



The Object-Oriented Framework for Motif and CDE Developers

ViewKit provides the premier framework for developing Motif and CDE applications using C++. It allows developers to focus on application specific issues, and not spend time with generic tasks that are handled by the framework.

A Foundation for Development

ViewKit™ is a mature application framework for developing OSF/Motif™ and CDE (Common Desktop Environment) applications using C++. ViewKit was originally written by Doug Young and others at Silicon Graphics and builds on the C++ design concepts defined by Young in his book, *Object-Oriented Programming with C++ and OSF/Motif*.™ The ViewKit framework is the commercial realization of these concepts, providing a high level application architecture along with ready-to-use components. This framework is a starting point for individual application development, and more importantly, is the base for a customized, corporate development environment. This combination provides a solid foundation for object-oriented development, genuine code reuse, and an overall improvement in application quality.

Application Frameworks

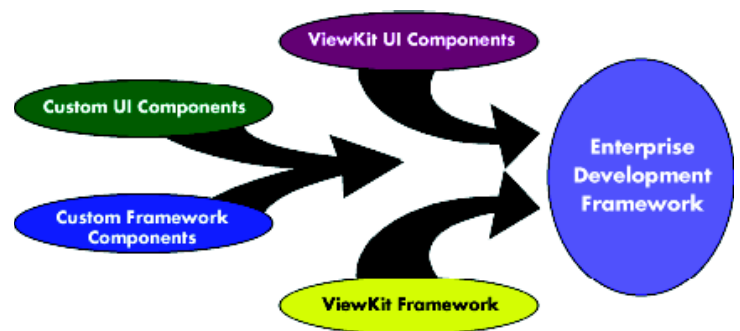
An application framework provides a pre-built application with all of its generic features already implemented. Developers use object-oriented techniques, including subclassing and inheritance, to build on the generic application and tailor components to the new application. Because they no longer need to spend time implementing basic features and functions, developers are then free to focus on the elements particular to their application. This architectural approach will save time in the initial stages of development, and just as importantly, in the ongoing maintenance and enhancement of the application.

Enterprise Development Frameworks

Project groups, departments, and corporations can use the ViewKit framework to create their own customized development frameworks. By subclassing the ViewKit core framework classes and extending the framework with domain-specific, reusable components, an engineering organization can tailor a development environment to its specific needs. An enterprise framework allows developers to focus on problems unique to the application at hand. Elements commonly used in the organization's software are handled in the framework. This also eliminates the cloning, or code copying, that makes maintenance a nightmare.

Key Features

- Provides the framework for a customized, corporate development environment
- Reduces the amount of new code written
- Enables wide-scale software reuse
- Reduces application maintenance costs
- Simplifies programming Motif with C++
- Eases the transition from C to C++





The Object-Oriented Framework for Motif and CDE Developers

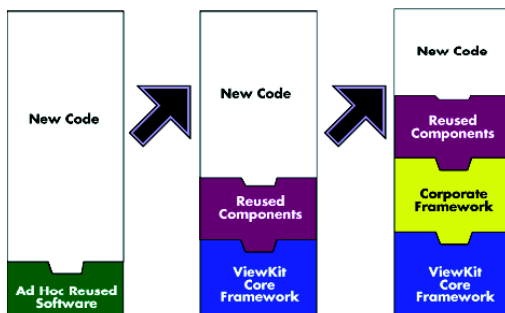
Reusable Software

Many software reuse programs fail because multiple developers create multiple objects with multiple APIs, each assuming different software bases. So, while individual objects may be reusable, as a set—across various groups and projects—they are not. ViewKit solves this problem. ViewKit supports a strong stylistic approach to building new components. This approach helps ensure that different developers will extend ViewKit in the same fashion. ViewKit offers all of the crucial ingredients developers need to make software reuse a reality. It provides an essential, common base for all development, as well as specific methods with which to extend that base through the creation of reusable components.

