Публичный

CM-LYNX 2.10.00

Графическая оболочка
Руководство пользователя

Версия	**2.6.0**
Дата версии	**24.09.2026**
Имя файла	**ML_GUI_Руководство_пользователя_R_V2.6.0**

ИСТОРИЯ ИЗМЕНЕНИЙ

Версия	Дата	Автор	Описание
2.5.0	24.09.2026	Новосёлов А.В.	Подготовка к поставке 2.10.00. Внесены уточнения в п. 6.12.3.1.
2.5.0	09.06.2025	Кузин И.А.	Подготовка к поставке 2.09.00. Внесены уточнения в п. 6.11.7, 6.11.8, 6.11.12.5, 6.12.1.4.2.1. Добавлен раздел 6.12.2.3.3 «Настройки организации подключаемых заголовков».
2.4.0	29.03.2024	Новосёлов А.В.	Подготовка к поставке 2.08.00. Внесены уточнения в п. 5, 5.4.2.5, 5.4.3.4, 6.2.1, 6.2.5, 6.4.1, 6.6.13, 6.10.2, 6.11, 6.11.1, 6.11.4, 6.11.7, 6.11.7.3, 6.11.8, 6.12.1.1, 6.12.2.2.2, 6.12.3 Добавлены разделы 5.4.3.9 «Вкладка «Trace Core»», 6.6.14 «Подсветка синтаксиса в ассемблерных файлах», 6.11.17 «Отладка с помощью буфера трассировки», 6.12.1.4.2.1 «Настройки редактора ассемблерных файлов», 6.12.2.4.2 «Настройки отладочных конфигураций», 6.12.2.4.7 «Настройки буфера трассировки».
2.3.1	24.10.2023	Смирнова Ю.И.	Подготовка к поставке 2.07.00. Внесены уточнения в п. 5.1.8, 5.2, 5.4.3.1, 5.4.3.6, 6.11.5, 5.1.8, 6.4.1, 6.9.2.4.1.1, 6.9.2.4.1.6.4, 5.1.1, 5.3, 6.2.1, 6.1, 6.11.1, 5.4.3.4, 6.11.12.3, 6.11.12.4, 6.11.7, 6.11.8.
2.3.0	02.10.2023	Новосёлов А.В.	Добавлен п. 6.11.16. Уточнено примечание в п.6.11.7.
2.2.0	06.04.2023	Смирнова Ю.И.	Подготовка к поставке 2.06.00. Добавлены уточнения в п. 6.11.5, 6.9.2.4.1.5.6, 5.4.3.2, 5.4.3.3, 5.4.3.5, 5.4.3.8, 6.11.1, 6.1, 6.4.5, 6.11.1.2, 6.11.12.7, 6.9.2.4.1.2, 6.9.2.4.1.4.1. Добавлены разделы 6.12.2.4.4 «Настройки приложения MACSProf», 6.11.15 «Просмотр результатов профилирования МАКС», 6.6.13 «Автоматическое переименование меток в asm-файлах».
2.1.0	14.10.2022	Смирнова Ю.И.	Подготовка к поставке 2.05.00. Добавлены разделы 5.4.3.8, 6.11.14. Внесены уточнения в п. 6.11.12.7, 6.2, 6.4.1. Внесены исправления по тексту документа и обновлены скриншоты в связи с переходом новую версию Eclipse.
2.0.11	20.05.2022	Токарева Г.И.	Доработка ранее внесенных изменений.
2.0.10	16.05.2022	Токарева Г.И.	Внесены изменения в п. 6.2, 6.2.2, 6.2.3, 6.2.5. Добавлено описание примера кода для 1967VN044 процессора.
2.0.9	24.04.2022	Смирнова Ю.И.	Подготовка к поставке 2.04.00.
2.0.8	24.01.2022	Колесникова Т.В.	Внесены уточнения в таблицы 6, 9, 10, 12, 17, 24, 44, 46, 56, 57, 58, 67, 70, 91, 103, 132, 134, 143, п.6.2.2, п. 6.11.6, п. 6.11.12. Исправлены опечатки в таблице 3, п. 5.4.2.2, п. 6.2.1, п. 6.2.5, п. 6.4.5.1. Обновлены скриншоты рис. 1, 9, 12, 13, 35, 37, 46, 83, 119, 262.
2.0.7	15.11.2021	Новосёлов А.В.	Исправления опечаток в таблице 58, п. 6.2.5, 6.9.2.4.1.7, 6.11.1.4.
2.0.6	20.10.2021	Новосёлов А.В.	Подготовка к поставке 2.03.00.

Версия	Дата	Автор	Описание
			Добавлены разделы: 6.9.2.4.1.2 - Оборудование, 6.9.2.4.1.3 - Окружение, 6.9.2.4.1.7 - Компоновщик библиотек, 6.11.1.4 - Соответствие между конфигурациями отладки и проектами, 6.11.12.7 - Построение графиков, 6.12.3 - Настройки запуска и отладки. Добавлен перезапуск отладки в п. 6.11.3. Добавлен «Instruction Stepping» в п. 6.11.1, 6.11.5. Актуализирован п. 5.4.3. Внесены уточнения в п. 1, 2, 5.2, 5.3, 6.9.2.4.1.5.1, 6.9.2.4.1.5.6, 6.9.2.4.1.6.3, 6.9.2.4.1.8, 6.9.3.3, 6.11.5, 6.11.9-6.11.11, 6.11.12.2, 6.12.2.6, Таблицу 39.
2.0.5	28.05.2021	Смирнова Ю.И.	Внесены небольшие исправления в п. 5.4.3.4.
2.0.4	28.05.2021	Смирнова Ю.И.	Внесены уточнения в п. 5.4.3.3 и п. 5.4.3.4.
2.0.3	19.03.2021	Смирнова Ю.И.	Подготовка к поставке 2.02.00. Скорректированы разделы 6.11.7, 6.11.8.
2.0.2	01.02.2021	Смирнова Ю.И.	Подготовка к поставке 2.01.02.
2.0.1	24.11.2020	Смирнова Ю.И.	Скорректировано описание функционала работы с WP.
2.0.0	11.11.2020	Смирнова Ю.И.	Подготовка к поставке 2.01.01. Заменены некоторые скриншоты. Флаг «Align arrays by quad word» включен по умолчанию.
1.10.0	21.09.2020	Смирнова Ю.И.	Подготовка документа к поставке версии 2.01.00. Внесены уточнения в п. 5.4.3.7.
1.9.2	25.06.2020	Стрельцова О.В.	Обновление документа по замечаниям. Правка номеров таблиц.
1.9.1	22.06.2020	Стрельцова О.В.	Обновление документа по замечаниям.
1.9.0	18.06.2020	Стрельцова О.В.	Подготовка документа к поставке версии 2.00.00. Изменение разделов 6.10.1, 6.4.5, 6.11.9. Добавлен раздел 6.5
1.8.0	23.03.2020	Стрельцова О.В.	Подготовка документа к поставке версии 1.09.00. Изменение разделов 6.9.2.4.1, 5.1.8, 6.11.8 и 6.11.9.
1.7.2	16.01.2020	Смирнова Ю.И.	Устранены опечатки по тексту
1.7.1	23.12.2019	Смирнова Ю.И.	Внесены корректировки по тексту документа. Подготовка документа к поставке
1.7.0	12.09.2019	Стрельцова О.В.	Описание функционала актуализировано в соответствии с версией CM-LYNX 1.08.00. Изменены разделы: 5.4.3.6 Вкладка «Disassembly» 6.2.1 Шаг 1. Создание проекта 6.4.1Создание проекта 6.4.5 Импорт существующего проекта 6.9.2.4.1 Настройки компонентов тулчейн 6.11.9 Просмотр и редактирование значений регистров Добавлены разделы: 6.11.8 Установка точки наблюдения
1.6.0	05.09.2019	Смирнова Ю.И.	Существенно переработана структура документа и расширено его содержание. Описание функционала актуализировано в соответствии с версией CM-LYNX 1.07.00
1.5.0	19.04.2019	Смирнова Ю.И.	Добавлено уточнение в описание в п. 7.5.2.8. Добавлены уточнения процедуры изменения значений переменных в окне «Variables» и «Expressions» В п. 7.4.2.4.1.3.7 добавлено описание новой настройки

Версия	Дата	Автор	Описание
1.4.1	15.01.2019	Смирнова Ю.И.	Исключено описание функционала профилировщика и визуализатора.
1.4.0	11.01.2019	Смирнова Ю.И.	Добавлено уточнение в п. 7.5.2.8 Уточнены требования к поддерживаемым ОС Подготовка документа к поставке
1.3.1	22.11.2018	Смирнова Ю.И.	Добавлено описание панели «Compiler» для компилятора Clang (см. п. 7.4.2.4.1.3)
1.3.0	13.11.2018	Смирнова Ю.И.	Уточнено описание в разделе 6.4. Добавлено описание импорта проектов (см. п. 6.4.5) и смены рабочего пространства (см. п. 6.4.12)
1.2.1	31.10.2018	Смирнова Ю.И.	Скорректирован п. 7.6.2.1
1.2.0	26.10.2018	Смирнова Ю.И.	Добавлено уточнение в раздел 7.6.2
1.1.4	25.10.2018	Стрельцова О.В.	Обновлён раздел 7.2.2.2.1. Добавлен раздел 7.6.2.1
1.1.3	23.10.2018	Погибельский Д.А.	Добавлено описание дополнительных опций компоновщика, которые могут быть использованы при конструировании файла LDF (п. 7.4.4)
1.1.2	17.10.2018	Смирнова Ю.И.	Обновлено описание создания проекта (п. 7.2)
1.1.1	25.09.2018	Смирнова Ю.И.	Подготовка документа к поставке
1.1.0	14.09.2018	Стрельцова О.В.	Исправленная версия
1.0.0	24.08.2018	Стрельцова О.В.	Первая версия

ОГЛАВЛЕНИЕ

1. Введение 8

2. Термины и сокращения 9

3. Назначение программы 11

3.1. Функциональное назначение программы 11

3.2. Эксплуатационное назначение программы 11

4. Условия выполнения программы 12

4.1. Минимальный состав аппаратных средств 12

4.2. Минимальный состав программных средств 12

4.3. Требования к персоналу (пользователю) 12

5. Интерфейс CM-LYNX 13

5.1. Основное меню приложения 14

5.1.1. Меню «File» 14

5.1.2. Меню «Edit» 16

5.1.3. Меню «Source» 18

5.1.4. Меню «Refactor» 19

5.1.5. Меню «Navigate» 20

5.1.6. Меню «Search» 22

5.1.7. Меню «Project» 23

5.1.8. Меню «Run» 24

5.1.9. Меню «Window» 27

5.1.10. Меню «Help» 29

5.2. Панель инструментов 29

5.3. Панель «Project Explorer» 31

5.4. Рабочая область 33

5.4.1. Редактор C/C++ 33

5.4.2. Режим разработки (перспектива «C/C++») 34

5.4.3. Режим отладки (перспектива «Debug») 43

6. Выполнение программы 61

6.1. Загрузка и запуск программы 61

6.2. Начало работы 62

6.2.1. Шаг 1. Создание проекта 63

6.2.2. Шаг 2. Создание C++ файла и его редактирование 66

6.2.3. Шаг 3. Изменение конфигурации и параметров сборки 69

6.2.4. Шаг 4. Сборка проекта 71

6.2.5. Шаг 5. Отладка проекта 72

6.3. Выбор и настройка режима работы (перспективы) 87

6.3.1. Режимы работы (перспективы) 87
6.3.2. Переключение между режимами (перспективами) 89
6.3.3. Настройка набора инструментов режима (перспективы) 90
6.4. Работа с ресурсами (проектами, папками и файлами) 94
6.4.1. Создание проекта 95
6.4.2. Создание папки 103
6.4.3. Создание файла 105
6.4.4. Импорт ресурсов из файловой системы 106
6.4.5. Импорт существующего проекта 108
6.4.6. Быстрый поиск ресурсов 111
6.4.7. Копирование ресурсов 112
6.4.8. Перемещение ресурсов 112
6.4.9. Переименование ресурсов 113
6.4.10. Удаление ресурсов 114
6.4.11. Закрытие проектов 114
6.4.12. Смена рабочего пространства 115
6.5. Работа с репозиторием Git 116
6.5.1. Подключение контроля версий Git 116
6.5.2. Базовые операции меню «Team» 116
6.5.3. Вкладка «Git Repositories» 118
6.5.4. Перспектива Git 119
6.6. Работа с редакторами 123
6.6.1. C/C++ редактор 124
6.6.2. Внешние редакторы 125
6.6.3. Открытие файла на редактирование 125
6.6.4. Настройка ассоциации редактора с типом файлов 125
6.6.5. Редактирование файлов вне рабочей среды 127
6.6.6. Преднастройка редактора C/C++ 128
6.6.7. Комментирование кода 128
6.6.8. Работа с функцией завершения кода (Content Assist) 129
6.6.9. Сдвиг строчек кода вправо или влево 132
6.6.10. Переход к объявлениям C / C++ 132
6.6.11. Переименование объектов C / C++ 133
6.6.12. Переход к определению функции 134
6.6.13. Автоматическое переименование меток в ассемблерных файлах 134
6.6.14. Подсветка синтаксиса в ассемблерных файлах 136
6.7. Использование локальной истории 138
6.7.1. Сравнение текста файла с локальной историей 138
6.7.2. Замена ресурса локальной историей 139
6.7.3. Восстановление ресурса из локальной истории 140
6.7.4. Настройки локальной истории 140
6.8. Использование меток 140
6.8.1. Закладки 141

6.8.2. Задания 142
6.9. Настройка свойств проекта 144
6.9.1. Настройки ресурсов проекта 144
6.9.2. Настройка параметров сборки проекта 151
6.9.3. Настройки общих свойств проекта C/C++ 209
6.9.4. Настройки Git 219
6.10. Сборка проекта 220
6.10.1. Управление автосборкой 220
6.10.2. Запуск ручной сборки 220
6.10.3. Обработка ошибок сборки 221
6.11. Отладка 222
6.11.1. Настройка конфигурации и запуск отладки 222
6.11.2. Операции при отладке 233
6.11.3. Управление сессией отладки 234
6.11.4. Просмотр стека вызовов 235
6.11.5. Пошаговое выполнение инструкций программы 236
6.11.6. Выгрузка данных дисассемблера в текстовый файл 238
6.11.7. Установка/снятие точек останова 239
6.11.8. Установка точки наблюдения 250
6.11.9. Просмотр и редактирование значений регистров 255
6.11.10. Просмотр и редактирование значений переменных 258
6.11.11. Просмотр и редактирование значений выражений 260
6.11.12. Просмотр и редактирование содержимого памяти целевого процессора 261
6.11.13. Управление процессорами при многопроцессорной отладке 270
6.11.14. Просмотр карты памяти процессора 270
6.11.15. Просмотр результатов профилирования МАКС 275
6.11.16. Отладка с помощью нескольких сред разработки 276
6.11.17. Отладка с помощью буфера трассировки 278
6.12. Персональные настройки 279
6.12.1. Общие настройки 280
6.12.2. Настройки режима работы C/C++ 310
6.12.3. Настройки запуска и отладки 358
6.13. Завершение работы 361

7. Диагностические сообщения 362

1. Введение

Настоящий документ является руководством пользователя автоматизированного рабочего места разработки программного обеспечения для высокопроизводительных процессоров цифровой обработки сигнала серии 1967ВН0XX (далее – процессор 1967ВН0XX) при работе с интегрированной средой разработки CM-LYNX (далее - CM-LYNX или программа).

В документе рассматриваются:

- назначение и функциональные возможности графической оболочки CM-LYNX;
- графический интерфейс пользователя CM-LYNX;
- последовательность действий пользователя;
- сообщения пользователю, возникающие в процессе работы с программой.

2. Термины и сокращения

В документе применяются следующие термины и сокращения:

Таблица 1. Термины и сокращения

Сокращение	Описание
ANSI	American national standards institute – Американский национальный институт стандартов. ANSI-кодировка используется для не-юникодных Windows-программ.
ASCII	American standard code for information interchange – название таблицы (кодировки, набора), в которой некоторым распространённым печатным и непечатным символам сопоставлены числовые коды.
CDT	C/C++ Development Tools – модуль среды разработки Eclipse для языков C и C++.
CM-LYNX	Интегрированная среда разработки, разработанная на основе Eclipse.
CVS	Concurrent Versions System – программный продукт, относящийся к разряду систем управления версиями.
DXE	Расширение исполняемых бинарных файлов, генерируемых компоновщиком.
EBCDIC	Extended Binary Coded Decimal Interchange Code – стандартный восьмибитный код, разработанный корпорацией IBM для использования на мейнфреймах собственного производства и совместимых с ними.
GDB	GNU debugger (отладчик GNU).
HAL	Hardware Abstraction Layer – набор программных функций, предназначенный для взаимодействия с периферийными устройствами процессора.
IDE	Integrated Development Environment – интегрированная среда разработки.
IEEE	Institute of Electrical and Electronics Engineers – международная некоммерческая ассоциация специалистов в области техники, разработчик стандартов по радиоэлектронике, электротехнике и аппаратному обеспечению вычислительных систем и сетей.
JPEG	Joint Photographic Experts Group – растровый графический формат, применяемый для хранения фотоизображений и подобных им изображений.
JTAG	Joint Test Action Group – специализированный аппаратный интерфейс на базе стандарта IEEE 1149.1
LDF	Linker Description File –файл сценария для компоновщика.
LLVM	Low Level Virtual Machine – универсальная система анализа, трансформации и оптимизации программ, реализующая виртуальную машину с RISC-подобными инструкциями.
LST	Listing files – файлы листинга ассемблера.
MAP	Файл с картой памяти, содержащий информацию о памяти и символах для исполняемого файла(ов). Предоставляет список абсолютных адресов всех символов.
OLE	Object Linking and Embedding – технология связывания и внедрения объектов в другие документы и объекты, разработанная корпорацией Майкрософт.

Сокращение	Описание
RTTI	Run-time type information специальный механизм определения типа объекта во время выполнения.
TCP	Transmission Control Protocol – протокол передачи данных интернета, предназначенный для управления передачей данных.
Workbench	Обозначение окон приложения CM-LYNX, предоставляющих инструментарий для разработки.
XML	Язык структурирования некоторых генерируемых файлов.
ОЗУ	Оперативное запоминающее устройство — энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором.
ОС	Операционная система.
ПК	Персональный компьютер.
ПО	Программное обеспечение.
Тулчейн (toolchain)	Набор программ (компилятор, ассемблер, компоновщик, библиотеки), необходимых для компиляции и генерации выполняемого кода из исходных текстов.

3. Назначение программы

3.1. Функциональное назначение программы

Графическая оболочка CM-LYNX обеспечивает следующую функциональность:

- Управление проектами и конфигурациями разрабатываемого программного обеспечения (в том числе интеграция с Git);
- Синтаксически ориентированное редактирование исходных текстов на языках C++, C и ассемблер для 1967BH0XX;
- Автоматическая проверка синтаксической корректности исходных текстов в процессе редактирования;
- Отображение подсказок, автоматическое дополнение и рефакторинг на основе структурного анализа исходного текста;
- Поддержка процесса сборки проекта с использованием утилит командной строки (компилятор C++, ассемблер, компоновщик и др.);
- Структурированное отображение диагностических сообщений по результатам сборки проекта (ошибок, предупреждений) с привязкой к позициям в исходном тексте;
- Обеспечение процесса интерактивной (высокоуровневой и низкоуровневой) отладки ПО;
- Поддержка режимов программной симуляции и отладки с использованием аппаратного отладчика;
- Отображение и редактирование данных, содержащихся в памяти и регистрах общего и специального назначения 1967BH0XX.

3.2. Эксплуатационное назначение программы

Графическая оболочка CM-LYNX базируется на функциональности, предоставляемой Eclipse CDT, и предназначена для поддержки процесса разработки ПО для высокопроизводительных процессоров цифровой обработки сигнала серии 1967BH0XX.

4. Условия выполнения программы

4.1. Минимальный состав аппаратных средств

Для работы с графической оболочкой CM-LYNX необходим ПК, удовлетворяющий следующим требованиям:

- Совместимость с ОС Microsoft Windows 7 (64-bit)/ Windows 10 (64-bit);
- Свободное место на диске не менее 1 Гбайт;
- ОЗУ не менее 4 Гбайт.

4.2. Минимальный состав программных средств

Для работы с графической оболочкой CM-LYNX необходим следующий минимальный состав программных средств:

- ОС Microsoft Windows 7 (64-bit)/ Windows 10 (64-bit).

4.3. Требования к персоналу (пользователю)

Для работы с графической оболочкой CM-LYNX от пользователя требуются общие навыки работы с персональным компьютером, командной строкой ОС, знания в области разработки программного обеспечения для микропроцессорной техники.

5. Интерфейс CM-LYNX

После запуска приложения открывается главное окно CM-LYNX (см. Рисунок 1).

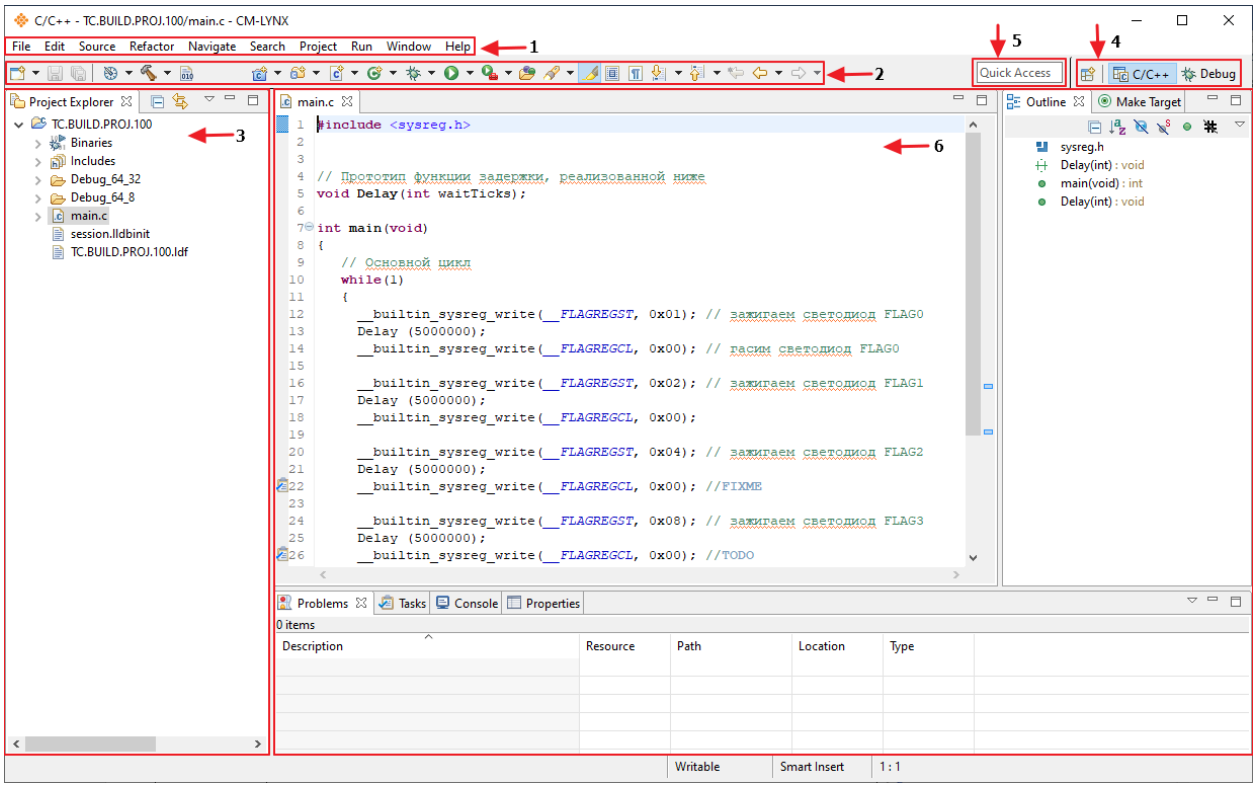


Рисунок 1. Главное окно приложения

Окно приложения делится на области:

1. Основное меню приложения – меню, содержащее базовые операции работы с приложением. Подробнее см. п. 5.1. Основное меню приложения;

2. Панели инструментов – панель, предоставляющая инструментарий для разработки, отладки, в зависимости от выбранного режима (*перспективы*). Доступные инструменты для каждого из режимов (каждой из *перспектив*) отличаются. Подробнее см. п. 5.2. Панель инструментов;

Примечание: панель инструментов может быть скрыта. Для её отображения следует выбрать в основном меню **Window → Show Toolbar.**

3. Панель «**Project Explorer**» – на панели представлена иерархия ресурсов, древовидное представление всех папок и файлов проекта. Подробнее см. п. 5.3. Панель «Project Explorer». Панель недоступна в режиме отладки (в перспективе «**Debug**»);

4. Кнопки переключения режимов (перспектив) – кнопки для выбора одного из доступных режимов: разработка (перспектива «C/C++»), отладка (перспектива «Debug») и др. Подробнее см. п. 6.3 Выбор и настройка режима работы;

5. Панель «**Quick Access**» - поле для быстрого поиска элементов интерфейса по их названию. Для доступа к элементу интерфейса достаточно начать набирать его название;

6. Рабочая область – панели и вкладки для работы в среде разработки в соответствии с выбранным режимом (перспективой), т.е. вкладки и панели для разных режимов (перспектив) отличаются. Подробнее см. п. 5.4 Рабочая область.

Внешний вид окна программы настраивается в меню «Window». Подробнее см. п.5.1.9.

5.1. Основное меню приложения

5.1.1. Меню «File»

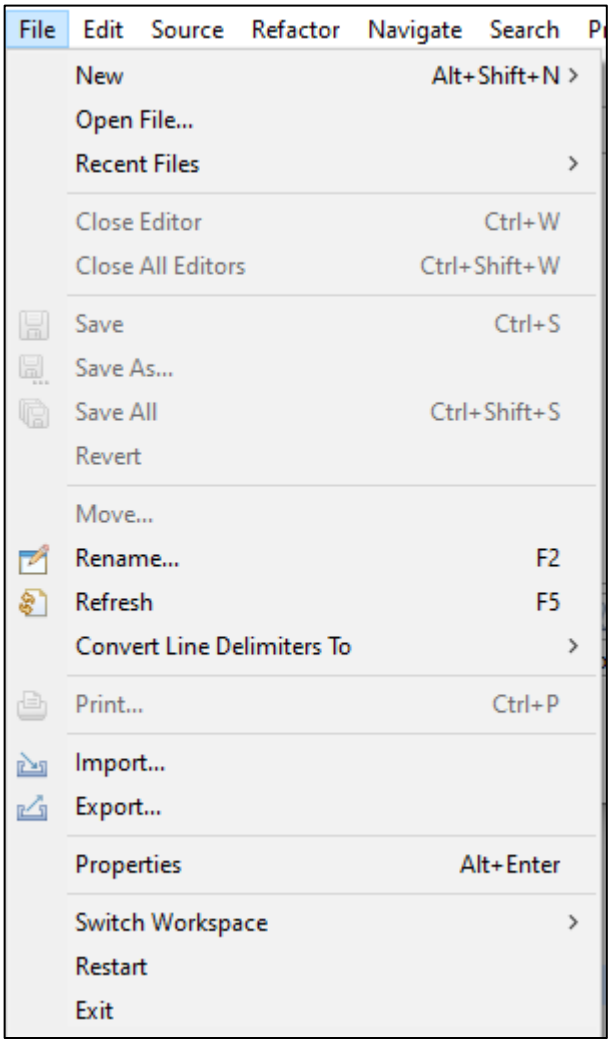


Рисунок 2. Меню «File»

Таблица 2. Команды меню «File»

Название команды	Описание функционирования	Горячие клавиши
New	Создать новый проект, файл, папку или иной объект.	Alt+Shift+N
Open File...	Открыть существующий файл.	–
Recent Files	Произвести операции с последними файлами: открыть один из последних файлов или очистить историю (**Clear History**).	
Close Editor	Закрыть текущий редактор. Если редактор содержит несохраненные данные, отобразится диалоговое окно с предложением сохранить данные.	Ctrl+W Ctrl+F4

Название команды	Описание функционирования	Горячие клавиши
Close All Editors	Закрыть все редакторы. Если редактор содержит несохраненные данные, отобразится диалоговое окно с предложением сохранить данные.	Ctrl+Shift+W Ctrl+Shift+F4
Save	Сохранить содержимое текущего редактора. Недоступно, если в редакторе нет несохраненных изменений.	Ctrl+S
Save As	Сохранить содержимое текущего редактора под новым именем.	–
Save All	Сохранить содержимое текущего редактора. Недоступно, если в редакторе нет несохраненных изменений.	Ctrl+Shift+S
Revert	Вернуть содержимое текущего редактора к предыдущей сохраненной версии файла. Недоступно, если в редакторе нет несохраненных изменений.	–
Move…	Переместить ресурс.	–
Rename…	Переименовать ресурс.	F2
Refresh	Обновить содержимое текущего элемента с локальной файловой системой. При запуске без указания конкретного места обновляет все проекты.	F5
Convert Line Delimiters to	Изменить ограничитель строки в формат Windows (по умолчанию) или UNIX.	–
Print…	Распечатать содержимое текущего редактора. Доступно, когда в редакторе есть фокус.	Ctrl+P
Import…	Открыть диалоговое окно «**Import»** и показать мастер импорта.	–
Export…	Открыть диалоговое окно «**Export»** и показать мастер экспорта.	–
Properties	Открыть страницу свойств выбранных элементов.	Alt+Enter
Switch workspace...	Перезапустить с новым рабочим пространством	–
Restart	Перезапустить приложение.	–
Exit	Выйти из CM-LYNX.	–

5.1.2. Меню «Edit»

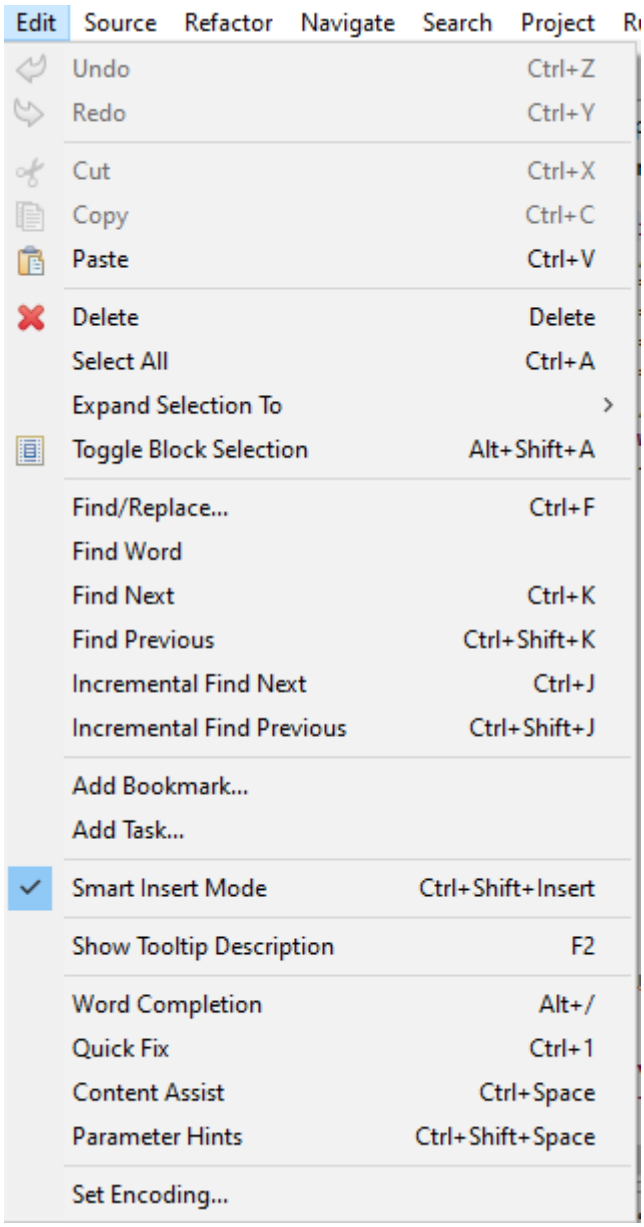


Рисунок 3. Меню «Edit»

Таблица 3. Команды меню «Edit»

Название команды	Описание функционирования	Горячие клавиши
Undo	Отменить последнее действие в редакторе.	Ctrl+Z
Redo	Вернуть отмененное действие.	Ctrl+Y
Cut	Вырезать выбранный текст или элемент в буфер обмена. Элемент не удаляется из предыдущего месторасположения, пока его не вставят в новое.	Ctrl+X
Copy	Скопировать выбранный элемент или текст в буфер обмена.	Ctrl+C
Paste	Вставляет текущую информацию в виде текста в редактор как родственный или дочерний элемент в выбранный.	Ctrl+V

Название команды	Описание функционирования	Горячие клавиши
Delete	Удалить выбранный текст или элемент.	Delete
Select All	Выбрать всё содержимое в редакторе.	Ctrl+A
Expand Selection To	Увеличить область выделенного текста: ▪ **Enclosing Element** - до завершающего элемента; ▪ **Next Element** – до следующего элемента; ▪ **Previous Element** – до предыдущего элемента ▪ **Restore Last Selection** - повторить последнее выделение.	
Find / Replace	Открыть диалоговое окно «**Find/Replace**» поиска или замены элементов. Только для редактора.	Ctrl+F
Find Word	Найти слово. Только для редактора.	–
Find Next	Найти искомое слово в редакторе далее по тексту. Только для редактора.	Ctrl+K
Find Previous	Найти искомое слово в редакторе ранее по тексту. Только для редактора.	Ctrl+Shift+K
Incremental Find Next	Запустить режим поиска без строки поиска далее по тексту, т.е. поиск выполняется по мере ввода искомого текста. Только для редактора.	Ctrl+J
Incremental Find Previous	Запустить режим поиска без строки поиска ранее по тексту, т.е. поиск выполняется по мере ввода искомого текста. Только для редактора.	Ctrl+Shift+J
Add Bookmark…	Добавить закладку в месте выбранного текста или элемента.	–
Add Task…	Добавить пользователю определенное задание к выбранному тексту или элементу.	–
Smart Insert Mode	Включить/отключить режим умной вставки.	Ctrl+Shift+Insert
Show Tooltip Description	Показать всплывающие подсказки, где применимо.	F2
Word Completion	Закончить текущее слово в редакторе.	Alt+/
Quick fix	Вызвать диалоговое окно быстрого устранения выбранной ошибки/предупреждения.	Ctrl+1
Content Assist	Включить помощник подбора содержимого.	Ctrl+Space
Parameters Hints	Включить подсказки работы с параметрами.	Ctrl+Shift+Space
Set Encoding...	Переключить кодировку содержимого показываемого текста.	–

5.1.3. Меню «Source»

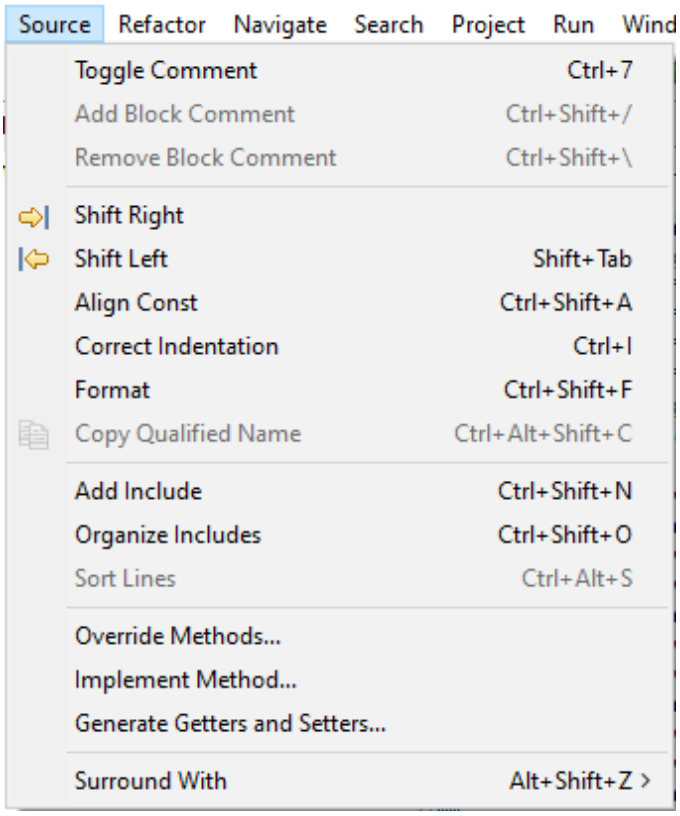


Рисунок 4. Меню «Source»

Таблица 4. Команды меню «Source»

Название команды	Описание функционирования	Горячие клавиши
Toggle Comment	Добавить или убрать одиночные строки, стили C++-, отметки комментариев на каждой строке или в строке с установленным курсором, если ничего не выделено.	Ctrl+7
Add Block Comment	Добавить блок комментариев вокруг выбранного места.	Ctrl+Shift+/
Remove Block Comment	Убрать отметку комментария вокруг блока комментариев в выбранном месте.	Ctrl+Shift+\
Shift Right	Увеличить отступы на каждой выбранной строке или на строке с курсором.	_
Shift Left	Уменьшить отступы на каждой выбранной строке или на строке с курсором.	Shift+Tab
Align Const	Выровнять константы.	Ctrl+Shift+A
Correct Indentation	Применить простые правила отступа к выбранному месту или к строке с курсором.	Ctrl+I
Format	Применить полные свойства ко всему содержимому активного редактора или к выбранному месту.	Ctrl+Shift+F
Copy Qualified Name	Копировать полное имя.	Ctrl+Alt+Shift+C

Название команды	Описание функционирования	Горячие клавиши
Add Include	Добавить оператор включения и, при необходимости, утвердить выбранное имя.	Ctrl+Shift+N
Organize Includes	Добавить требуемые включения для выбранного файла.	–
Sort Lines	Упорядочить выбранные строки.	Ctrl+Alt+S
Override Methods…	Открыть диалоговое окно «**Override Method**» для переопределения метода.	
Implement Method...	Открыть диалоговое окно «**Implement Method**» для описания метода.	–
Generate Getters and Setters...	Открыть диалоговое окно «**Generate Getters and Setters**», которое позволяет создавать получающий и устанавливающий методы для полей текущего текста. Доступно на полях или выбранном тексте.	–
Surround With	Окружить выбранные утверждения шаблоном кода.	Alt+Shift+Z

5.1.4. Меню «Refactor»

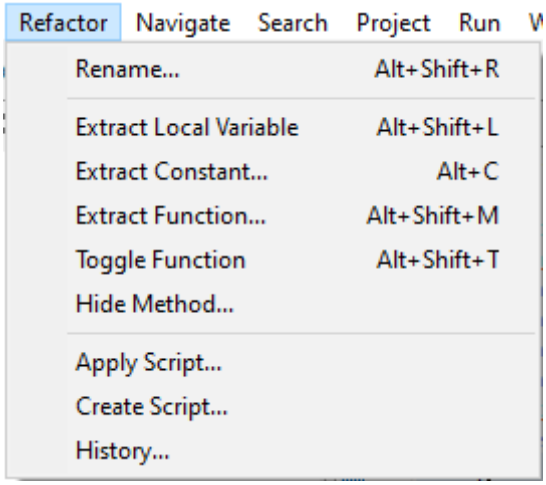


Рисунок 5. Меню «Refactor»

Таблица 5. Команды меню «Refactor»

Название команды	Описание функционирования	Горячие клавиши
Rename...	Переименовать выбранный объект (переменную, метод и т.д.) и распространить изменения на другие файлы в проекте.	Alt+Shift+R
Extract Local Variable	Извлечь локальную переменную.	Alt+Shift+L
Extract Constant…	Извлечь константу.	Alt+C
Extract Function…	Извлечь функцию.	Alt+Shift+M
Toggle Function	Переключить функцию.	Alt+Shift+T

Название команды	Описание функционирования	Горячие клавиши
Hide Method…	Скрыть метод.	–
Apply Script...	Применить сценарий (скрипт).	–
Create Script…	Создать сценарий (скрипт).	–
History…	Открыть журнал реорганизации кода.	–

5.1.5. Меню «Navigate»

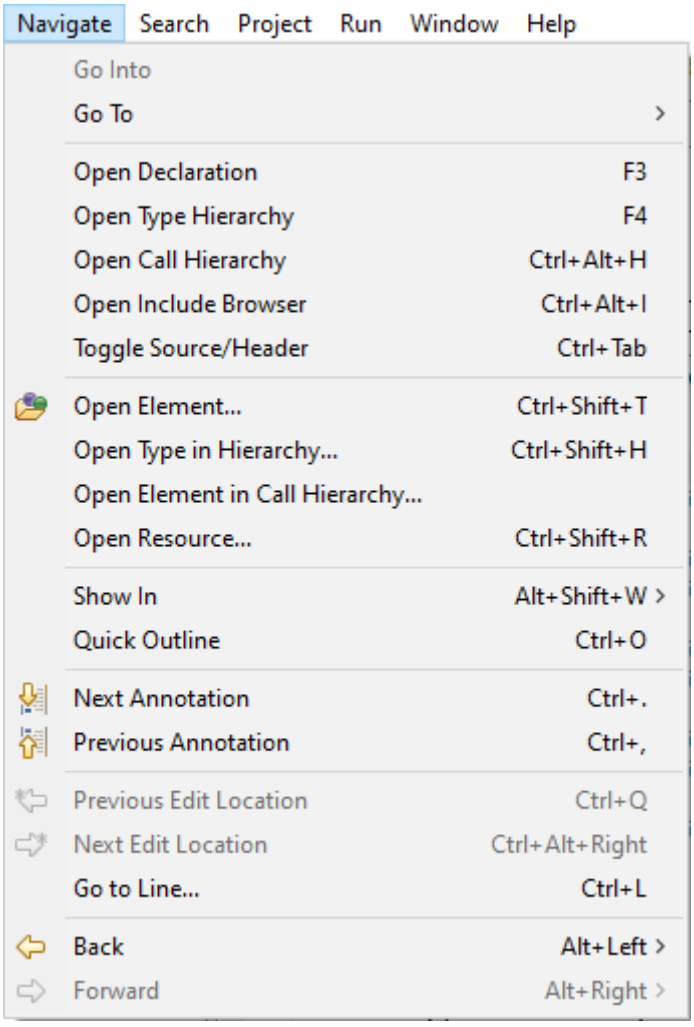


Рисунок 6. Меню «Navigate»

Таблица 6. Команды меню «Navigate»

Название команды	Описание функционирования	Горячие клавиши
Go Into	Обновляет активное представление, отображая вложенную структуру текущего элемента. При этом выбранный объект иерархии выводится в качестве корневого.	–

Название команды	Описание функционирования	Горячие клавиши
Go To	Изменяет текущее представление следующим образом: ▪ **Back** – осуществляет переход к предыдущему представлению. Например, если с помощью **Go Into** перейти внутрь объекта иерархии, то последующая команда **Back** вернет к тому представлению, из которого команда **Go Into** была вызвана (аналогично кнопке **Back** в браузере); ▪ **Forward** – осуществляет переход к иерархии, следующей за текущей. Например, если была выполнена команда **Back**, то последующая команда **Forward** вернет к тому виду иерархии, из которого команда **Back** была активирована (аналогично кнопке **Forward** в браузере); ▪ **Up one level** – отображает иерархию родительского элемента текущего объекта; ▪ **Resource** …– открывает окно **Go To Resource** для быстрого перехода к нужному ресурсу; ▪ **Next Member** – осущетсвляет переход к следующему члену; ▪ **Previous Member –** – осущетсвляет переход к предыдущему члену; ▪ **Go to Mathing Bracket** –осущетсвляет переход к парной скобке; ▪ **Next Bookmark** –осуществляет переход к следующей метке.	Ctrl+Shift+Up Ctrl+Shift+Down Ctrl+Shift+P
Open Declaration	Открыть вкладку «**Declaration»** для выбранного объекта.	F3
Open Type Hierarchy	Открыть вкладку «**Type Hierarchy»** для выбранного объекта.	F4
Open Call Hierarchy	Открыть вкладку «**Call Hierarchy»** для выбранного объекта.	Ctrl+Alt+H
Open Include Browser	Открыть вкладку «**Include Browser»** для выбранного объекта.	Ctrl+Alt+I
Toggle Source/Header	Переключиться между исходниками/заголовками.	Ctrl+Tab
Open Element...	Вызвать диалоговое окно «**Open Element»** для открытия элемента в редакторе. Это диалоговое окно показывает все типы, существующие в рабочем пространстве.	Ctrl+Shift+T
Open Type in Hierarchy…	Открыть вкладку «**Type in Hierarchy»** для выбранного объекта.	Ctrl+Shift+H
Open Element in Call Hierarchy…	Открыть вкладку «**Element in Call Hierarchy»** для выбранного объекта.	_
Open Resource	Вызвать диалоговое окно для выбора любого ресурса в рабочем пространстве для открытия в редакторе.	Ctrl+Shift+R
Show In	Найти и выбрать указанный ресурс в другом виде. Если редактор активен, эти команды используются для выбора ресурса, редактируемого в другом виде.	Ctrl+Shift+W
Quick Outline	Получить краткую структуру кода в редакторе.	Ctrl+O
Next Annotation	Перейти на следующий объект относительно текущего фокуса в окне. Например, во время поиска эта запись становится следующим совпадением**.**	Ctrl+.

