the tools you need to progress quickly and reliably from the specification through to the finished product. You will feel at home with EasyCODE right away, as you will achieve results immediately and increase productivity from day one. The unique combination of structure diagrams and statecharts is the key to projects with no overheads.

## Analyze existing files when you import them

The EasyCODE project environment enables you to import files from previous projects into a new one and to examine their relationships.

EasyCODE runs an integrity and name validation when you create the information file for the browser. This gives you initial indications of any missing includes and multiple defined symbols.



All the information about the individual modules, together with their dependencies and attributes, is then displayed to you in the browser for a systematic analysis. This means that you can see immediately which elements have been defined and where changes have an effect. Even at this early phase, you have all the information you need about your new project.

## Optimized workflows with version control

The version control systems Microsoft *Team Foundation Server*, Microsoft *SourceSafe*® and *Subversion* are seamlessly integrated in the EasyCODE project environment. All the critical functions in these systems are accessible from the shortcut menu, and can for the most part be automated in a convenient dialog window.

## Work with the compiler of your choice

EasyCODE functions with any compiler and also supports the utilization of multiple compilers or configurations in a single project. The interface is designed to allow you to manage configurations for different target systems and import previously created configurations from other projects.

## Create structure diagrams for a clear overview from the very start

Structure diagrams are still the number one choice for programs or sub-programs that cannot be realized as a state machine. Files developed in a structure diagram have clearly defined structures right from the beginning. Design errors in the file can be identified early on and rapidly remedied. This allows the basic framework of the file to be created quickly and with the correct syntax. The framework takes you almost automatically to an even more detailed level, until finally the structure diagram contains perfectly commented and valid source code.

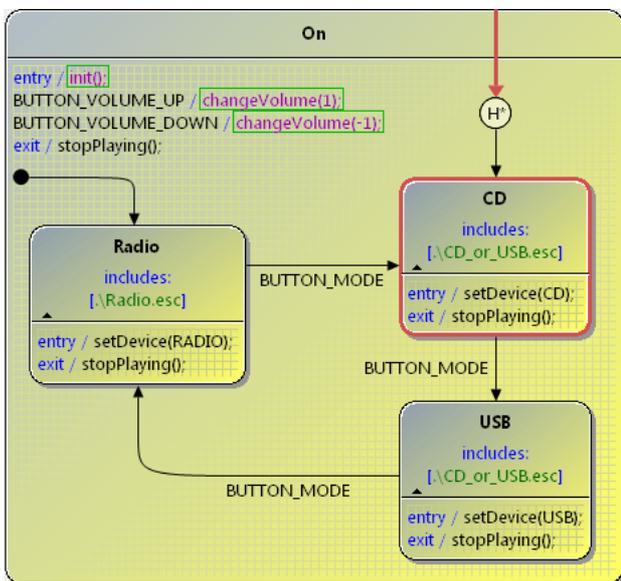
There is no dispute about the benefits of structure diagrams for analyzing complex algorithms and functions or resolving nested if-then-else queries. EasyCODE is capable of displaying the base metrics and adapting the format of the source files to comply with valid guidelines. The automatic code completion function helps you when you're writing your new code.

```
if ( m_stateFunc )
then
// stop the active state's submachine:
TRIGGER_INTERNAL_EVENT(m_stateFunc, eStopSubmachine);
else
endif

#ifdef USE_MULTITHREADING
if ( m_bOwnThread )
then
if ( m_bDoThreadLoop )
else
m_bRunning = false;
endif
endif
endif
```

## Create a sophisticated system using the layer technique

In the structure diagram editor, you can move sections of a diagram out of the view you are working on. For instance, in a larger function you can transpose single logical units to deeper layers of the structure diagram to make your workspace much less cluttered. The layers can be viewed, edited and printed individually, and they can also be easily expanded and collapsed again. The layer architecture of the structure diagram can be viewed in the structure overview.



## Design your software using statecharts

Use EasyCODE as early as the design phase! Statecharts are singularly suitable for designing products from within a higher graphic abstraction level. Many regulators or control and communications systems are more or less nothing other than complex statecharts. What could be more obvious than to use statecharts consistently from the initial design through to the finished product?

The concept can be designed in the diagram, leaving much more scope for ideas. Save time on descriptions or drawings. Discover new possibilities for implementing your specifications and avoid any errors in the architecture.

## Easily generate made-to-measure source code from your statecharts

EasyCODE generates source code from your statecharts with all the benefits of automatic code production. If the generated source code doesn't quite meet your specifications, then just quickly configure the code generator to get the code you need. This is a crucial aspect if you're developing security-critical applications.

## Integration of other state machines

States can contain references to other statechart files whose state machines are to be integrated as sub-machines when the code is generated.

## Simulations help to optimize your software more quickly

EasyCODE is capable of automatically generating common source code for multiple statecharts compiled in a single project to facilitate the interworking and management of the statecharts.

Additionally, EasyCODE incorporates a function for creating a simulation application to run the generated code independently of the target environment. The EasyCODE SimulationController can be used to trigger the events being used in the statecharts either per mouse-click or using an event batch file.

This means that state transitions are shown directly in the animated statechart. Actions that are dependent upon the target environment are simulated by pseudo-actions. All triggered events and actions are recorded in a log.

