

MasterObjects®

PlanBoard for Omnis Studio

An amazing component entirely written in Studio

MasterObjects PlanBoard is a ready-to-use object for Omnis developers. The object allows you to easily add interactive scheduling functionality to your Omnis applications. Since its first release in 1998, many Omnis developers world wide have discovered the power of PlanBoard to enhance their applications.

Independent of data model and application interface, PlanBoard can easily be used not only to graphically display time slots but to also create, modify and delete them using a drag & drop interface. MasterObjects PlanBoard was proven very successful in a variety of scheduling and visualization applications including planning of people, machines, rooms, broadcasts, advertising, satellite traffic, medical appointments, project planning, car rental etc.

PlanBoard runs on all Microsoft Windows and Mac OS versions supported by Omnis, and will easily integrate into your existing or new applications.

Version 3.0 was rewritten from the ground up for Studio 3.x using advanced design patterns allowing unprecedented enhancement flexibility through subclassing and delegation. Please refer to our web site for more information and additional screen shots.



\$1,499 Limited-time Offer
see <http://www.masterobjects.com> for details

PlanBoard for Omnis Classic

\$1,259 Classic Developer Pack
includes window source code, sample library,
PDF Developer's Guide, 50-user Deployment License,
1 Year of Maintenance and E-Mail Support,
30-day Money-Back Guarantee

Contact

MasterObjects
Reigerskamp 393
3607 HX Maarssen
The Netherlands

Phone +31 (0) 346 285 134
Fax +31 (0) 346 285 754
info@masterobjects.com
<http://www.masterobjects.com>

MD Framework

Ultimate Object-Oriented Development in Omnis Studio

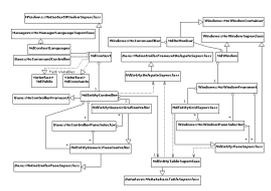
MasterObjects is developing a next-generation framework for Omnis Studio. It is ideally suited for rapid development of Client-Server applications with complex nested master-detail windows.



We are working closely with several Omnis developers to build "MD Framework for Omnis Studio™". It is due for commercial release during the second half of 2002.

What's so special about the MD Framework?

The MD Framework™ was designed from the ground up and is completely free of legacy code. After studying successful object-oriented frameworks written in Java including Apple's Enterprise Objects Framework™, we created the object model using the Unified Modeling Language.



These are just some of the goals of MD framework:

- Provide the highest level of reuse possible. Abstract superclasses add functionality missing from Omnis Studio's built-in classes, while still taking advantage of Omnis' strengths and remaining fully compatible with existing Omnis code and even other Omnis frameworks. Subwindows are nested recursively into "framesets", allowing developers to modify window layout at the flip of a single attribute.
- Specify business rules in the model and have the user interface automatically adjust itself accordingly. MD Framework adds XML-formatted business rules to Omnis SQL classes. Rules can easily be replicated from a central database. For example, if the model defines a field to be "mandatory", any view that displays the field uses a corresponding platform dependent field style.
- Avoiding code generation. MD Framework allows you to modify database rules or adjust parts of the UI without needing to rebuild any code. The framework dynamically adjusts itself and your applications remain fully compatible with future enhancements.
- Be completely language-independent without forcing developers to store their language strings in a framework-specific way. Model-dependent strings (such as field labels) are defined where they belong: in the model. Any views and user messages automatically use translated strings, allowing live language switching.
- Don't force a "look" on the developer. By strictly applying the Model-View-Controller design pattern, developers can easily subclass or replace parts of framework views for seamless integration into the rest of their application.
- Adhere to user interface guidelines. MD Framework automatically adjusts spacing, background colors, and even the order of "Cancel" and "OK" buttons for correct positioning on Windows and Mac.
- Allow for easy version control and deployment. The framework uses a Class Broker allowing classes to be stored in any library, resolving their location at runtime. The framework allows classes to be subdivided and managed in any number of sub libraries. At deployment, classes can be moved into a single library without additional recoding.