Название команды	Описание функционирования	Горячие клавиши
Previous Annotation	Перейти на предыдущий объект относительно текущего фокуса в окне. Например, во время поиска эта запись становится предыдущим совпадением.	Ctrl+,
Previous Edit Location	Переместить курсор на строку, которая содержит последнее изменение. Только для редактора.	Ctrl+Q
Next Edit Location	Переместить курсор на строку, которая содержит следующее изменение. Только для редактора.	Ctrl+Alt+Right
Go to Line	Перейти к строке с указанным номером. Только для редактора.	Ctrl+L
Back	Переместить фокус на предыдущий файл. Только для редактора.	Alt+Left
Forward	Вернуть фокус с предыдущего файла. Только для редактора.	Alt+Right

5.1.6. Меню «Search»

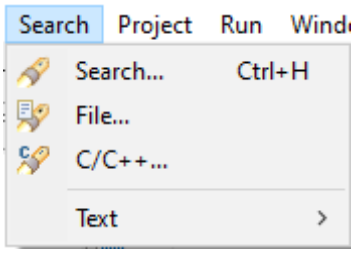


Рисунок 7. Меню «Search»

Таблица 7. Команды меню «Search»

Название команды	Описание функционирования	Горячие клавиши
Search...	Открыть окно поиска для текущего редактора.	Ctrl + H
File...	Открыть окно поиска на странице поиска файла.	–
C/C++...	Открыть окно поиска на странице поиска C/C++.	–
Text	Открыть окно поиска для полнотекстового поиска в данной области: ▪ **Workspace** - рабочее пространство; ▪ **Project** – проект; ▪ **File** – фал; ▪ **Working Set** – выбранный рабочий набор.	Ctrl+Alt+G (для поиска по рабочему пространству)

5.1.7. Меню «Project»

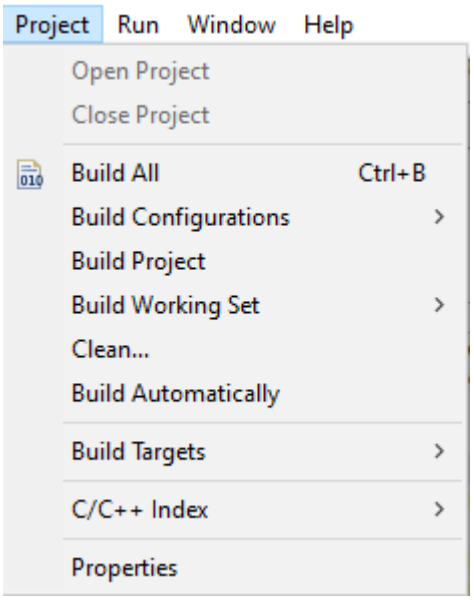


Рисунок 8. Меню «Project»

Таблица 8. Команды меню «Project»

Название команды	Описание функционирования	Горячие клавиши
Open Project	Вызвать диалоговое окно для выбора и открытия закрытого проекта.	–
Close Project	Закрыть выбранный проект.	–
Build All	Собрать все проекты в рабочем пространстве. Это полная сборка всех файлов.	Ctrl+B
Build Configurations	Выбрать операцию по работе параметров сборки: ▪ **Set Active** - установить активную конфигурацию сборки; ▪ **Manage...**- управлять (добавить, удалить, переименовать) конфигурациями сборки; ▪ **Build by Working Set** – собрать из выбранного рабочего набора; ▪ **Set Active by Working Set** - установить активную конфигурацию из рабочего набора; ▪ **Manage Working Sets...** - управлять (добавить, удалить, переименовать) рабочими наборами.	–
Build Project	Собрать выбранный проект. Эта полная сборка, участвуют все файлы.	–
Build Working Set	Выбрать текущий и собрать текущий рабочий набор.	–
Clean…	Провести очистку, определенную в создателе файлов.	–
Build Automatically	Выполнить сборку при проверке, когда бы не был сохранен файл в проекте. Эту функцию следует отключить в больших проектах.	–

Название команды	Описание функционирования	Горячие клавиши
Build Targets	Выполнить операцию с целями сборки (Build Targets): ▪ **Create…** – вызвать диалог «Create Build Target» для создания цели сборки (Build Targets); ▪ **Build…**- вызвать диалог «Build Targets» для запуска сборки выбранной цели; ▪ **Rebuild Last Target** – собрать повторно последнюю цель.	Shift+F9 (Build Targets-> Build…)
C/C++ Index	Выполнить операцию по работе с индексами: ▪ **Rebuild** – пересобрать индексы; ▪ **Freshen All Files** – ; ▪ **Update with Modified Files** – ; ▪ **Re-solve Unresolved Includes - ;** ▪ **Search for Unresilved Includes** - ; ▪ **Create Parser Log File** - .	–
Properties	Открыть диалоговое окно свойств, где можно посмотреть свойства ресурсов Info, External Tools Builders, C/C++ Build (managed only) File Types, Indexer options, C/C++ Make Project (standard only), C/C++ Project Paths (standard only), Include Paths and Symbols (standard only) и Project References.	–

5.1.8. Меню «Run»

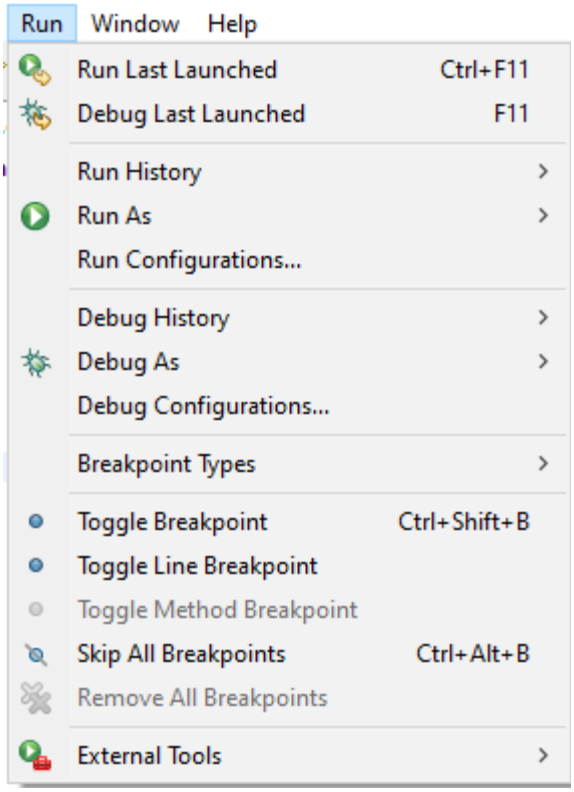


Рисунок 9. Меню «Run» в режиме разработки

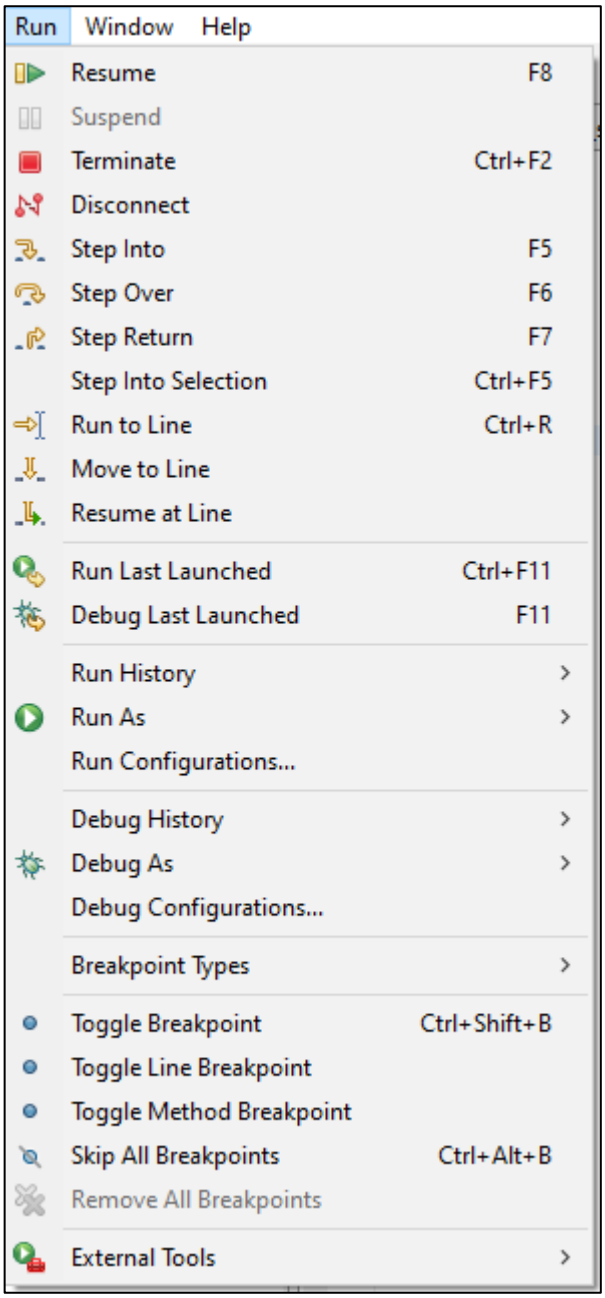


Рисунок 10. Меню «Run» в режиме отладки

Таблица 9. Команды меню «Run» в режиме разработки

Название команды	Описание функционирования	Горячие клавиши
Run Last Launched	Повторить последний запуск в режиме разработки.	Ctrl+F11
Debug Last Launched	Повторить последний запуск в режиме отладки.	F11
Run History	Повторить один из предыдущих запусков в режиме разработки.	
Run As	Выполнить запуск по горячим клавишам (в подменю) в режиме разработки.	
Run Configurations…	Открыть диалоговое окно конфигурации для управления конфигурациями запуска режима разработки.	

Название команды	Описание функционирования	Горячие клавиши
Debug History	Повторить один из предыдущих запусков в режиме отладки.	–
Debug As	Выполнить запуск по горячим клавишам (в подменю) в режиме отладки. Горячие клавиши запуска обеспечивают поддержку среды разработки проекта или чувствительный запуск выбора активного редактора.	–
Debug Configurations...	Открыть диалоговое окно конфигурации для управления конфигурациями запуска режима отладки.	–
Breakpoints Types	Перейти к работе с точками останова различных типов.	
Toggle Breakpoint	Установить или удалить точку останова в данном месте исходного текста или дизассемблерного кода.	Ctrl+Shift+B
Toggle Line Breakpoint	Установить или удалить контрольную строку в данном месте исходного текста или дизассемблерного кода.	–
Toggle Method Breakpoint	Установить или удалить контрольный метод в данном месте исходного текста или дизассемблерного кода.	–
Skip All Breakpoints	Пропустить все точки останова.	Ctrl+Alt+B
Remove All Breakpoints	Удалить все точки останова.	–
External Tools	Вызов внешних приложений	–

Таблица 10. Дополнительные оманды меню «Run» в режиме отладки

Название команды	Описание функционирования	Горячие клавиши
Resume	Возобновить отладку.	F8
Suspend	Приостановить отладку.	
Terminate	Завершить выбранную сессию отладки.	Ctrl+F2
Disconnect	Отключиться.	
Step Into	Выбрать для выполнения текущую строку, включая все функции, и перейти к следующему утверждению.	F5
Step Over	Выбрать для выполнения текущую строку, следуя за выполнением в программе.	F6
Step Return	Продолжить выполнение до конца текущей программы, затем перейти к выполнению вызывающей подпрограммы.	F7
Step Into Selection	Выбрать выделенную строку.	Ctrl+F5
Run to Line	Возобновить выполнение программы до достижения указанной строки. Это удобный способ приостановить выполнение на строчке без установки точки останова.	Ctrl+R
Move To Line	Перенести выполнение на указанную строку.	

Название команды	Описание функционирования	Горячие клавиши
Resume At Line	Запустить выполнение с указанной строки.	

5.1.9. Меню «Window»

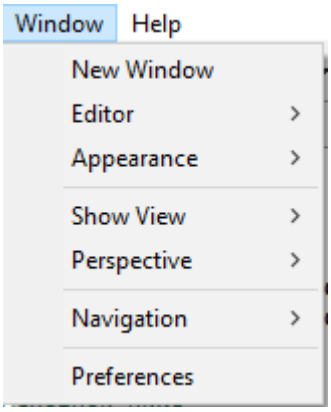


Рисунок 11. Меню «Window»

Таблица 11. Команды меню «Window»

Название команды	Описание функционирования	Горячие клавиши
New Window	Открыть новое окно.	
Editor	Настроить окно редактора в текущем окне программы: ▪ **Toggle Split Editor (Horizontal)** –закрепить редактор горизонтально; ▪ **Toggle Split Editor (Vertical)** – закрепить редактор вертикально; ▪ **Clone** – клонировать; ▪ **Toggle Word Wrap** – включить перенос слов; ▪ **Show Whitespace Characters** – показать пробелы; ▪ **Zoom In** –уменьшить; ▪ **Zoom Out** – увеличить	 Ctrl+_ Ctrl+{ - Alt+Shift+Y - Ctrl++ Ctrl+-
Appearance	Настроить представление: ▪ **Hide Toolbar** – скрыть панель инструментов; ▪ **Hide Status Bar** – скрыть строку состояния; ▪ **Toggle Full Screeen** –развернуть на всь экран; ▪ **Maximize Active View or Editor** – максимизировать активную вкладку или редактор; ▪ **Minimize Active View or Editor**– минимизировать активную вкладку или редактор.	 Alt+F11 Ctrl+M

Название команды	Описание функционирования	Горячие клавиши
Show View	Показать выбранные вкладки в текущем режиме.	
Perspective	Настроить внешний вид окна для текущего режима: **Open Perspective** – перейти в выбранный режим работы в текущем окне программы;	
	Customize Perspective – настроить вид окна в текущем режиме, показав или скрыв различные наборы действий; **Save Perspective As** – сохранить текущий режим, создав новый режим; **Reset Perspective** – вернуть вид окна в текущем режиме к исходной конфигурации; **Close Perspective** – закрыть активные режимы; **Close All Perspectives** – закрыть все открытые режимы в окне программы.	
Navigation	Отобразить определенные элементы интерфейса: ▪ **Show System Menu** – раскрыть меню, используемое для изменения размера, закрытия или закрепления текущей вкладки или редактора; ▪ **Show View Menu** – раскрыть меню, доступное на панели инструментов активной вкладки; ▪ **Find Actions –** открыть окно поиска описания команды**;** ▪ **Activate Editor** – сделать текущий редактор активным; ▪ **Next Editor** – активировать следующий активный редактор в списке недавно используемых редакторов; ▪ **Previous Editor** – активировать предыдущий активный редактор в списке недавно используемых редакторов; ▪ **Switch to Editor… -** открыть диалоговое окно выбора редактора для перехода; ▪ **Next View** – активирует следующую отрытую вкладку в списке недавно открытых; ▪ **Previous View** – активирует предыдущую отрытую вкладку в списке недавно открытых; ▪ **Next Perspective** – активирует следующий отрытый режим в списке недавно открытых; ▪ **Previous Perspective** – активирует предыдущий отрытый режим в списке недавно открытых.	Alt+- Ctrl+F10 Ctrl+3 F12 Ctrl+F6 Ctrl+Shift+F6 Ctrl+Shift+E Ctrl+F7 Ctrl+Shift+F7 Ctrl+F8 Ctrl+Shift+F8
Preferences	Открыть окно настройки «**Preferences**».	

5.1.10. Меню «Help»

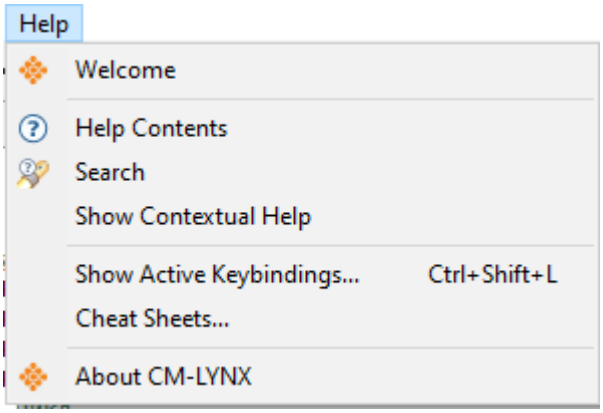


Рисунок 12. Меню «Help»

Таблица 12. Команды меню «Help»

Название команды	Описание функционирования	Горячие клавиши
Welcome	Открыть приветственную страницу справки.	–
Help Contents	Открыть содержание справки.	–
Search	Выполнить поиск по справке.	–
Show Contextual Help	Показать конекстную подсказку.	–
Show Active Keybindings…	Открыть подсказку с горячими клавишами.	Ctrl+Shift+L
Cheat Sheets…	Открыть файл со справочными материалами.	–
About CM-LYNX	Открыть окно с версией приложения.	–

5.2. Панель инструментов

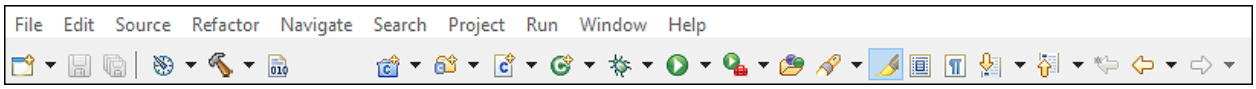


Рисунок 13. Панель инструментов в режиме разработки



Рисунок 14. Панель инструментов в режиме запущенной отладки

Таблица 13. Кнопки панели инструментов

Кнопка	Название	Описание функционирования
	Create New	Создать новый проект, папку или файл.
	Save	Сохранить содержимое текущего редактора. Недоступно, если в редакторе нет несохраненных изменений.

Кнопка	Название	Описание функционирования
	Save All	Сохранить содержимое всех редакторов. Недоступно, если в редакторе нет несохраненных изменений.
	Manage configurations	Показать диалоговое окно управления конфигурацией сборки «**Manage Configurations**».
	Build selected	Выполнить сборку всех активных конфигураций или выбранных проектов.
	Build All	Выполнить сборку всех проектов.
	New C/C++ project	Создать новый проект C/C++ .
	Create Folder	Создать папку в текущем проекте.
	Create File	Создать файл в текущем проекте.
	Create C++ Class	Создать класс C++ в текущем проекте.
	Debug	Запустить приложение в режиме отладки.
	Run	Запустить приложение.
	External Tools	Запустить диалоговое окно «**External Tools**».
	Open Element	Запустить диалоговое окно «**Open Element selection**» для открытия элемента в редакторе. Диалоговое окно «**Open Elements selection**» отображает все элементы, существующие в рабочем пространстве.
	Search	Запустить окно поиска C/C++.
	Toggle Mark Occurences	Включить/выключить режим подсветки всех вхождений выбранного элемента в текущем файле.
	Toggle Block Selection Mode	Включить/выключить режим выделения блоков.
	Show Whitespace Characters	Отобразить непечатаемые символы в редакторе.
	Next annotation	Перейти к следующей аннотации в редакторе.
	Previous annotation	Перейти к предыдущей аннотации в редакторе.
	Go to Last Edit Location	Вернуть вид редактора к последней редактируемой строке, если последний редактируемый файл был закрыт, он откроется заново.
	Back	Перейти назад по открытым файлам.
	Forward	Перейти вперед по открытым файлам.
	Skip All Breakpoints	Пропускать все точки останова.

Кнопка	Название	Описание функционирования
	Resume	Возобновить выполнение приостановленной отладки.
	Suspend	Приостановить выполнение выбранной отладки.
	Terminate	Завершить выбранную сессию отладки.
	Disconnect	Отключиться.
	Step Into	Выбрать текущую строку для выполнения, включая все методы и переход к следующему сообщению.
	Step Over	Выбрать текущую строку для выполнения после выполнения в программе.
	Step Return	Продолжить выполнение до конца текущей программы, затем перейти к выполнению вызывающей подпрограммы.
	Run to Line	Возобновить выполнение программы до достижения указанной строки. Это удобный способ приостановить выполнение на строчке без установки точки останова.
	Move to Line	Перенести выполнение на указанную строку (см. п. 6.11.5).
	Resume at Line	Запустить выполнение с указанной строки (см. п. 6.11.5).
	Multisession Mode	Включить режим «Multisession» (см. п. 6.11.13.2)
	Multicore Commands Mode	Включить режим «Multicore» (см. п. 6.11.13.1)
	Drop to Frame	Выполнить повторный вход в стек вызовов в режиме отладки.
	Instruction Stepping Mode	Активировать возможность выполнения инструкции в пошаговом режиме для проверки программы так, как она входит в дизассемблированный код.
	Use Step Filters	Включить/выключить использование шаговых фильтров.

5.3. Панель «Project Explorer»

Панель «**Project Explorer**» расположена на левой панели окна программы и предназначена для более удобного просмотра ресурсов в иерархическом представлении.

Примечание: панель может быть скрыта в режиме отладки.

На панели «**Project Explorer**» предоставлены возможности открытия файлов, редактирования и выбора ресурсов для операций.

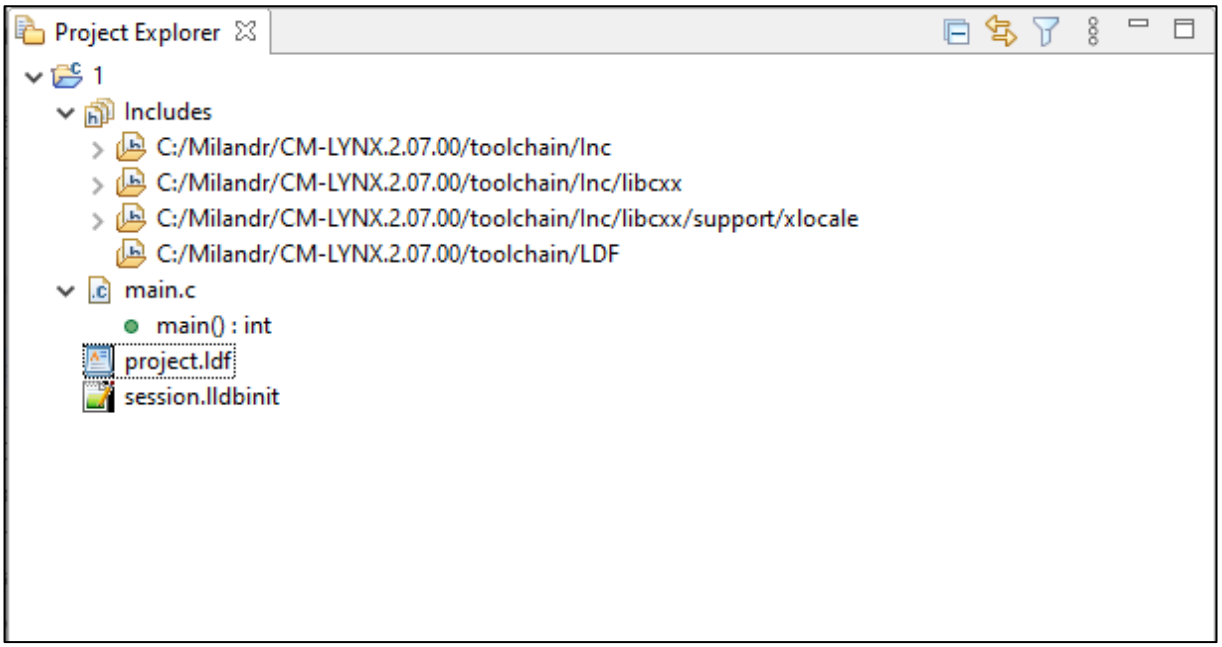


Рисунок 15. Вкладка «Project Explorer»

На панели доступно контекстное меню, предоставляющее базовый функционал для работы с приложением. Контекстное меню вызывается щелчком правой клавиши мыши на строке любого ресурса.

Кнопки на панели инструментов вкладки «**Project Explorer**» описаны в таблице ниже.

Таблица 14. Кнопки панели инструментов вкладки «Project Explorer»

Кнопка	Название	Описание функционирования
	Collapse All	Свернуть дерево всех ресурсов.
	Link with Editor	Установить связь вкладки с активным редактором. Когда выбрана эта опция, изменение активного редактора автоматически обновит выбор «**Project Explorer**» до редактируемого ресурса.
	View Menu	Показать пункты меню, которые позволяют сортировать или фильтровать содержимое «**Project Explorer**», а также выбирать рабочий набор: ▪ **Top level elements** (пункт меню позволяет задать корневой элемент; доступны варианты**: Projects** – проект и **Working set** – рабочий набор); ▪ **Select Working set…** (пункт меню вызывает окно «**Select Working set**», с помощью которого можно выбрать рабочий набор); ▪ **Deselect Working set** (пункт меню позволяет отменить выбор рабочего набора); ▪ **Edit Active Working set…** (пункт меню вызывает окно «**Edit Working set**», с помощью которого можно изменить содержимое рабочий набор); ▪ **Customize View…** (пункт меню вызывает окно «**Available Customizations**» для настройки активных фильтров и отображаемого содержимого окна); ▪ **Link with Editor** (пункт меню позволяет установить связь вкладки с активным редактором; совпадает по назначению с кнопкой).
	Minimize	Свернуть вкладку.
	Maximize	Развернуть вкладку на всю рабочую область окна C/C++.

5.4. Рабочая область

Рабочая область содержит редактор(ы) (по умолчанию в центральной части окна) и вкладки, состав которых определяется режимом работы.

Вкладки поддерживают редакторы и обеспечивают альтернативное представление информации в CM-LYNX. Например, «**Project Explorer**» и другие навигационные вкладки отображают проекты и другие ресурсы, с которыми работает пользователь.

Меню вкладок открывается двойным щелчком по значку, расположенному слева от панели заголовка вкладки. У некоторых вкладок есть своя панель инструментов. Действия, представленные кнопками на панелях вкладки, распространяются только на элементы в этой вкладке.

Вкладка может отображаться отдельно или располагаться рядом с другими вкладками. Вид окна программы можно изменить, открывая и закрывая вкладки, размещая их в разных областях окна.

5.4.1. Редактор C/C++

Область редактора C/C++ расположена в центральной части окна программы и позволяет редактировать файлы C/C++.

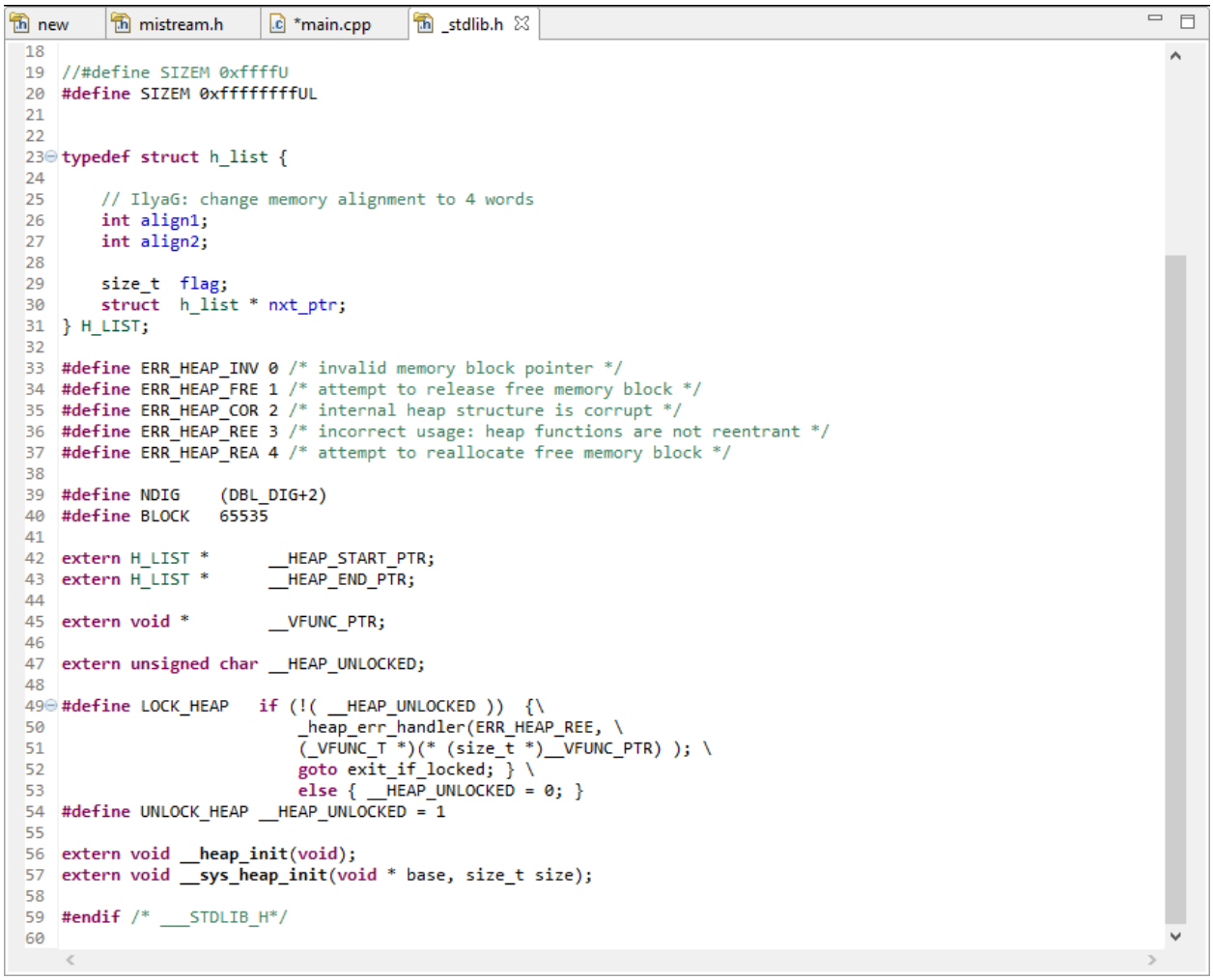


Рисунок 16. Рабочая область редактора C/C++

Файл, связанный с редактором – это представление структуры C/C ++, которое отображает структуру активного C, C++ или make-файла. Он обновляется при редактировании этих файлов.

Редактор включает в себя следующие функции:

- выделение синтаксиса;
- помощник кода/содержимого;
- встроенные функции отладки.

Наиболее распространенный способ вызвать редактор C/C++ – это выбрать файл из списка во вкладках «**Project Explorer**» или «**C/C++ Project**», используя контекстное меню, или щелчком мыши на строке файла (одинарный или двойной щелчок в зависимости от пользовательских настроек).

Редактор C/C++ не содержит собственной панели инструментов. Для работы с редактором могут быть использованы следующие инструменты:

- основная панель инструментов (см. п. 5.2. Панель инструментов);
- пункты главного меню «**Edit**» (см. п. 5.1.2. Меню «Edit»);
- пункты главного меню «**Search**» (см. п. 5.1.6. Меню «Search»);
- горячие клавиши.

5.4.2. Режим разработки (перспектива «C/C++»)

5.4.2.1. Вкладка «Outline»

Вкладка «**Outline**» расположена на правой панели окна программы. Вкладка отображает сводную структуру файла C/C++, который в данный момент открыт в области редактора, выводя список структурных элементов.

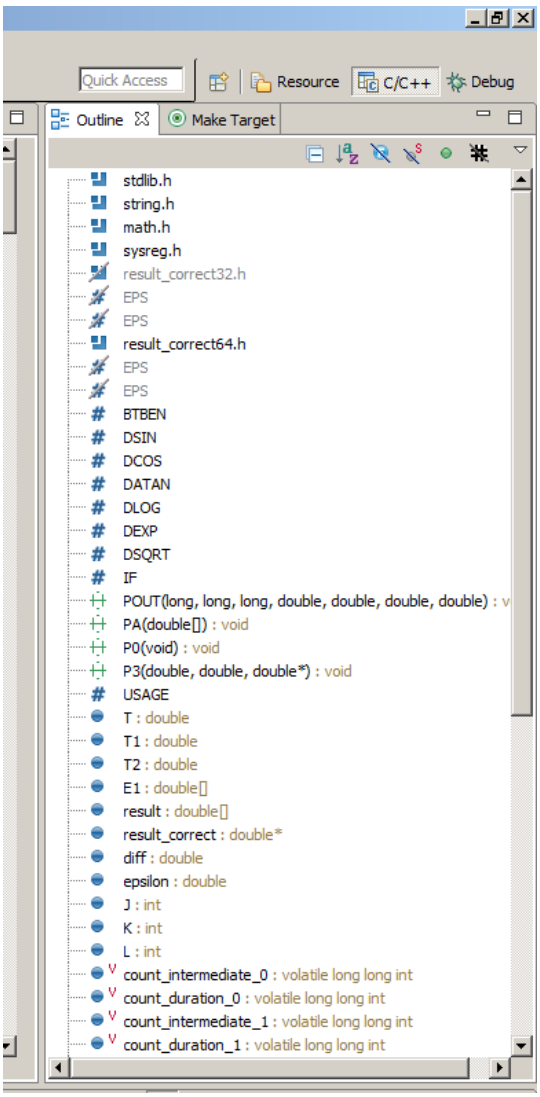


Рисунок 17. Вкладка «Outline»

Таблица 15. Кнопки панели инструментов вкладки «Outline»

Кнопка	Описание функционирования
	Свернуть вкладку.
	Упорядочить элементы списка в алфавитном порядке.
	Скрыть поля.
	Скрыть статические члены.
	Скрыть закрытые члены.
	Скрыть неактивные элементы списка.

Структурные элементы файла C/C++ в зависимости от категории обозначаются соответствующими значками.

Таблица 16. Значки для различных категорий структурных элементов файла C/C++

Значок	Описание
	Class
	Namespace
	Macro Definition
	Enum
	Enumerator
	Variable
	Field private
	Field protected
	Field public
	Include
	Method private
	Method protected
	Method public
	Struct
	Type definition
	Union
	Function

5.4.2.2. Вкладка «Make Target»

Вкладка «**Make Target**» расположена на правой панели окна программы; предназначена для создания и управления целевыми объектами.

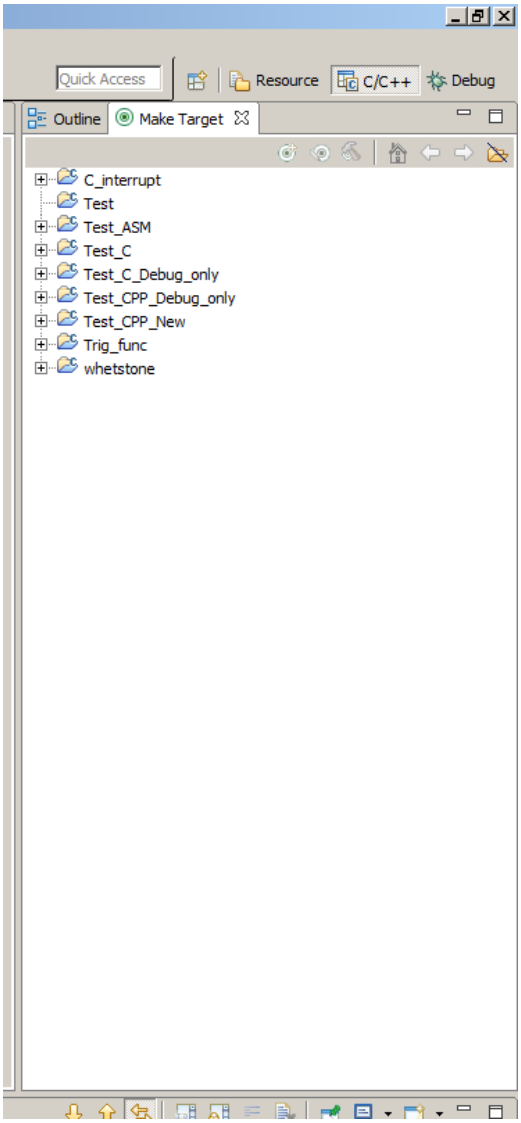


Рисунок 18. Вкладка «Make Targets»

Таблица 17. Кнопки панели инструментов вкладки «Make Targets»

Кнопка	Название	Описание функционирования
	New Make Target	Открыть диалоговое окно «**Create Make Target**» для создания нового целевого объекта.
	Edit Make Target	Открыть диалоговое окно «**Modify Make Target**» для редактирования выбранного объекта.
	Build Make Target	Выполнить сборку выбранного объекта.
	Home	Переместить на верхний уровень.
	Back	Переместить на предыдущий уровень.
	Forward	Переместить на следующий уровень.
	Hide Empty Folders	Переключить отображение пустых папок.

5.4.2.3. Вкладка «Problems»

Вкладка «**Problems**» расположена на нижней панели окна программы. На вкладке отображается информация о найденных ошибках, выдаваемых C++ компилятором и/или ассемблером:

- Описание ошибки;
- Наименование ресурса, где была найдена ошибка;
- Путь к ресурсу, где была найдена ошибка;
- Местоположение ошибки (номер строки);
- Тип ошибки.

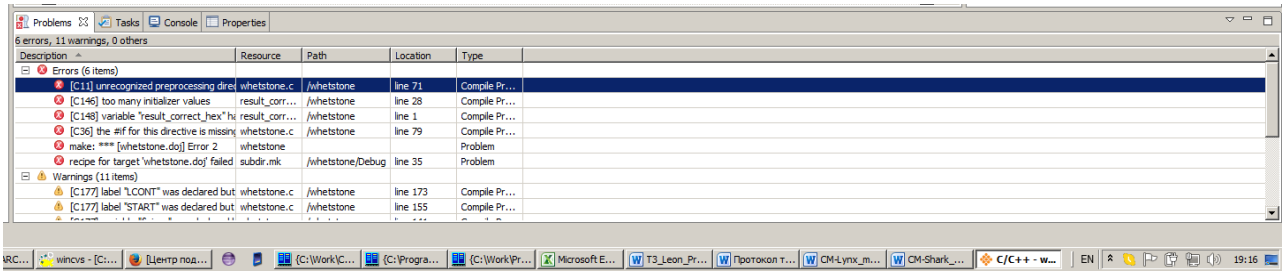


Рисунок 19. Вкладка «Problems»

При выборе ошибки из списка в поле редактора C/C++ откроется связанный файл, содержащий данную ошибку. При этом строка, в которой обнаружена ошибка, будет выделена цветом.

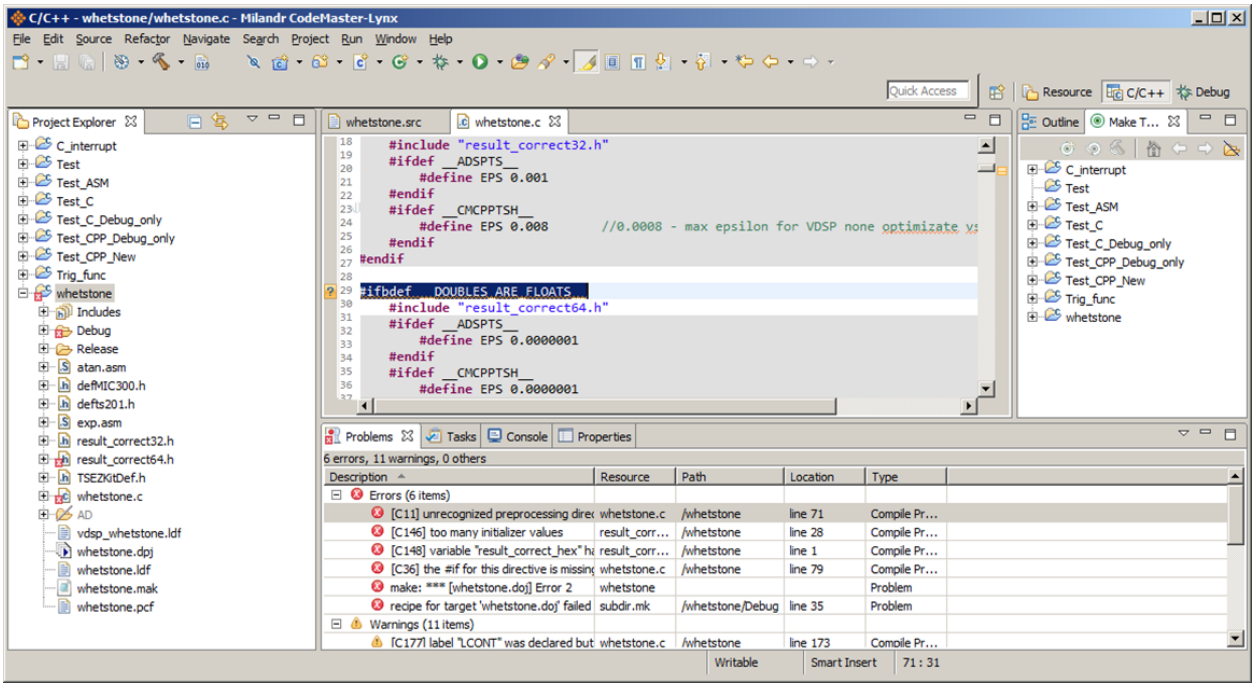


Рисунок 20. Строка с обнаруженной ошибкой в поле редактора C/C++

Таблица 18. Кнопки панели инструментов вкладки «Problems»

Кнопка	Название	Описание функционирования
▽	View Menu	Открыть меню, содержащее следующие команды: ▪ **Show** – раскрывает список для выбора вида отображаемых ошибок; ▪ **Group By** – раскрывает список для выбора параметра для группировки ошибок; ▪ **Sort By** – раскрывает список для выбора способа сортировки ошибок; ▪ **New Problems View** – открывает окно «**New Problems View**» для создания новой вкладки «**Problems**»; ▪ **Configure Contents...** – открывает окно «**Configure Contents**» для настройки содержимого вкладки; ▪ **Configure Columns...** – открывает окно «**Configure Columns**» для выбора отображаемых и скрытых параметров ошибок, а также для настройки их представления на вкладке.

5.4.2.4. Вкладка «Tasks»

Вкладка «**Tasks**» расположена на нижней панели окна программы. Данная вкладка позволяет создать собственные задания, а также автоматически добавляет задания **`TODO`** и **`FIXME`** из комментариев кода.

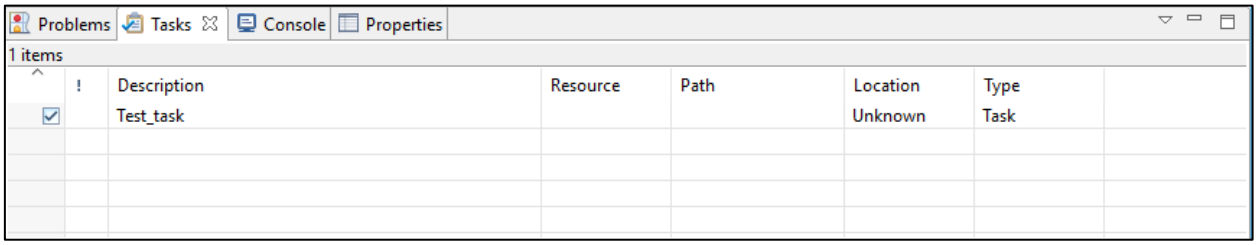


Рисунок 21. Вкладка «Tasks»

Информация о заданиях отображается в табличном виде, таблица имеет следующие столбцы:

- Первый столбец указывает, завершено ли задание. Выполненные задания помечаются флажком. Флажок можно добавить вручную;
- Второй столбец указывает на приоритет задания (высокий, нормальный или низкий приоритет) Высокий приоритет помечается значком ; низкий - ;
- Столбец «**Description**» содержит описание задания. Можно изменить описание задания, выбрав пункт «**Properties**» в контекстном меню;
- Столбцы «**Resource**» и «**Path**» содержат имя и расположение ресурса, связанного с заданием;
- Столбец «**Location**» указывает на номер строки ресурса, соответствующего заданию.

Таблица 19. Кнопки панели инструментов вкладки «Tasks»

Кнопка	Название	Описание функционирования
	View Menu	Отобразить меню для работы с вкладкой «**Tasks**». Меню содержит пункты: ▪ **Show** – меню позволяет выбрать какие задания показывать (**All** - все или только **TODOs**); ▪ **Group By** - меню позволяет выбрать способ сортировки заданий; ▪ **Sort By** – меню позволяет выбрать способ сортировки заданий; ▪ **New Tasks View** – открывает окно создания нового задания; ▪ **Configure Contents...** - открывает окно настройки содержимого; ▪ **Configure Columns...** - открывает окно настройки колонок.
	Minimize	Свернуть вкладку
	Maximize	Развернуть вкладку во всю рабочую область
	Restore	Восстановить размер вкладки

Таблица 20. Команды контекстного меню вкладки «Tasks»

Значок	Горячие клавиши	Описание
Go to		Перейти…
Copy	Ctrl+C	Скопировать.
Delete	Delete	Удалить.
Select All	Ctrl+A	Выделить все.
Show In		Выбрать вариант просмотра задания. Доступны варианты: ▪ Include Browser; ▪ C/C++ Projects; ▪ Project Explorer; ▪ System Explorer; ▪ Properties.
Add Task…		Открыть окно добавления задания
Delete Completed Tasks		Удалить выполненные задания
Mark Completed		Пометить задания как выполненные
Properties	Alt+Enter	Открыть окно свойств задания

5.4.2.5. Вкладка «Console»

Вкладка «**Console**» расположена на нижней панели окна программы. Вкладка отображает выходные данные процесса и позволяет вводить данные с клавиатуры. Возможно закрепление нескольких вкладок «**Console**». Список доступных вкладок открывается кнопкой на панели инструментов вкладки.

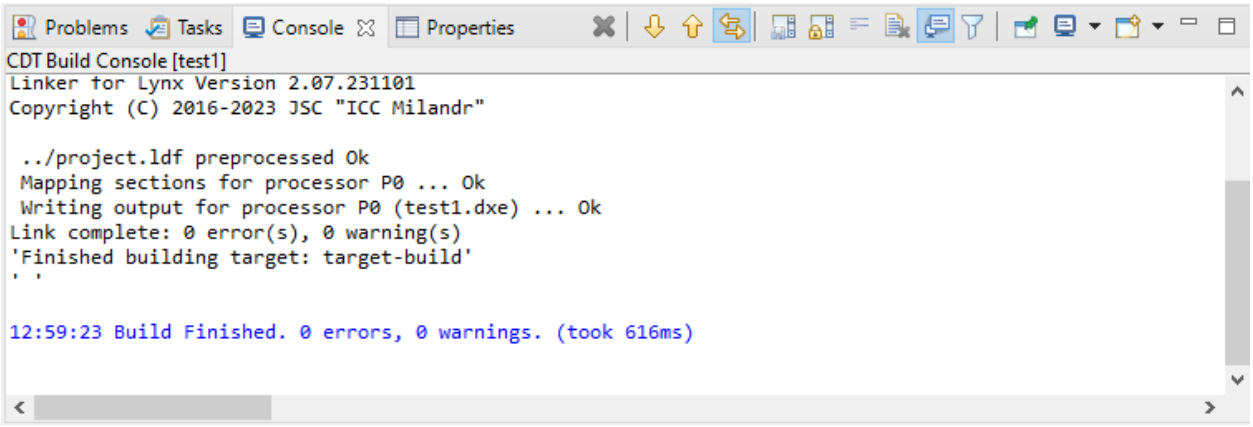


Рисунок 22. Вкладка «Console»

На вкладке «**Console**» выходных данных отображается текст нескольких видов, отмеченных разным цветом:

- стандартные выходные данные;
- стандартная ошибка;
- стандартные входные данные.

