

SansGUI®

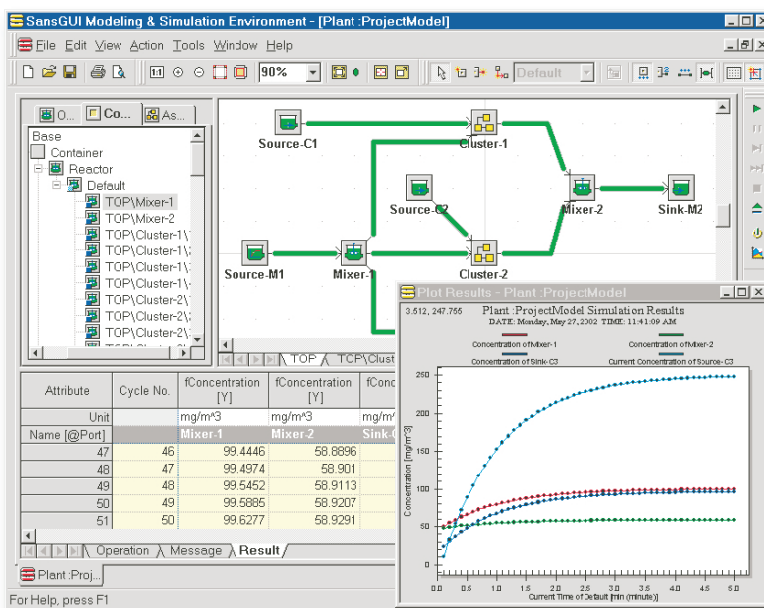
Modeling and Simulation Environment for Developing and Deploying Scientific and Engineering Simulators *without* Writing Any **Graphical User Interface** Code

Why SansGUI?

How do you develop the graphical user interface (GUI) for your scientific or engineering simulators? Do you find it hard to develop and deploy windows-based simulation software? Do you spend more time on developing the GUI front-end than the actual simulator core routines? Do you create an *ad hoc* GUI for each and every simulator? How do you let your users know what input ranges they should use? How do you resolve the unit discrepancies between the user input/output and your simulator? How do you synchronize the changing input specifications between your simulator and the GUI? Does the GUI development force you to learn yet another programming language? Do your users request GUI environments for your legacy simulation code? Is the GUI only a shell to invoke your simulator without run-time user controls? Do you outsource your GUI project to consultants but find it already obsolete as soon as it is delivered to you? Does the GUI cost you too much to develop, maintain, and support?

The SansGUI Solution

SansGUI has been developed to address all of these issues for you. Using the innovative SansGUI software, you get a modern, highly interactive, object-oriented modeling and simulation environment for your simulators *without writing a single line of GUI code*.



For simulation developers, research scientists or engineers, SansGUI is a framework for modeling physical or abstract systems. You can use SansGUI to create models with different levels of complexity, from a few simple data forms, to a single level network of parts and links, or to as complex as multiple levels of assemblies with interconnections and reference data in many scientific and engineering disciplines.

The Application Domains

The following are some sample application domains that can take advantage of SansGUI's modeling capabilities:

- Physical Systems
- Mechanical Dynamics
- Hydraulic Pipeline Systems
- Control Systems
- Communications Networks
- Electrical/Electronic Circuits
- Artificial Neural Networks
- Other Simulation Fields

SansGUI currently implements continuous, cycle-driven simulation control for highly interactive, tightly-coupled, in-process simulators. Stand-alone, external process simulators are not limited to built-in simulation flow controls. SansGUI provides a mechanism to integrate legacy simulators written in any programming language.

How SansGUI Works

One of the many innovative concepts introduced by SansGUI is the *SansGUI Data Object*, a simple and easy-to-implement protocol for simulation engines to communicate with SansGUI. One data object format and one application programming interface (API) are the only requirements for creating simulators tightly coupled with SansGUI. Such simplicity implies that it requires only minimal effort from developers to master the whole system and be ported to other platforms or translated to other protocols easily.

