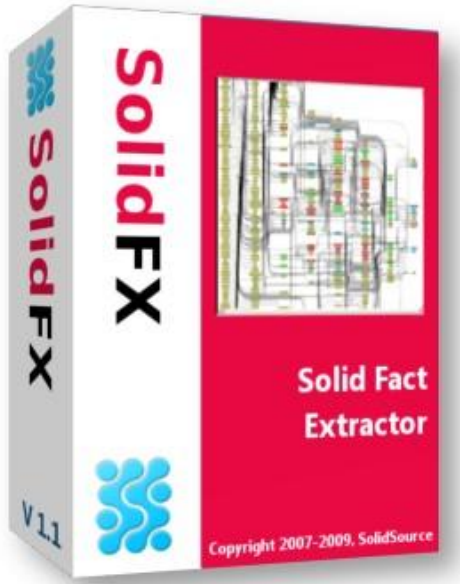


# Solid FX for C/C++

*Code faster, cleaner, safer*



## Description

Solid Fact eXtractor (SolidFX) is a standalone, non-intrusive solution for analyzing industry-size projects written in the C and C++ programming languages. SolidFX uses proprietary technology to analyze even the most complex C/C++ code bases efficiently and robustly.

SolidFX offers detailed design extraction, as well as predefined analysis scenarios and metrics to measure C and C++ code quality, maintainability, modularity, and defect potential -- all at your fingertips to allow you to code faster, cleaner, safer.

**SolidFX supports a wide range of tasks.** SolidFX supports most C and C++ dialects and platforms and is able to quickly analyze multi-million line projects, even if the code is incorrect and/or incomplete. The extracted information can be queried at levels ranging from detail information over each line of code to automatic reverse-engineered, full system architectural diagrams.

**SolidFX comes with several visualization plug-ins** showing combined UML and metric diagrams, large call graphs, and dozens of quality metric trends ranging from individual lines of code to entire subsystems. New plug-ins can be easily added to compute client-specific “what if” checks and queries and interface with third-party tools.

**SolidFX is easy to use.** The code analysis, reverse engineering, and quality assessment can be performed in one place via the SolidFX IDE and its various interactive visualization plug-ins quickly and easily.

## Intended audience

SolidFX addresses several roles within a project:

- **Project managers:** enables informed decision making for software development, refactoring, migration
- **Software architects:** detects crosscutting concerns, quality problems, and deficient design code areas
- **Developers:** detects potential problems, code duplication, low-quality code areas

## Benefits

### Cost reduction

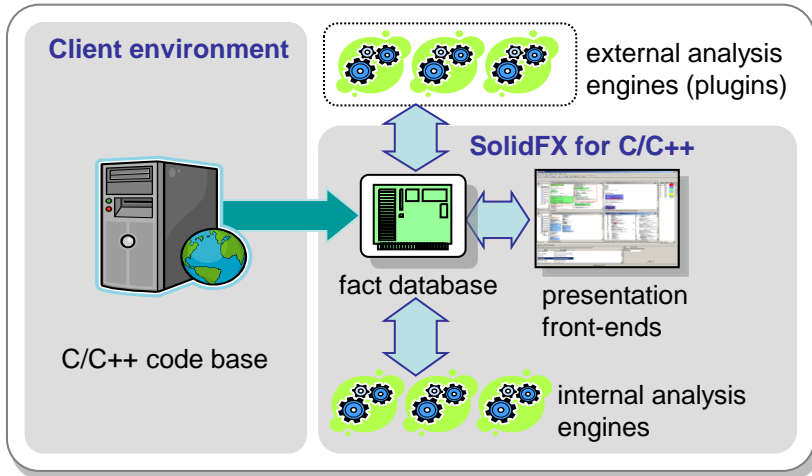
Plan maintenance and migration more effectively  
Reduce understanding and development time  
Code faster, cleaner, safer

### Quality improvement

Measure and monitor software quality  
Support targeted refactoring and maintenance  
Detect potential defects and safety problems in code  
Check reverse-engineered design against source code

### Decision making support

Easy-to-understand visual presentations of analysis  
Documentation and analysis report generation  
Support “what if” analysis scenarios  
Assess system maturity and stability  
Support outsourcing buy/sell decisions



**SolidFX interaction with the client environment**

## Integration in client environment

SolidFX is a non-intrusive tool, easy to integrate in the client environment. SolidFX requires no changes to the existing development process. SolidFX can be installed side-by-side with existing compilers such as Visual Studio or gcc, but can also run standalone. Custom analysis engines can be added as plug-ins during a hot integration. There are no development risks associated with removing SolidFX from the client environment.

## Example use cases

### Maintenance support

**Problem:** How can our system quality be improved to reduce overall maintenance costs?  
**Approach:** Assess the quality of code and architecture by computing several quality metrics. Visualize the metrics correlated with the (reverse engineered) system architecture. Detect weak spots. Understand the interplay of concerns in large, complex code bases at a glance.  
**Added value:** Supports maintenance by targeting effort where it is most needed.

### Refactoring support

**Problem:** How can we perform refactoring activities in a cost-effective and safe manner?  
**Approach:** Inspect the reverse engineered system structure. Compute and visualize code duplication and system modularity. Use these facts to plan and execute refactoring scenarios with reduced risks and costs. Iterate the tool-supported refactoring process during the entire product life-cycle.  
**Added value:** Supports low-cost, low-risk refactoring by providing structural overviews and detailed information.

### Test coverage assessment

**Problem:** How do test scenarios cover the software source code stack?  
**Approach:** Visualize the test scenario data combined with the (reverse-engineered) system architecture. Detect improperly covered areas at a glance. Adapt test scenarios accordingly.  
**Added value:** Facilitates test management.

## System Requirements

- PC compatible workstation (800MHz CPU, 1GB RAM, 50 GB HDD, depending on the analyzed project). A fast graphics card with at least 128 MB of RAM is recommended.
- Operating system: MS Windows NT / 2000 / XP / Vista

**Marketed in North America by:**

**Vizlogix**

ATTN: Ken Boyer  
 info@vizlogix.com