Цвета для разных видов текста можно настроить с помощью элементов панели инструментов вкладки «**Console**».

На вкладке выделяются проблемы сборки. Двойной щелчок на выделенной строке открывает код в поле редактора, в случае, когда анализаторы ошибки могут определить файл и строку выходных данных сборки.

Панель инструментов вкладки «**Console**» содержит кнопки, описанные в таблице ниже. Обратите внимание, что набор инструментов может отличаться в различных вкладках «**Console**».

Таблица 21. Кнопки панели инструментов вкладки «Console»

Кнопка	Название	Описание функционирования
	Close Console	Закрыть вкладку
	Next Error	Перейти к следующей ошибке.
	Previous Error	Вернуться к предыдущей ошибке.
	Show Error in Editor	Включить/выключить синхронизацию выбранных ошибок с областью редактора (по умолчанию синхронизация включена).
	Copy Build Log	Скопировать выходные данные сборки в файл.
	Scroll Lock	Включить/выключить блокирование прокрутки.
	Wrap Lines	Включить/выключить перенос длинных строк.

Кнопка	Название	Описание функционирования
	Clear Console	Очистить текущую вкладку «**Console**».
	Bring console to the top when building	Отобразить консоль поверх остальных окон при сборке проекта.
	Show compact log filtered by toolchain	Показывать компактный журнал сборки, см. также Рисунок 23 ниже. Также есть возможность ограничить компактный журнал до 1 ошибки и предупреждения на файл – см. настройку «**Limit errors and warnings to 1 per source file (if supported)**» в п. 6.12.2.2.2.
	Pin Console	Закрепить вкладку «**Console**» поверх остальных вкладок на нижней панели окна C/C++.
	Display Selected Console	Раскрыть список открытых вкладок «**Console**» для выбора одной из них.
	Open Console	Открыть новую вкладку «**Console**» из раскрывающегося списка: ▪ **New Console View** – новое представление вкладки «**Console**»; ▪ **Program Output Console** – открыть вкладку выходных данных программы; ▪ **Windows Program Output Console** – открыть вкладку выходных данных Windows; ▪ CVS, и другие в соответствии с установленными опциями.

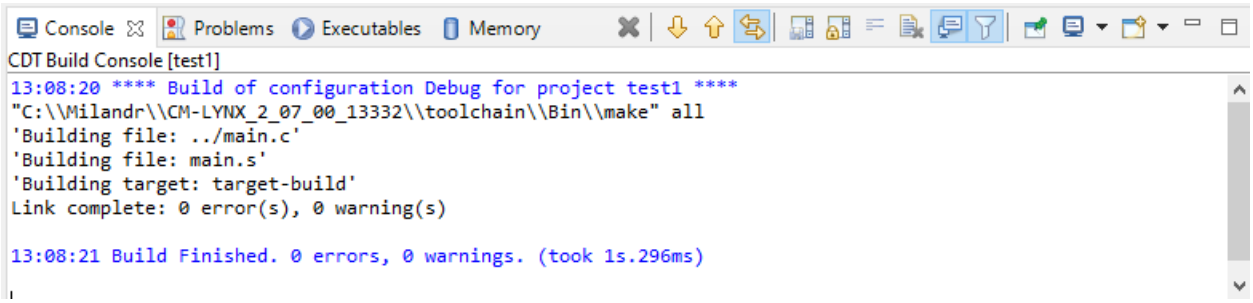


Рисунок 23. Вкладка «Console» при включенном компактном журнале сборке

При щелчке правой клавишей мыши в области вкладки «**Console**» (или при нажатии горячих клавиш **Shift+F10**) отображается контекстное меню, содержащее команды, описанные в таблице ниже. Набор команд контекстного меню может отличаться в различных вкладках «**Console**».

Таблица 22. Команды контекстного меню вкладки «Console»

Название	Описание функционирования
Clear	Очистить текущую вкладку.
Editor options: Cut, Copy, Paste, Select All	Выполнить стандартные операции редактирования. Набор доступных операций зависит от того, где на вкладке «**Console**» находился курсор. Например, невозможно вставить текст в выходные данные программы, но можно вставить текст внизу файла.
Go to Line	Открыть окно, которое перемещает фокус на указанную пользователем строку. Также оно показывает общее число строк в буфере вкладки.
Find/Replace...	Открыть окно «**Find/Replace**» для работает в текстовом поле вкладки.

Название	Описание функционирования
Open Link	Перейти по обнаруженной гиперссылке для различных вкладок «**Console**», где поддерживается данная команда.
Scroll Lock	Включить/выключить блокирование прокрутки.

5.4.2.6. Вкладка «Properties»

Вкладка «**Properties**» расположена на нижней панели окна программы. Вкладка отображает наименования и значения свойств для выбранного ресурса.

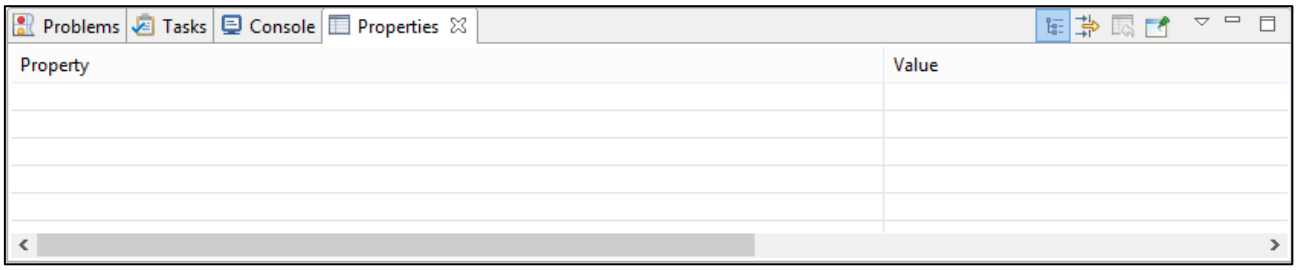


Рисунок 24. Вкладка «Properties»

Для просмотра более подробной информации о ресурсе, помимо отображаемой на вкладке «**Properties**», следует щёлкнуть правой клавишей мыши на строке ресурса во вкладке «**Project Explorer**» и в открывшемся контекстном меню выбрать пункт **Properties**.

Таблица 23. Кнопки панели инструментов вкладки «Properties»

Кнопка	Название	Описание функционирования
	Pin Properties View	Закрепить окно за текущим выбранным контекстом.

5.4.3. Режим отладки (перспектива «Debug»)

5.4.3.1. Вкладка «Debug»

Вкладка «**Debug**» расположена на левой верхней панели окна и содержит информацию о цели отладки в иерархическом порядке.

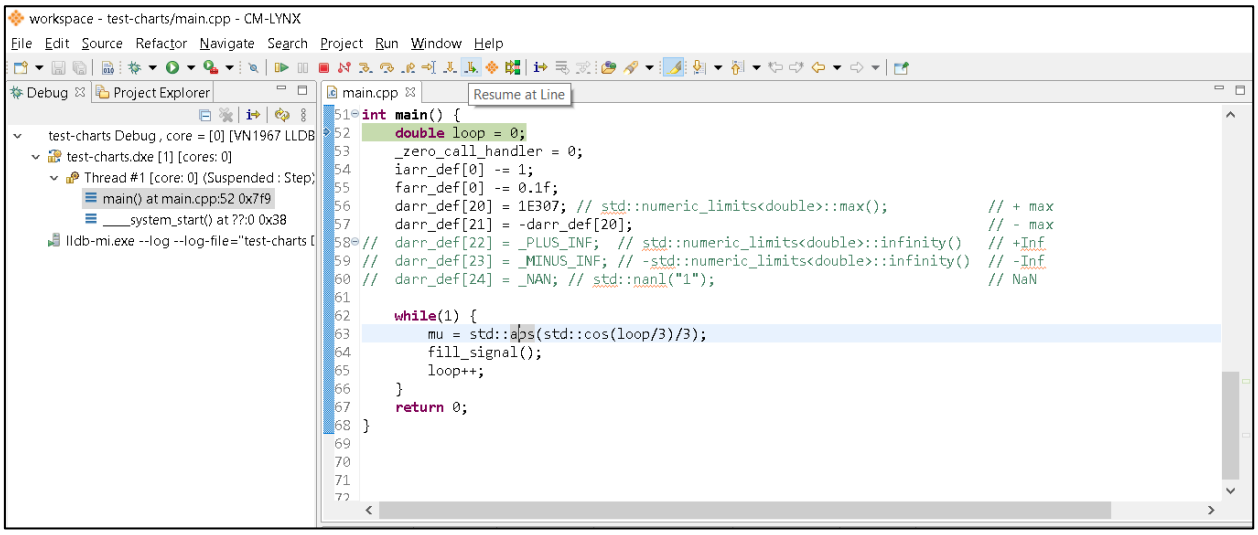


Рисунок 25. Вкладка «Debug»

Таблица 24. Кнопки панели инструментов вкладки «Debug»

Кнопка	Название	Описание функционирования
	Remove All Terminated Launches	Очистить законченные запуски вкладки «**Debug**».
	Instruction Stepping Mode	Активировать возможность выполнения инструкции в пошаговом режиме для проверки программы так, как она входит в дизассемблированный код.
	Refresh	Обновить содержимое вкладки.
	View Menu	Раскрыть меню, содержащее следующие команды: ▪ **Layout** – изменяет представление вкладки «**Debug**» (древовидное представление или навигационная цепочка); ▪ **Show Debug Toolbar** – отображает/скрывает полный набор кнопок панели управления; ▪ **View Management...** – открывает окно для управления вкладками; ▪ **Show Full Path** – активирует/деактивирует отображение потомков во вкладке.

Таблица 25. Команды контекстного меню вкладки «Debug»

Значок	Название	Описание
	Copy Stack	Скопировать выбранный стек приостановленных задач, в том числе и состояние запущенных целевых объектов, в буфер обмена (аналогично копированию запущенных задач в буфер обмена).
–	Find...	Открыть диалоговое окно поиска элементов.
	Drop To Frame	Вернуться к выбранному стеку вызовов на вкладке «**Debug**».
	Step Into	Выбрать для выполнения текущую строку, включая все функции, и перейти к следующему утверждению.

Значок	Название	Описание
	Step Over	Выбрать для выполнения текущую строку, следуя за выполнением в программе.
	Step Return	Продолжить выполнение до конца текущей программы, затем перейти к выполнению вызывающей подпрограммы.
	Instruction Stepping Mode	Выбрать пошаговый режим выполнения инструкций для изучения шагов программы в дизассемблированном коде.
	Use Step Filters	Переключить пошаговый фильтр «**Вкл/Выкл**». При нажатой кнопке функции применяется пошаговый фильтр.
	Resume	Возобновить выполнение приостановленной цели отладки.
	Suspend	Приостановить выполнение выбранного целевого объекта отладки.
	Terminate	Завершить выбранную сессию отладки.
	Terminate and Relaunch	Завершить выбранный целевой объект отладки и затем перезапустить его.
	Disconnect	Отключиться.
	Remove All Terminated	Очистить все завершенные запуски во вкладке «**Debug**».
	Terminate and Remove	Завершить выбранную цель отладки и удалить ее из вкладки.
	Terminate/Disconnect All	Завершить/отключиться от всех сессий отладки.
–	Properties	Открыть окно «**Properties**» для отображения свойств процесса.

5.4.3.2. Вкладки «Registers»

Вкладки «**Registers <Core>**» и «**Registers <Peripherals>**», расположенные на правой верхней панели окна, содержит информацию о регистрах ядра процессора и регистрах периферии соответственно.

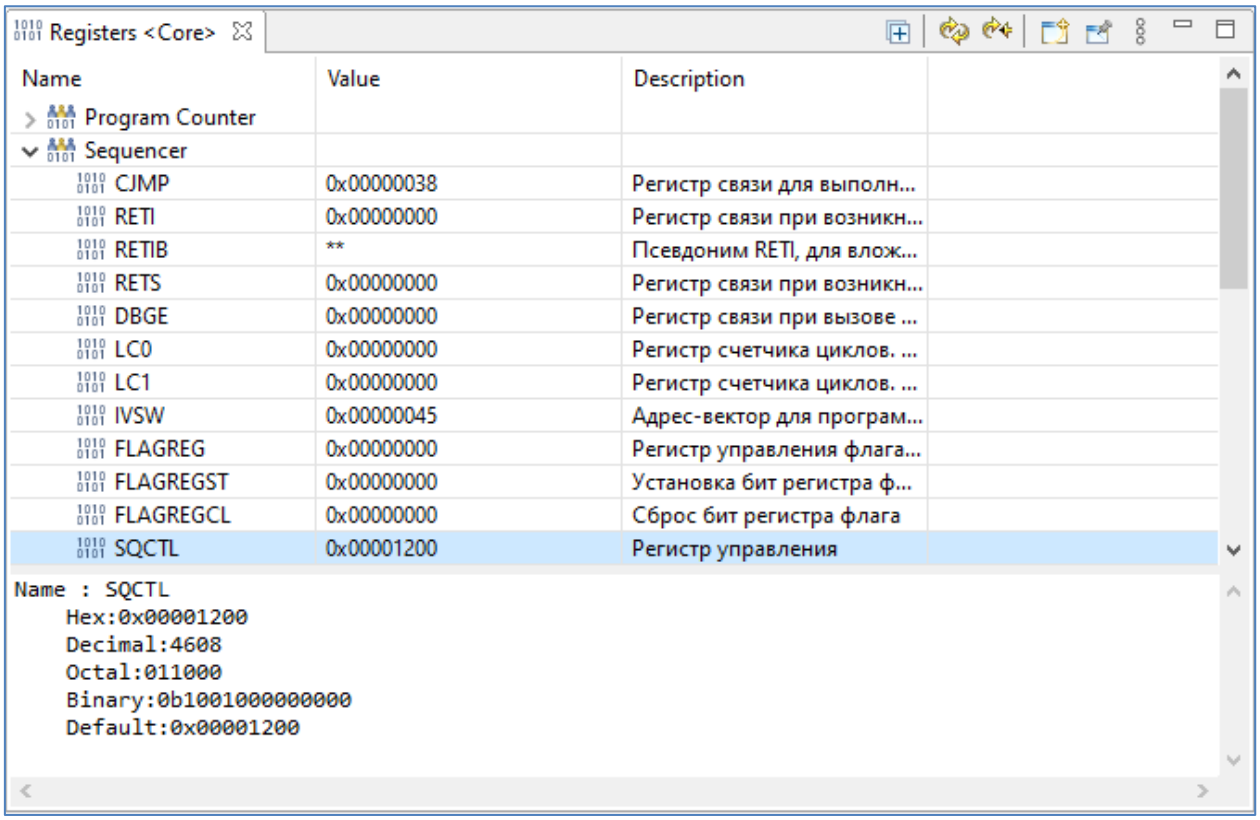


Рисунок 26. Вкладка «Registers <Core>»

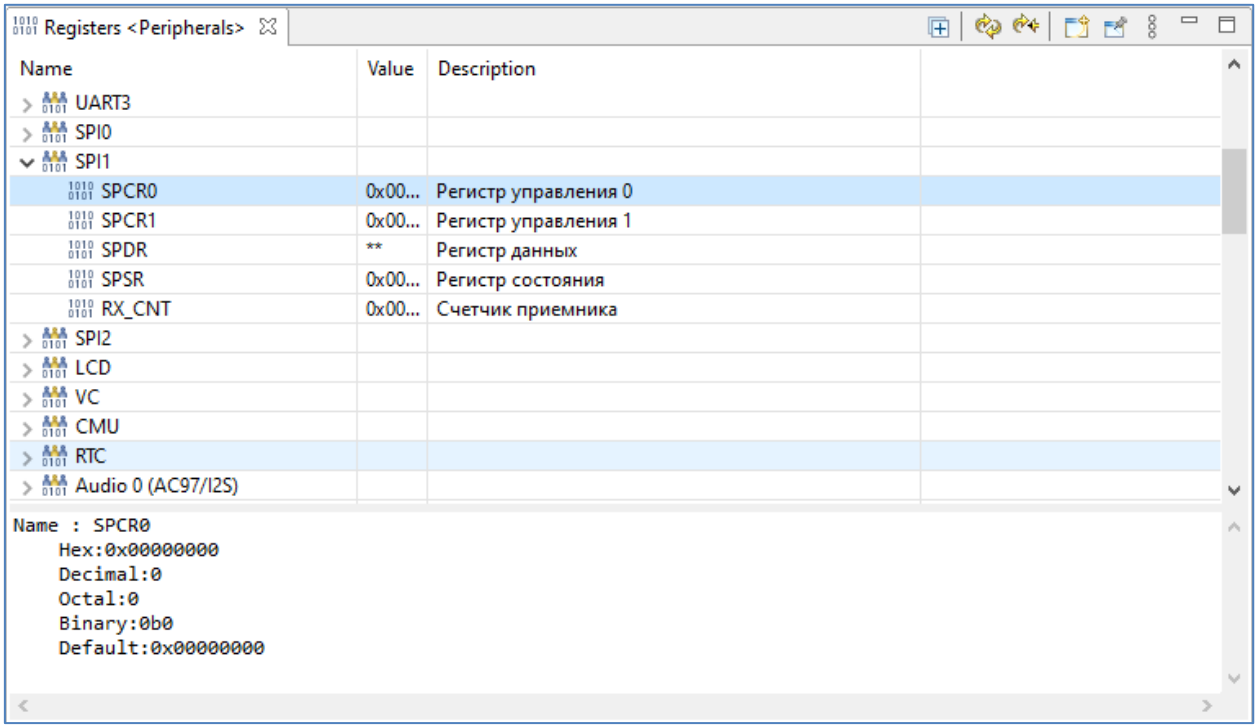


Рисунок 27. Вкладка «Registers <Peripherals>»

Таблица 26. Кнопки панели инструментов вкладок «Registers<Core>» и «Registers<Peripherals>»

Кнопка	Название	Описание функционирования
	Collapse egisters	Свернуть все развернутые регистры.
	Expand registers	Развернуть все свернутые регистры.
	Refresh	Обновить регистры.
	Refresh selected	Обновить выделенные регистры.
	Open New View	Открыть новую вкладку «**Registers**» на нижней панели окна в режиме отладки.
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.
	View Menu →Number Format	Открыть список для выбора формата отображения значений регистров.
	View Menu → Layout	Открыть список для выбора представления параметров во вкладке: ▪ **Show Columns** – позволяет отобразить/скрыть столбцы таблицы; ▪ **Select Columns** – позволяет настроить отображаемые столбцы в таблице; ▪ **Select Columns Alignment** – позволяет настроить выравнивание («Left», «Center», «Right») для любого столбца таблиц.

Таблица 27. Команды контекстного меню вкладок «Registers<Core>» и «Registers<Peripherals>»

Значок	Название	Описание
	Select All	Выделить все содержимое редактора.
	Copy Registers	Скопировать названия регистров и содержимое в буфер обмена.
	Number Format	Выбрать тип формата из списка: ▪ **Default** - по умолчанию ▪ **Hex** - шестнадцатеричный; ▪ **Decimal** - десятеричный; ▪ **Octal** - восьмеричный; ▪ **Binary** - бинарный; ▪ **String** - строковый.
	Find...	Открыть диалоговое окно «**Find**» для поиска определенных элементов.
	Watch	Преобразовать выбранные регистры в отслеживаемое выражение.
	Refresh selected	Обновить выбранные регистры.

5.4.3.3. Вкладка «Variables»

Вкладка «**Variables**», расположенная на правой верхней панели окна, отображает информацию о глобальных переменных и переменных, ассоциируемых со стеком вызовов, выбранным во вкладке «**Debug**».

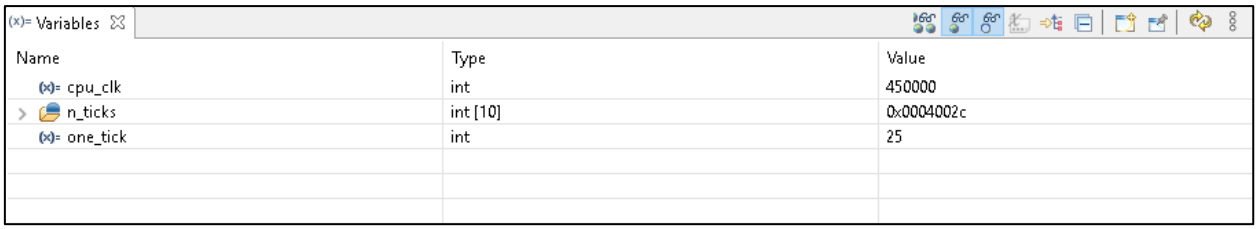


Рисунок 28. Вкладка «Variables»

Для включения отображения адресов переменных на вкладке «Variables» необходимо включить отображение колонки «Location» в настройках представления параметров на вкладке.

Настройки представления вызываются командой «View Menu (пиктограмма) ->Layout -> Select Columns».

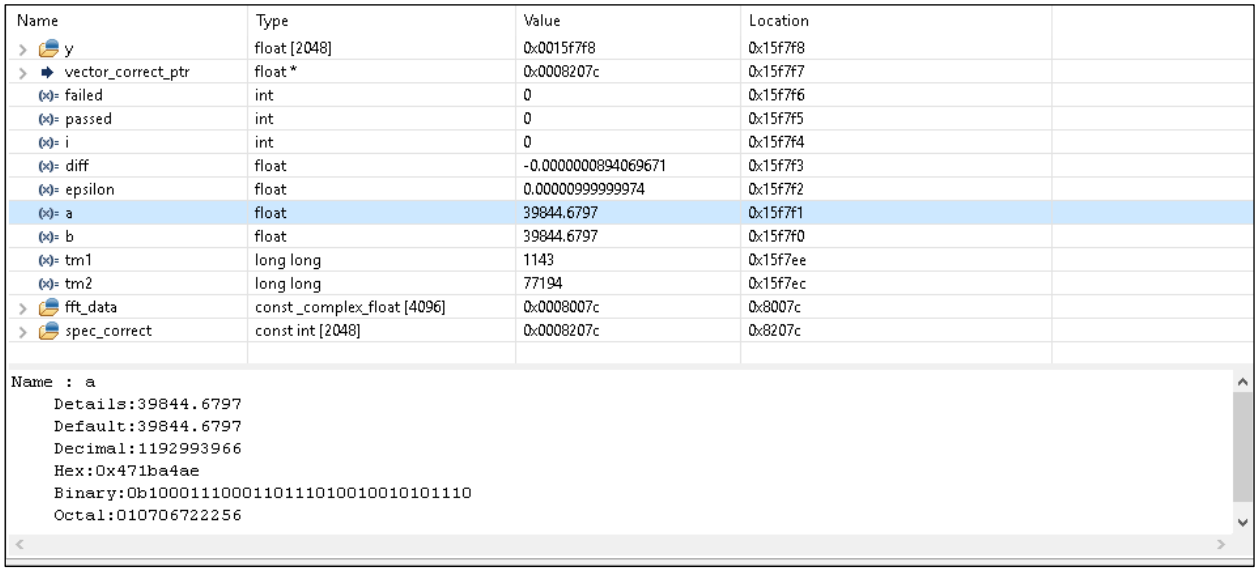


Рисунок 29. Вкладка «Variables» с включенной колонкой «Location»

Таблица 28. Кнопки панели инструментов вкладки «Variables»

Кнопка	Название	Описание функционирования
	Show Module Global Variables	Добавить все видимые общие глобальные переменные текущего исполняемого модуля.
	Show File Global Variables	Добавить видимые глобальные не статические переменные только текущего исходного файла.
	Show Static Variables	Добавить все видимые статических переменных (локальные и глобальные).
	Show Type Names	Включить отображение типа рядом со значением переменных (для отключенного режима «**Show Columns**»)

Кнопка	Название	Описание функционирования
	Show Logical Structure	Включить отображение логической структуры переменных.
	Collapse All	Свернуть все переменные.
	Open New View	Открыть новую вкладку «**Variables**» на нижней панели окна в режиме отладки.
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.
	Refresh	Обновить данные на вкладке.
	View Menu →Number Format	Открыть список для выбора формата отображения значений переменных.
	View Menu → Layout	Открыть список для выбора представления параметров во вкладке. Команда «**Layout -> Select Columns**» позволяет включить отображение колонки «Location», которая будет показывать адреса всех переменных. Команда «**Layout -> Select Columns Alignment**» позволяет настроить выравнивание («Left», «Center», «Right») для любого столбца таблиц.

Таблица 29. Команды контекстного меню вкладки «Variables»

Значок	Название	Описание
	Select All	Выделить все переменные во вкладке.
	Copy Variables	Скопировать выбранные переменные в буфер обмена.
	Enable	Включить выключенную переменную.
	Disable	Выключить включенную переменную.
	Number Format	Выбрать тип формата из списка: ▪ по умолчанию; ▪ десятеричный; ▪ шестнадцатеричный; ▪ восьмеричный; ▪ бинарный. Изменение формата влияет в том числе на элементы структур и массивов.
	Cast To Type…	Открыть диалоговое окно «**Cast To Type**…» для преобразования типа переменной.
	View Memory	Создать монитор памяти для выбранной переменной.
	Find...	Открыть диалоговое окно «**Find**» для поиска элементов во вкладке.

Значок	Название	Описание
	Add Watchpoints (C/C++)...	Открыть диалоговое окно «**Properties for C/C++ Watchpoint**» для добавления контрольных точек.
	Watch	Создать отслеживаемое выражение для выбранной переменной.

5.4.3.4. Вкладка «Breakpoints»

Вкладка «**Breakpoints**», расположенная на правой верхней панели окна, отображает список точек останова и точек наблюдения.

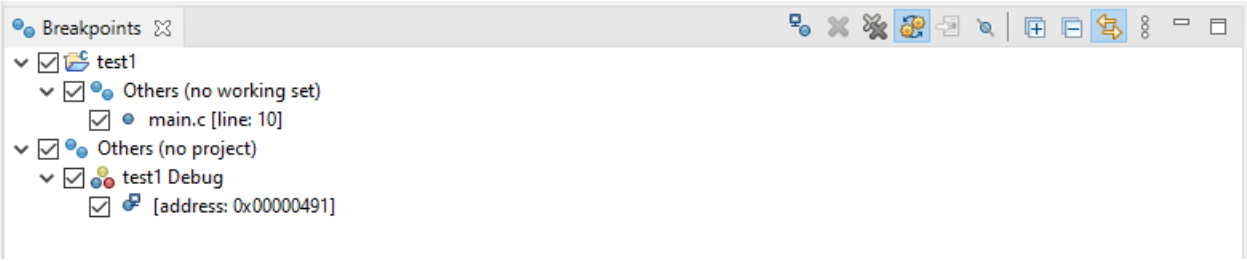


Рисунок 30. Вкладка «Breakpoints»

Таблица 30. Кнопки панели инструментов вкладки «Breakpoints»

Кнопка	Название	Описание функционирования
	Add address breakpoint	Добавить адресную точку останова.
	Remove Selected Breakpoints	Удалить выбранную(ые) точку(и) останова.
	Remove All Breakpoints	Удалить все точки останова. В том числе и те, которые привязаны к другим проектам и отладочным конфигурациям.
	Show Breakpoints Supported by Selected Target	Отображать только точки останова и точки наблюдения, активные для текущей отладки (привязанные к отлаживаемому проекту и не привязанные к проектам). Если кнопка выключена – будут отображаться все точки останова и наблюдения в текущем рабочем пространстве.
	Go to File	Перейти в место, соответствующее точке останова, на вкладке редактора.
	Skip All	Включить режим, при котором все точки останова будут пропущены.
	Expand All	Развернуть все подпункты вкладки.
	Collapse All	Свернуть все подпункты вкладки.
	Link with Debug View	Установить связь точек останова с вкладкой «**Debug**».
	View Menu → Layout	Открыть список для выбора представления параметров во вкладке.

Таблица 31. Команды контекстного меню вкладки «Breakpoints»

Значок	Название	Описание
	Go to File	Перейти в место, соответствующее точке останова, на вкладке редактора.
	Enable	Включить выбранную(ые) точку(и) останова.
	Disable	Выключить выбранную(ые) точку(и) останова.
	Remove	Удалить выбранную(ые) точку(и) останова.
	Remove All	Удалить все точки останова.
	Select All	Выделить все точки останова, содержащиеся на вкладке.
	Copy	Скопировать выбранные точки останова в буфер обмена.
	Paste	Вставить скопированные точки останова во вкладку.
	Import Breakpoints...	Открыть диалоговое окно «**Import Breakpoints**» для импорта точек останова.
	Export Breakpoints...	Открыть диалоговое окно «**Export Breakpoints**» для экспорта точек останова.
	Breakpoint Properties...	Открыть диалоговое окно свойств точек останова.

5.4.3.5. Вкладка «Expressions»

Вкладка «**Expressions**», расположенная на правой верхней панели окна, содержит информацию о выражениях – фрагментах кода, которые могут быть оценены для получения результата.

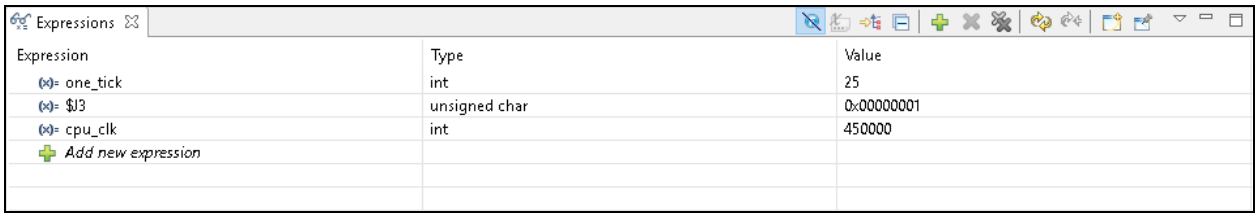


Рисунок 31. «Вкладка Expressions»

Таблица 32. Кнопки панели инструментов вкладки «Expressions»

Кнопка	Название	Описание функционирования
	Expressions for selected debug context	Отображать выражения, привязанные к выбранному контексту отладки выражения. См. также п. 6.11.11.
	Show Type Names	Включить отображение названий типов.
	Show Logical Structure	Показать/Скрыть логическую структуру выражений.

Кнопка	Название	Описание функционирования
	Collapse All	Свернуть все выражения.
	Add Watch Expressions	Разрешить добавлять отслеживаемое выражение.
	Remove Expressions	Удалить выбранное(ые) выражение(я) из вкладки.
	Remove All Expressions	Удалить все выражения из вида.
	Refresh	Обновить данные на вкладке.
	Refresh selected	Обновить выделенные выражения.
	Open New View	Открыть новую вкладку «**Expressions**».
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.
	View Menu → Layout	Открыть список для выбора представления параметров во вкладке. Команда «**Layout -> Select Columns**» позволяет включить отображение колонки «Address», которая будет показывать адреса всех переменных. Команда «**Layout -> Select Columns Alignment**» позволяет настроить выравнивание («Left», «Center», «Right») для любого столбца таблиц.

Таблица 33. Команды контекстного меню вкладки «Expressions»

Значок	Название	Описание
	Select All	Выделить все выражения во вкладке.
	Copy Expressions	Скопировать выбранное выражение в буфер обмена.
	Remove	Удалить выбранное(ые) выражение(я) из вкладки.
	Remove All	Удалить все выражения из вкладки.
	Number Format	Выбрать тип формата из списка: ▪ по умолчанию; ▪ десятеричный; ▪ шестнадцатеричный; ▪ восьмеричный; ▪ бинарный. Изменение формата влияет в том числе на элементы структур и массивов.
	Add Watch Expressions	Разрешить добавлять отслеживаемое выражение.
	Add Expression Group	Добавить одну из групп выражений: ▪ Registers – все регистры;

Значок	Название	Описание
		▪ Local variables – все локальные переменные;
	Find...	Открыть диалоговое окно «**Find**» для поиска элементов во вкладке.
	Disable	Выключить включенное выражение.
	Enable	Выключить выключенное выражение.
	Edit Watch Expression	Разрешить редактирование существующего отслеживаемого выражения.
	Add Watchpoints (C/C++)...	Открыть диалоговое окно «**Properties for C/C++ Watchpoint**» для добавления контрольных точек.
	Cast To Type...	Открыть диалоговое окно «**Cast To Type**...» для преобразования типа переменной.
	Refresh Selected	Обновить выделенные выражения.

5.4.3.6. Вкладка «Disassembly»

Вкладка «**Disassembly**», расположенная на центральной правой панели окна, отображает загруженную программу в виде инструкций ассемблера вместе с исходным кодом для сравнения. Выполняемая строка отмечена стрелкой и выделена цветом. На вкладке «**Disassembly**» можно выполнить следующее:

- установить точки останова в начале любой строки;
- включить и выключить точки останова и установить их свойства;
- пошагово пройти все инструкции дизассемблирования программы;
- перейти к определенным инструкциями программы.

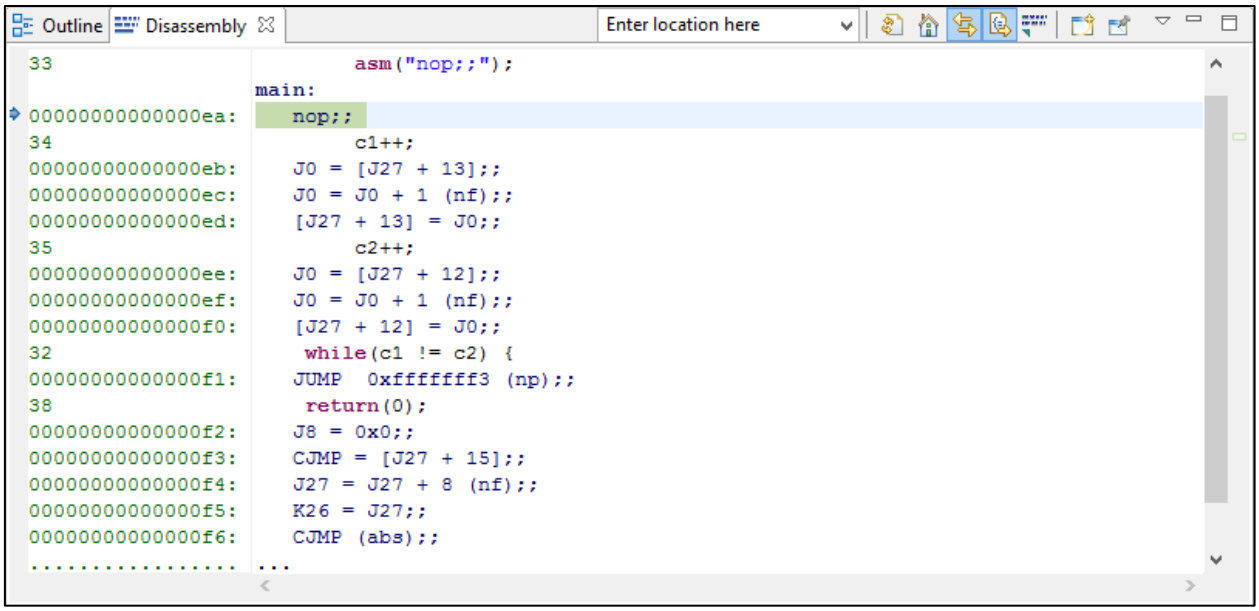


Рисунок 32. Вкладка «Disassembly»

Таблица 34. Команды на панели инструментов меню вкладки «Disassembly»

Значок	Название	Описание
	Refresh View	Обновить содержимое вкладки «Disassembly».
	Go to Current Program	Поставить курсор в то место, где произошла остановка программы.
	Link with Active Debug Context	Включить/отключить отслеживание предыдущих мест остановки программы.
	Show Source	Показать/скрыть исходный текст.
	Export To Text File	Выгрузить данные дисассемблера в текстовый файл.
	Open New View	Открыть новую вкладку.
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.

Таблица 35. Команды контекстного меню вкладки «Disassembly»

Значок	Название	Описание
	Show Source	Показать/скрыть исходный текст.
	Show Symbols	Показать/скрыть метки.
	Run to Line	Возобновить выполнение программы до достижения указанной строки. Это удобный способ приостановить выполнение на строчке без установки точки останова.
	Move To Line	Перенести выполнение на указанную строку (см. п. 6.11.5).
	Resume At Line	Запустить выполнение с указанной строки (см. п. 6.11.5).

Таблица 36. Команды контекстного меню области «Address»

Название	Описание
Disable/Enable Breakpoint	Отключить/включить точку останова.
Toggle Breakpoint	Поставить/внять точку останова.
Add Breakpoint	Открыть диалоговое окно для редактирования новой точки останова.
Breakpoint Properties	Открыть диалоговое окно для редактирования точки останова.
Show Addresses	Показать/скрыть адреса.
Show Opcodes	Показать/скрыть коды операций.

5.4.3.7. Вкладка «Memory»

Вкладка «**Memory**», расположенная на нижней панели окна, позволяет наблюдать и изменять содержимое ячеек памяти.

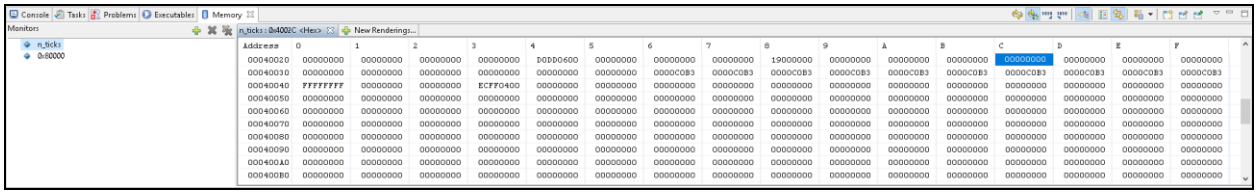


Рисунок 33. Вкладка «Memory»

Вкладка «**Memory**» содержит две панели: «**Monitors**» и «**Renderings**». Панель «**Monitors**» отображает список мониторов памяти, добавленных в сессию отладки, выбранную на вкладке «**Debug**». Содержимое панели «**Renderings**» определяется выбором монитора памяти на панели «**Monitors**» и представляет собой содержимое ячеек памяти в определенном формате. На панели «**Renderings**» одновременно могут отображаться несколько разных представлений данных в разных форматах.

Таблица 37. Кнопки панели инструментов вкладки «Memory»

Кнопка	Название	Описание функционирования
	Refresh	Обновить содержимое монитора.
	Automatic refresh	Автоматически обновлять монитор после каждой команды пошаговой отладки или останова.
	Import	Импортировать из других видов файлов во вкладку «**Memory**». Импорт доступен только в области памяти, размеченные как RAM.
	Export	Экспортировать содержимое вкладки «**Memory**» в файлы других типов.
	Toggle Memory Monitors Pane	Показать/скрыть панель «**Monitors**».
	Toggle Split Pane	Переключить разделение панели «**Renderings**».
	Link Memory Rendering Panes	Синхронизировать выбор адреса и формат представлений.
	Switch Memory Monitor	Переключить на другой монитор для просмотра, когда активны более одного монитора памяти.
	New Memory View	Создать новую вкладку «**Memory**».
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.
	Pin Memory Monitor	Закрепить монитор памяти поверх всех других мониторов памяти.
	View Menu → Layout	Переключить отображение панелей «**Monitors**» и «**Rendering**» с горизонтального вида на вертикальный и наоборот.

Кнопка	Название	Описание функционирования
–	View Menu → Preferences	Открыть окно «**Preferences**», позволяя выполнить следующие настройки: ▪ **Reset Memory Monitor** – перезагрузка только видимых или всех рендерингов при перезагрузке монитора памяти; ▪ **Padded String** – указывается неосвобождаемая память; ▪ **Select Codepages** – указывается стандарт для конвертирования памяти ASCII и EBCDIC строки.
–	View Menu → Table Rendering Preferences	Установить режим загрузки памяти, чтобы использовать при прокрутке рендера к концу буфера.
–	View Menu → Traditional Rendering Preferences	Открыть панель «**Traditionally Memory Rendering**» в диалоговом окне «**Preferences**».
–	View Menu → Floating Point Rendering Preferences	Открыть панель «**Floating Point Rendering**» в диалоговом окне «**Preferences**».

Таблица 38. Команды контекстного меню панели «Monitors»

Значок	Название	Описание
	Add Memory Monitor	Добавить новый адрес или переменную памяти на панель «**Monitors**».
	Remove Memory Monitor	Удалить выбранный адрес или переменную памяти на панели «**Monitors**».
	Reset	Перезагрузить панель монитора памяти.

Таблица 39. Команды контекстного меню панели «Renderings»

Значок	Название	Описание
	Add Rendering	Добавить вкладку «**Renderings**» для отображения монитора памяти по другому адресу или в другом формате.
	Remove Rendering	Удалить выбранный рендеринг на панели «**Memory**».
	Panes	Только для «**Traditional**». Включить или отключить отображение частей представления - адреса, данные, текст.
	Endian	Только для «**Floating Point**» и «**Traditional**». Установить порядок байт числа: **Little** или **Big Endian**. Применяется для заданной разрядности числа.
	Floating Point Type	Только для «**Floating Point**». Установить разрядность числа: **32-bit Single Precision** или **64-bit Double Precision**.
	Displayed Precision	Только для «**Floating Point**». Установить точность отображения.
	Text	Только для «**Traditional**». Выбрать кодировку текстовой части представления.
	Cell Size	Только для «**Traditional**». Установить разрядность столбца (чисел) данных: (1, 2, 4, 8) байт.

Значок	Название	Описание
	Radix	Только для «**Traditional**». Установить формат чисел (**Hex**, **Decimal Signed**, **Decimal Unsigned**, **Octal**, **Binary**).
	Columns	Только для «**Floating Point**» и «**Traditional**». Установить количество чисел в строке.
	Update Mode	Только для «**Floating Point**» и «**Traditional**». Задать режим обновления при пошаговой отладке: ▪ **Always** (автоматический), ▪ **On Breakpoint** (по точке останова), ▪ **Manual** (вручную по команде **Refresh**).
	Reset To Base Address	Перезапустить окно «**Renderings**» до оригинальных базовых адресов.
	Go To Address	Открыть окно редактирования для ввода адреса памяти. Нажмите **Enter**, чтобы перейти по этому адресу и отобразить его на панели «**Renderings**».
	Format	Кроме «**Floating Point**» и «**Traditional**». Установить разрядность отображаемой строки и столбца данных в 32-х битных словах
	Hide/Show Address Column	Показать/скрыть первую колонку, в которой указан адрес памяти.
	Copy To Clipboard	Скопировать выбранную часть окна «**Renderings**» в буфер обмена.
	Add Watchpoint (C/C++)...	Открыть диалоговое окно «**Properties for C/C++ Watchpoint**».
	Endianness	Кроме «**Floating Point**» и «**Traditional**». Установить порядок байт данных: **Little Endian** (по умолчанию) или **Big Endian**. В представлениях **HEX** и **ASCII** данные отображаются как байтовый или символьный массив, а установка порядка байт применяется к 32-битным словам. В представлениях **Hex Integer**, **Signed Integer** и **Unsigned Integer** порядок применяется к заданной разрядности столбца.

5.4.3.8. Вкладка «Memory Map»

Вкладка «**Memory Map**», расположенная на нижней панели окна, позволяет работать с картами памяти. Работа с картой виртуальной памяти поддерживается только для 1967ВН058, работа с картами физической памяти доступна также для 1967ВН028 и 1967ВН044.

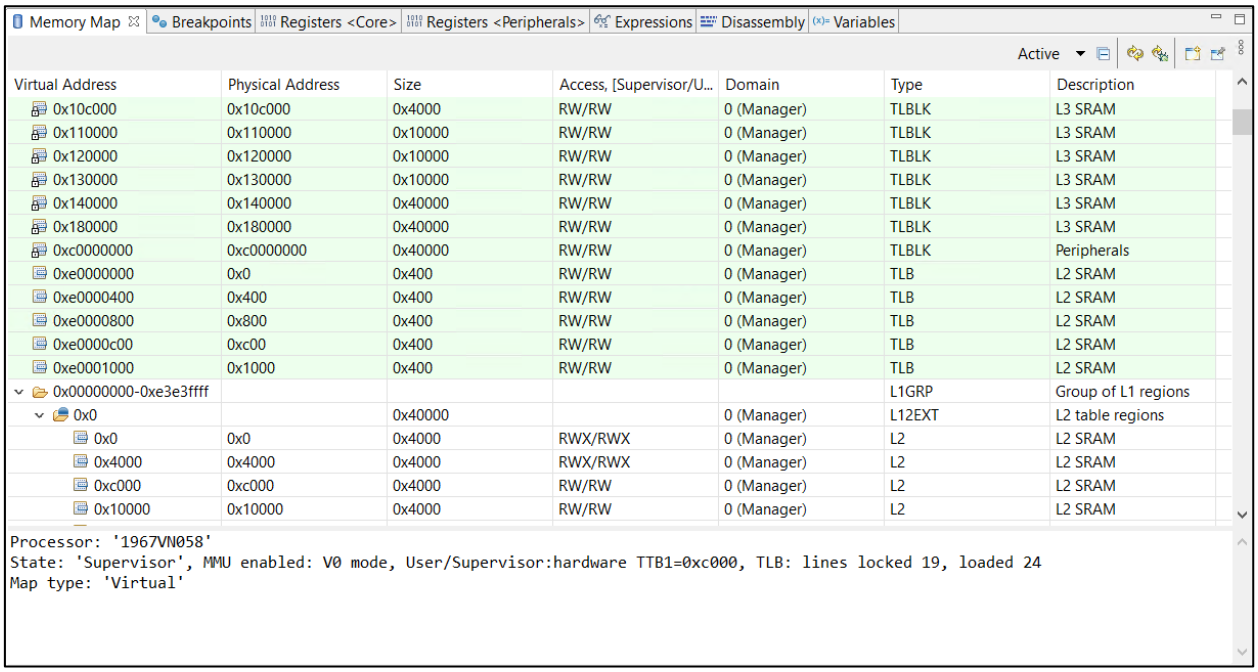


Рисунок 34. Вкладка «Memory Map»

Таблица 40. Кнопки панели инструментов вкладки «Memory Map»

Кнопка	Название	Описание функционирования
	Memory Map Type	Переключить режим отображения карты памяти. Доступны следующие режимы: ▪ Active – активная карта паяти; ▪ Physical – карта физической памяти; ▪ Virtual – карта виртуальной памяти. По умолчанию выбран режим отображения активной карты памяти «Active».
	Refresh	Обновить содержимое вкладки.
	Automatic refresh	Автоматически обновлять содержимое после каждой команды пошаговой отладки или останова.
	Open New View	Создать новую вкладку «**Memory Map**».
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.
	View Menu → Layout	Открыть список для выбора представления параметров во вкладке. Команда «**Layout** -> **Select Columns Alignment**» позволяет настроить выравнивание («Left», «Center», «Right») для любого столбца таблиц.