A simulation developer uses SansGUI to specify the classes of model building blocks in a target system as a *Schema Definition* and compile it into an *Object Library* and a set of source routines, in supported programming languages, for implementing the behavior of the classes. A simulation user installs the Object Library and the associated simulator executables from the simulation developer to create a *Project Model* with a model configuration, a certain input data set and parameters. SansGUI takes care of user interaction, hierarchical

model configuration, object sharing, data entry assistance and validation, unit conversion, dynamic memory allocation, multi-thread/process simulator execution, simulation control, result logging, displaying, plotting, and, most importantly, evolving the user's model data in case that the simulator has Schema Definition changes between versions. An external process simulator receives a textual *Model File*, in Tabular Data Blocks or XML Model Data formats, with a flattened model configuration. Dealing with hierarchical models and object-orientation is highly recommended, but not required.

SansGUI Features

Features	Benefits
SansGUI Development Environment	
Object-oriented development	Encourage well-organized data and code modules
Hierarchical class schema	Create subclasses with attribute and behavior inheritance
Multi-level class sharing	Reuse classes among simulators, workgroups or companies
Rich GUI control types	Enhance user interface for data entry and editing
Rich data validation rules	Reduce or eliminate errors from user input
Programmable data validation	Validate data entries across multiple fields
Automatic unit conversion	Guarantee correct units in data
Aggregate data types	Support collections, tables, matrices and more
Row or column major order	Meet programming language storage requirements
Dynamic memory allocation	Request data storage as needed at run-time
Single data object format	Unify the protocol for program module communication
Single API function prototype	Reduce learning time for simulator development
Source file framework	Generate simulator routine skeleton for DLL creation
Multi-programming languages	Choose familiar programming languages for development
Schema version control	Manage changes in class schema and keep track of versions
In-process simulator support	Build highly interactive simulation control
External process support	Support legacy code integration and XML file format
Robust debugging support	Trace through simulator code in question
On-line documentation	Add on-line documentation to simulators

SansGUI Run-Time Environment

Object-oriented model building	Reduce redundant data entries with multi-level assemblies
User extensible unit conversion	Choose data units according to user's preferences
Symbolic parametric values	Test variations of models with case parameters
Qualitative input assistant	Convey value semantics from the developer to the user
User overridable routines	Initialize data arrays and/or extend the simulator's behavior
Simulation run control	Start, pause, stop, step, or fast forward simulation run
Data modification and visualization	Change values and plot data dynamically during run-time
Version evolution/update	Keep Project Models in synchronization with simulators

License Options

Option	Description	Develop	Run-Time	Notes
SGpro	Professional Edition	Yes	Yes	Full features
SGstu	Student Edition	Yes*	Yes	*Limited to 10 classes, 20 attributes each
SGrun	Run-Time Edition	No	Yes	No class schema creation
SGdemo	Demonstration Only	No	Yes**	**No file saving capability



ProtoDesign, Inc.
2 Sandalwood Court
Bolingbrook, IL 60440-1573
USA

For more information:
<http://protodesign-inc.com>
Phone: (630)759-9930
E-mail: info@protodesign-inc.com

Run-Time Environment System Requirements

- Pentium® class or compatible computer system
- 16 MB physical memory, 32 MB or more recommended
- 800x600 SVGA display with 256 colors, higher recommended
- 15 MB minimum hard disk space for installation
- Microsoft® Windows® NT 4.0 (SP3 or later), Windows® 2000, XP, or Windows® 95/98/Me

Developer Environment System Requirements

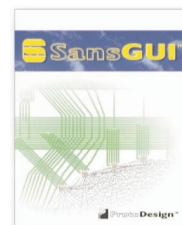
- Pentium® class or compatible computer system
- 32 MB physical memory, 64 MB or more recommended
- 1024x768 SVGA display with 256 colors, higher recommended
- 20 MB minimum hard disk space for installation
- Microsoft Windows® NT 4.0 (SP3 or later), Windows® 2000, XP, or Windows® 95/98/Me

Development Environment for In-Process Simulators

- Microsoft Visual C/C++® 6.0+
- Compaq Visual Fortran® 6.1+
- Any programming language or environment that can generate Win32 DLLs callable by Microsoft Visual C/C++®

Development Environment for External Process Simulators

- Any programming language or environment that can generate stand-alone executables



SansGUI and its logo is a registered trademark and ProtoDesign and its logo is a registered service mark of ProtoDesign, Inc. Pentium is a trademark of Intel Corporation. Microsoft, Windows, Visual C/C++ are trademarks of Microsoft Corporation. Compaq Visual Fortran is a trademark of Compaq Computer Corporation. All specifications subject to change without notice.