

DinkClass Road Map

This document is intended to provide a starting point on the learning curve to using DinkClass. It is also intended to present the spirit of the product. One of the first things you need to keep in mind is that this is not a product which has a tutorial on Macintosh programming or object oriented design/development built into it. It may be of help in these areas, but it is not that kind of product. The documentation in some places is sparse, and will slowly improve. It represents an application frame work which is small, non-trivial, and maintainable. Attempts to make the product more complete in documentation and features is an on-going effort (why do you think we called it DinkClass anyway?)

DinkClass is a (System 7 only) class library with which to dink. Comprehensibility and ease of use are critical qualities a class library needs to support the dinkability goal set for this product. It also can't be too big. So to provide such a product, Applied Technical Software needed to figure out a way of conveying the concepts which went into the implementation of a piece of code. Provide a documentation allowing the user to follow the thought processes involved as the design and implementation the code where done.

The solution at which ATS arrived was to provide a model of the development process as a kind of reference point from which a user could infer what the developer was thinking about while developing. Along with this model a documentation "standard" was developed to complement the process model. This model need not provide the full life-cycle of a multi-person development effort, but rather it needs to provide the point of view of the individual developer.

The individual developer tends to implement features. They typically think in terms of "make the program jump through that hoop" and what it takes to make that program do it. The developer at this point has two concerns -- one is "make it work", the other is "make it maintainable." A third concern of the developer is to improve his/her performance for the next task. The process model and accompanying documentation standard presented in this package attempt to support these three concerns of the individual developer.

This type of development seems to be what the software engineering methodologies do NOT address. We shouldn't fault them too much: the major backers of software engineering methodology development

efforts (DOD, Management Consulting Firms...) are entities primarily concerned with the MANAGEMENT of large multi-person development efforts over several years. Because of this large-scale project management bias, it is typically difficult for most organizations and individuals to apply the methodologies to their efforts.

To get started with using DinkClass you need to first have a high-level understanding of how the class library is organized. DinkClass is made up of two different conceptual architectures. One is the event-handling architecture which addresses how the events decoded within the event loop get to where they need to go. A second is that an application made with DinkClass has 4 components to them (the main function, the DApplication subclass, the DDocument subclass and the DWindow subclass) all of which determine the behavior of an application.

The next document you should look at is the "Application implementation docs", then the "Scribble application doc" and the "TextEdit app docs" to get the idea of how applications can be built out of DinkClass. You should also look at the demo applications, and build from them to make your own demo applications. Then as you implement your own applications with DinkClass, look at the other documentation as needed.

Applied Technical Software hopes that you feel that DinkClass is a good product, despite its rough edges, and hopes that you will feel free to make recommendations with respect to the in-line documentation, the off-line documentation and the application framework in general.

Best Wishes,
Mark Gross
Applied Technical Software.