5.4.3.9. Вкладка «Trace Core»

Вкладка «**Trace Core**», расположенная на нижней панели окна, позволяет работать с буфером трассировки.

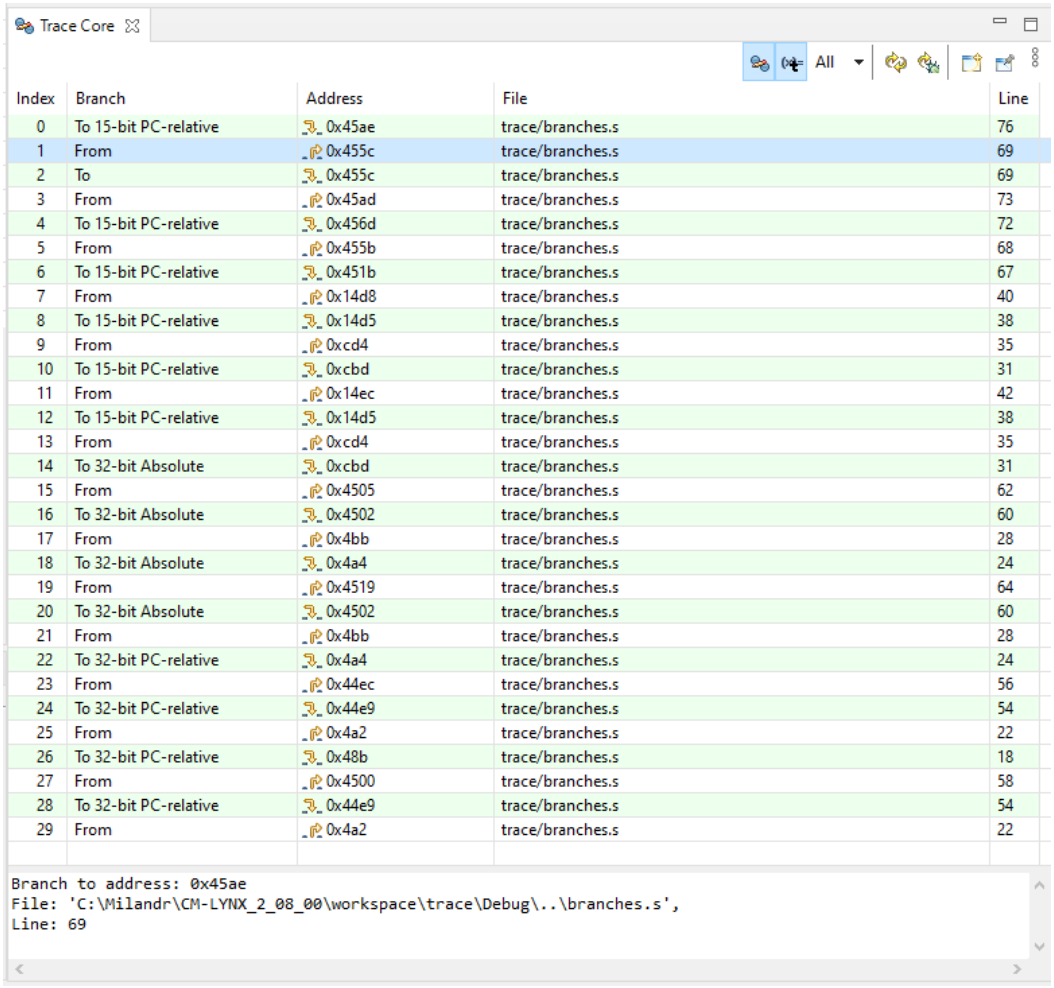


Рисунок 35. Вкладка «Trace Core»

Таблица 41. Кнопки панели инструментов вкладки «Trace Core»

Кнопка	Название	Описание функционирования
	State and control of core tracing for selected debug context	Кнопка состояния и управления работой буфера трассировки.
	Append calculated branch destination addresses if available	Дополнять переходы по абсолютному или относительному адресу (для которых в аппаратном буфере есть только запись «откуда») адресами куда произошёл переход.
	Type of data to display	Переключить режим отображения содержимого буфера. Доступны следующие варианты: ▪ All – все записи; ▪ Branch instructions – только адреса, откуда произошёл переход; ▪ Branch destinations – только адреса, куда произошёл переход. По умолчанию выбрано отображение всех записей «All».
	Refresh	Обновить содержимое вкладки.
	Automatic refresh	Автоматически обновлять содержимое после каждой команды пошаговой отладки или останова.
	Open New View	Создать новую вкладку «**Trace Core**».

Кнопка	Название	Описание функционирования
	Pin to Debug Context	Закрепить окно за текущим выбранным контекстом.
	View Menu → Layout	Открыть список для выбора представления параметров во вкладке. Команда «**Layout** -> **Select Columns Alignment**» позволяет настроить выравнивание («Left», «Center», «Right») для любого столбца таблиц.

6. Выполнение программы

6.1. Загрузка и запуск программы

Приложение CM-LYNX запускается из исполняемого файла **cm-lynx.exe**. При запуске приложения необходимо указать путь для сохранения файла рабочего пространства (см. Рисунок 36).

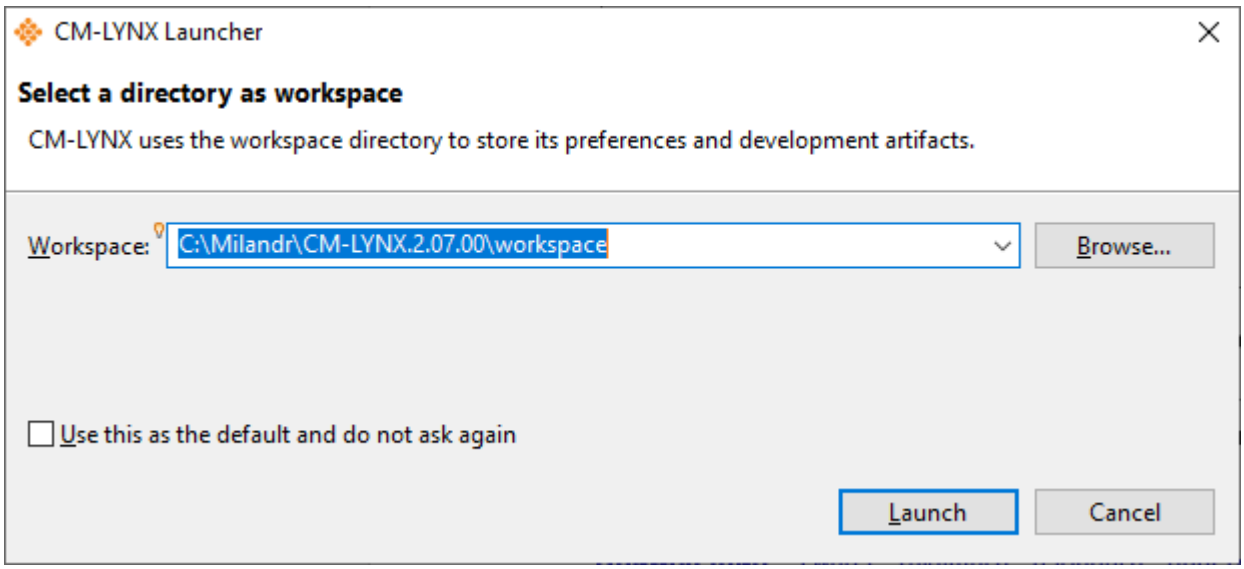


Рисунок 36. Выбор рабочего пространства при запуске приложения

По умолчанию директория «**Workspace**», в которой будут храниться проекты пользователя, располагается в корне директории с установленным приложением. Для изменения пути следует нажать кнопку «**Browse**», выбрать другую папку на ПК пользователя и нажать кнопку «**Выбор папки**».

Примечание: смена текущего рабочего пространства на рабочее пространство одной из предыдущих версий CM-LYNX настоятельно не рекомендуется. Для работы с проектами, созданными с помощью более ранних версий, необходимо воспользоваться стандартной процедурой импорта проектов (см. п. 6.4.5. Импорт существующего проекта). При импорте рекомендуется установить флаг «**Copy projects into workspace**». В результате импорта будет выполнена автоматическая конвертация проекта «устаревшего» формата. Кроме того, можно импортировать отладочные конфигурации из прежней директории «**Workspace**» (см. п. 6.11.1.2).

Если пользователь намерен и в дальнейшем использовать эту папку, то следует отметить опцию «**Use this as default and do not ask again**». В этом случае при последующих запусках приложения это окно больше не будет показываться пользователю.

После этого для загрузки среды разработки необходимо нажать на кнопку **«Launch»**.

Откроется главное окно CM-LYNX (см. Рисунок 37).

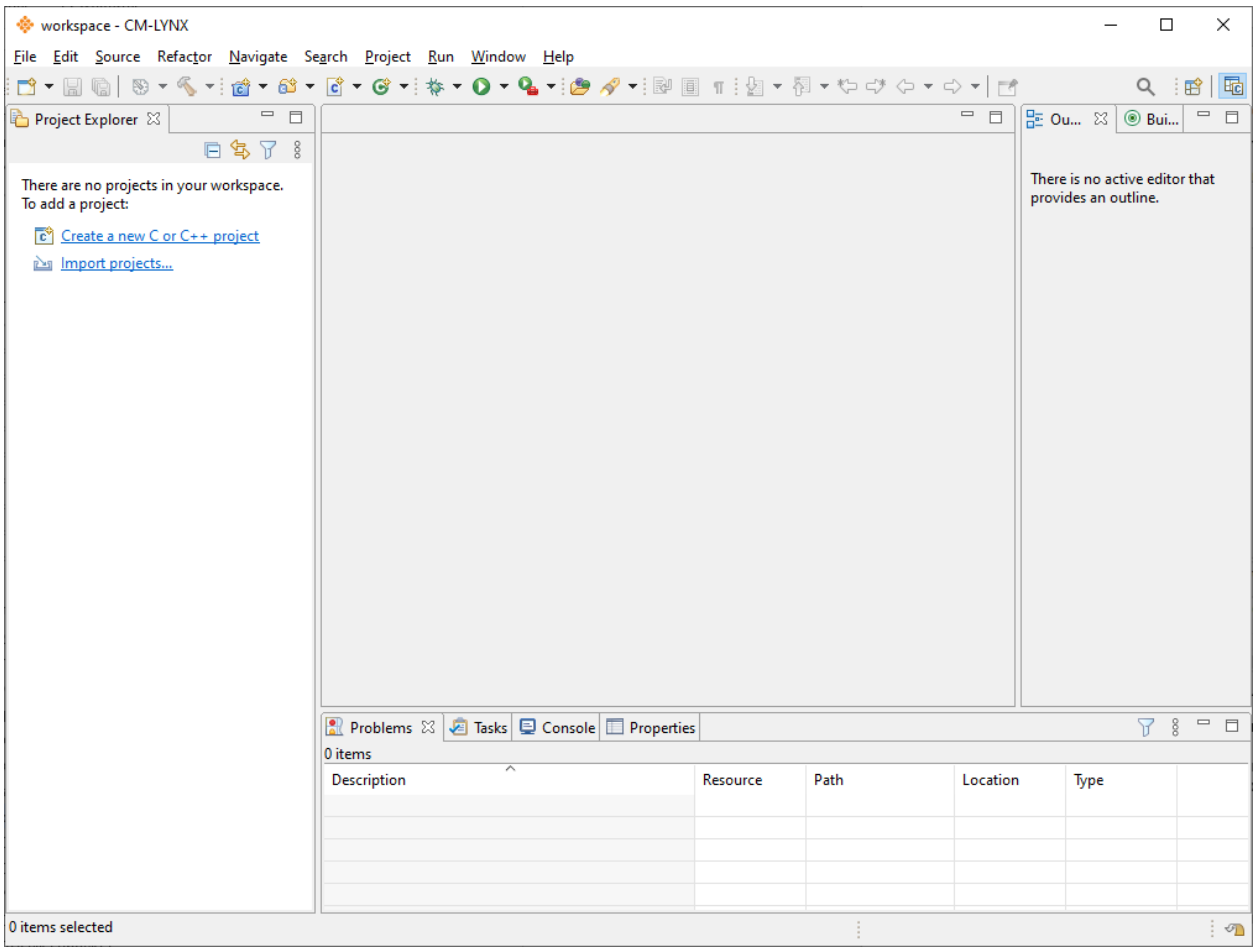


Рисунок 37. Главное окно CM-LYNX

В процессе работы пользователь может сменить рабочее пространство. Подробнее см. п. 6.4.12. Смена рабочего пространства.

6.2. Начало работы

Рассмотрим на простейшем примере процесс разработки приложения для процессоров 1967ВН028/1967ВН044 с помощью CM-LYNX и его отладки с помощью отладочной платы.

Чтобы создать и отладить простейшее приложение с помощью среды, необходимо выполнить следующие шаги:

1. Создать проект;
2. Написать код (например, создать C++ файл);
3. Настроить конфигурацию оборудования;
4. Проверить код и собрать проект;
5. Отладить проект.
6. Собрать рабочую версию.

6.2.1. Шаг 1. Создание проекта

Для того чтобы создать новый проект, необходимо:

1. Воспользоваться ссылкой **«Create a new C or C++ project»** на панели **«Project Explorer»** (см. Рисунок 37) или выбрать в меню **File-**> **New -**> **C/C++ Project** (см.).

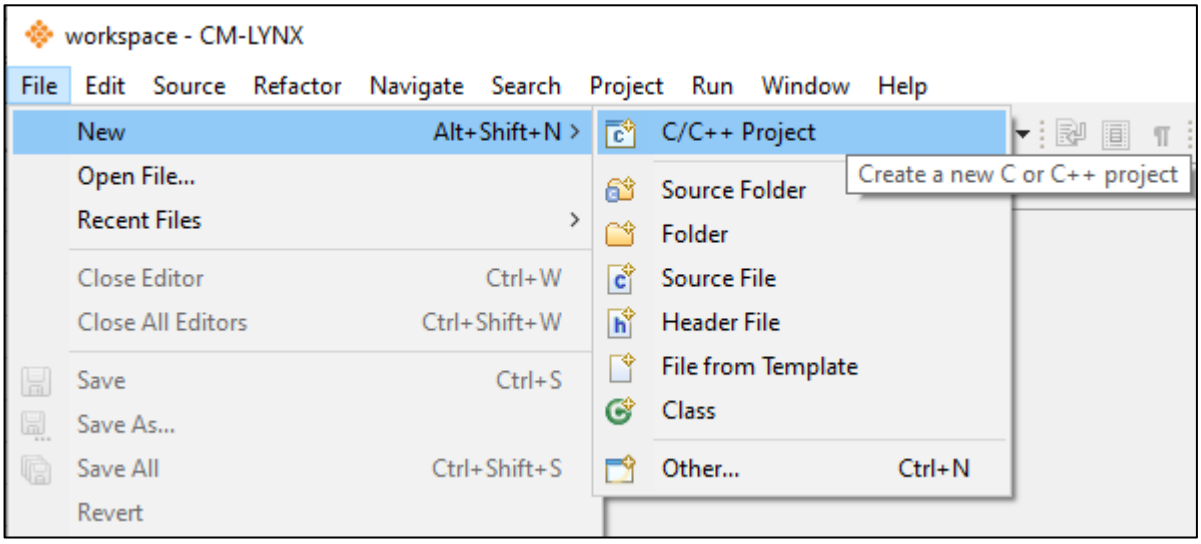


Рисунок 38. Создание проекта из меню

2. Выбрать шаблон и нажать кнопку «**Next**».

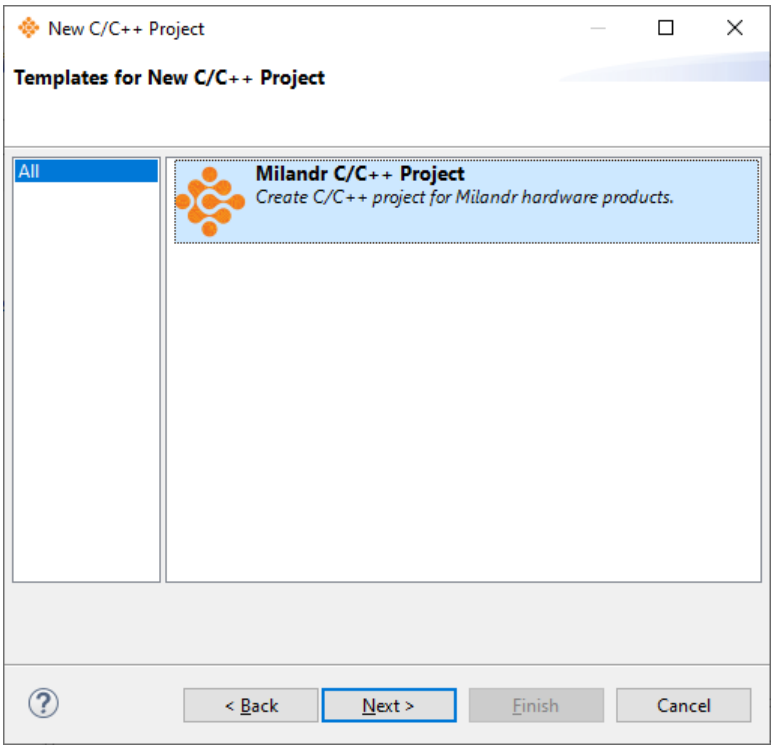


Рисунок 39. Выбор шаблона проекта

3. Ввести наименование проекта (в нашем примере - **Example**) и нажать кнопку «**Next**».

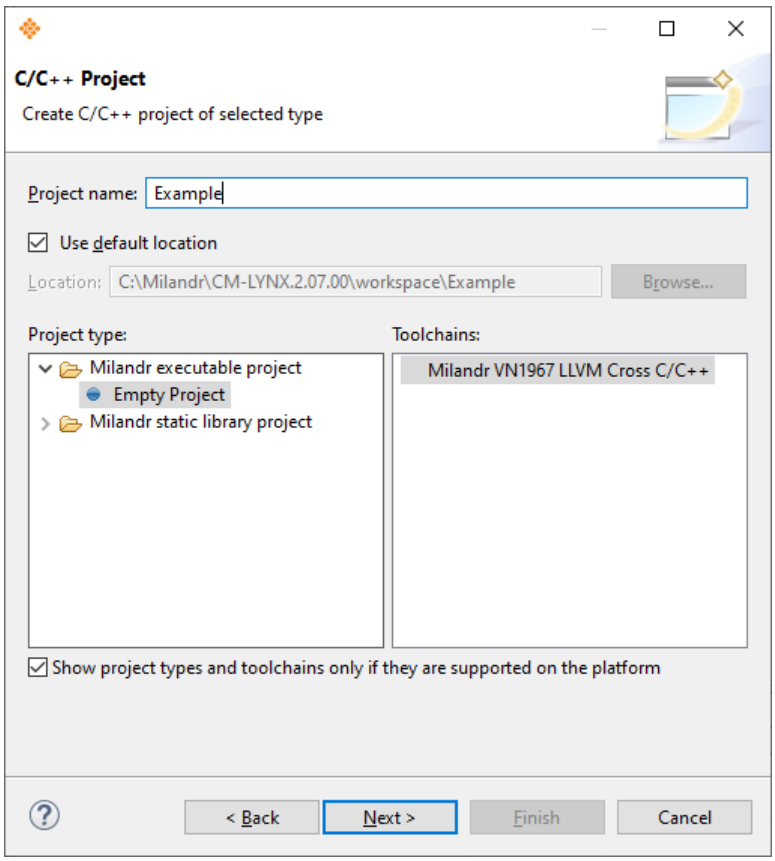


Рисунок 40. Добавление проекта «Example»

4. При необходимости изменить настройкиконфигурации и нажать кнопку «**Next**».

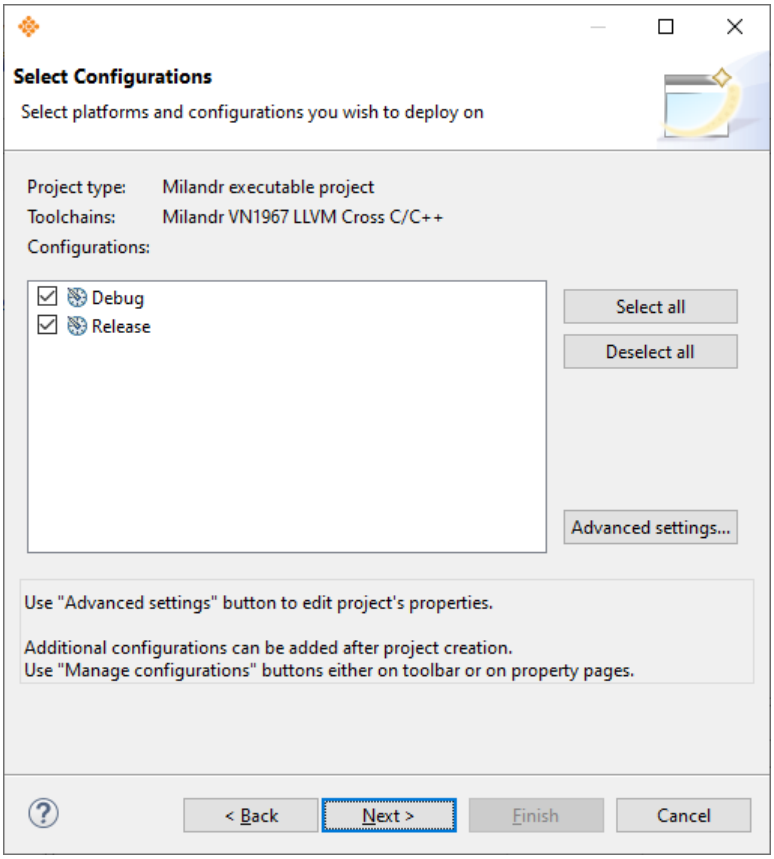


Рисунок 41. Выбор конфигурации

5. Выбрать процессор (по умолчанию указан **«1967VN044 Rev.3»**), способ подключения HAL, тип файла main и нажать кнопку **«Finish»**.

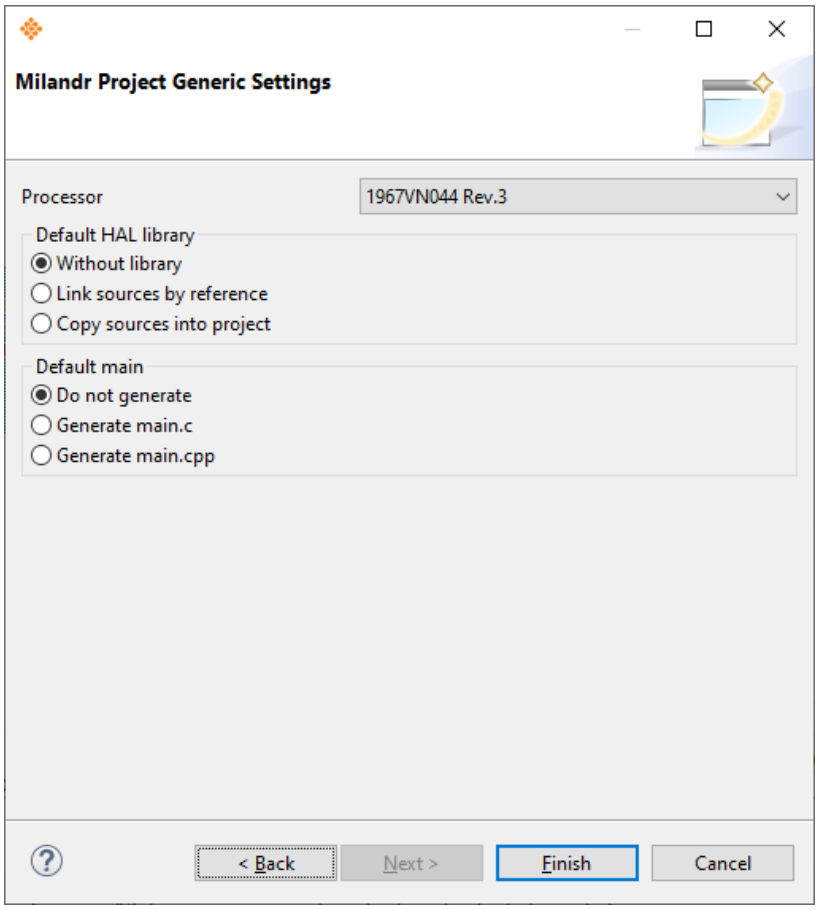


Рисунок 42. Выбор типа процессора, способа подключения HAL и типа файла main

Будет создан проект с настройками по умолчанию (см. Рисунок 43).

В каждом проекте присутствуют следующие специальные файлы:

- **`project.ldf`** – ldf-файл по умолчанию, описывающий процесс компоновки (при необходимости, можно изменить ldf-файл; подробнее см. в документе ML_MCLINK-TSH Руководство пользователя);
- **`session.lldbinit`** – файл, содержащий команды инициализации периферии до запуска отладки (при необходимости, можно изменить этот файл; подробнее см. в документе ML_LLDB Руководство пользователя).

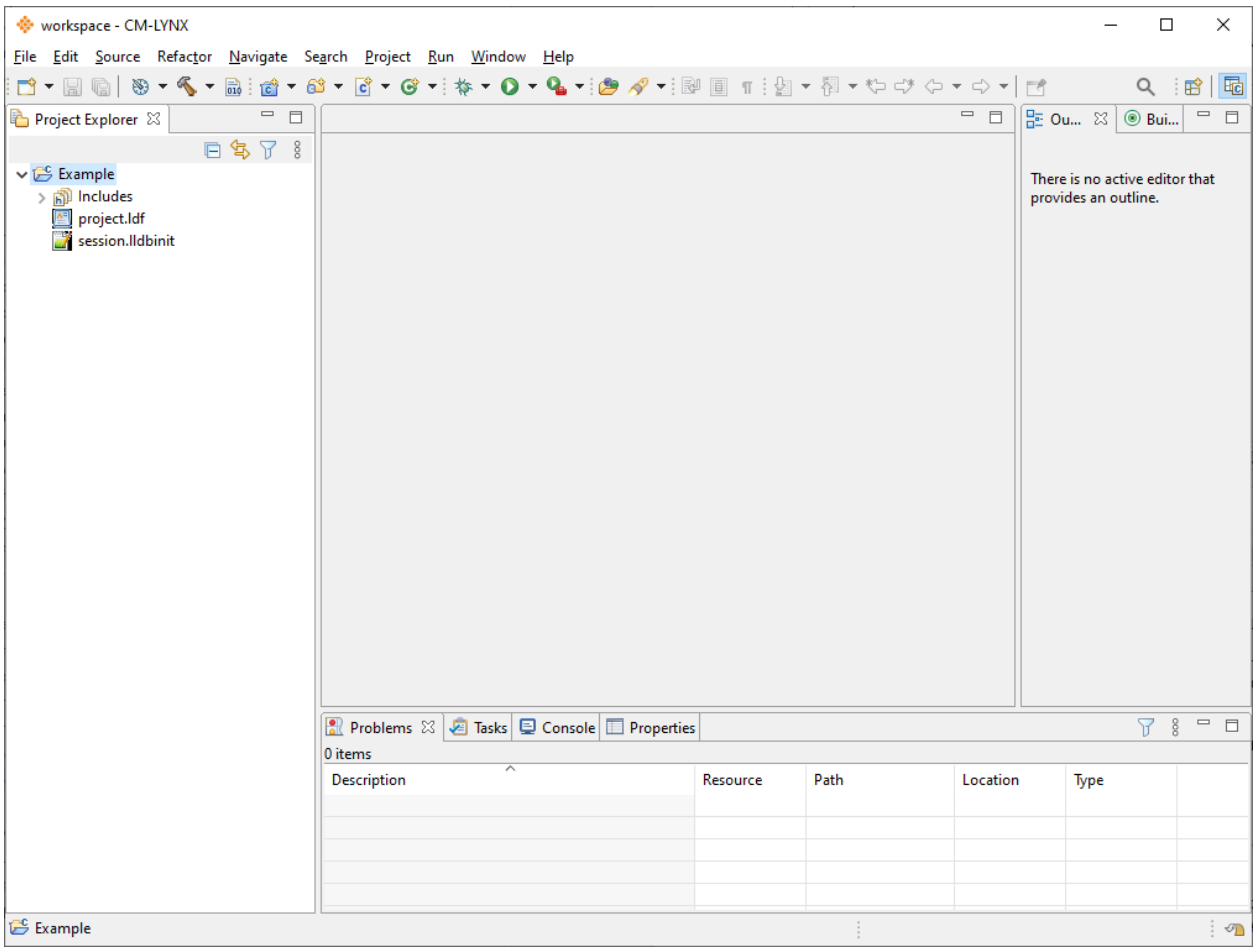


Рисунок 43. Результат Шага 1

Подробнее о создании проекта см. п. 6.4.1. Создание проекта. Также можно импортировать в среду существующий проект (см. 6.4.5. Импорт существующего проекта).

6.2.2. Шаг 2. Создание C++ файла и его редактирование

Файл main.cpp можно создать на этапе создания проекта, выбрав вариант **«Generate main.cpp»** для настройки **«Default main»** (см. Рисунок 42).

Если же на этапе создания проекта файл main.cpp не был создан или нужно создать другой файл, то его можно создать следующим способом:

1. На вкладке **«Project Explorer»** щелкнуть правой кнопкой мыши по папке с созданным в шаге 1 проектом и в контекстном меню выбрать команду **«New > Source File»**.

2. В поле «**Source file**» ввести имя файла (в нашем примере - «**main.cpp**»).

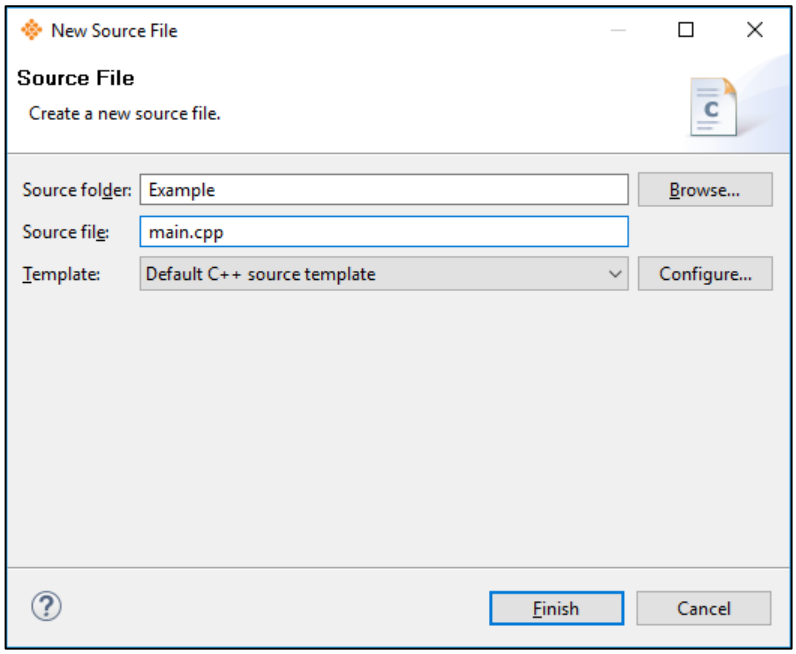


Рисунок 44. Создание файла main.cpp

3. Нажать кнопку «**Finish**».

   В редакторе появится пустой файл **main.cpp**.

Для создания простейшего приложения для процессора **1967ВН028 /1967ВН044** необходимо ввести код в окне редактора или скопировать его туда из данного руководства.

Пример кода простешего приложения для процесора **1967ВН028:**

```
#include <sysreg.h>
// Прототип функции задержки, реализованной ниже
void Delay(int waitTicks);
int main(void)
{
   int count = 0;
   // Основной цикл
   while(1)
   {
      __builtin_sysreg_write(__FLAGREGST, 0x01); // зажигаем светодиод FLAG0
      Delay (5000000);
      __builtin_sysreg_write(__FLAGREGCL, 0x00); // гасим светодиод FLAG0

      __builtin_sysreg_write(__FLAGREGST, 0x02); // зажигаем светодиод FLAG1
      Delay (5000000);
      __builtin_sysreg_write(__FLAGREGCL, 0x00);

      __builtin_sysreg_write(__FLAGREGST, 0x04); // зажигаем светодиод FLAG2
      Delay (5000000);
      __builtin_sysreg_write(__FLAGREGCL, 0x00);

      __builtin_sysreg_write(__FLAGREGST, 0x08); // зажигаем светодиод FLAG3
      Delay (5000000);
      __builtin_sysreg_write(__FLAGREGCL, 0x00);
      count++;
   }
}

//Функция задержки
void Delay(int waitTicks)
```

```
{
  volatile int i;
  for (i = 0; i < waitTicks; i++);
}
```

Ниже представлен пример кода прриложения для процессора **1967ВН044**:

```
#define base_GPC                    0x80001080
#define GPC_DR_SET_LOC              (base_GPC + 1)
#define GPC_DR_CLEAR_LOC            (base_GPC + 2)
#define GPC_DDR_SET_LOC             (base_GPC + 5)
#define GPC_DDR_CLEAR_LOC           (base_GPC + 6)
#define GPC_ALT_CLEAR_LOC           (base_GPC + 26)

// Прототип функции задержки, реализованной ниже
void Delay(int waitTicks);

int main(void)
{
      int count = 0;
      // Настройка GPIO порта
      * (int *) GPC_ALT_CLEAR_LOC = 0xF;
      * (int *) GPC_DDR_CLEAR_LOC = 0xF;
      * (int *) GPC_DR_CLEAR_LOC  = 0xF; // Светодиод инвертирован
отновистельно значения регистра DDR
      * (int *) GPC_DDR_SET_LOC   = 0xF; // Здесь все 4 светодиода на плате
загорятся
      // Основной цикл
      while(1)
      {
            * (volatile int *) GPC_DR_SET_LOC   = 0x1; // гасим светодиод
FLAG0
            Delay (5000000);
            * (volatile int *) GPC_DR_CLEAR_LOC = 0x1; // зажигаем светодиод
FLAG0
            * (volatile int *) GPC_DR_SET_LOC   = 0x2; // гасим светодиод
FLAG1
            Delay (5000000);
            * (volatile int *) GPC_DR_CLEAR_LOC   = 0x2;
            * (volatile int *) GPC_DR_SET_LOC   = 0x4; // гасим светодиод
FLAG2
            Delay (5000000);
            * (volatile int *) GPC_DR_CLEAR_LOC   = 0x4;
            * (volatile int *) GPC_DR_SET_LOC   = 0x8; // гасим светодиод
FLAG3
            Delay (5000000);
            * (volatile int *) GPC_DR_CLEAR_LOC   = 0x8;
            count++;
            }
}

//Функция задержки
void Delay(int waitTicks)
{
      volatile int i;
      for (i = 0; i < waitTicks; i++);
}
```

Для сохранения изменений нажмите комбинацию клавиш «**Ctrl+S**».

Файл **main.cpp** и входящие в него C++ объекты отобразятся в виде иерархии на вкладке «**Project Explorer**», также основные C++ объекты отобразятся в виде иерархии на вкладке «**Outline**» (см. Рисунок 45).

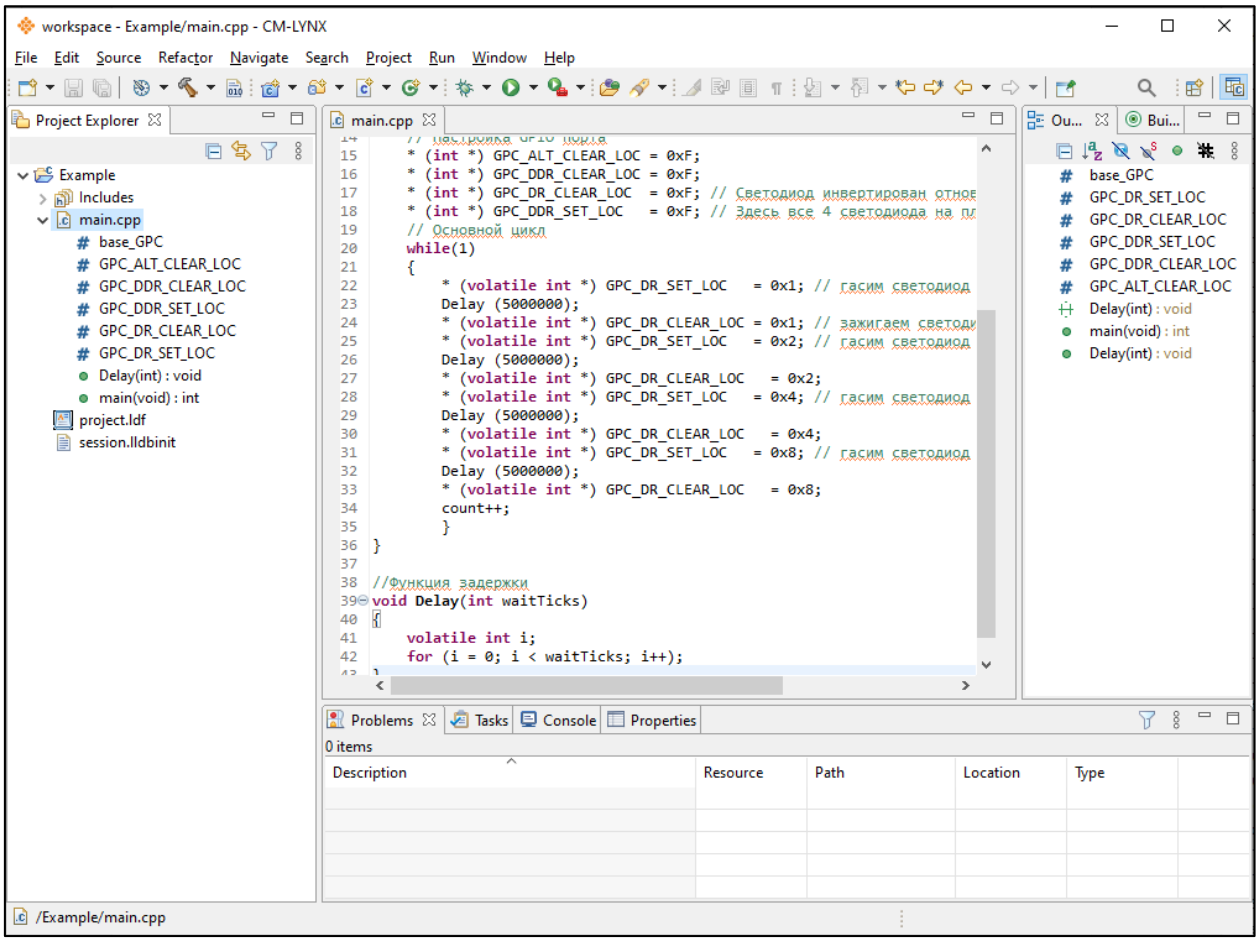


Рисунок 45. Результат Шага 2 (добавлен код для процессора 1967ВН044)

Проект может содержать любое количество файлов, которые могут быть сгруппированы в папки. О том, как создать файл, см. также п. 6.4.3. Создание файла; папку – см. п. 6.4.2. Создание папки.

Для редактирования файлов используются редакторы; подробнее о работе с редакторами см. п. 6.6. Работа с редакторами.

6.2.3. Шаг 3. Изменение конфигурации и параметров сборки

Целевой процессор и способ подключения HAL задаются на шаге создания прокта (см. 6.2.1. Шаг 1. Создание проекта).

Для того чтобы сменить целевой процессор или изменить настройки конфигурации, необходимо:

1. На вкладке «**Project Explorer**», установив курсор на проект, вызвать контекстное меню правой клавишей мыши и выбрать пункт «**Properties**».

2. Выбрать группу настроек «**C/C++ Build->Settings->Tool Settings->Processor**».

3. В поле «**Processor**» выбрать значение «**1967VN028 Rev.2**» (Рисунок 46). Здесь же можно изменить размер используемых типов (по умолчанию, **char=32** и **double=64**).

   Для процессора 1967ВН044 также можно изменить размер используемых типов (по умолчанию, char=32 и double=64) (Рисунок 47).

4. Нажать кнопку «**Apply**».

5. Нажать кнопку «**OK**».

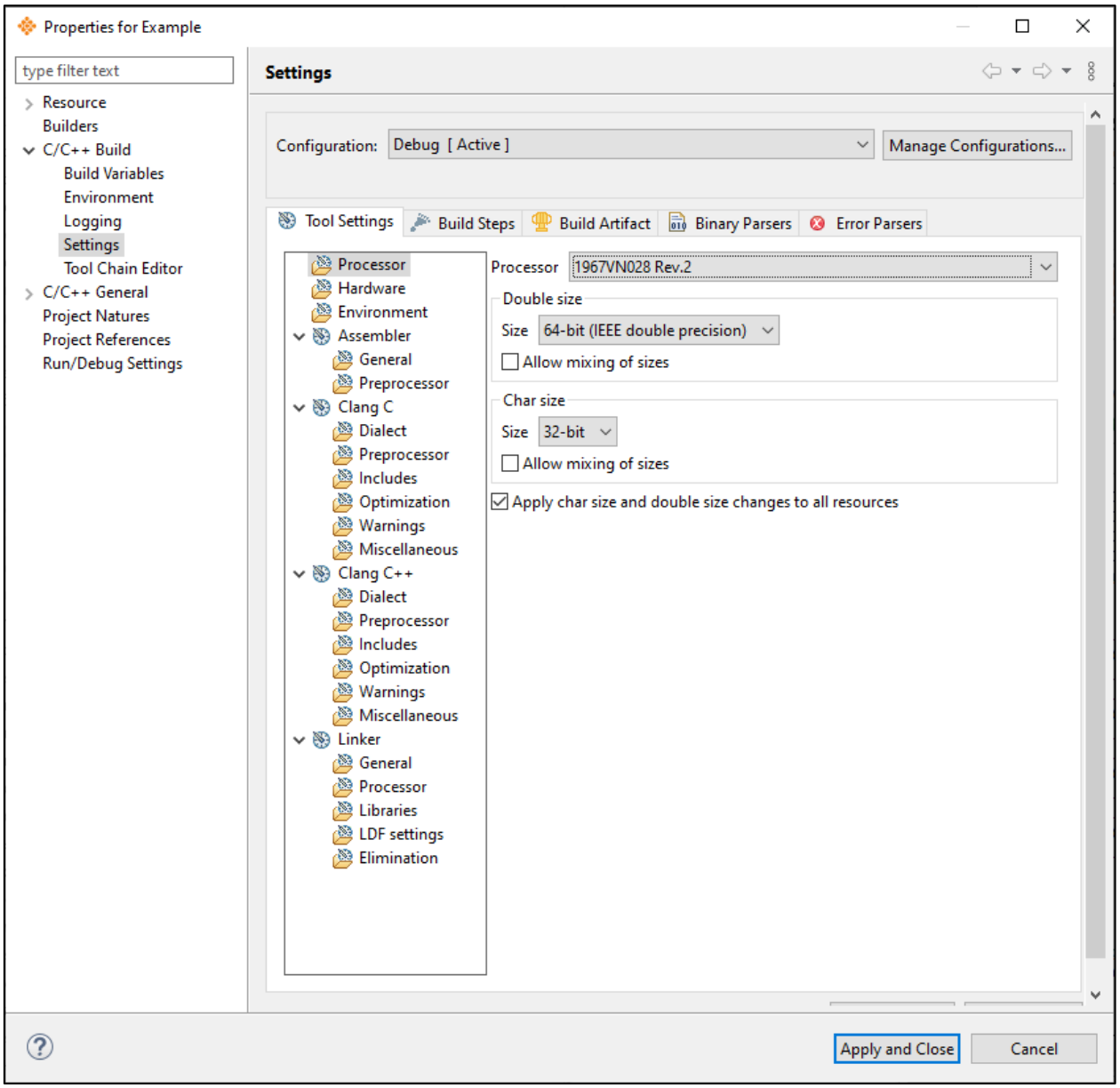


Рисунок 46. Настройка конфигурации оборудования для 1967ВН028 процессора

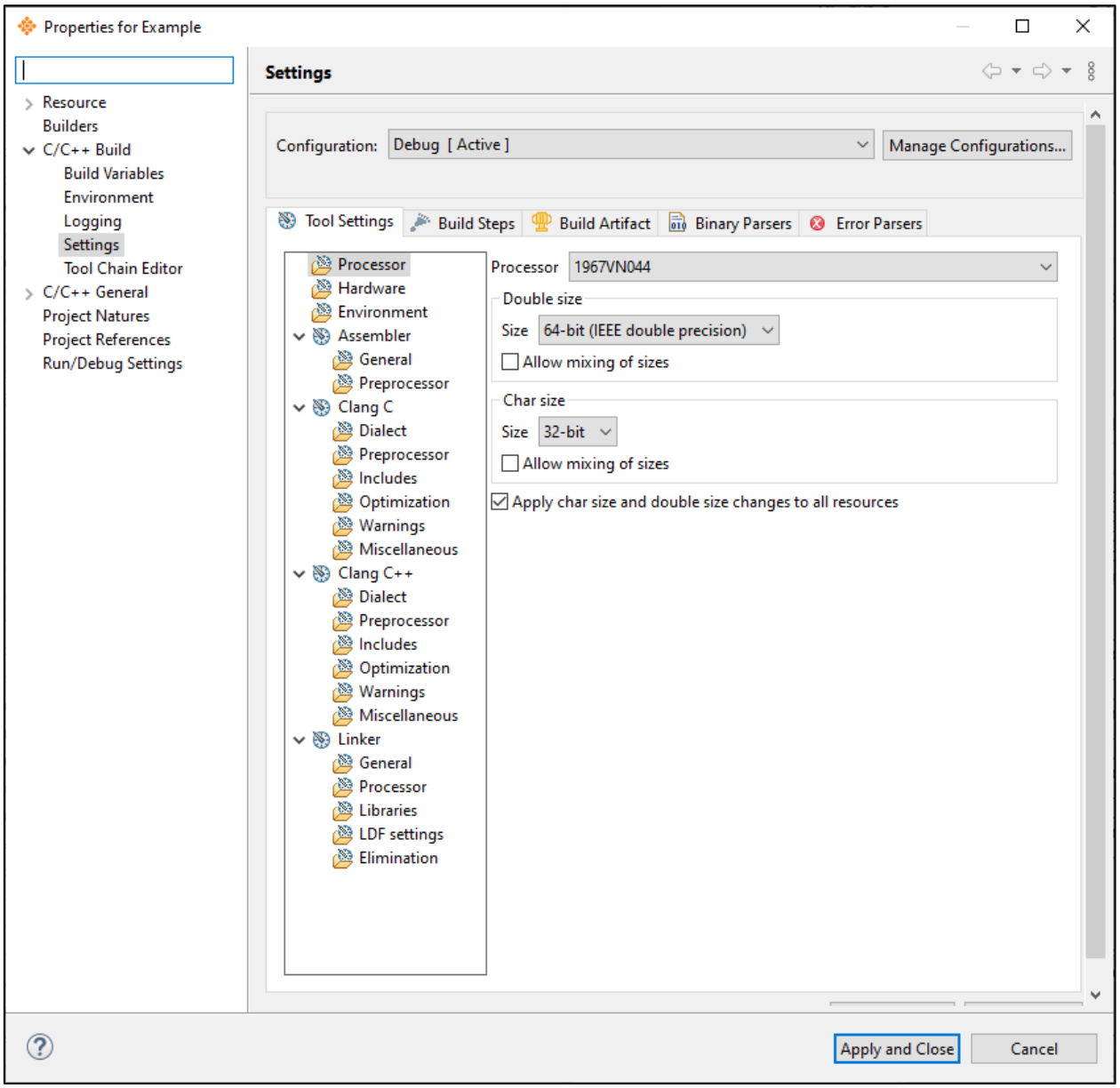


Рисунок 47. Настройка конфигурации оборудования для 1967ВН044 процессора

Помимо настройки конфигурации оборудования, может потребоваться выполнить, например, следующие настройки проекта:

- изменить тип оптимизации компилятора (по умолчанию, указана оптимизация **O0**);
- подключить **HAL**, **libdsp** или иные библиотеки (по умолчанию, подключена только библиотека **clib**);
- включить функционал исключений C++;
- отключить исключение неиспользуемого кода.