Reuse and Application Quality

Using an application framework and reusing software will significantly increase the quality of an application, reduce development time, and cut costs. A study at The University of Maryland found significant benefits to using MotifApp, the first generation predecessor to ViewKit. The study found that for every 10% increase in the reuse rate:

- Productivity increases by nearly 20 LOC (lines of code) per hour
- Error density drops by nearly 1 error per KLOC (thousand lines of code)
- Rework drops by 7% (in this study, that 7% translated to 6.3 man-hours)



ViewKit Components

ViewKit contains over 95 classes that comprise the core framework and subsystems. These classes include UI elements, menus, dialogs, preferences, inter-application messaging, and UNIX process control. These classes are used as is, and as the basis to extend ViewKit into a customized development framework for your company.

Core Framework Components

ViewKit provides a set of classes that are essential for all applications. The most basic of these classes is the `VkComponent` class, which defines the basic structure of all user interface components. Coupled with the `VkApp` class, which provides a complete generic application, developers will find a significant portion of their work completed before they begin.

The ViewKit framework classes also include support for application management and initialization, top-level windows, menus, and dialog management. Other basic supported services are: single and multilevel undo, interruptible tasks, and an application-level callback mechanism that allows C++ classes to dynamically register member functions to be invoked by other C++ classes.

Preference System

ViewKit supports an easy-to-use collection of classes for building preference dialogs. Rather than dealing directly with widgets, their placement, callbacks, and so on, programmers who use ViewKit can simply create groups of *preference items*. These items maintain their own states, which allows an application to simply query each item to see if it has been changed. Layout is handled automatically, and ViewKit provides the ability to apply or revert all preferences to their previous state.

Frameworks significantly reduce the amount of new code written for an application, cutting development time, increasing reliability, and reducing maintenance costs.



The Object-Oriented Framework for Motif and CDE Developers

Menu System

The ViewKit Menu components provide a simplified interface for Motif menu creation and manipulation. Among other things, they allow the developer to display, activate, and deactivate items with a single function call. The menu system also provides single and multi-level undo support.

Dialog System

The ViewKit Dialog system provides a number of pre-built dialogs. Dialogs in ViewKit are optimized for performance and memory use with automatic caching on a per application or per window basis. ViewKit also gives the developer complete control of the posting mode and positioning in multiwindow applications.

UNIX Process Control

ViewKit provides components to make it easier to use and manage the various UNIX process control functions. With these classes, developers can create “run once” server applications, application sub-processes, and individual classes to accept input and display output from another process.

ToolTalk Messaging Components

ViewKit includes components which will automatically enable an application to support ToolTalk. Sending applications create, fill in, and send a message; the ToolTalk service determines the recipients and delivers the message to the receiving applications. Receiving applications retrieve messages, examine the information in the message, and then either discard the message or perform an operation and reply with the results.

User Interface Components

ViewKit includes a number of useful user interface components. The components include items such as a text field improved to provide automatic input completion, a tabbed deck container familiar to Windows users, a directed graph component, and many others.

Convenience Utilities

ViewKit provides various utility functions and classes for convenience. These utilities include simple functions that make it easier to handle cut, copy, paste, and drag and drop operations; load resources (including automatic type conversion); register fallback resources; configure the window manager; manipulate menus; support the use of icons, and more.

BX PRO

Although a powerful tool by itself, ViewKit is also part of BX PRO™, a larger visual reuse environment that maximizes developer productivity. BX PRO consists of the following components:

- Builder Xcessory™, the industry’s most advanced user interface builder for OSF/ Motif™ and Java™
- EnhancementPak™, a set of reusable components including business graphs and sophisticated user interface controls

Windows Support

The BX PRO environment supports the delivery of applications built with ViewKit on Microsoft Windows NT platforms through the use of an optional software development kit (SDK). Applications are first built using BX PRO and ViewKit on high performance UNIX workstations. Then in many cases, all that is needed to port the developed applications to Windows NT is to re-compile and link using standard NT libraries and the libraries and tools provided by the SDK. This “single source tree” approach saves effort during both development and maintenance by eliminating the need for parallel development teams. Furthermore, developers do not have to learn a new API or become experts in Windows.



**Integrated Computer
Solutions Incorporated**

54 Middlesex Turnpike • Bedford, MA • 01730
www.ics.com • info@ics.com • 617.621.0060