

# Руководство пользователя утилиты Pro2M

Pro2M v1.7

Copyright (c) 2001-2004 Швец Дмитрий [subml@mail.ru](mailto:subml@mail.ru)

09.02.2004

## Введение

Графический пользовательский интерфейс транслятора O2M (рис. 1.1) разработан для упрощения работы с транслятором и проектами O2M.

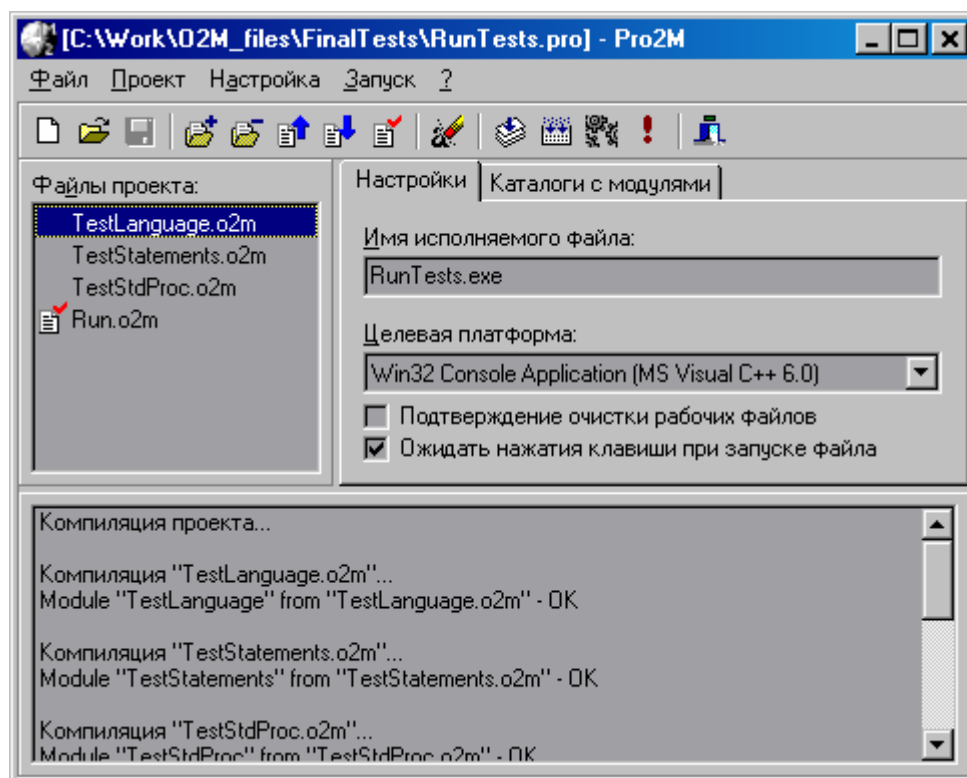


Рис. 1.1. Графический пользовательский интерфейс транслятора O2M

Графический пользовательский интерфейс выполнен в виде приложения Pro2M, предназначенного для работы в операционных системах Windows 95/98/NT/2000/XP. Приложение Pro2M обеспечивает функции настройки и компиляции проектов O2M. Поддерживается работа с проектами O2M (\*.pro), запуск компилятора O2M, линковщика Link2M, утилиты Make2M, внешнего компилятора C++ и запуск полученного в результате компиляции исполняемого файла на выполнение. В качестве внешнего компилятора могут использоваться Microsoft Visual C++ 6.0, MinGW 2.95 (использование данных компиляторов поддерживается по умолчанию) или другой ANSI C++ совместимый компилятор (требуется дополнительные настройки).

## Панель инструментов

Панель инструментов содержит следующие кнопки:

- 1 *Создать новый проект* – создание нового пустого проекта O2M.
- 2 *Открыть проект* – открытие существующего файла проекта O2M.
- 3 *Сохранить проект* – сохранение текущего проекта
- 4 *Добавить файл в проект* – добавление файла с исходными текстами O2M в текущий проект.
- 5 *Убрать файл из проекта* – убрать выделенный файл из проекта.

- 6 *Повысить приоритет файла* – повысить приоритет выделенного файла в списке файлов проекта. Выделенный файл будет перемещен в списке на 1 позицию вверх.
- 7 *Понизить приоритет файла* – понизить приоритет выделенного файла в списке файлов проекта. Выделенный файл будет перемещен в списке на 1 позицию вниз.
- 8 *Назначить главный файл* – назначение выделенного файла в списке файлов проекта главным файлом проекта.
- 9 *Очистить проект* – удаление временных файлов проекта, возникающих при трансляции.
- 10 *Компиляция выделенного файла* – компиляция выделенного файла в списке файлов проекта.
- 11 *Компиляция проекта* – компиляция всех файлов проекта в порядке их перечисления в списке.
- 12 *Сборка исполняемого файла* – запуск утилиты nmake из комплекта Microsoft Visual C++ 6.0 или утилиты make из MinGW для получения исполняемого файла.
- 13 *Запуск исполняемого файла* – запуск исполняемого файла, полученного в результате компиляции проекта O2M.
- 14 *Выход* – выход из программы.