Подробнее о настройках проекта см. п. 6.9. Настройка свойств проекта. О том, как изменить настройки компилятора, ассемблера, компоновщика – см. п. 6.9.2.4.1. Настройки компонентов тулчейн.

6.2.4. Шаг 4. Сборка проекта

Для сборки проекта необходимо нажать комбинацию клавиш **«CTRL+B»** или в контекстном меню на вкладке **«Project Explorer»** выбрать команду **«Project -> Build Project»**.

Результат сборки можно проверить на вкладке «Console»:

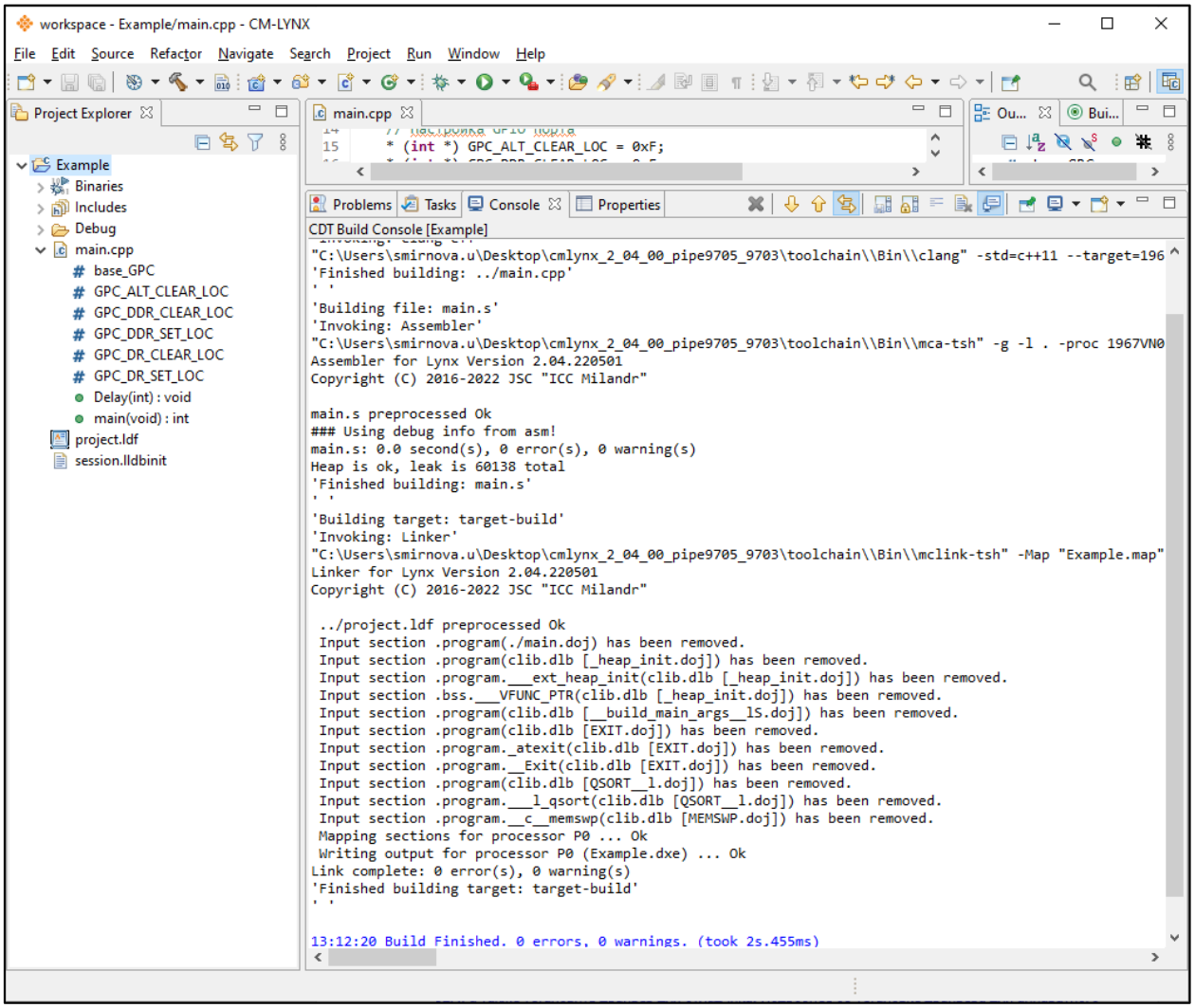


Рисунок 48. Результат Шага 3

Подробнее о сборке проектов см. п. 6.10. Сборка проекта.

6.2.5. Шаг 5. Отладка проекта

Примечание: перед отладкой необходимо подключить отладочную плату с аппаратным отладчиком JEM, а также установить драйвер для отладчика. Подробнее об установке драйвера для аппаратного отладчика JEM см. Инструкцию по установке.

Для того чтобы отладить проект для процессора **1967ВН028**, необходимо:

1. В меню «**Debug**»  выбрать пункт «**Debug Configurations...**».

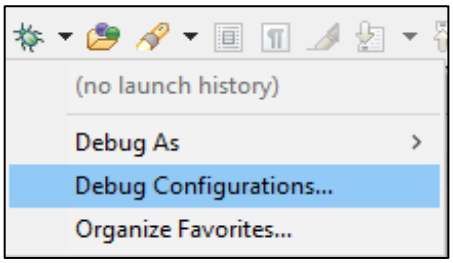


Рисунок 49. Команда настройки конфигурации отладки

2. В открывшемся окне выбрать автоматически созданную конфигурацию **«Example Debug»** или создать новую конфигурацию отладки, дважды щелкнув по наименованию конфигурации отладки **Milandr DSP1967 LLDB**.

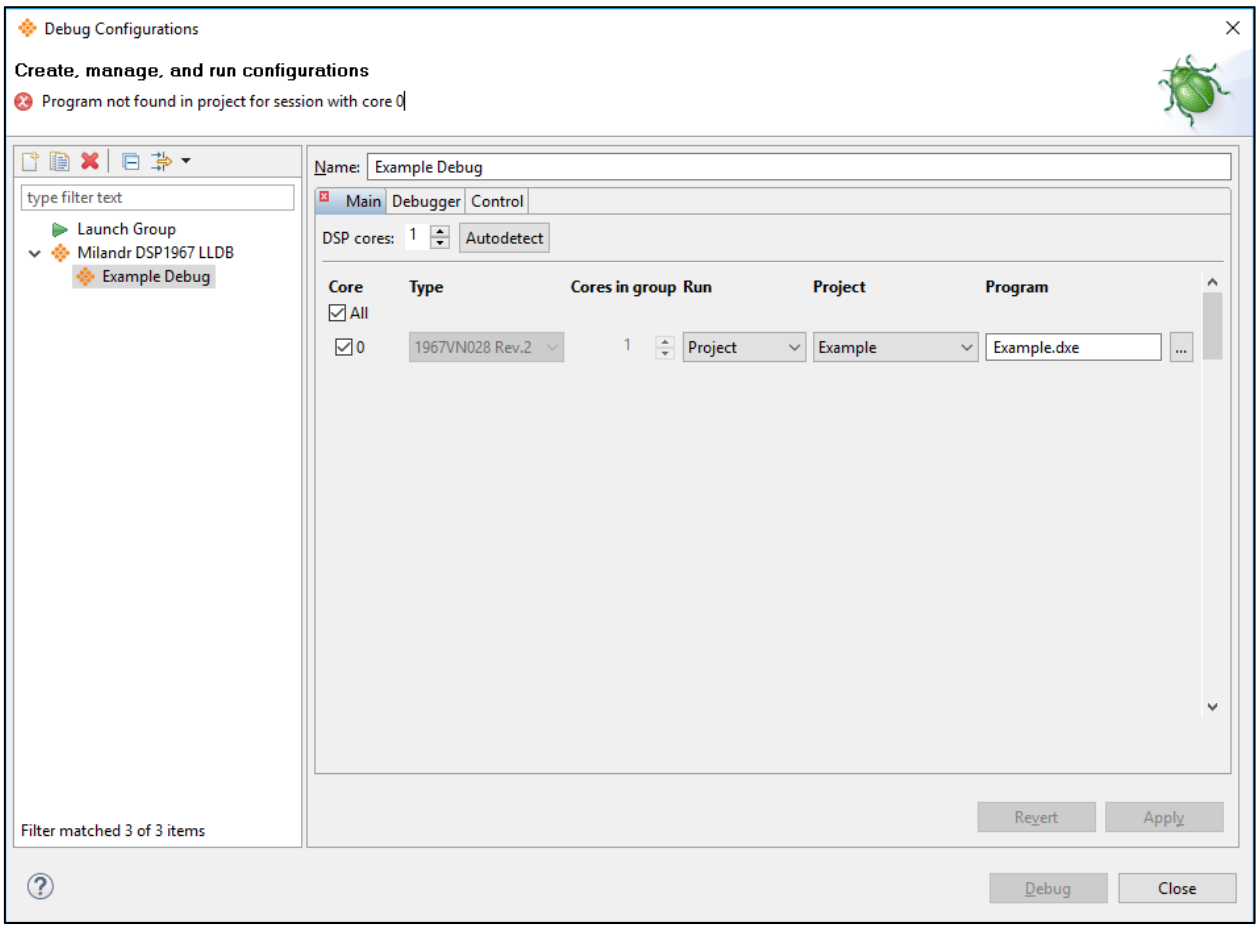


Рисунок 50. Создание новой конфигурации отладки

3. Нажать на кнопку «**Debug**» для запуска сессии отладки.

4. Нажать на кнопку «**Yes**» для перехода в режим отладки.

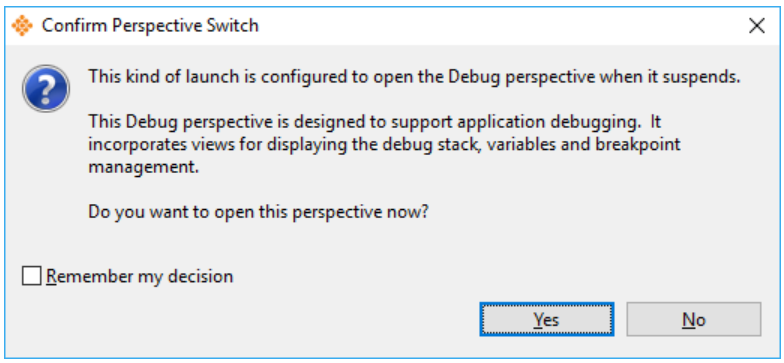


Рисунок 51. Окно подтверждения перехода в режим отладки

Произойдет переход в режим отладки: внешнее представление окна приложения изменится.

Примечание: рекомендуется, в режиме отладки настроить представление, показанное на Рисунок 52. Подробнее см. п. 6.3.1.2. Режим отладки (перспектива «Debug»).

На вкладке «**Debug**» можно увидеть, что отладка остановится на **main**:

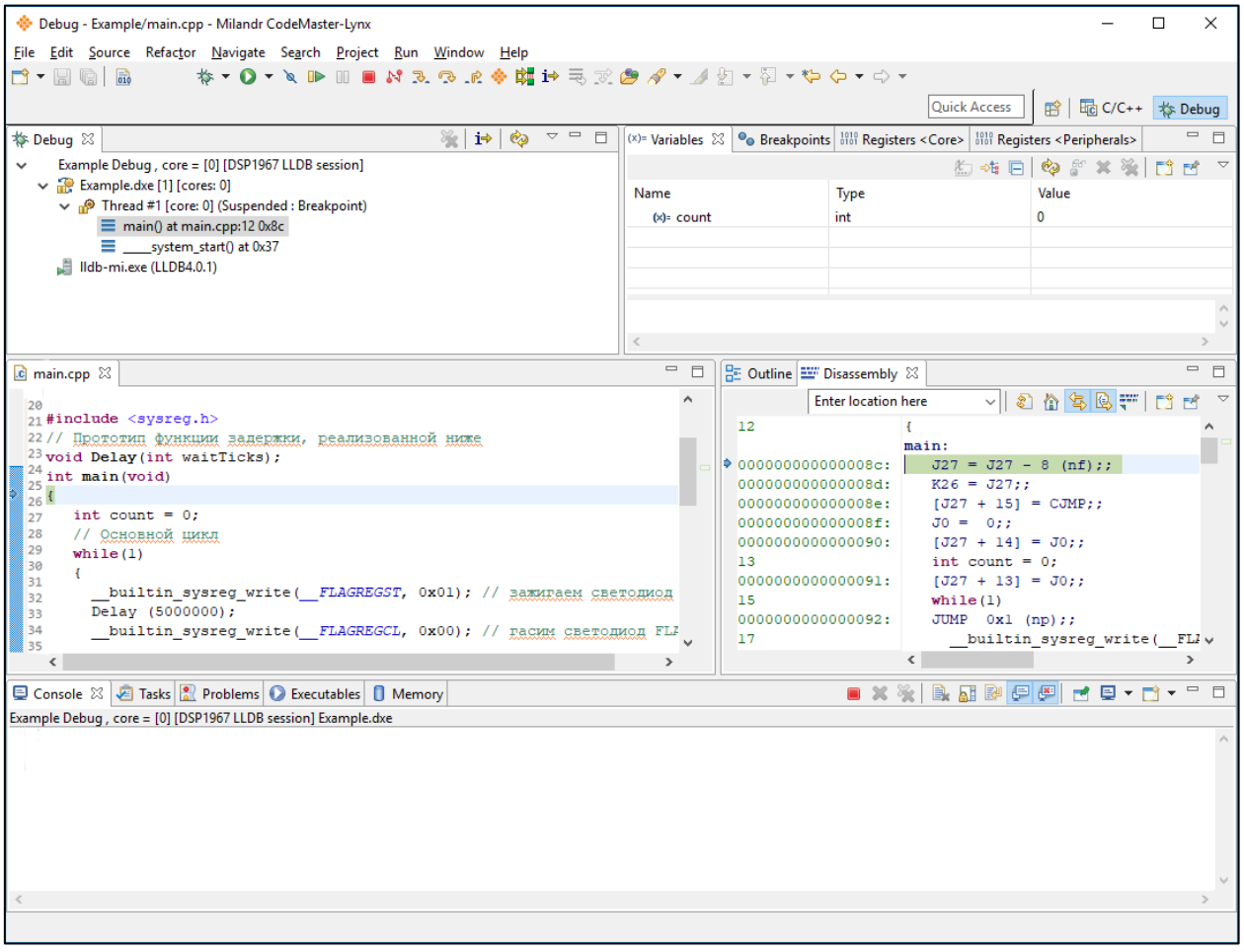


Рисунок 52. Результат запуска отладки

5. Установить точки останова (всего 4 шт.) на строки «**Delay (5000000)**». Для этого следует дважды щелкнуть по синей полосе напротив нужной строки в окне редактора. Слева появится синяя точка.

6. Нажать на кнопку (первый раз).

Убедиться, что:

- отладка остановится при достижении точки останова;
- на отладочной плате загорится первый светодиод;
- в окне «**Variables**» значение переменной **count**=0

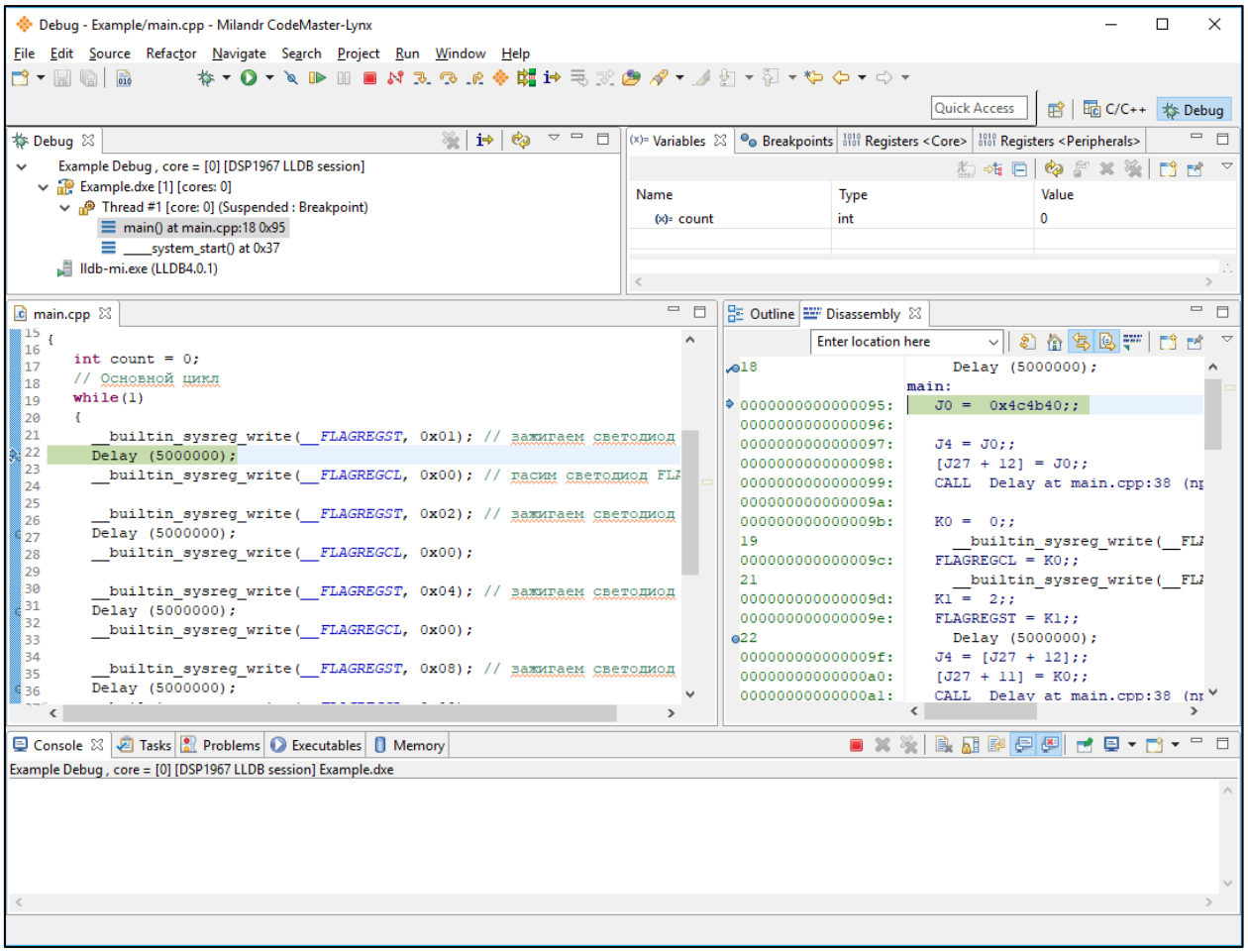


Рисунок 53. Состояние среды при останове на точки останова 1

Для удобства работы добавим переменную на вкладку «**Expressions**». Для этого в контекстном меню на вкладке «**Variables**» следует выбрать пункт «**Watch**». Будет добавлено выражение на вкладке «**Expresions**»:

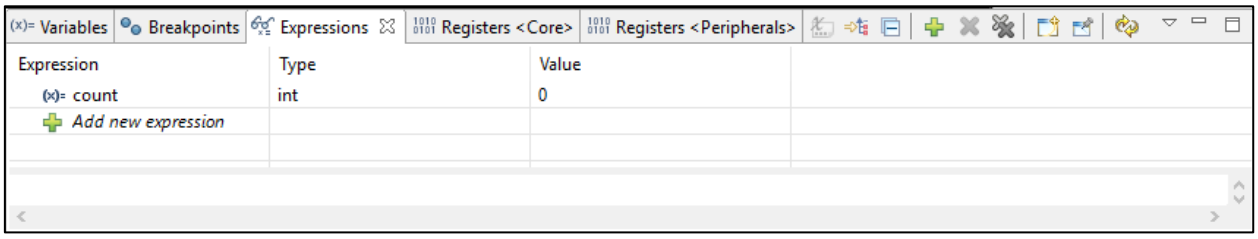


Рисунок 54. Добавление переменной на вкладку «Expresions»

7. Перейти на вкладку «**Registers <Core>**» и проверить, что значение регистра **FLAGREG** (из группы регистров **Sequence**r) равно **0x00000001**.

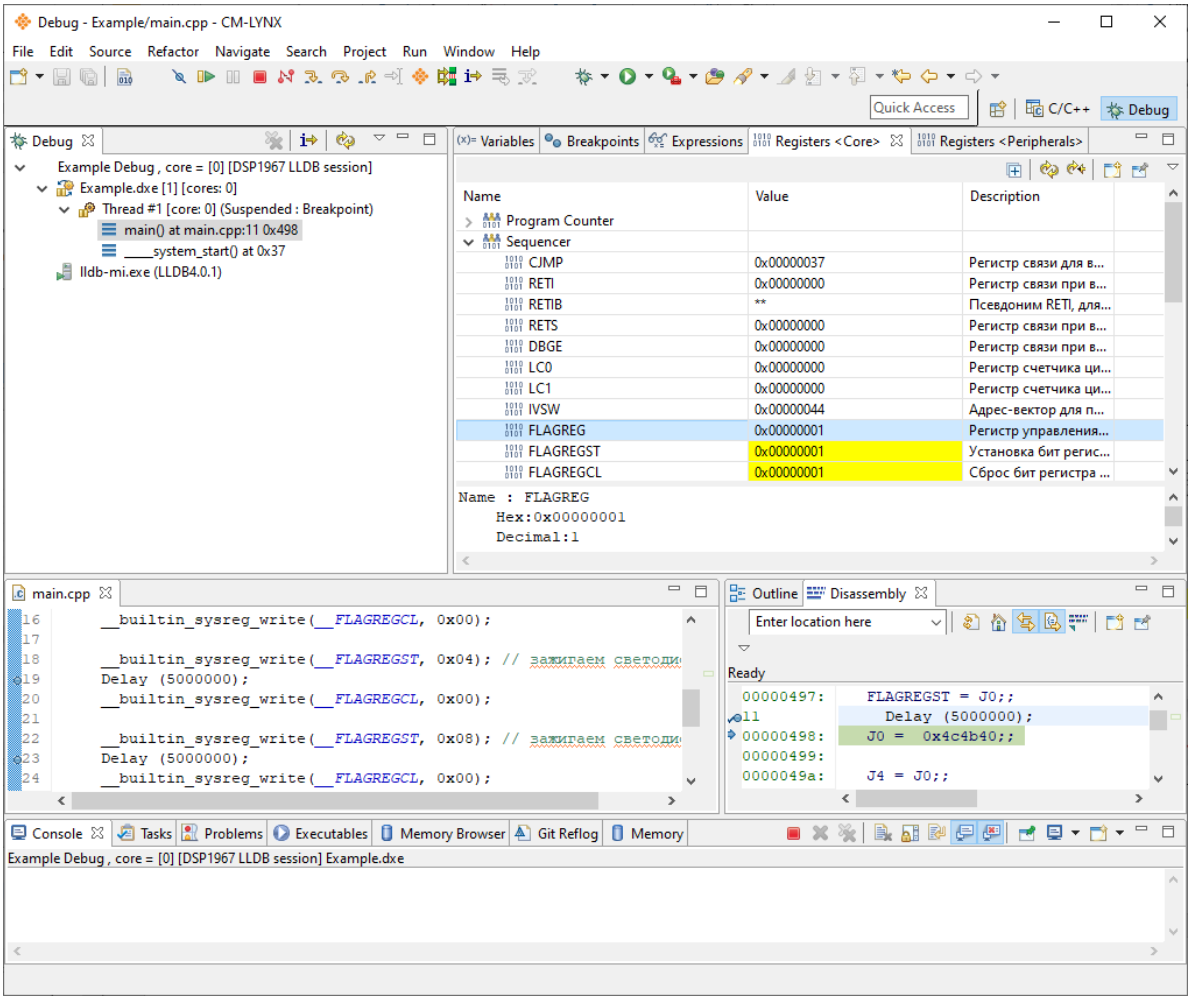


Рисунок 55. Состояние среды при останове на точки останова 1 (просмотр регистров)

Для удобства работы добавим регистр на вкладку «**Expressions**». Для этого в контекстном меню следует выбрать пункт «**Watch**». Будет добавлено выражение:

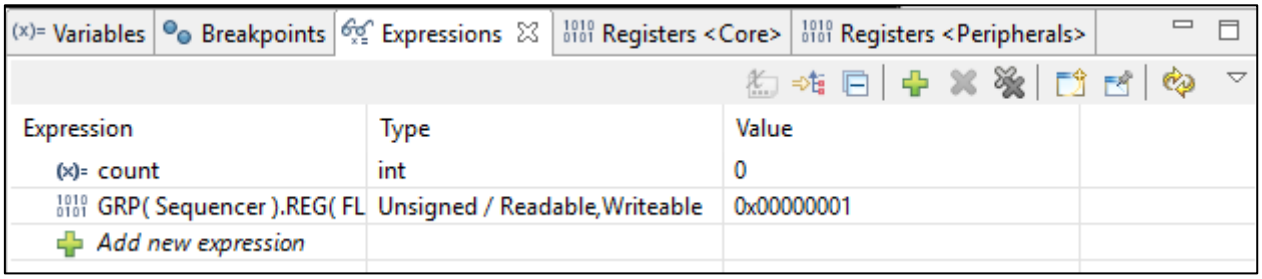


Рисунок 56. Добавление регистра на вкладку «Expresions»

8. Нажать на кнопку .

   Убедиться, что:

   - первый светодиод погас, а второй загорелся;
   - изменилось значение регистра;
   - значение переменной не изменилось.

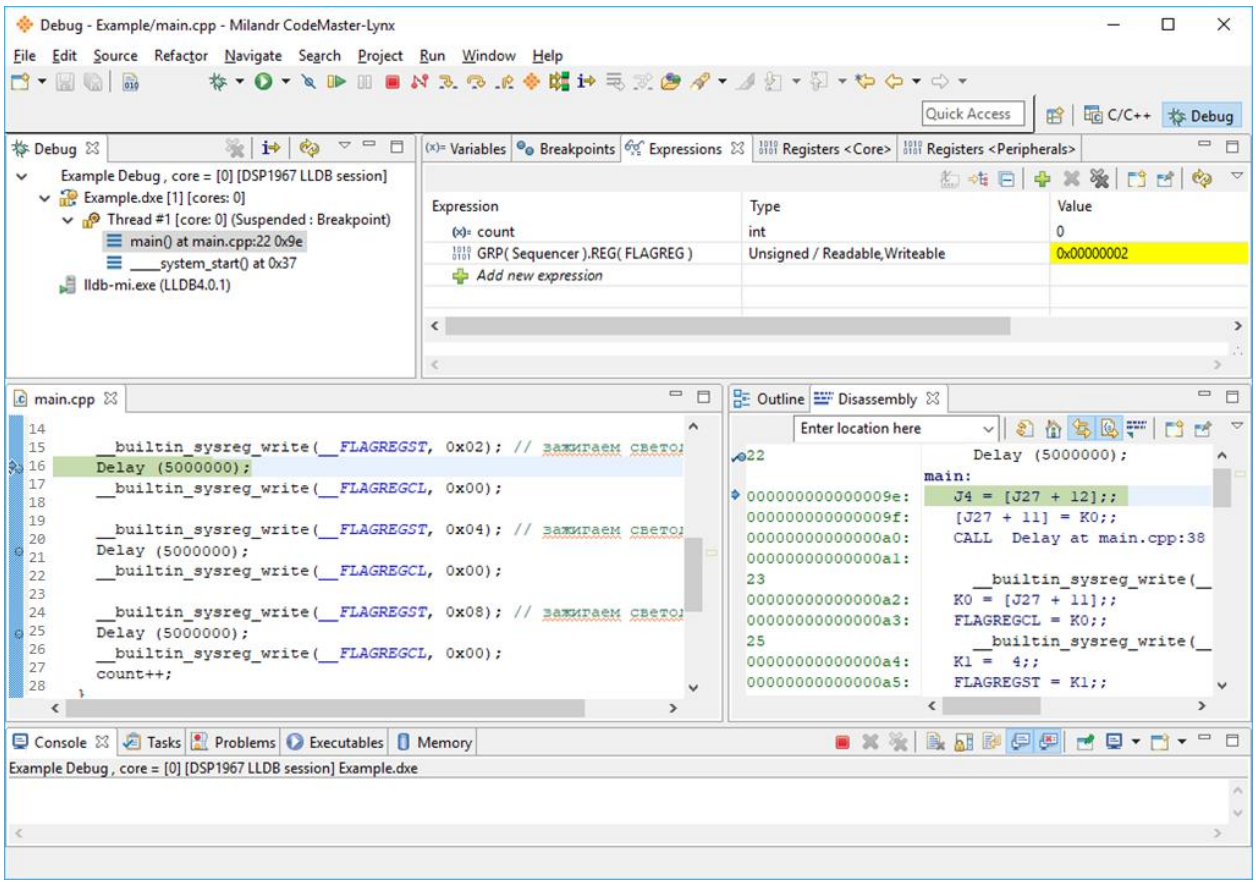


Рисунок 57. Состояние среды при останове на точки останова 2

9. Нажать на кнопку  третий раз.

   Убедиться, что:

   - второй светодиод погас, а третий загорелся;
   - изменилось значение регистра.

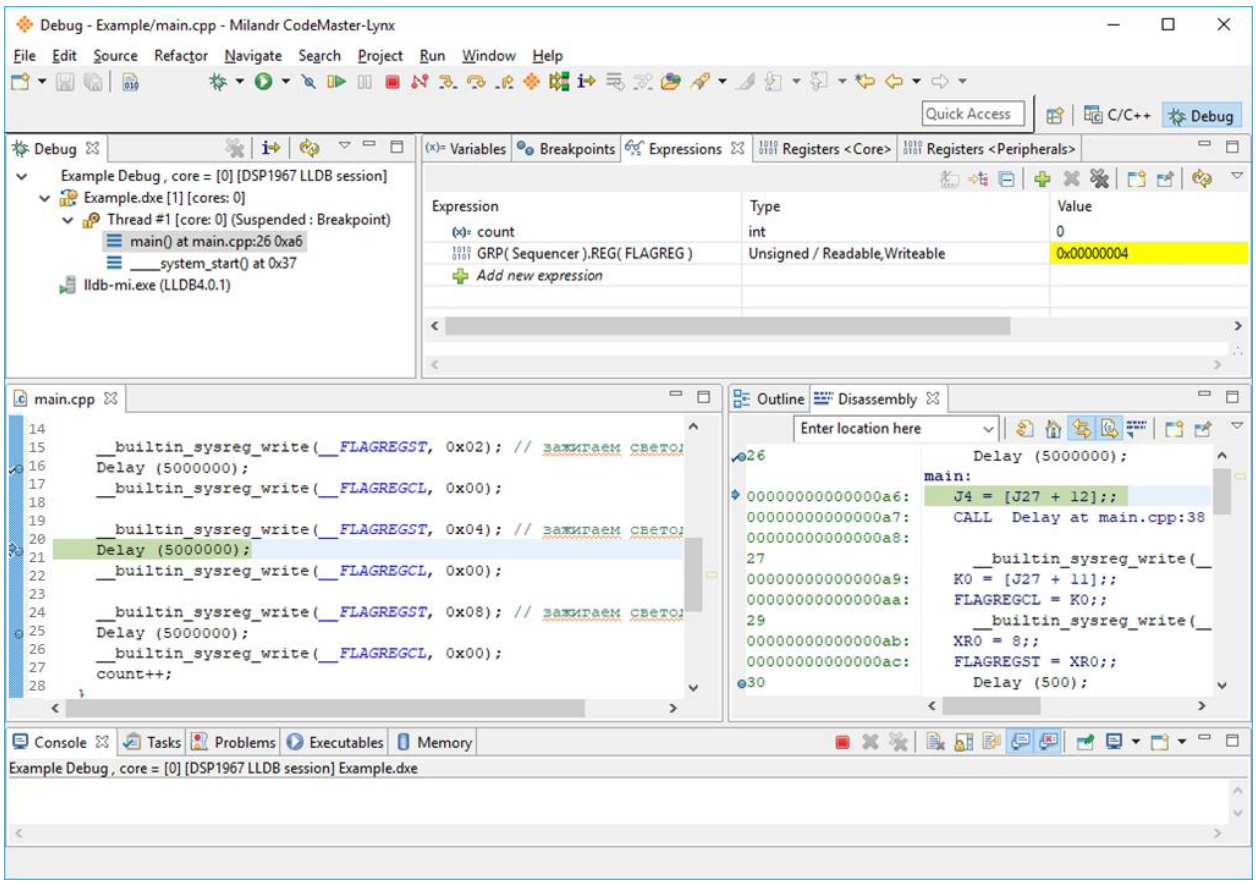


Рисунок 58. Состояние среды при останове на точки останова 3

10. Нажать на кнопку  четвертый раз.

    Убедиться, что:

    - третий светодиод погас, а четвертый загорелся;
    - изменилось значение регистра.

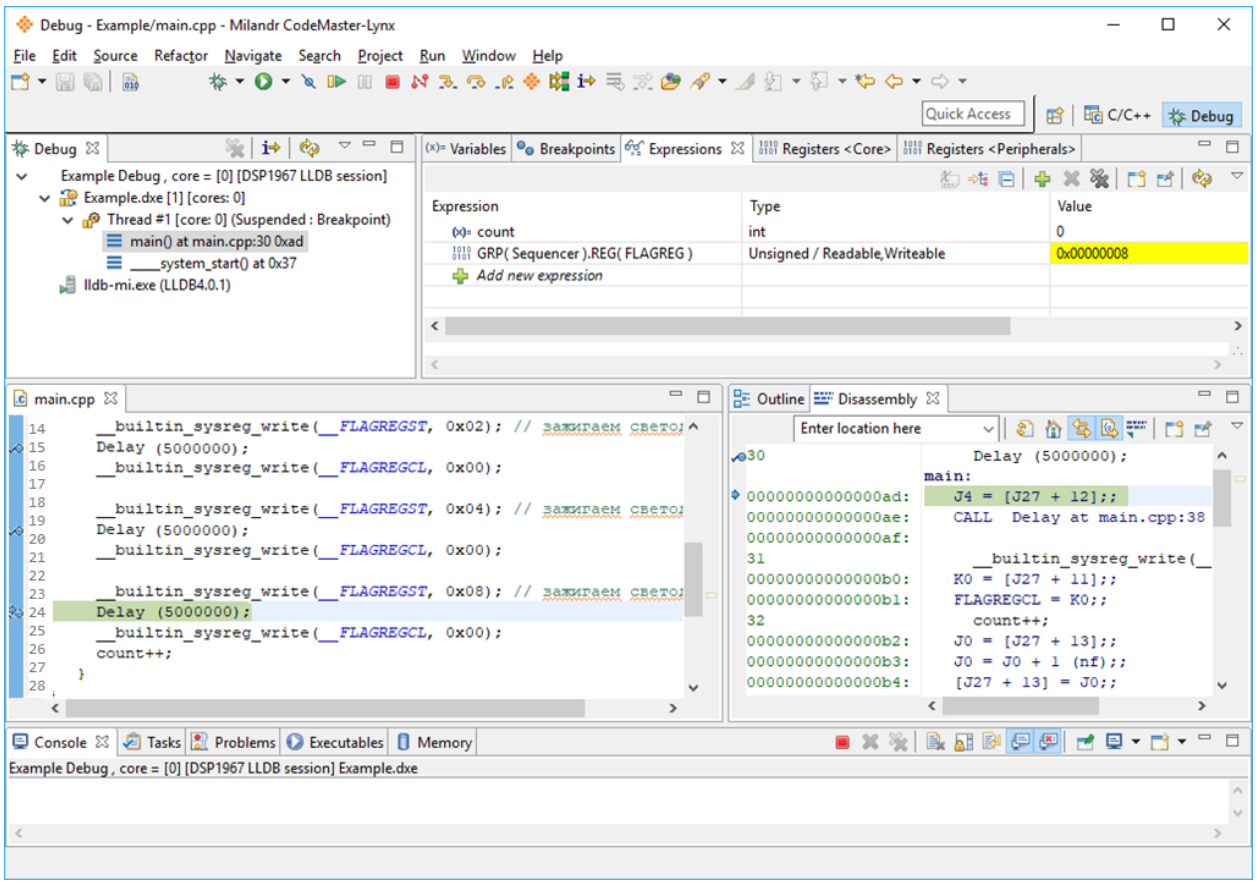


Рисунок 59. Состояние среды при останове на точки останова 4

11. Нажать на кнопку пятый раз.

    Убедиться, что:

    - четвертый светодиод погас, а первый опять загорелся;
    - значение переменной изменилось.

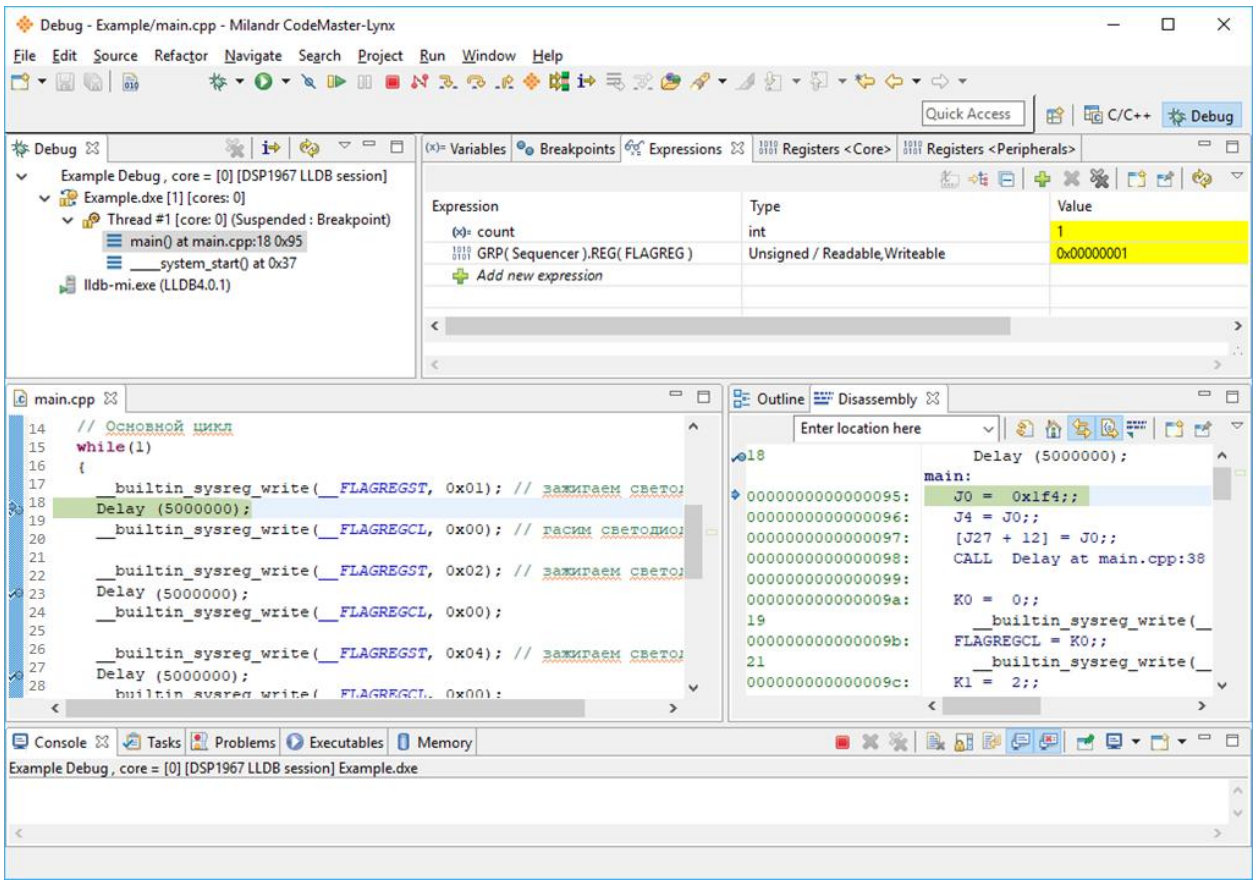


Рисунок 60. Состояние среды после полного прохода приложения

Подробнее о настройке отладочной конфигурации и запуске сессии отладки с использованием программного симулятора или отладочной платы см. п. 6.11.1. Настройка конфигурации и запуск отладки. О том, как перейти в режим отладки, см. п. 6.3.2. Переключение между режимами. Подробнее об операциях при отладке см. п. 6.11.2. Операции при отладке.

Для того чтобы отладить проект для процессора **1967ВН044**, необходимо:

1. В меню «Debug» выбрать пункт «Debug Configurations…» (Рисунок 49).

2. В открывшемся окне создать конфигурацию отладки, дважды щелкнув по наименованию конфигурации отладки **Milandr DSP1967 LLDB**.

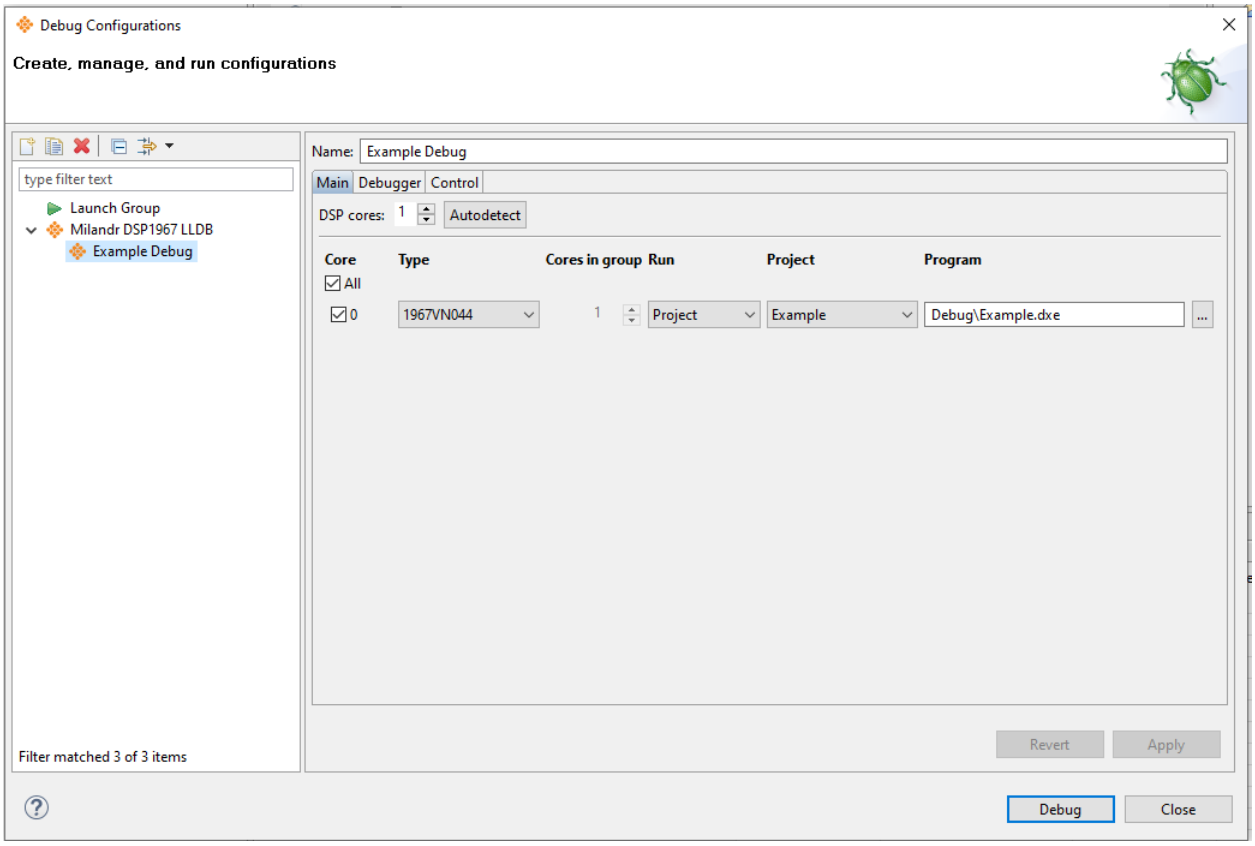


Рисунок 61. Создание новой конфигурации отладки

3. Нажать на кнопку «Debug» для запуска сессии отладки.

4. Нажать на кнопку «OK» для перехода в режим отладки.

Произойдет переход в режим отладки: внешнее представление окна приложения изменится.

На вкладке «**Debug**» можно увидеть, что отладка остановится на **main**:

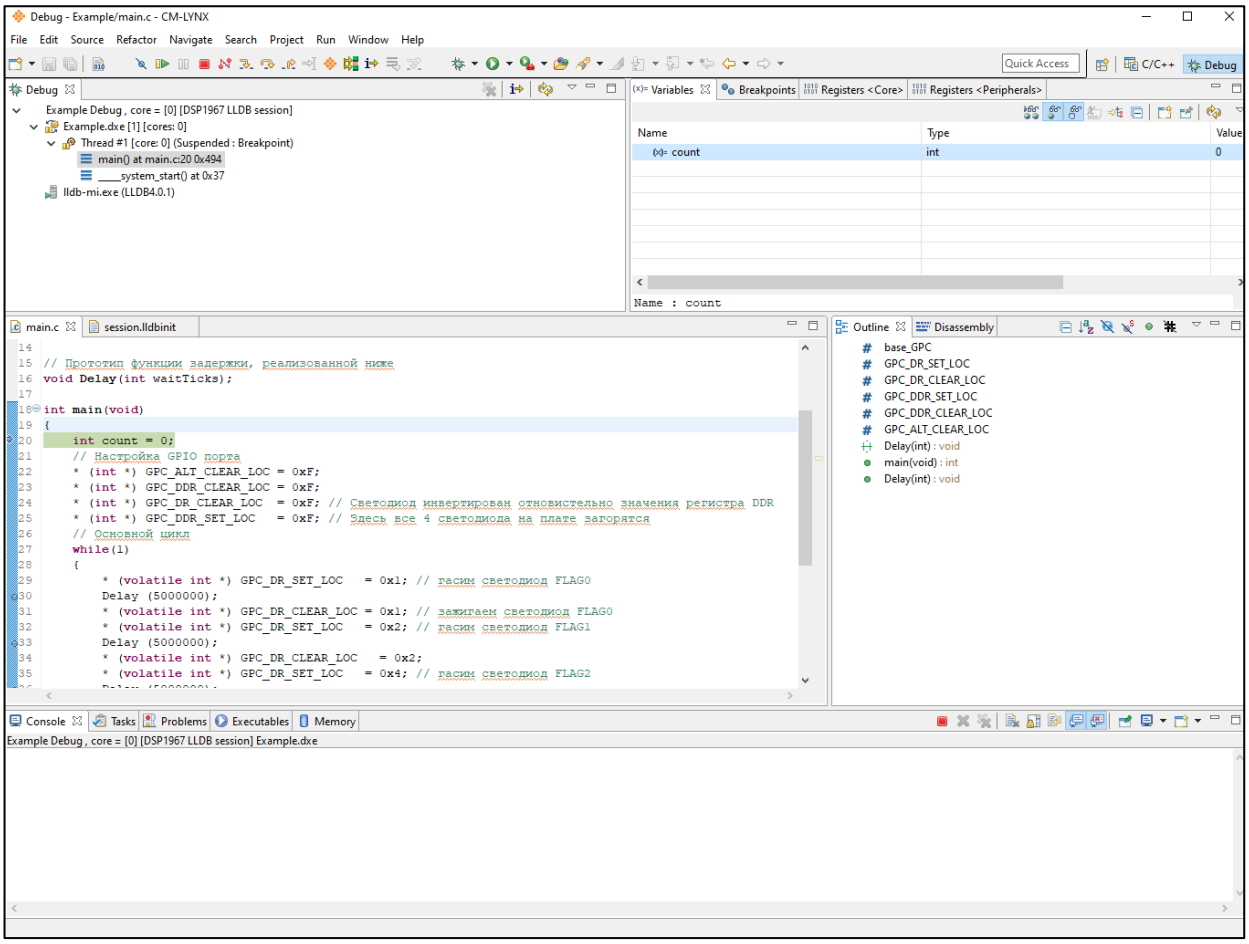


Рисунок 62. Результат запуска отладки

5. Установить точки останова (всего 4 шт.) на строки «**Delay (5000000)**». Для этого следует дважды щелкнуть по синей полосе напротив нужной строки в окне редактора. Слева появится синяя точка.

6. Нажать на кнопку  (первый раз).

Убедиться, что:

- отладка остановится при достижении точки останова;
- на отладочной плате погас первый светодиод;
- в окне «Variables» значение переменной count=0.

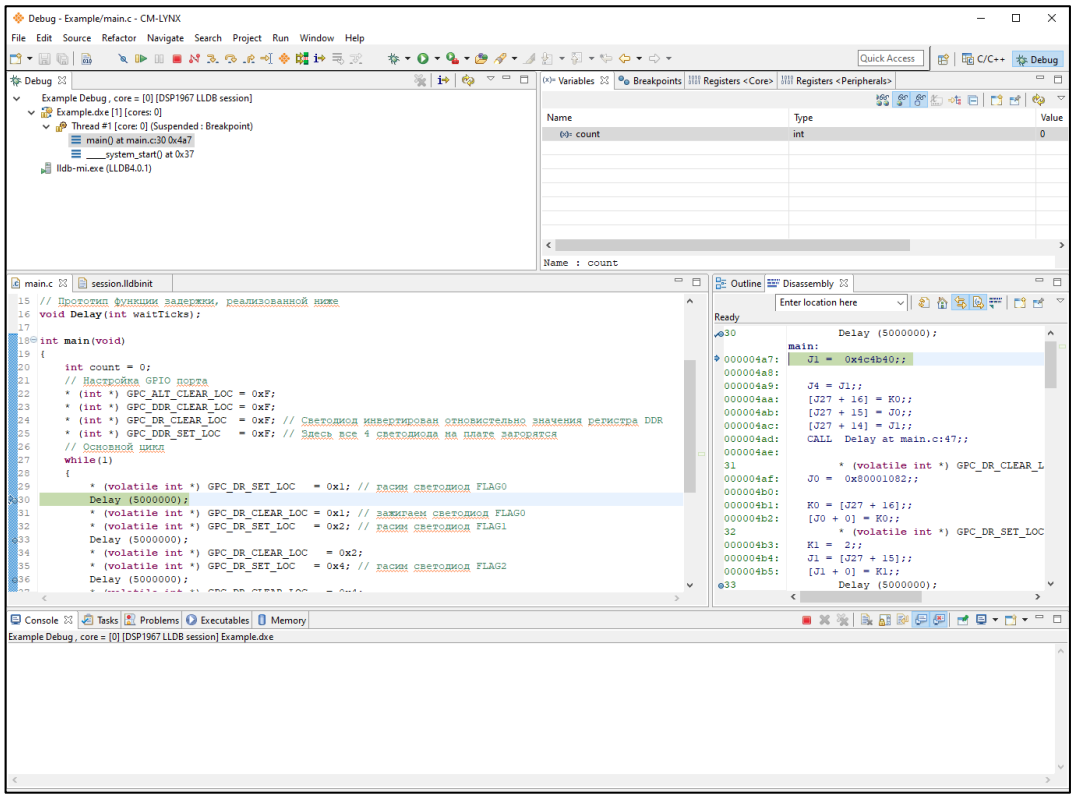


Рисунок 63. Состояние среды при останове на точки останова 1

7. Нажать на кнопку (второй раз).

   Убедиться, что:

   - первый светодиод загорелся, а второй погас;
   - значение переменной не изменилось.

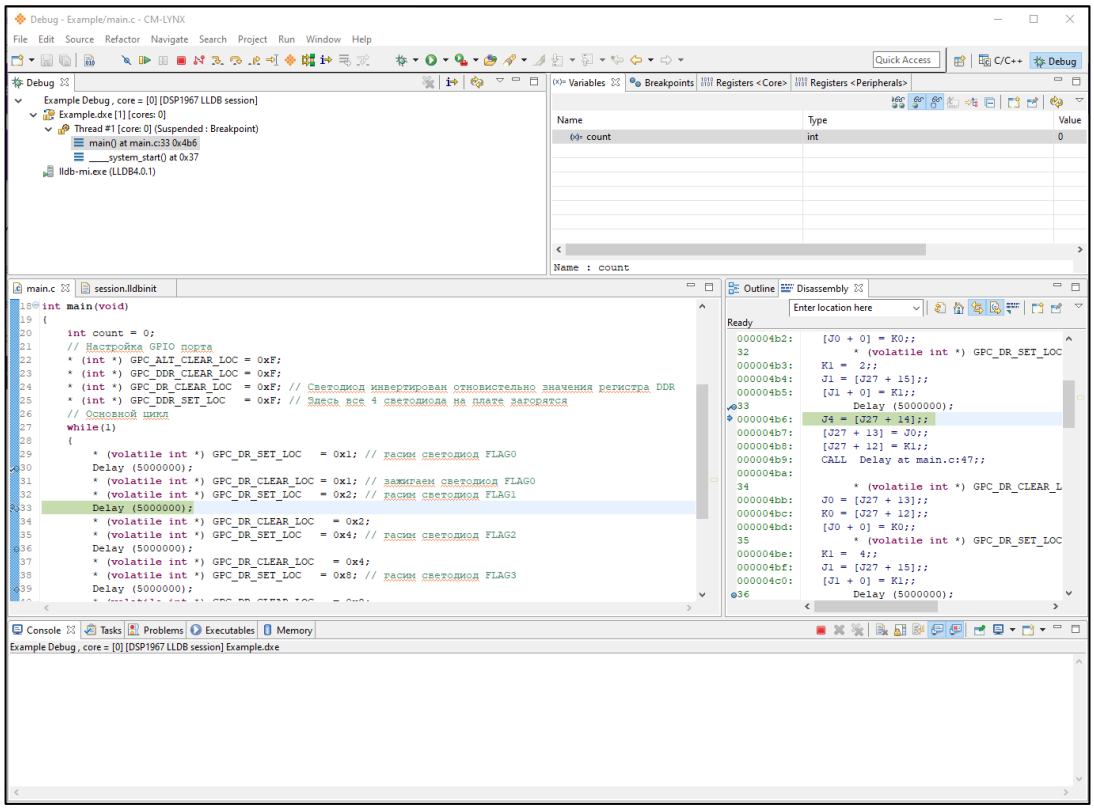


Рисунок 64. Состояние среды при останове на точки останова 2

8. Нажать на кнопку  (третий раз).

   Убедиться, что:

   - второй светодиод загорелся, а третий погас;
   - значение переменной не изменилось.

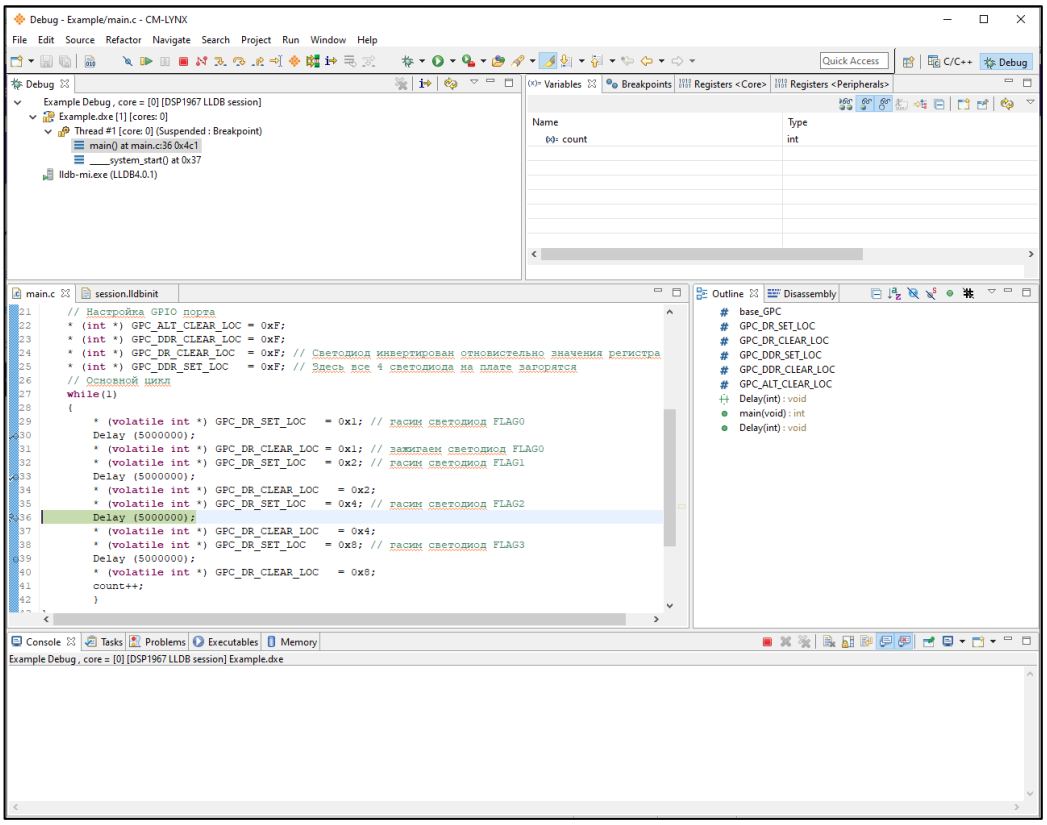


Рисунок 65. Состояние среды при останове на точки останова 3

9. Нажать на кнопку  (четвертый раз).

   Убедиться, что:

   - третий светодиод загорелся, а четвертый погас;
   - значение переменной не изменилось.

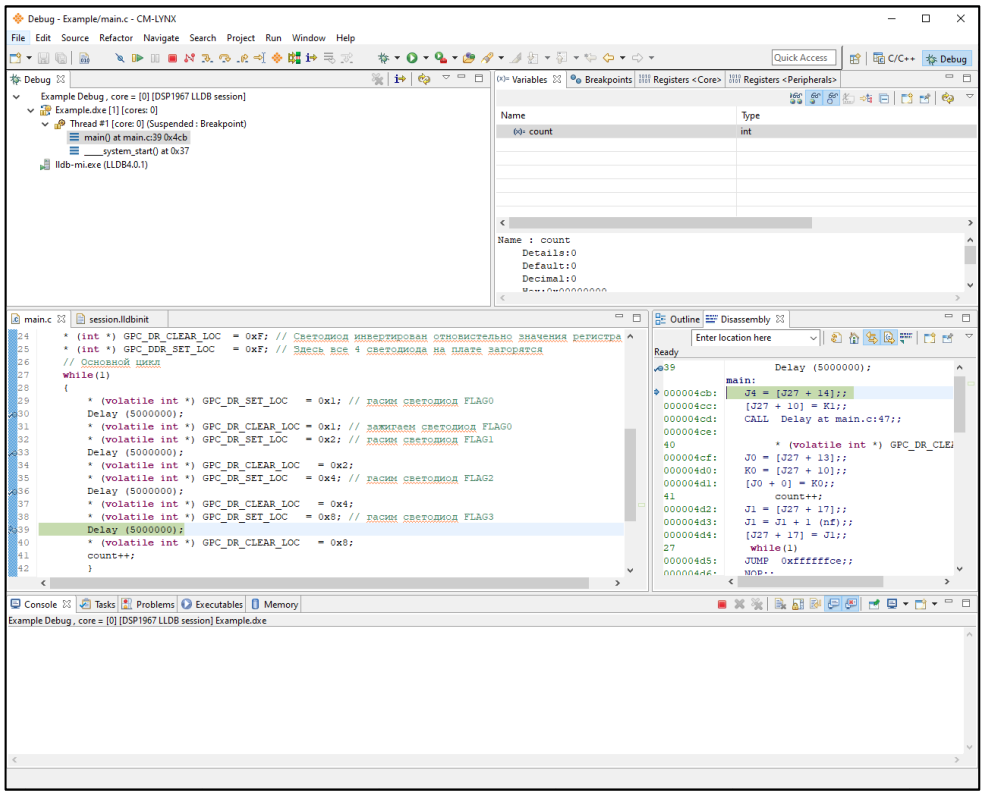


Рисунок 66. Состояние среды при останове на точки останова 4

10. Нажать на кнопку (пятый раз).

    Убедиться, что:

    - четвертый светодиод загорелся, а первый опять погас;
    - значение переменной изменилось.

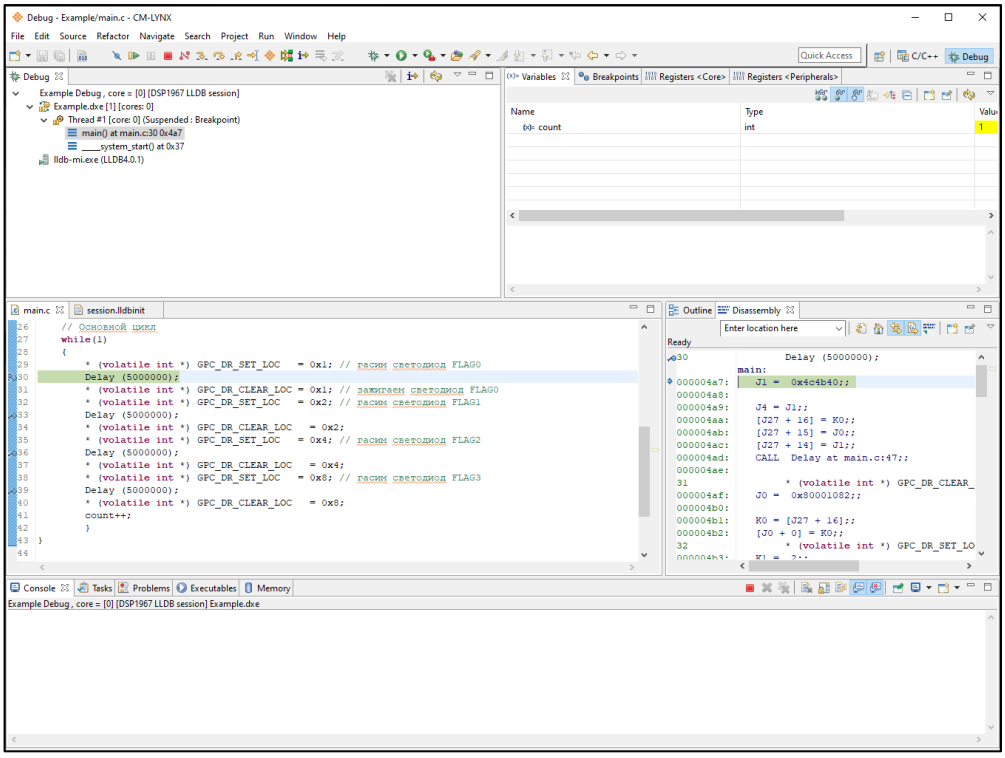


Рисунок 67. Состояние среды после полного прохода приложения

6.3. Выбор и настройка режима работы (перспективы)

Для каждого из режимов работы (перспективы) характерен индивидуальный набор вкладок (см. п. 6.3.1. Режимы работы) и инструментов управления (строка меню и панель инструментов отличаются для каждого из режимов).

Вкладку можно закрыть, изменить ее размеры или расположение, перетащив в удобную область окна.

При необходимости, можно добавить дополнительные вкладки с помощью меню «**Window->Show View**».

При необходимости, можно изменить набор команд строки меню и панели инструментов для режима (см. п. 6.3.3. Настройка набора инструментов режима).

После настройки внешнего вида окна для режима работы пользователь может сохранить текущий профиль для дальнейшей работы с ним при помощи команды «**Save Perspective as**» в меню «**Window**».

Возврат к настройкам по умолчанию для режима работы по умолчанию доступен с помощи команды «**Reset Perspective**» в меню «**Window**».

В процессе работы пользователь может переключаться между доступными режимами работы (см. п. 6.3.2. Переключение между режимами).

Список доступных режимов работы настраивается на панели «**Perspectives**» в окне «**Prerefences**». Подробнее см. раздел п. 6.12.1.6. Настройка режимов работы.

6.3.1. Режимы работы (перспективы)

Для каждого созданного проекта по умолчанию возможны два режима работы:

- *Режим разработки* (режим, в котором выполнение программы не может быть приостановлено и проконтролировано);
- *Режим отладки* (режим, в котором выполнение программы может быть приостановлено/возобновлено, проконтролировано изменение значений переменных, вычислены выражения и т.д.).

Также существуют дополнительные режимы работы:

- *Режим работы с ресурсами*;
- *Режим синхронизации работы команды*.

6.3.1.1. Режим разработки (перспектива «C/C++»)

Например, в режиме разработки (в перспективе «C/C++») в окне приложения (Рисунок 68) по умолчанию содержатся:

- вкладка «**Project Explorer**» (см. П. 5.3. Панель «Project Explorer»);
- редактор **C/C++** (см. п. 5.4.1. Редактор C/C++);
- вкладка «**Outline**» (см. п. 5.4.2.1. Вкладка «Outline»);
- вкладка «**Make Target**» (см. п. 5.4.2.2. Вкладка «Make Target»);
- вкладка «**Problems**» (см. п. 5.4.2.3. Вкладка «Problems»);
- вкладка «**Tasks**» (см. п. 5.4.2.4. Вкладка «Tasks»);
- вкладка «**Console**» (см. п. 5.4.2.5. Вкладка «Console»);
- вкладка «**Properties**» (см. п. 5.4.2.6. Вкладка «Properties»).

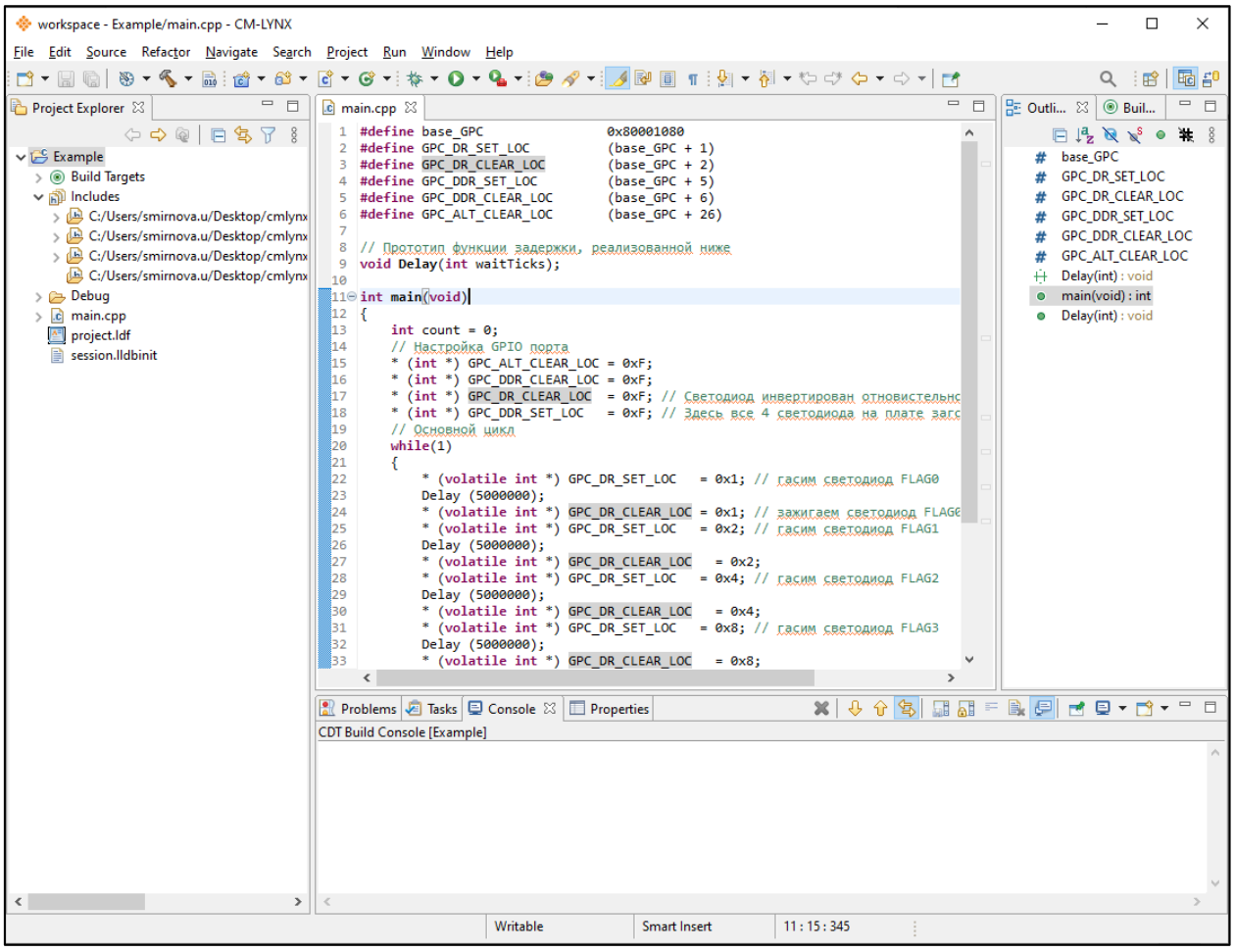


Рисунок 68. Окно программы в режиме разработки

6.3.1.2. Режим отладки (перспектива «Debug»)

В режиме отладки (в перспективе «Debug») в окне приложения по умолчанию содержатся:

- вкладка «**Debug**» (см. п. 5.4.3.1. Вкладка «Debug»);
- вкладка «**Variables**» (см. п. 5.4.3.3. Вкладка «Variables»);
- вкладка «**Breakpoints**» (см. п. 5.4.3.4. Вкладка «Breakpoints»);
- вкладка «**Registers <Core>**» и «**Registers <Peripherals>**» (см. п. 5.4.3.2. Вкладки «Registers»);
- редактор **C/C++** (см. п. 5.4.1. Редактор C/C++);
- вкладка «**Outline**» (см. п. 5.4.2.1. Вкладка «Outline»);
- вкладка «**Problems**» (см. п. 5.4.2.3. Вкладка «Problems»);
- вкладка «**Tasks**» (см. п. 5.4.2.4. Вкладка «Tasks»);
- вкладка «**Console**» (см. п. 5.4.2.5. Вкладка «Console»);
- вкладка «**Executables**».

Рекомендуется, в режиме отладки настроить представление, показанное на Рисунок 69, т.е. подключить вкладки:

- вкладка «**Disassembly**» (см. п. 5.4.3.6. Вкладка «Disassembly»);
- вкладка «**Expressions**» (см. п. 5.4.3.5. Вкладка «Expressions»);
- вкладка «**Memory**» или «**Memory browser**» (см. п. 5.4.3.7. Вкладка «Memory»).

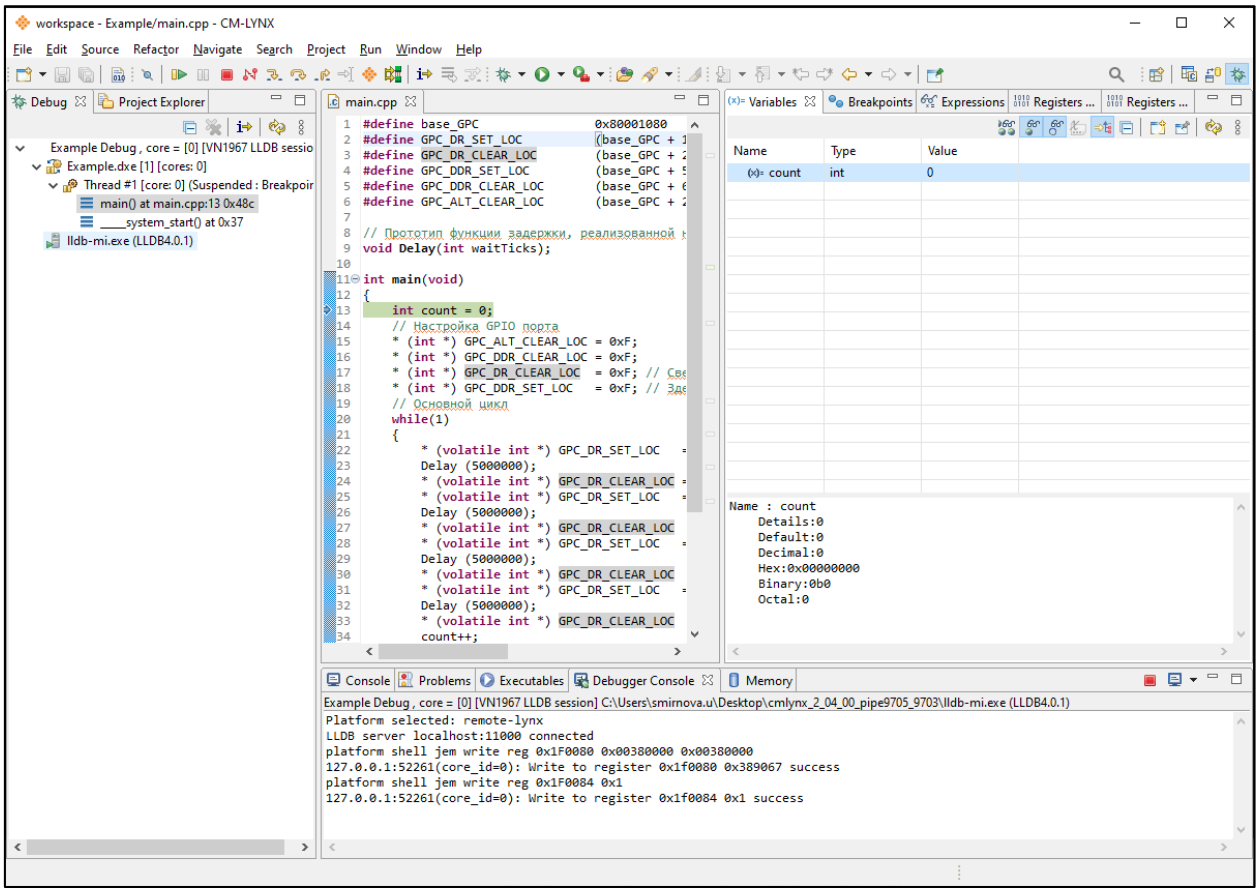


Рисунок 69. Окно программы в режиме отладки

6.3.2. Переключение между режимами (перспективами)

Переключение между режимами (перспективами) осуществляется кнопками на панели инструментов в правом верхнем углу окна программы:



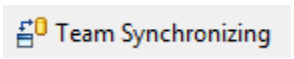
– переключение в режим разработки (в перспективу «C/C++»);



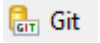
– переключение в режим отладки (в перспективу «Debug»);



– переключение в режим работы с ресурсами (по умолчанию, не отображается);



– переключение в режим синхронизации работы команды (по умолчанию, не отображается).



– переключение в режим работы с Git (по умолчанию, не отображается).

Примечание: текстовые пояснения у кнопок выбора режима могут отсутствовать. Чтобы добавить отображение текстовых пояснений для кнопок выбора режима, необходимо щелкнуть правой кнопкой мыши по любой кнопке выбора режима и выбрать пункт «**Show Text**» в контекстном меню.

Если на панели инструментов нет кнопки для перехода к нужному режиму, то следует:

1. Открыть в основном меню **Window → Perspective→Open perspective→Other...** или нажать значок «**Open Perspective**» (см. Рисунок 70) на панели инструментов в правом верхнем углу она программы.



Рисунок 70. Значок «Open Perspective»

2. В открывшемся диалоговом окне (см. Рисунок 71) выбрать нужный режим.

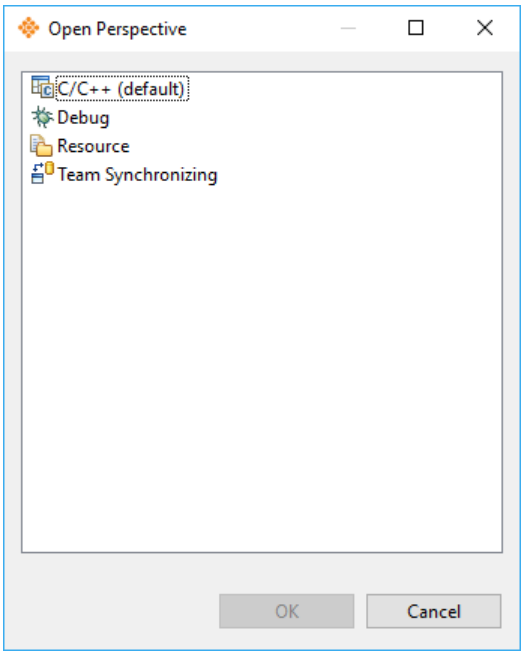


Рисунок 71. Окно открытия режима

3. Нажать на кнопку «**ОК**».

6.3.3. Настройка набора инструментов режима (перспективы)

Каждый режим включает предустановленный набор действий, доступных в строке меню и на панели инструментов окна. Связанные действия сгруппированы в наборы действий. Пользователь может настроить свой набор инструментов для режима. Для этого следует:

1. Выбрать режим работы, для которого требуется выполнить настройку. Например, это можно сделать с помощью кнопок на панели инструментов в правом верхнем углу окна программы.

2. Выбрать в основном меню команду **Window → Perspective→Customize perspective.**

3. Задать в открывшемся диалоговом окне нужные инструменты для выбранного режима.

   На вкладке «**Tool Bar Visibility**» (Рисунок 72) можно изменить внешний вид Панели инструментов. Для добавления на панель инструментов дополнительных инструментов необходимо установить флаг у нужного инструмента; для удаления – снять флаг. Инструменты с наименованиями, помеченными серым шрифтом, недоступны для добавления.

Для удобства поиска нужного инструмента можно воспользоваться флагом «**Filter by action set**», который позволяет отсортировать инструменты по набору доступных действий.

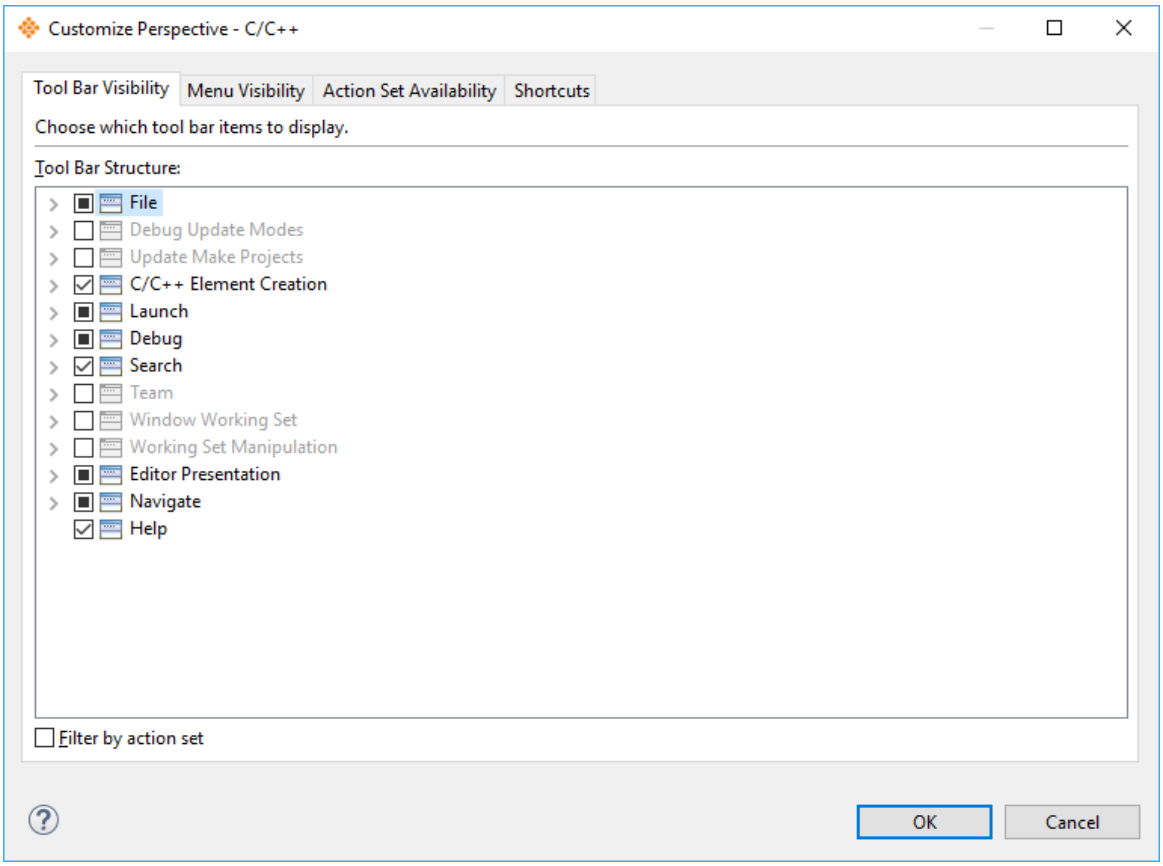


Рисунок 72. Настройка внешнего вида режимов. Вкладка «Tool Bar Visibility»

На вкладке «**Menu Visibility**» (Рисунок 73) можно изменить внешний вид строки меню для выбранного режима. Для добавления в строку меню дополнительного меню (пункта меню) необходимо установить флаг у нужного меню (пункта меню); для удаления – снять флаг.

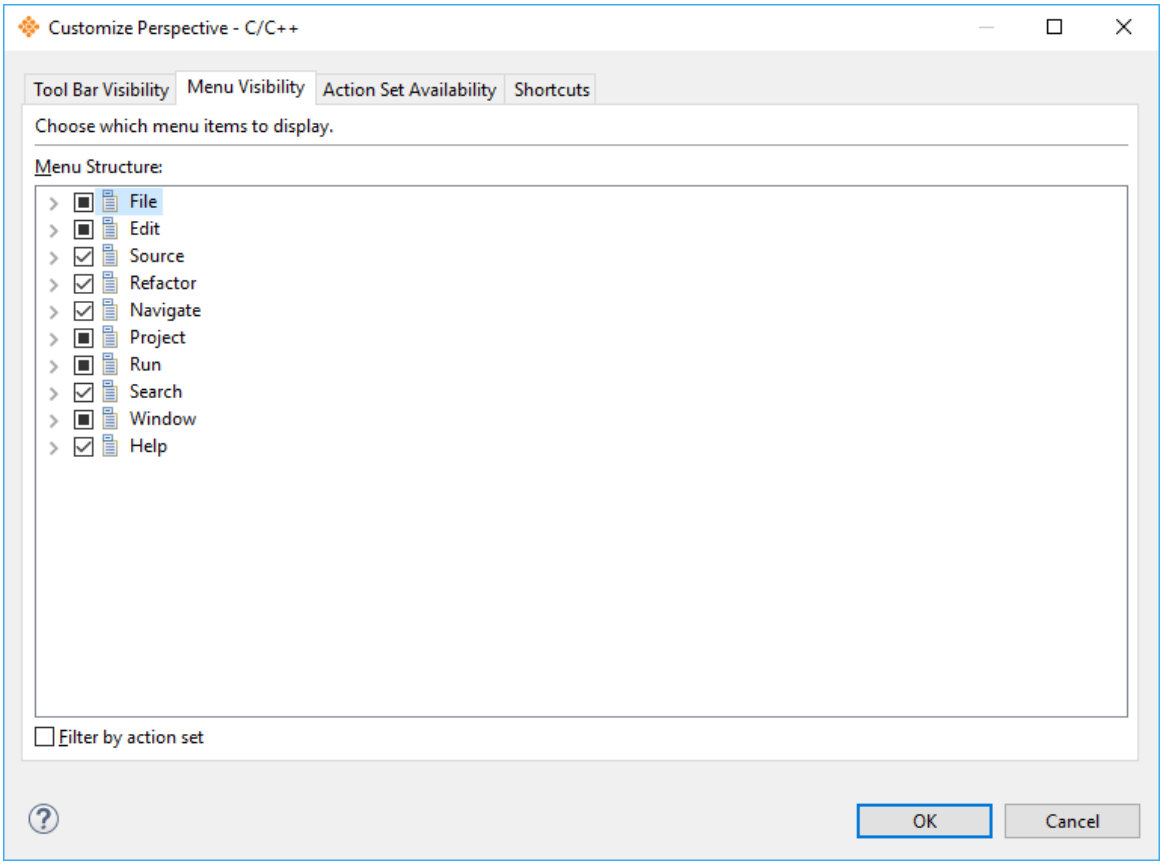


Рисунок 73. Настройка внешнего вида режимов. Вкладка «Menu Visibility»

На вкладке «**Action Set Visibility**» (Рисунок 74) можно изменить список доступных наборов действий для выбранного режима. Для добавления дополнительного набора действий необходимо установить флаг у нужного набора; для удаления – снять флаг. При выборе набора действий щелчком мыши можно получить информацию о меню и инструментах, которые будут добавлены при добавлении этого набора действий:

- в колонке «**Menubar detail**» отображается соответствующее выбранному набору действий меню, включая вложенные пункты меню;
- в колонке «**Toolbar detail**» отображаются соответствующие выбранному набору действий инструменты.

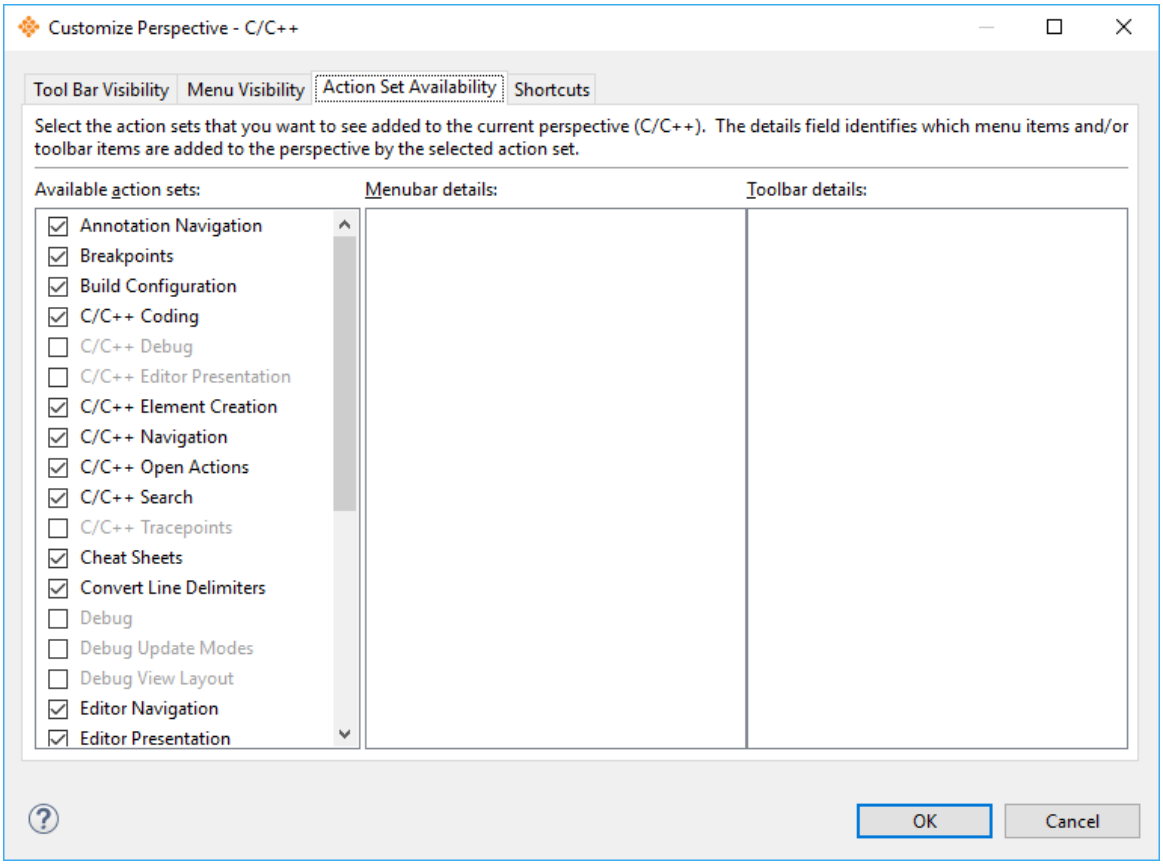


Рисунок 74. Настройка внешнего вида режимов. Вкладка «Action Set Availability»

На вкладке «**Shortcuts**» (Рисунок 75) можно изменить доступные для выбранного режима пункты вложенных меню «**New**», «**Show View**», «**Open Perspective**».