## **Поля ввода и информационные поля**

Поля ввода и информационные поля располагаются на трех панелях, которые разделены сплиттерами, что позволяет изменять размеры панелей при помощи мыши. Pro2M содержит следующие информационные поля и поля ввода:

- 1 *Файлы проекта* – список файлов текущего проекта O2M. Порядок следования файлов в списке определяет порядок их компиляции при компиляции проекта. Главный файл проекта в списке помечен при помощи специальной иконки. Главный файл проекта должен импортировать один из модулей O2M, зависящих от операционной системы и позволяющих сгенерировать исполняемый файл. К таким модулям относятся Console.dfn, Linux.dfn, Win.dfn. Данные модули содержат точку входа в программу, что позволяет компилятору C++ сгенерировать исполняемый файл.
- 2 *Настройки проекта* – панель содержит две вкладки – «*Настройки*» и «*Каталоги с модулями*»
  - 2.1 *Вкладка «Настройки»* - содержит следующие элементы управления:
    - 2.1.1 *Имя исполняемого файла проекта* – имя исполняемого файла, если имя файла не задано, имя генерируемого исполняемого файла будет совпадать с названием проекта.
    - 2.1.2 *Целевая платформа* – компилятор, в расчете на который будет генерироваться make-файл (например, консольное приложение Windows, оконное приложение Windows, консольное приложение MinGW). В будущем возможно расширение списка типов приложений и непосредственно поддерживаемых компиляторов. Данная информация сохраняется в файле проекта, и используется при генерации make-файла, необходимого для компиляции текстов C++. Целевая платформа зависит от того, какой из модулей, обеспечивающих генерацию исполняемого файла, был выбран (см. *Главный файл проекта*).
    - 2.1.3 *Подтверждение очистки рабочих файлов* – установка данного флажка приведет к запросу на подтверждение при попытке очистки временных файлов проекта (\*.2mk, \*.2ml, \*.cpp, \*.h, \*.bat, и т.п.).
    - 2.1.4 *Ожидать нажатие клавиши при запуске файла* - установка данного флажка приведет к ожиданию дополнительного нажатия клавиши при запуске сгенерированного исполняемого файла на исполнение.
  - 3 *Вкладка «Каталоги с модулями»* - содержит список путей к каталогам, содержащим файлы определений O2M (\*.dfn) и набор кнопок для редактирования списка путей. Файлы определений содержат описания объектов (типов, переменных, констант), экспортируемых модулем. После инсталляции O2M в системе должен иметься каталог Import, содержащий основные модули O2M, необходимые для разработки и компиляции программ O2M. Файлы определений также создаются при компиляции исходных файлов O2M. Для подключения к проекту модуля необходимо иметь собственно файл определений (\*.dfn) и реализацию объектов, описанных в данном файле. Реализация

может предоставляться как в виде файлов с исходными текстами O2M, так и в виде файлов с исходными текстами C++. Возможно формирование собственных библиотек модулей.

- 3.1 *Добавить каталог с модулями в список* – добавление каталога с файлом определений O2M в список каталогов с модулями.
- 3.2 *Убрать выделенный каталог с модулями* – убрать выделенный каталог из списка каталогов с модулями.
- 3.3 *Повысить приоритет каталога* – повысить приоритет выделенного каталога в списке каталогов с модулями. Выделенный каталог будет перемещен в списке на 1 позицию вверх.
- 3.4 *Понизить приоритет каталога* – понизить приоритет выделенного каталога в списке каталогов с модулями. Выделенный каталог будет перемещен в списке на 1 позицию вниз.
- 4 *Лог* – список сообщений, выдаваемых различными утилитами в процессе работы.

## **Настройка инструментальных средств**

Окно настройки инструментальных средств может быть вызвано через главное меню – “Настройка” – “Инструментальные средства...”. Данное окно позволяет настроить работу с утилитами O2M, Link2M, Make2M, NMake, Make. Окно содержит следующие поля ввода:

- 1.1 *Путь к компилятору O2M* – путь к компилятору O2M.
- 1.2 *Путь к компоновщику Link2M* – путь к компоновщику Link2M
- 1.3 *Путь к Make2M* – путь к утилите Make2M.
- 1.4 *Файл скрипта для запуска C++* - Название файла скрипта, в который утилитой Make2M будут записаны параметры вызова внешнего компилятора C++. Имя этого файла должно использоваться в командных строках для вызова утилит NMake и Make.
- 1.5 *Размер табуляции* – количество пробелов, соответствующее одному символу табуляции (символ с кодом 09). Данный параметр используется компилятором при выдаче сообщений об ошибках. Количество пробелов не может быть меньше нуля.
- 1.6 *Строка для запуска NMake* – командная строка для запуска утилиты NMake, используется при создании консольного и графического приложений Windows.
- 1.7 *Строка для запуска Make* – командная строка для запуска утилиты Make, используется при создании консольного приложения при помощи GNU C++.
- 1.8 *Перекодировать для Make названия файлов из Win1251 в OEM Cyrillic* – данный флажок включает перекодировку путей и названий файлов в утилите Make2M, данная функция может потребоваться при использовании MinGW 2.95, если подключаемые модули располагаются в каталогах, содержащих в своих названиях русские буквы.