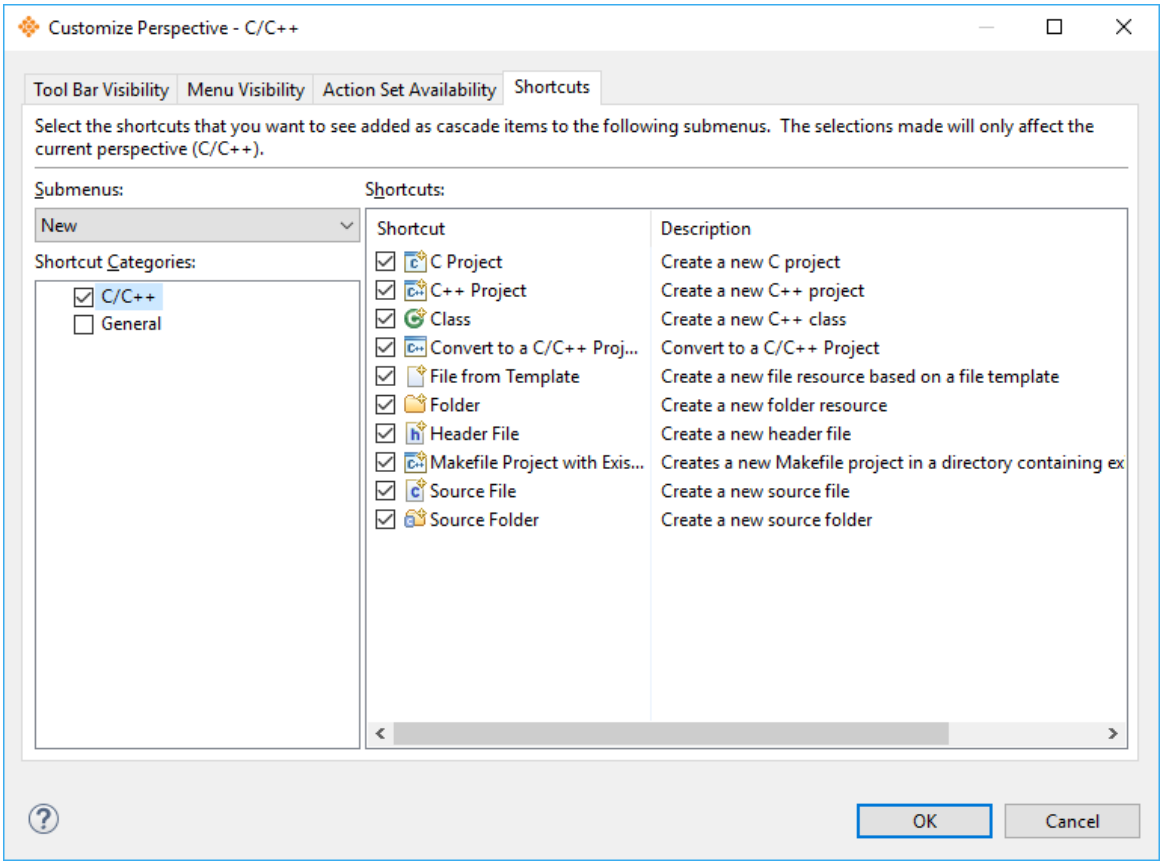


Рисунок 75. Настройка внешнего вида режимов. Вкладка «Shortcuts»

4. Нажать кнопку «**OK**».

После настройки внешнего вида окна режимов пользователь может сохранить текущий профиль для дальнейшей работы с ним при помощи команды «**Save Perspective as**» в меню «**Window**».

6.4. Работа с ресурсами (проектами, папками и файлами)

Ресурсы – это общее название для проектов, папок и файлов, существующих в среде. Файлы и папки в среде аналогичны файлам и папкам в файловой системе. Проекты используются для организации ресурсов и возможности поделиться ими. Как и папки, проекты располагаются в папках файловой системы. Проект может содержать папки и файлы. Папки содержатся в проектах или других папках; могут содержать файлы и другие папки.

Проект может быть открыт или закрыт. Закрытый проект нельзя изменить. Ресурсы для закрытого проекта не отображаются, но постоянно хранятся в локальной файловой системе. Закрытые проекты требуют меньше памяти, поэтому закрытый проект может ускорить процесс сборки. Открытый проект позволяет изменять структуру проекта и видеть его содержимое.

Папки и файлы могут располагаться в файловой системе вне места расположения проекта. Такие файлы и папки называются *связанными*.

Ресурсы хранятся и отображаются в виде иерархии со следующей структурой:

- *корневой ресурс* – верхний уровень содержимого;
- *родитель* – любой ресурс, содержащий другой ресурс. Только проекты и папки могут быть родительскими ресурсами;
- *потомок* (*дочерний ресурс*) – любой ресурс, содержащийся в другом ресурсе. Только файлы и папки могут быть дочерними ресурсами.

Связанные ресурсы – это файлы и папки, хранящиеся в файловой системе вне проекта.

Связанная папка, например, может быть использована для хранения выходных данных сборки отдельно от исходных файлов.

Связанные ресурсы также позволяют совместное использование ресурсов разных проектов в рабочем пространстве. Однако, при использовании пересекающихся ресурсов в рабочем пространстве необходимо учитывать, что изменение ресурса в одном проекте приведет к таким же изменениям этого ресурса в другом проекте.

Для работы с ресурсами по умолчанию используется панель «**Project Explorer**» (см. п. 5.3 Панель «Project Explorer»). Также работа с ресурсами доступна на вкладках «**C/C++ Projects**» и «**Navigator**».

6.4.1. Создание проекта

Для создания проекта необходимо выполнить следующую последовательность действий:

1. Выбрать в меню **File-> New** (см. Рисунок 76) или в контекстном меню на вкладке «**Project Explorer**» (см. Рисунок 77) тип создаваемого проекта: **C/ C++ Project** или **Project**.

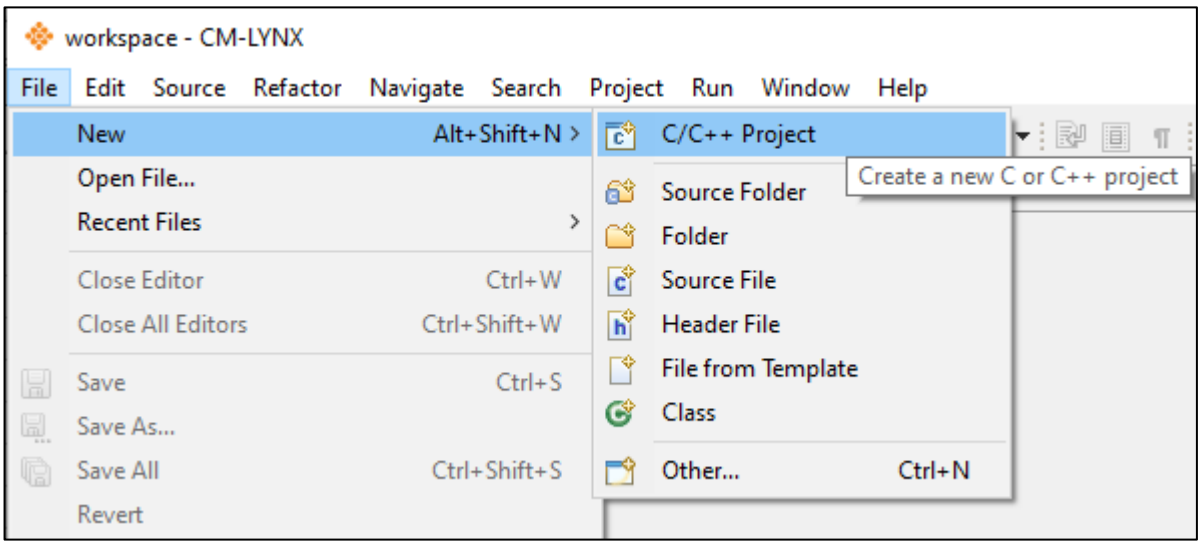


Рисунок 76. Создание проекта из меню

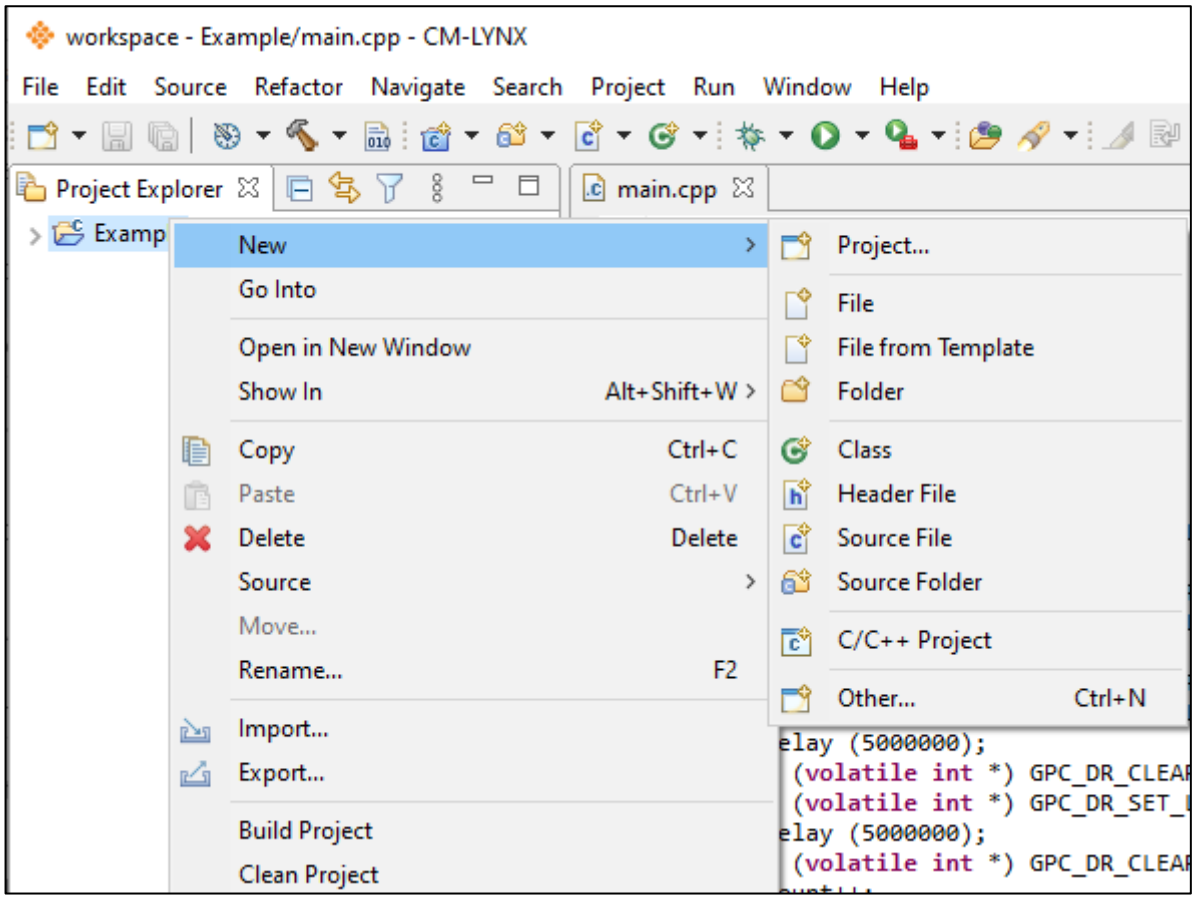


Рисунок 77. Создание проекта из всплывающего меню «View» Project Explorer

2. Для пункта «**Project**» следует выбрать тип проекта в появившемся диалоговом окне и нажать кнопку «**Next**» (см. Рисунок 78).

Примечание: при выборе пункта **C/C++ Project** на шаге 1, данное диалоговое окно не показывается, программа сразу переходит к шагу 3.

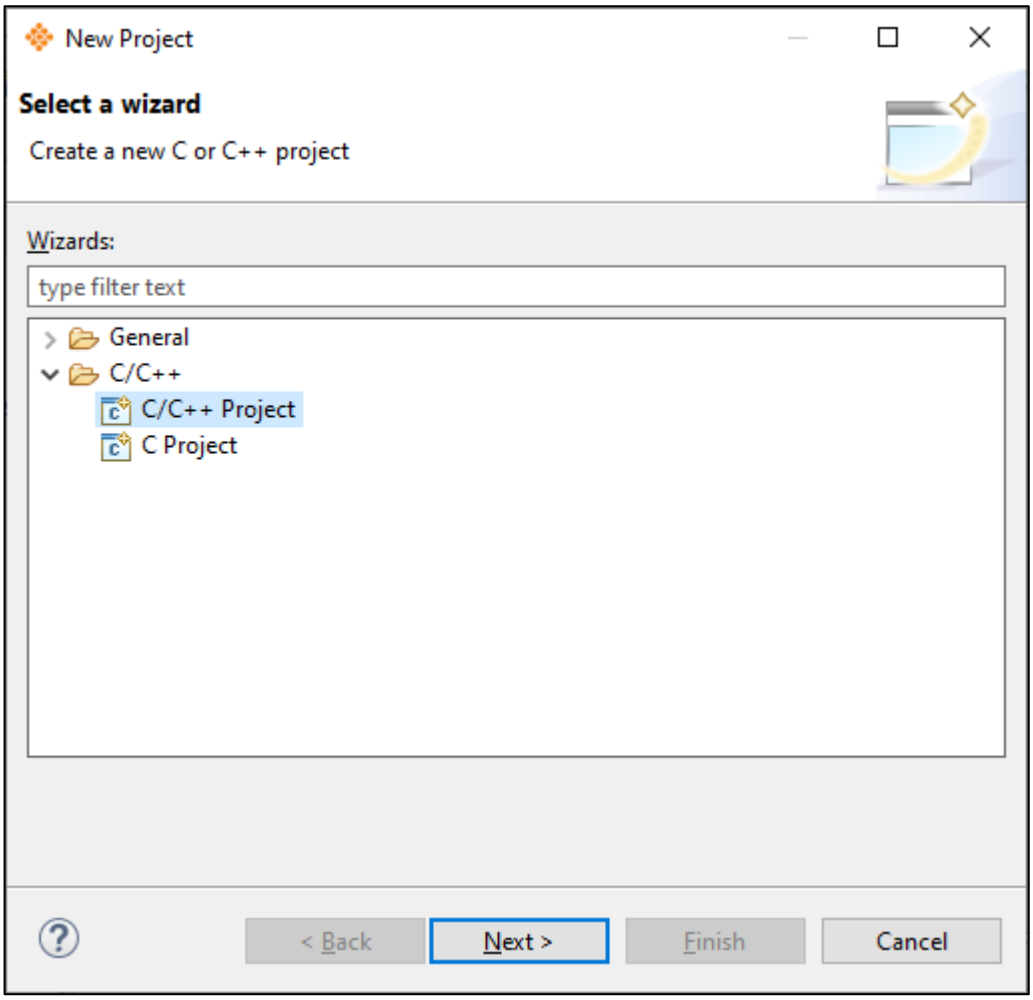


Рисунок 78. Выбор типа проекта

3. Выбрать шаблон, щелкнуть по строке «Milandr C/C++ Project» и нажать кнопку «**Next**» (см. Рисунок 79).

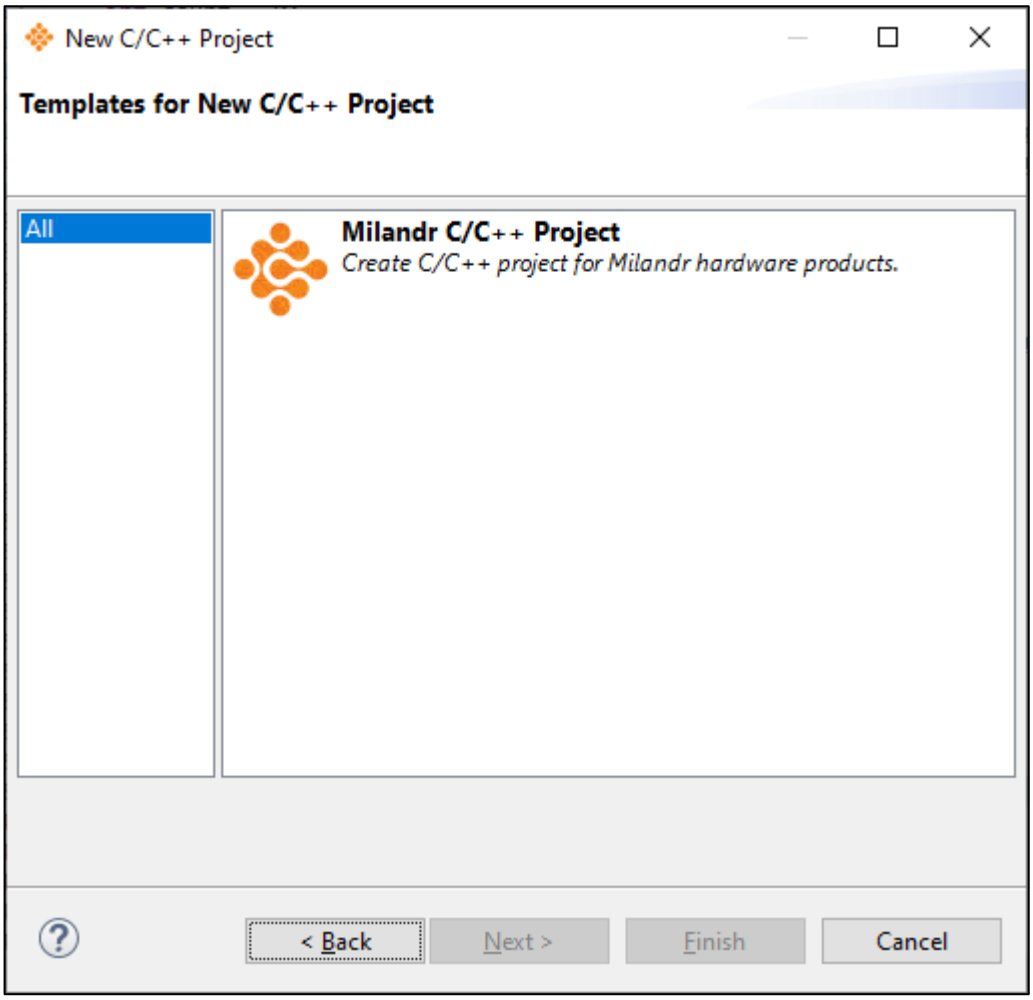


Рисунок 79. Выбор шаблона

4. Заполнить параметры создаваемого проекта (см. Рисунок 80):
    1) Указать в поле «**Project name**» имя создаваемого проекта;
    2) Указать путь для сохранения проекта:
        - Для сохранения проекта в каталоге по умолчанию следует отметить чек-бокс «**Use default location**»;
        - Для сохранения проекта в другом каталоге следует снять отметку с чек-бокса «**Use default locatio**n и ввести новый путь в поле **Location**».
    3) Выбрать тип проекта:
        - **Executable project** – исполняемый проект;
        - **Static Library project** – проект статической библиотеки.
    4) Выбрать тулчейн – набор программ, необходимых для компиляции и генерации выполняемого кода из исходных текстов.

        В текущей версии используется тулчейн **Milandr DSP1967 LLVM Cross C/C++**, в состав которого входят компилятор **Clang**, ассемблер **MCA-TSH** и компоновщик **MCLINK-TSH**.

    Подробнее о работе компилятора, ассемблера и компоновщика можно почитать в пользовательской документации на эти компоненты.

    В дальнейшем настройка тулчейна выполняется в свойствах проектах на панели «**Settings**». Подробнее см. п. 6.9.2.4.1 Настройки компонентов тулчейн.

5. Нажать кнопку «**Next**».

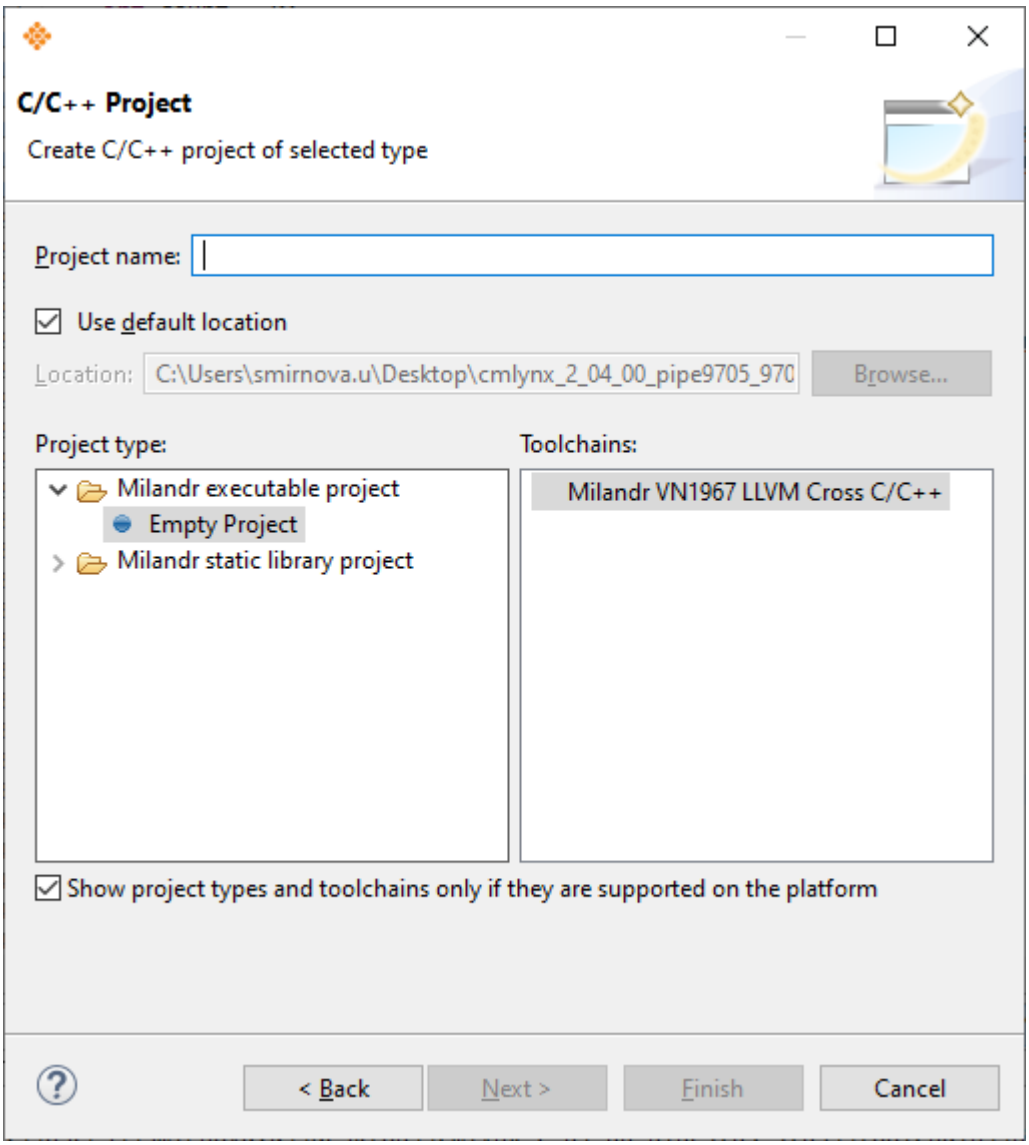


Рисунок 80. Заполнение параметров проекта

6. Утвердить список конфигураций по умолчанию (см. Рисунок 81) и нажать кнолпку «**Next**».

   Доступны конфигурации:

   - **Debug** – такая конфигурация будет содержать отладочную информацию;
   - **Release** – конфигурация без отладочной информации.

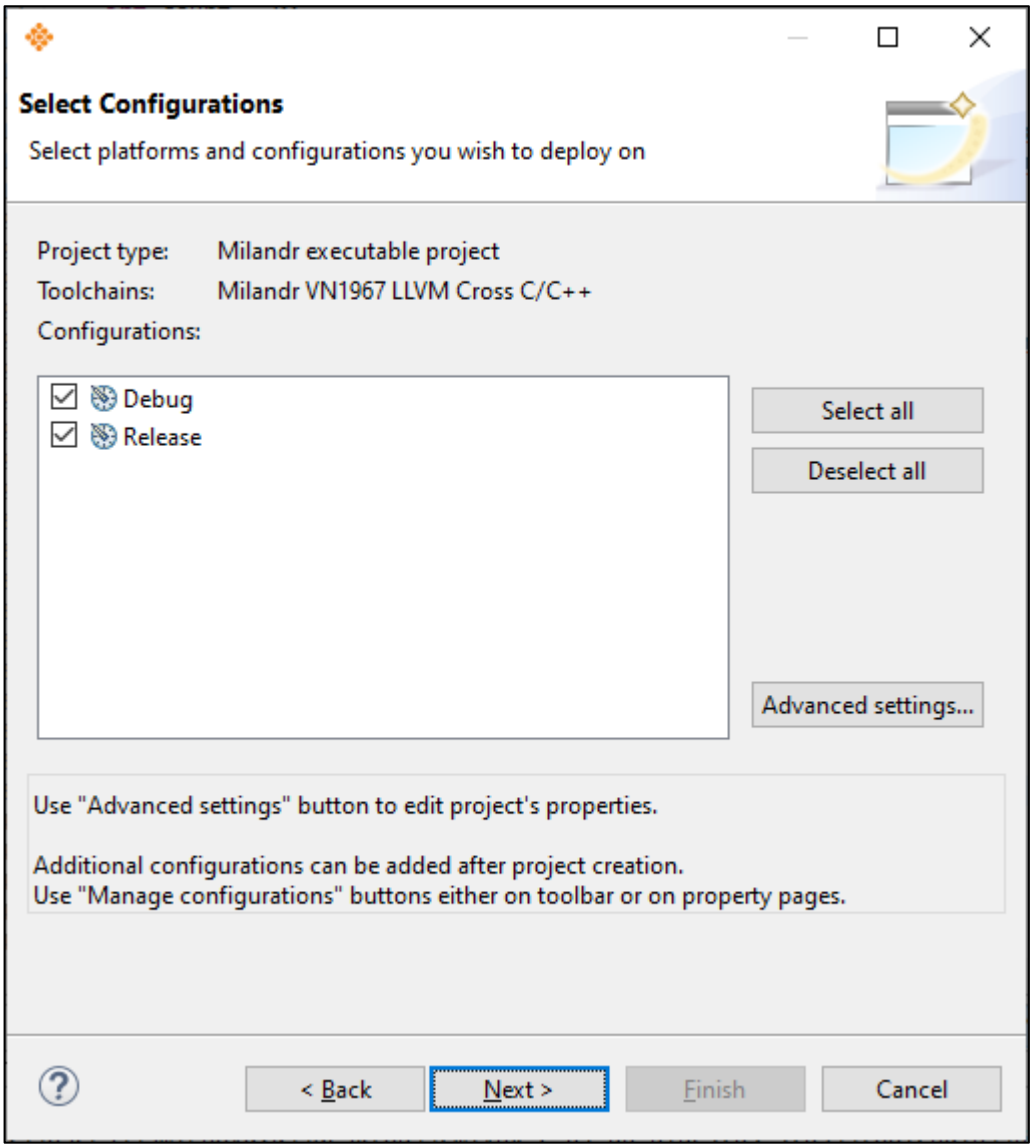


Рисунок 81. Выбор списка конфигураций

С помощью кнопки **«Advanced settings...»** можно настроить свойства проекта. Свойства проекта также можно настроить или изменить для проекта позднее. Подробнее о настройке проектов см. раздел п. 6.9. Настройка свойств проекта.

7. Выбрать процессор (по умолчанию – **«1967VN044 Rev.3»**), способ подключения HAL, тип файла main и нажать кнопку **«Finish»** (см. Рисунок 82).

   Настройка «**Default HAL library**» определяет способ подключения HAL к проекту:

Without library	Не использовать HAL в проекте
Link sources by reference	Подключить HAL по ссылке
Copy sources into project	Скопировать HAL в проект

Настройка «**Default main**» определяет режим генерации файла main:

Do not generate	Не добавлять файл main.c/main.cpp в проект
Generate main.c	Добавить файл main.c в проект
Generate main.cpp	Добавить файл main.cpp в проект

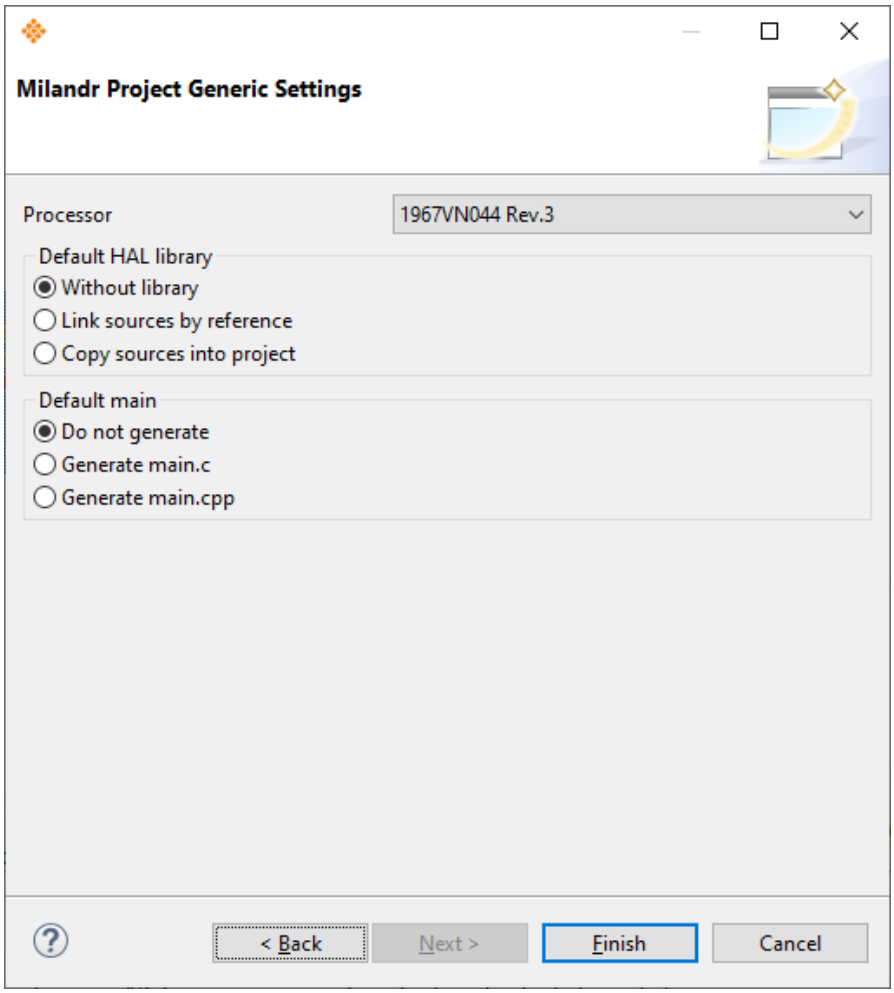


Рисунок 82. Выбор процессора, способа подключения HAL, типа файла main

Для процессора 1967ВН058 доступна настройка функций управления памятью процессора (см. Рисунок 83). Для отображения и редактирования доступных настроек управления памятью процессора 1967ВН058 необходимо установить флаг «Default memory management functions». После этого станут доступны следующие настройки:

Number of cores	Использовать конфигурацию памяти для заданного числа внутренних ядер процессора. Определяет выбор шаблонного LDF и распределения памяти L3 для одного или двух внутренних ядер процессора. Значение по умолчанию - одно ядро.
Cashes	Использовать конфигурацию памяти для выбранных кешей и включить их при запуске. Определяет выбор шаблонного LDF с учетом

занимаемой кэшами памяти, версию специализированного загрузчика, а также работу модуля управления памятью в части использования кешей. Возможные значения: L1, L1+L2. Значение по умолчанию - L1.

Management mode

Включить функции управления памятью MMU в заданном режиме. Определяет выбор версии специализированного загрузчика с включением функций управления памятью в заданном режиме. Возможные значения:

- Fixed TLB only - защищенный режим MMU. Доступная память автоматически конфигурируется при старте и соответствует только таблице TLB, с максимальным числом дескрипторов памяти – 32;
- Fixed TLB with hardware management- защищенный режим MMU. Доступная память автоматически конфигурируется при старте и соответствует фиксированной части TLB (в текущей версии 24 дескриптора), а также дополняется динамически загружаемой аппаратно из таблиц трансляций уровней L1 и L2 (до 8 дескрипторов);
- Fixed TLB with software management- защищенный режим MMU. Доступная память автоматически конфигурируется при старте и соответствует фиксированной части TLB (в текущей версии 24 сегмента), а также дополняется динамически загружаемой программно (до 8 сегментов). Модуль управления памятью из поставляемой стандартной библиотеки делает это также из таблиц трансляций уровней L1 и L2, для совместимости алгоритмов с аппаратным режимом подкачки дескрипторов;
- TA_mode – не защищенный, упрощенный режим.

Значение по умолчанию – Fixed TLB only.

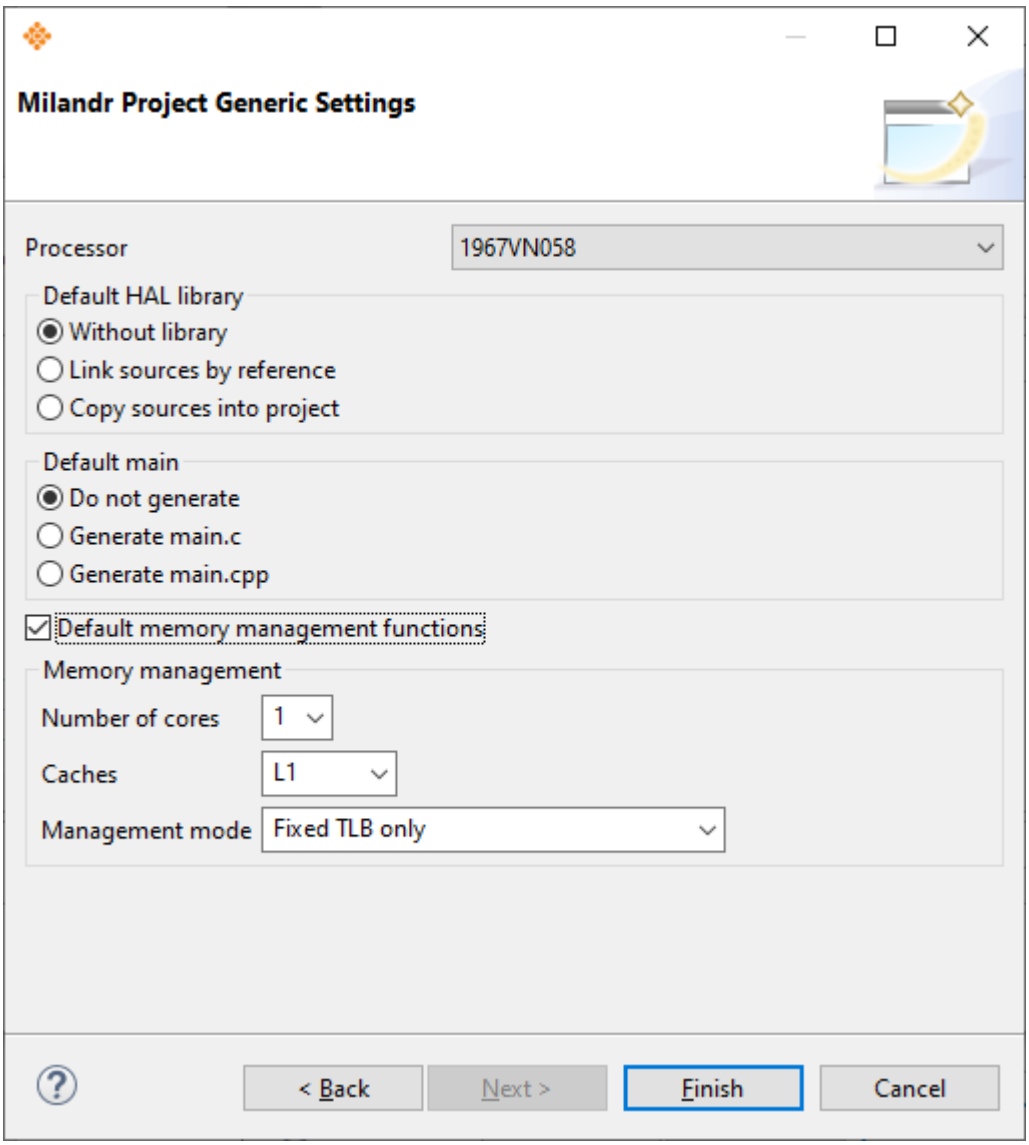


Рисунок 83. Настройки управления памятью (только для 1967ВН058)

6.4.2. Создание папки

Для того чтобы создать папку, необходимо:

1. Выбрать в меню **File-> New** (см. Рисунок 76) или в контекстном меню на вкладке «**Project Explorer**» (см. Рисунок 77) или с помощью кнопки на панели инструментов пункт «**Folder**».

   Откроется окно «**New Folder**» (см. Рисунок 84).

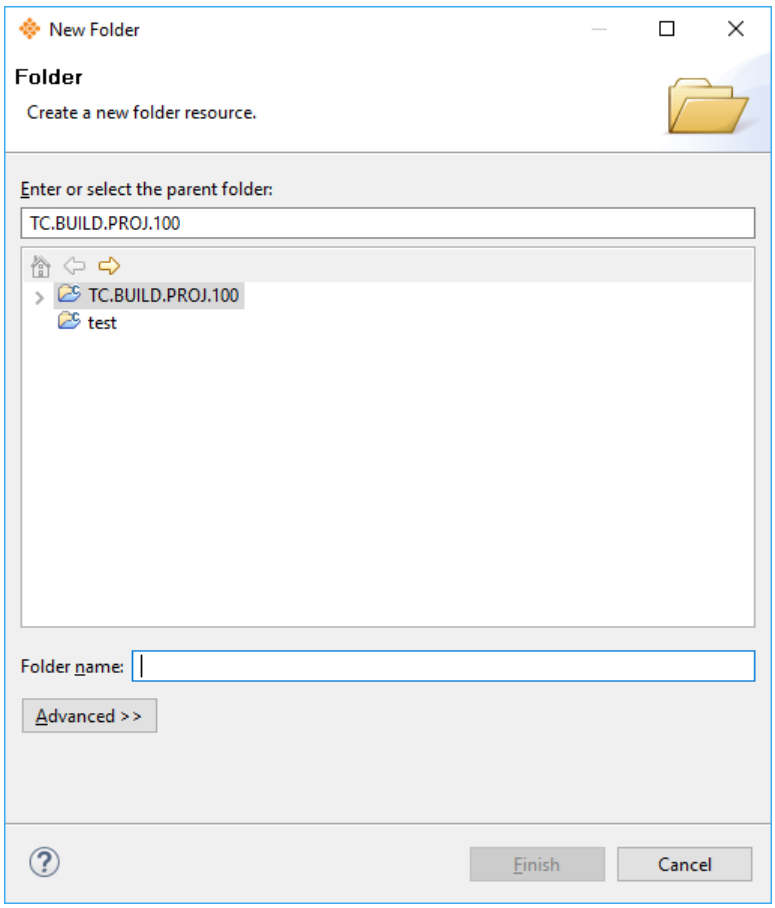


Рисунок 84. Окно «New Folder»

2. Указать родительскую папку.
3. Ввести наименование создаваемой папки в поле «**Folder name**».
4. По умолчанию папка создается в текущее место файловой системы.

   Для создания связанной папки необходимо:

   1) Нажать на кнопку «**Advanced**».

      Раскроется панель «**Advanced**» (см. Рисунок 85).

   2) Выбрать вариант «**Link to alternative location (Linked Folder)**».

   3) Указать место хранения папки, отличное от текущего. Для этого можно использовать кнопку «**Browse**» или задать место с помощью переменных (кнопка «**Variables...**»).

      Кнопка «**Resource Filters...**» позволяет перейти к редактированию фильтров ресурсов.

5. Нажать кнопку «**Finish**».

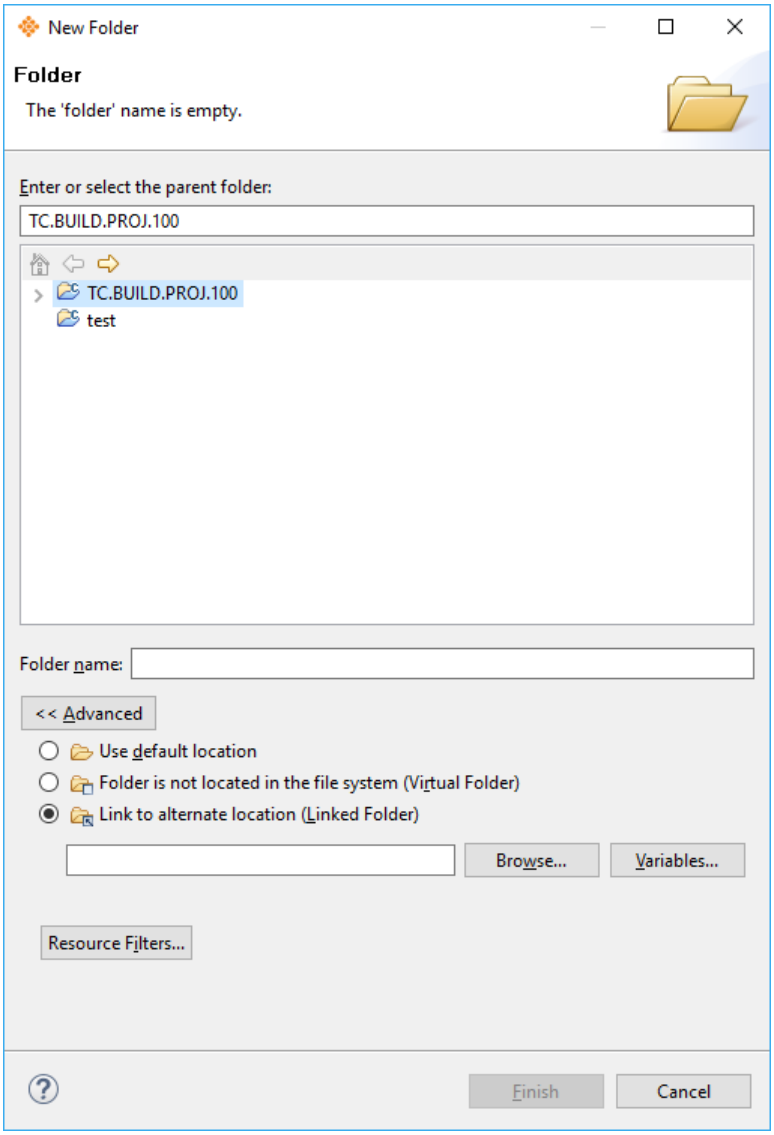


Рисунок 85. Окно «New Folder» с панелью «Advanced»

6.4.3. Создание файла

Для того чтобы создать файл, необходимо:

1. Выбрать в меню **File-> New** (см. Рисунок 76) или в контекстном меню на вкладке «**Project Explorer**» (см. Рисунок 77) или с помощью кнопки на панели инструментов пункт «**File from Template**».

   Откроется окно «**New File**» (см. Рисунок 86).

2. Указать родительскую папку.

3. Ввести имя файла в поле «**File name**».

Примечание: обратите внимание, что если расширение файла «*.C» указано большими буквами, то при сборке данный файл будет интерпретироваться как C++.

4. Для создания файла с использованием шаблона установить флаг «**Use template**» и выбрать шаблон из списка. При отсутствии нужного шаблона в списке можно добавить нужный шаблон с помощью редактора шаблонов (вызывается кнопкой «**Configure...**»).

5. По умолчанию файл создается в текущее место файловой системы.

   Для создания связанного файла необходимо:

   1) Нажать на кнопку «**Advanced**».
   2) Выбрать вариант «**Link to file in the file system**».
   3) Указать место хранения файла, отличное от текущего. Для этого можно использовать кнопку «**Browse**» или задать место с помощью переменных (кнопка «**Variables...**»).

6. Нажать кнопку «**Finish**».

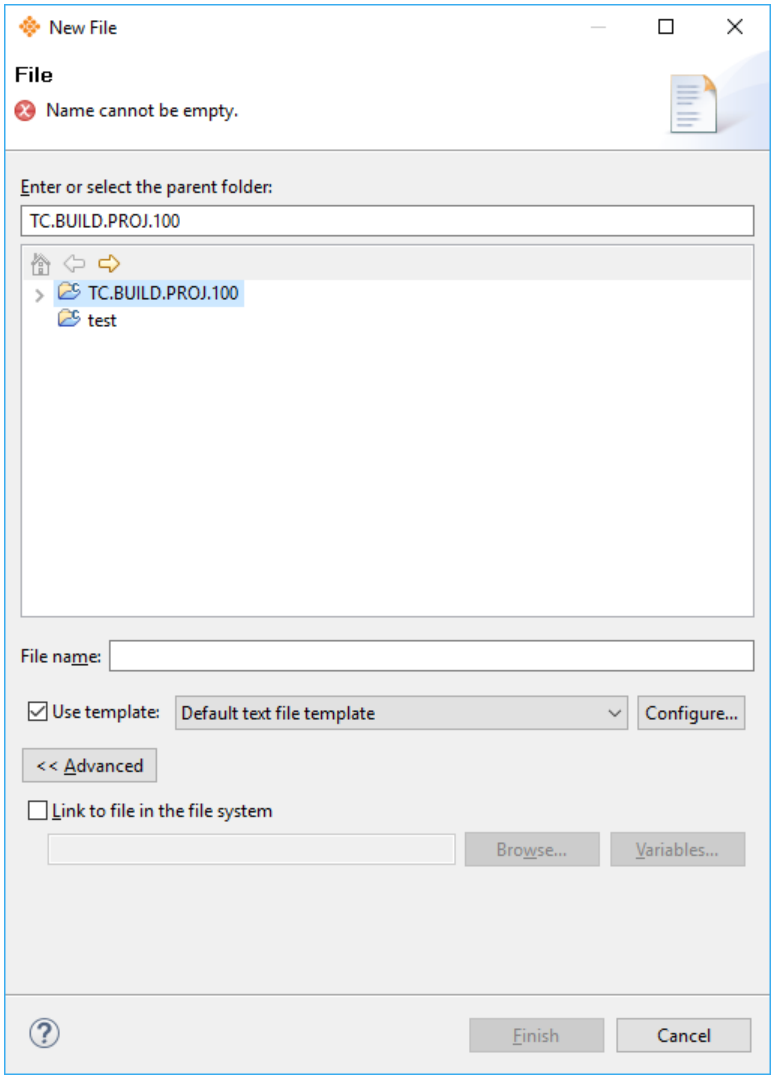


Рисунок 86. Окно «New File» с панелью «Advanced»

6.4.4. Импорт ресурсов из файловой системы

Для того чтобы импортировать ресурсы из файловой системы, необходимо:

1. Выбрать в меню «**File**» команду «**Import...**».

1. Выбрать команду «**General->File System**» и нажать кнопку «**Next**» (см. Рисунок 88).

2. Указать директорию, из которой требуется импортировать ресурсы, в поле «**From directory**». Для выбора местоположения директории можно использовать кнопку «**Browse**».

3. Отметить ресурсы, которые требуется импортировать, установив флаги напротив импортируемых ресурсов.

   Для этого можно использовать кнопки:

Filter Types...	Вызвать окно для настройки фильтрации по типам файлов
Select All	Выделить все
Deselect All	Снять выделение со всех

4. Указать папку, куда будут импортированы ресурсы, в поле «**Into folder**». Для выбора местоположения директории можно использовать кнопку «**Browse**».

5. Задать необязательные настройки на панели «**Options**» при необходимости:

Overwrite existing resources without warnings	Перезаписать поверх существующих ресурсов без предупреждения
Create top-level folder	Создать корневой каталог

6. Задать необязательные настройки на панели «**Advanced**» при необходимости хранения импортированных ресурсов как связанных.

   Для этого необходимо:

   1) Нажать на кнопку «**Advanced**».
   2) Установить флаг «**Create links in workspace**».

      Флаг «**Create link locations relative to**» будет установлен автоматически.
   3) Указать место хранения ресурсов, отличное от текущего местоположения папки, в которую импортируются ресурсы. Для этого следует нажать кнопку «**Edit Variables...**» и задать путь с помощью переменной пути.

7. Нажать кнопку «**Finish**».

Примечание: импорт папок и файлов также доступен посредством «перетаскивания» (метод «drag and drop») или путем копирования и вставки этих ресурсов из файловой системы на вкладку «**Project Explorer**»/«**C/C++ Project**»/ «**Navigator**».

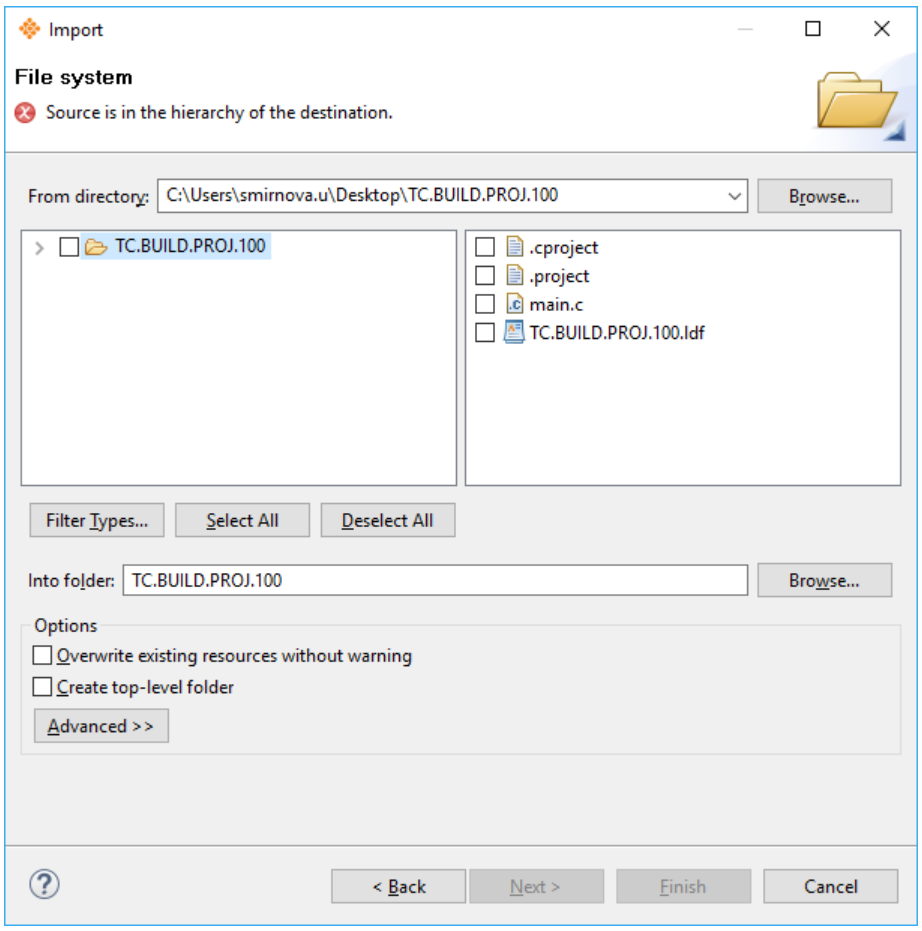


Рисунок 87. Окно «Import File system»

6.4.5. Импорт существующего проекта

Для импорта существующего проекта в текущее *рабочее пространство* (Workspace) нужно выполнить следующую последовательность действий:

1. Выбрать в меню «**File**» команду «**Import...**».

2. Выбрать команду «**General->Existing Projects into Workspace**» и нажать кнопку «**Next**» (см. Рисунок 88).

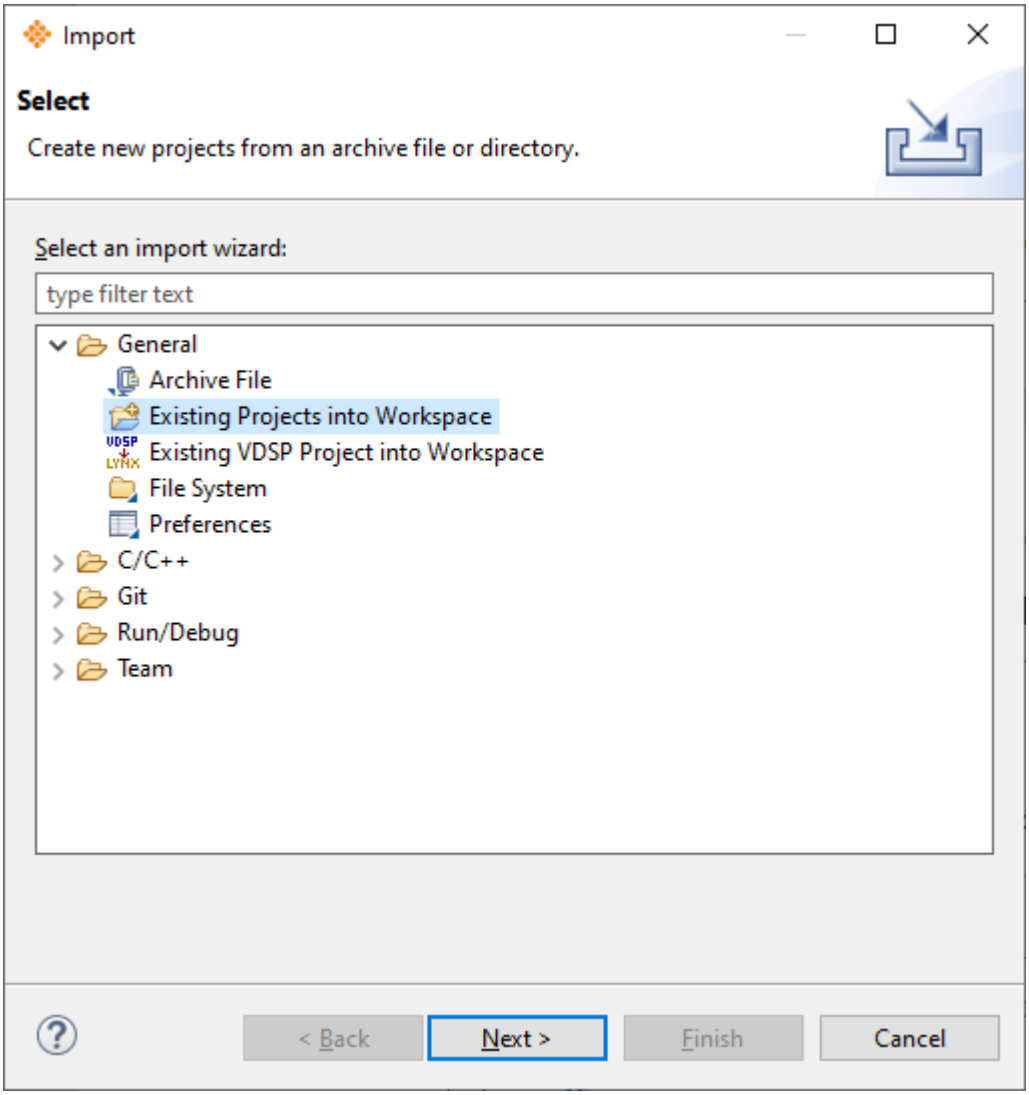


Рисунок 88. Импорт проекта, шаг 1

3. Выбрать импортируемые проекты и задать дополнительные опции при необходимости (см. Рисунок 89).

    1) Указать директорию или архив с проектами, которые необходимо импортировать.

        Для импорта проектов из директории необходимо установить переключатель в положение «**Select root directory**» и указать путь к директории с проектами.

        Для импорта проектов из архива необходимо установить переключатель в положение «**Select archive file**» и указать путь к директории с проектами.

    2) Выбрать проекты для загрузки, установив флаги напротив нужных наименований проектов.

        Можно воспользоваться кнопками:

Select All	Выбрать все
Deselect All	Отменить выбор для всех
Refresh	Обновить

    3) Для копирования файлов проекта в текущее рабочее пространство (Workspace) установить флаг «**Copy projects into workspace**».

При этом в Workspace будет создана копия проекта, включая все файлы, находящиеся в директории с проектом. Обратите внимание на то, что может потребоваться ручная правка путей к заголовочным файлам. Если в проекте присутствуют прилинкованные файлы вне директории проекта, ссылки на них сохраняются, но сами файлы не копируются.

Для добавления проекта в workspace по ссылке снять флаг «**Copy projects into workspace**». При этом проект не копируется, а добавляется в workspace "как есть". Рекомендуется использовать этот способ при импорте проектов со сложной структурой исходных файлов, хранящихся отдельно от директории проекта.

4. Нажать кнопку **«Finish»**. Проекты будут импортированы в текущее рабочее пространство (Workspace).

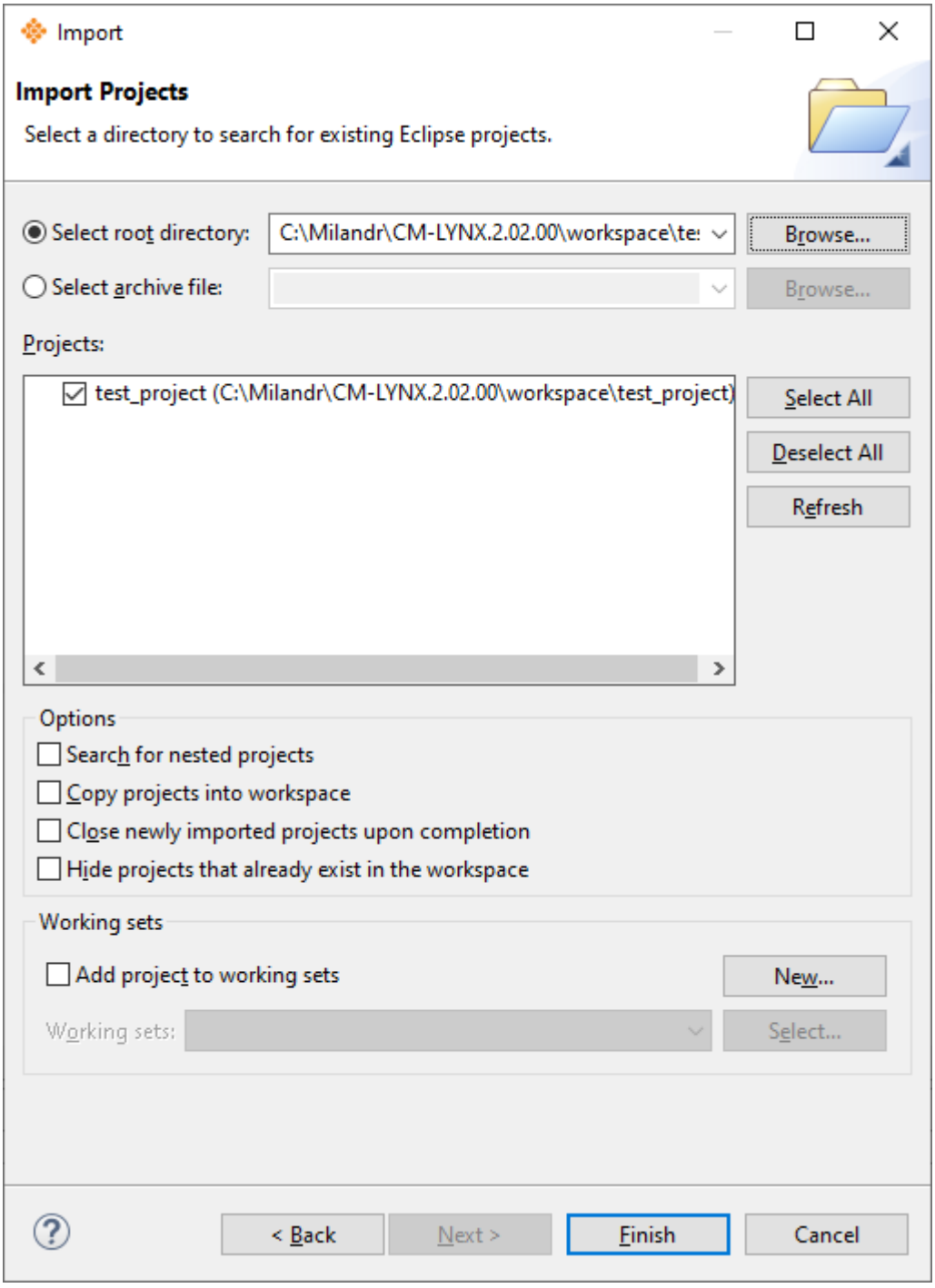


Рисунок 89. Импорт проекта, шаг 2

Примечание: при импорте проектов Git не рекомендуется выбирать опцию «Copy projects into workspace».

При переносе проектов C/C++ между разными рабочими местами (ПЭВМ) необходимо учесть следующие ограничения.

1. Максимальную степень переносимости имеют простые проекты без виртуальных ссылок на исходные файлы, без подключённых заголовочных и конфигурационных файлов за пределами папки проекта, за исключением стандартных папок тулчейна CM-Lynx. Такие проекты могут быть импортированы как с копированием ресурсов в текущее рабочее пространство, так и без него с подключением структуры исходного проекта к текущему рабочему пространству.

2. Проекты с виртуальными ссылками должны обязательно проверяться разработчиком на возможность переноса. Если все необходимые исходные файлы и папки проектов CM-Lynx копируются в общую папку с сохранением дерева вложенных ресурсов, то любые проекты из неё могут быть импортированы в режиме без копирования в текущее рабочее пространство CM-Lynx. При этом фактически ни один файл из исходного проекта не копируется. В IDE импортированный проект регистрируется как внешний, а его стандартная переменная **`PROJECT_LOC`** будет указывать на папку исходного проекта.

6.4.6. Быстрый поиск ресурсов

Для того чтобы быстро отыскать те или иные ресурсы, необходимо

1. В меню «**Navigate**» выбрать команду «**Go To-> Resource**».

2. В окне « **Go To Resource**» (Рисунок 90) начать ввод наименования ресурса, который необходимо найти. Достаточно указать первые несколько символов названия. Например, если ввести несколько первых символов названия файла, будут показаны все возможные совпадения на панели «**Matching Items**». Можно использовать специальные символы:

   ? Любой символ

   ***** Любая строка

3. Выбрать ресурс на панели «**Matching Items**» и нажать кнопку «**OK**».

На вкладках рабочей области будет отображен выбранный ресурс.

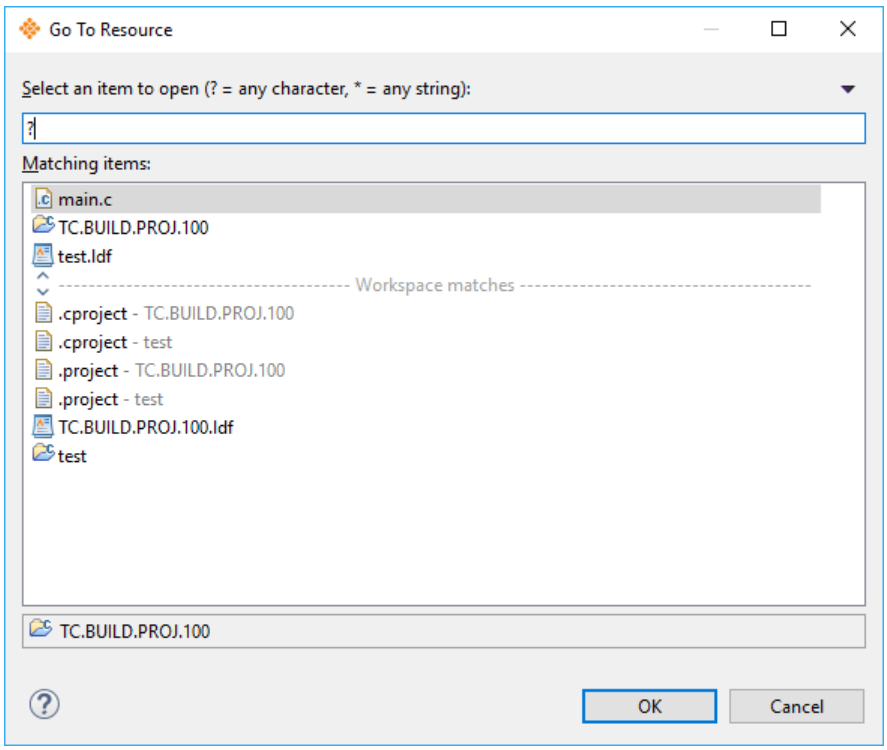


Рисунок 90. Окно «Go To Resource»

6.4.7. Копирование ресурсов

Пользователь может копировать ресурсы (например, из одного проекта в другой). Для того чтобы скопировать ресурсы, необходимо:

1. Выбрать ресурсы, которые необходимо скопировать, на вкладке навигации (например, на вкладке «**Project Explorer**»).

2. В контекстном меню вкладки выбрать команду «**Copy**».

3. На вкладке навигации (например, на вкладке «**Project Explorer**») выбрать проект или папку, куда следует скопировать ресурсы.

4. В контекстном меню вкладки выбрать команду «**Paste**».

Скопировать ресурсы можно также посредством «перетаскивания» нужных ресурсов из одного места в другое, удерживая клавишу **Ctrl**.

Примечание: при копировании связанных ресурсов в рабочей области физическое местоположение ресурсов в файловой системе не меняется. Например, при копировании связанной папки из одного проекта в другой физически расположение этой папки в файловой системе не изменится. Обратите внимание на то, что связанные ресурсы можно копировать только в проект.

6.4.8. Перемещение ресурсов

Пользователь может перемещать ресурсы (например, из одного проекта в другой). Для того чтобы переместить ресурсы, необходимо:

1. Выбрать ресурсы, которые необходимо переместить, на вкладке навигации (например, на вкладке «**Project Explorer**»).

2. В контекстном меню вкладки выбрать команду «**Move**».

3. В окне перемещения ресурсов (Рисунок 91) выбрать проект или папку, куда будут перемещены ресурсы и нажать кнопку «**OK**».

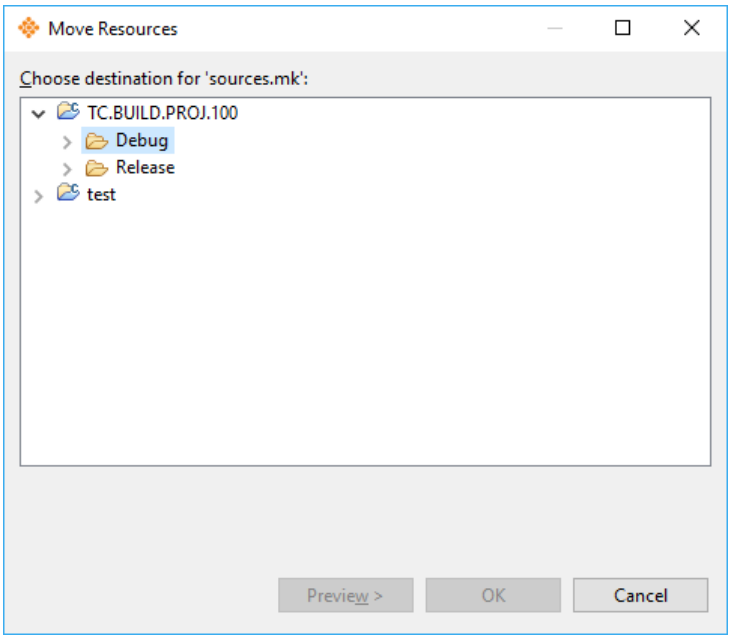


Рисунок 91. Окно «Move Resources»

Переместить ресурсы можно также посредством «перетаскивания» нужных ресурсов из одного места в другое.

Примечание: при перемещении связанных ресурсов в рабочей области физическое местоположение ресурсов в файловой системе не меняется. Например, при перемещении связанной папки из одного проекта в другой физически расположение этой папки в файловой системе не изменится. Обратите внимание на то, что связанные ресурсы можно перемещать только в проект.

6.4.9. Переименование ресурсов

Для того чтобы переименовать ресурс, необходимо:

1. Щелкнуть правой клавишей мыши по ресурсу, который следует переименовать
2. В контекстном меню выбрать пункт «**Rename**».
3. В окне «**Rename Resource**» (Рисунок 92) указать новое название ресурса и нажать кнопку «**OK**».

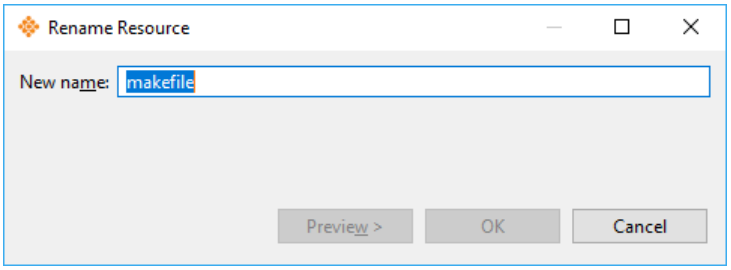


Рисунок 92. Окно «Rename Resource»

Примечание: Переименование проекта приводит к его переиндексации. Для больших проектов этот процесс может занять значительное количество времени. Новое имя должно быть уникальным в рабочей области.

6.4.10. Удаление ресурсов

Для того чтобы удалить ресурс, необходимо:

1. Выбрать ресурс, который следует удалить. Можно выбрать сразу несколько ресурсов для удаления, удерживая клавишу «**Ctrl**».

2. В контекстном меню выбрать пункт «**Delete**» или нажать на клавишу «**Delete**».

   При удалении проекта будет предложен выбор (см. Рисунок 93) варианта удаления проекта (с удалением содержимого проекта в файловой системе или без).

3. Выбрать один из вариантов:

   - Для удаления проекта вместе с содержимым уставить флаг «**Delete project contents on disk (cannot be undone)**» и нажать на кнопку «**OK**».
   - Для удаления проекта без удаления его содержимого или для удаления папки, файла подтвердить операцию, нажав на кнопку «**OK**».

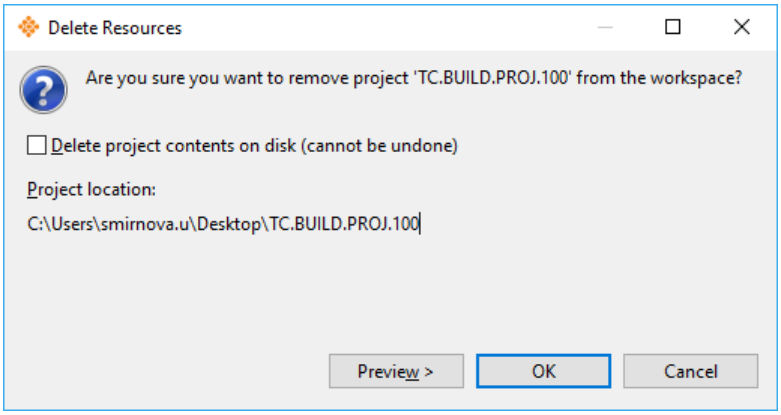


Рисунок 93. Окно «Delete Resources» при удалении проекта

Примечания:

1. При удалении ресурсов из рабочего пространства также удаляются соответствующие им файлы и папки из локальной файловой системы. Исключением является удаление связанных ресурсов: соответствующие им файлы и папки из локальной файловой системы не удаляются. Однако, удаление ресурсов, для которых связанная папка является родительской, вызывает их удаление и из файловой системы.

2. В большинстве случаев, есть возможность восстановить удаленные ресурсы из локальной истории.

6.4.11. Закрытие проектов

Когда проект закрыт, он больше не может быть изменен в рабочей среде и его ресурсы больше не отображаются, но они по-прежнему находятся в локальной файловой системе. Закрытые проекты требуют меньше памяти. Закрытые проекты не проверяются во время сборки, поэтому закрытие проекта может уменьшить время сборки.

Для того чтобы закрыть проект, необходимо:

1. Выбрать проект, который следует закрыть.

2. В контекстном меню выбрать пункт «**Close Project**».

После этого работа с ресурсами закрытого проекта будет недоступна, но его наименование будет отображаться в рабочей области.

Для того чтобы заново открыть проект, необходимо:

1. Выбрать проект, который следует открыть.

2. В контекстном меню выбрать пункт «**Open Project**».

6.4.12. Смена рабочего пространства

Примечание: Смена текущего рабочего пространства на рабочее пространство одной из предыдущих версий CM-LYNX настоятельно не рекомендуется. Для работы CM-LYNX с проектами, созданными с помощью более ранних версий, необходимо воспользоваться стандартной процедурой импорта проектов (см. п. 6.4.5. Импорт существующего проекта). При импорте рекомендуется установить флаг «**Copy projects into workspace**». В результате импорта будет выполнена автоматическая конвертация проекта «устаревшего» формата в формат новой версии CM-LYNX.

Все проекты сохраняются в директорию, называемую *Рабочее пространство* (Workspace). Рабочее пространство (Workspace) задается при запуске CM-LYNX (см. п. 6.1 Загрузка и запуск программы).

Для смены рабочего пространства нужно выполнить следующую последовательность действий:

1. Выбрать в меню «**File**» команду «**Switch workspace->Other...**».

2. Указать новую директорию для использования в качестве рабочего пространства текущей сессии (см. Рисунок 94).

3. При необходимости задать настройки «**Copy Settings**». Для копирования настроек Workbench установить флаг «**Workbench Layout**»; для копирования пользовательских настроек – флаг «**Working Sets**».

4. Нажать кнопку «**OK**».

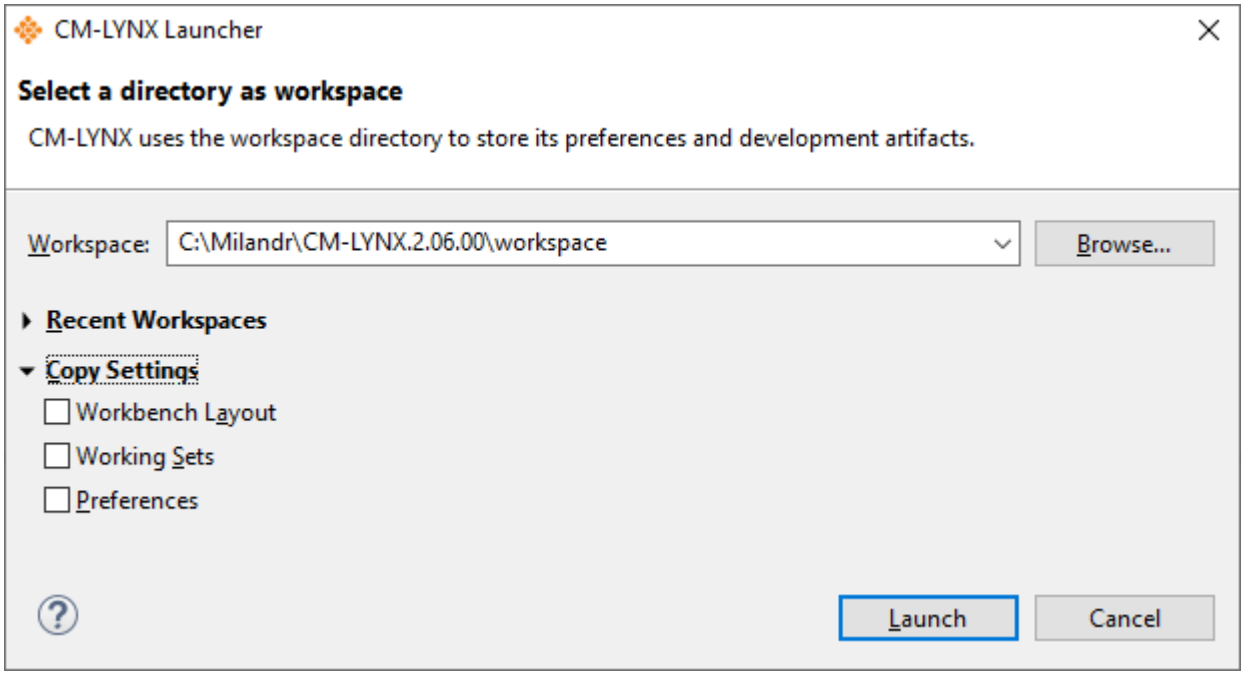


Рисунок 94. Смена Workspace

6.5. Работа с репозиторием Git

Начиная с версии 2.00.00 CM-LYNX поддерживает плагин Egit, предназначенный для контроля версий при работе с ресурсами Git.

Проекты с поддержкой Git отмечаются в дереве проектов значком .

6.5.1. Подключение контроля версий Git

Для того чтобы включить для проекта контроль версий на ресурсах Git, следует

1. В контекстном меню проекта выбрать **Team->Share Project.**

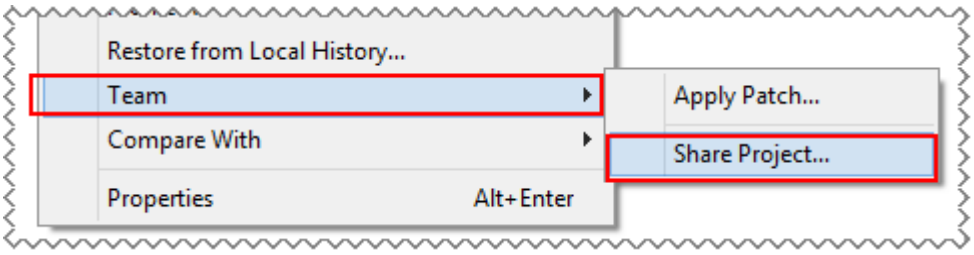


Рисунок 95. Подключение контроля версий

2. В открывшемся диалоговом окне задать репозиторий (Рисунок 96). Для этого выбрать уже существующий или создать новый при помощи кнопки **Create**.

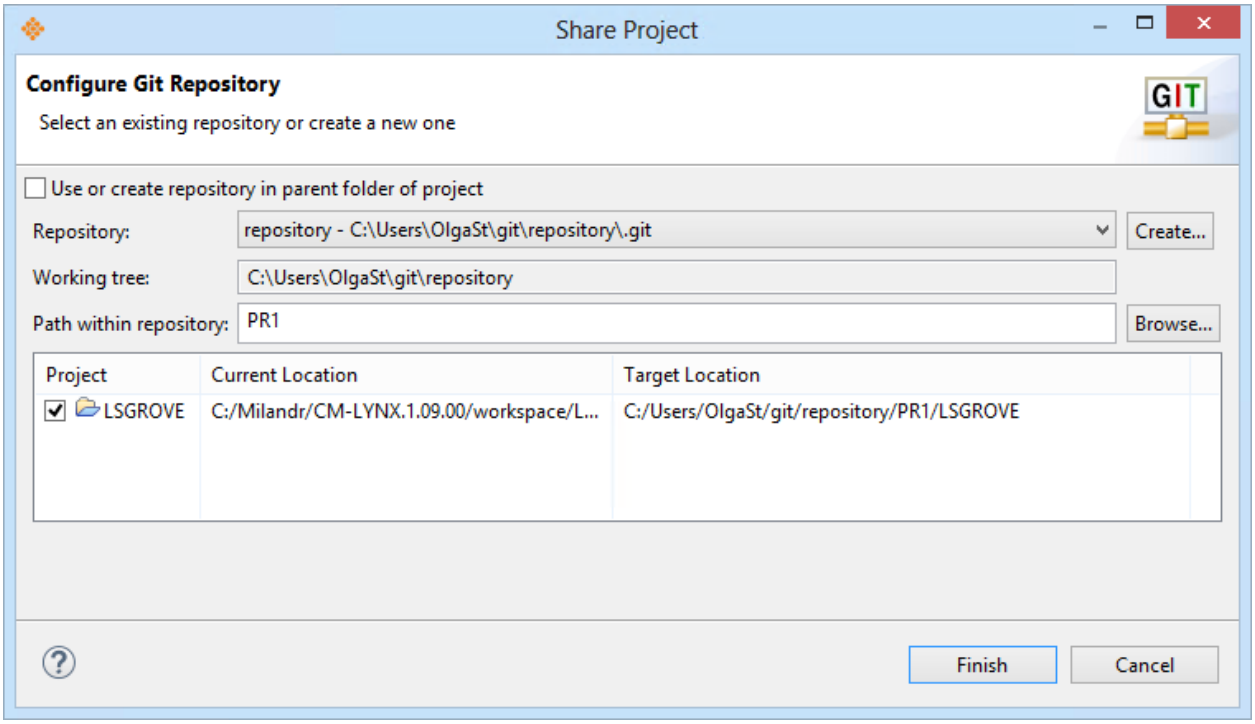


Рисунок 96. Подключение контроля версий 2

3. Нажать кнопку **Finish**.

6.5.2. Базовые операции меню «Team»

В результате для проекта станут доступны базовые операции Git, которые можно вызвать в контекстном меню проекта в пункте **Team**:

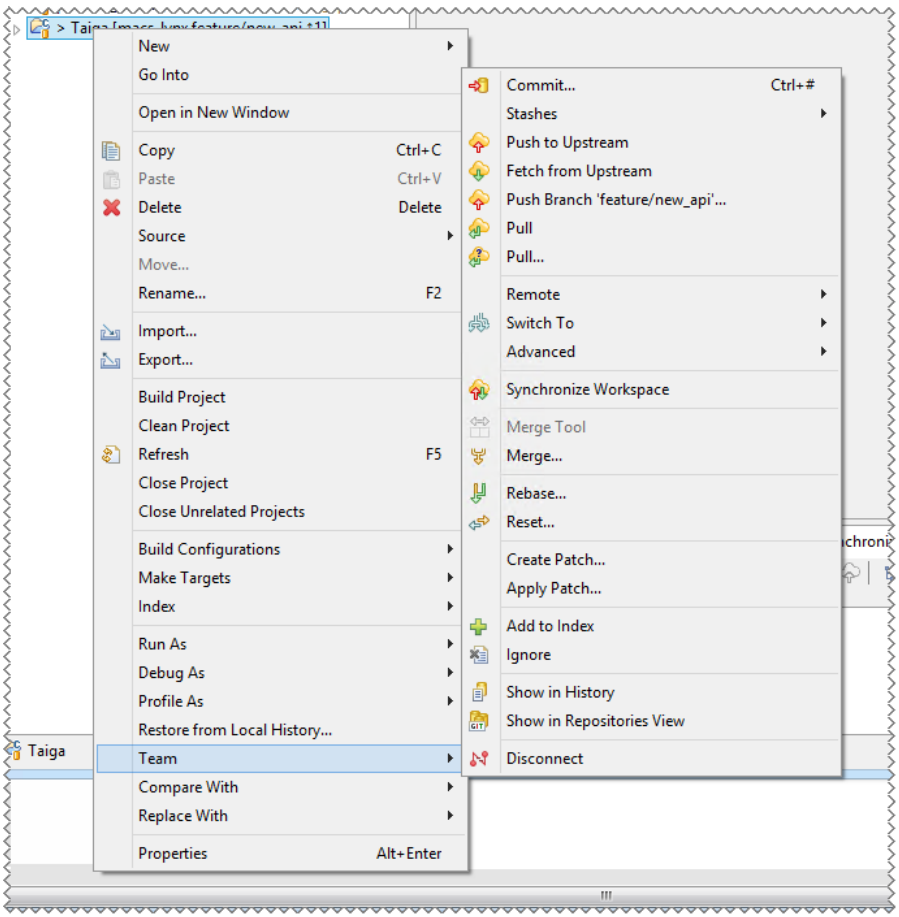


Рисунок 97. Контекстное меню операций плагина Egit

Таблица 42. Пункты подменю «Team»

Значок	Название	Описание функционирования
	Commit	Зафиксировать изменение в отслеживаемом файле
	Stashes	Внести правки в хранилище (stash)
	Push to Upstream	Отправить изменения, выполненные локально, на удалённый ресурс.
	Fetch from Upstream	Загрузить с удалённого ресурса
	Push Branch “<name>”	Отправить изменённую ветку на удалённый ресурс.
	Pull	Получить с удалённого ресурса
	Pull…	Получить определённый элемент с удалённого ресурса
	Remote	Удалённый режим
	Switch To	Перейти к
	Advanced	Расширенные настройки
	Synchronize Workspace	Синхронизировать локальный репозиторий с удалённым
	Merge Tool	Провести слияние инструментов

Значок	Название	Описание функционирования
	Merge…	Провести слияние
	Rebase…	Перебазировать коммиты…
	Reset…	Переподключиться…
	Create Patch…	Создать патч
	Apply Patch…	Применить патч
	Add to Index	Добавить в предметный указатель
	Remove from Index	Удалить из предметного указателя
	Ignore	Проигнорировать
	Show in History	Показать на вкладке «History»
	Show in Repository View	Показать на вкладке «Repository»
	Disconnect	Отключиться

6.5.3. Вкладка «Git Repositories»

Для работы с репозиторием следует перейти в меню **Window > Show View > Git Repositories…**.

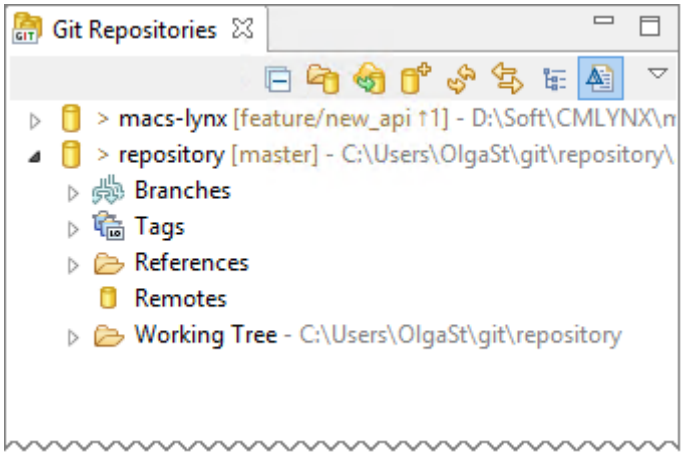


Рисунок 98. Вкладка «Git Repositories»

Таблица 43. Кнопки панели инструментов вкладки «Git Repositories»

Кнопка	Название	Описание функционирования
	Collapse All	Развернуть все.
	Add a Repository	Добавить на вкладку уже существующий локальный репозиторий.
	Clone a Repository	Скопировать уже существующий репозиторий.

Кнопка	Название	Описание функционирования
	Create a New Repository	Создать новый репозитоий git и добавить на эту вкладку.
	Refresh	Обновить.
	Link with Selection	Подключиться к выбранным элементам.
	Hierarchical Branch Layout	Показать иерархию веток.
	Display Latest Branch Commit	Показать последний коммит ветки.

6.5.4. Перспектива Git

Перспектива Git открывается при помощи меню: **Window > Perspective > Other...** и в диалоговом окне выбрать Git.

На вкладке «**History**» отображаются последние коммиты и изменения, сделенные в репозитории.

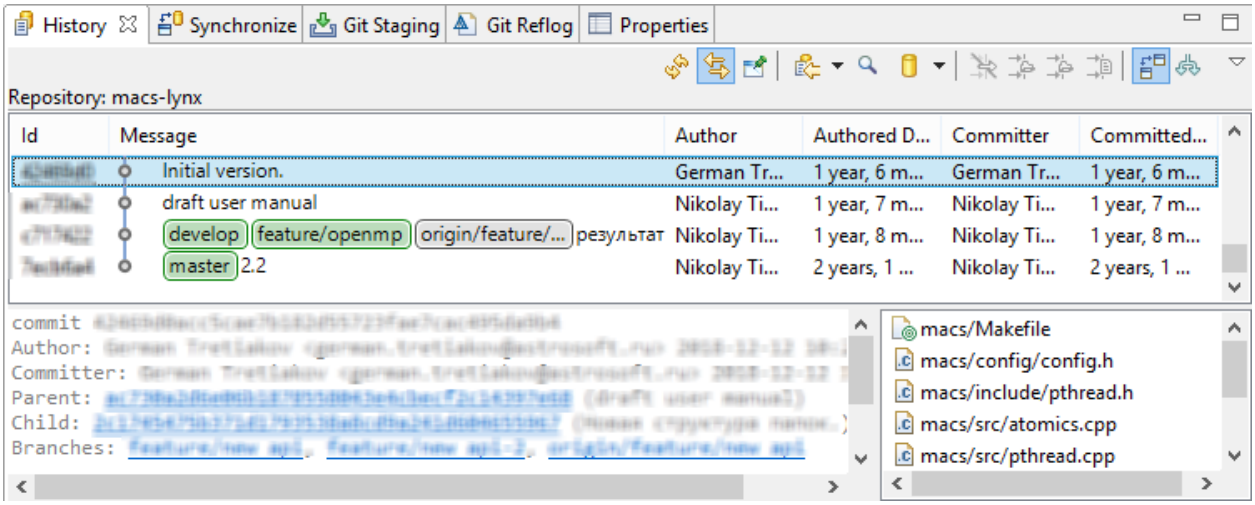


Рисунок 99. Вкладка «History»

Таблица 44. Кнопки панели инструментов вкладки «History»

Кнопка	Название	Описание функционирования
	Refresh	Обновить.
	Link with Editor and Selection	Подключиться к редактору и выбранным элементам.
	Pin This History View	Закрепить окно за текущим выбранным контекстом.
	Show Repository	Показать репозиторий.
	Show Find Toolbar	Показать строку поиска.
	Switch Repository	Подключиться к другому репозиторию.
	All Changes in Repository	Показать все изменения.

Кнопка	Название	Описание функционирования
	All Changes in Project	Показать все изменения в проекте.
	All Changes in Parent Folder	Показать все изменения в родительской папке.
	All Changes of Resource	Показать все изменения ресурса.
	Compare Mode	Включить режим сравнения.
	Show All Branches and Tags	Показать все ветки и теги.

На вкладке «**Synchronize**» отображается процесс синхронизации репозиториев и отслеживаются конфликты.

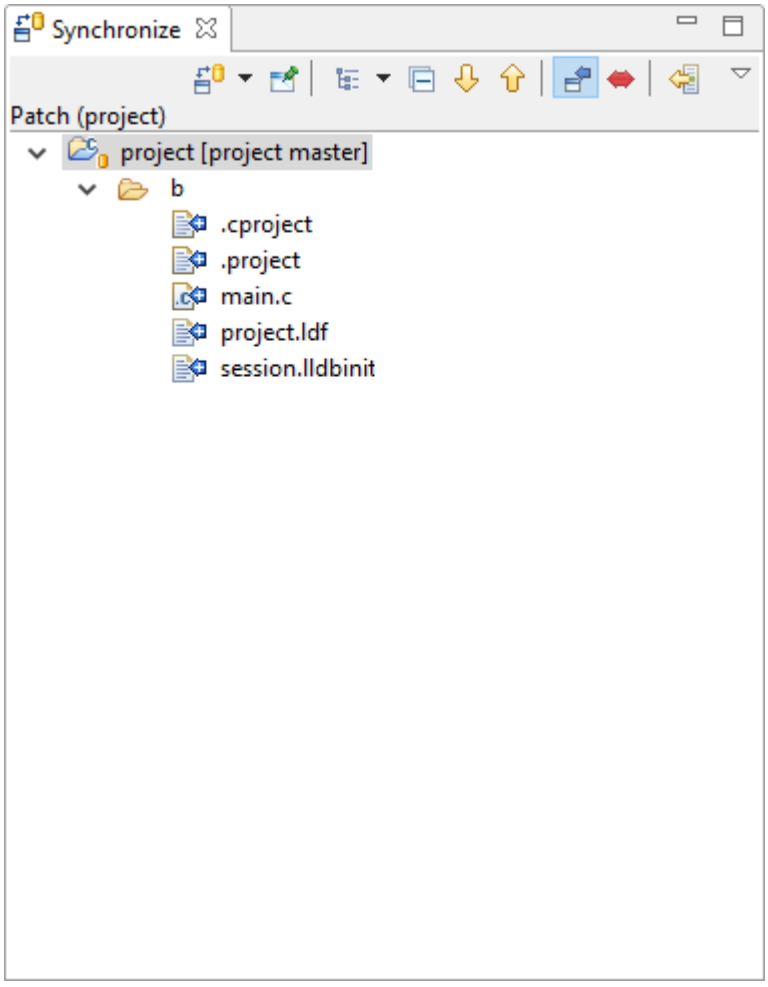


Рисунок 100. Вкладка «Synchronize»

Таблица 45. Кнопки панели инструментов вкладки «Synchronize»

Кнопка	Название	Описание функционирования
	Synchronize Git	Синхронизировать локальный и удалённый репозитории.
	Pin This Synchronize View	Закрепить окно за текущим выбранным контекстом.

Кнопка	Название	Описание функционирования
	Pull	Забрать с удалённого репозитория.
	Push	Отправить на удалённый репозиторий.
	Show File System Resources	Показать ресурсы файловой системы.
	Expand All	Всё свернуть.
	Collapse All	Всё развернуть.
	Next Difference	Следующее отличие.
	Previous Difference	Предыдущее отличие.
	Incoming Mode	Включить режим входящих изменений.
	Outcoming Mode	Включить режим исходящих изменений
	Incoming/Outcoming Mode	Включить режим входящих и исходящих изменений.
	Conflicts Mode	Включить режим отслеживания конфликтов.
	Merge All Non-Condlicting Changes	Провести слияние всех неконфликтующих изменений.

На вкладке «**Git Staging**» отслеживаются файлы, которые ещё не попали в коммит.

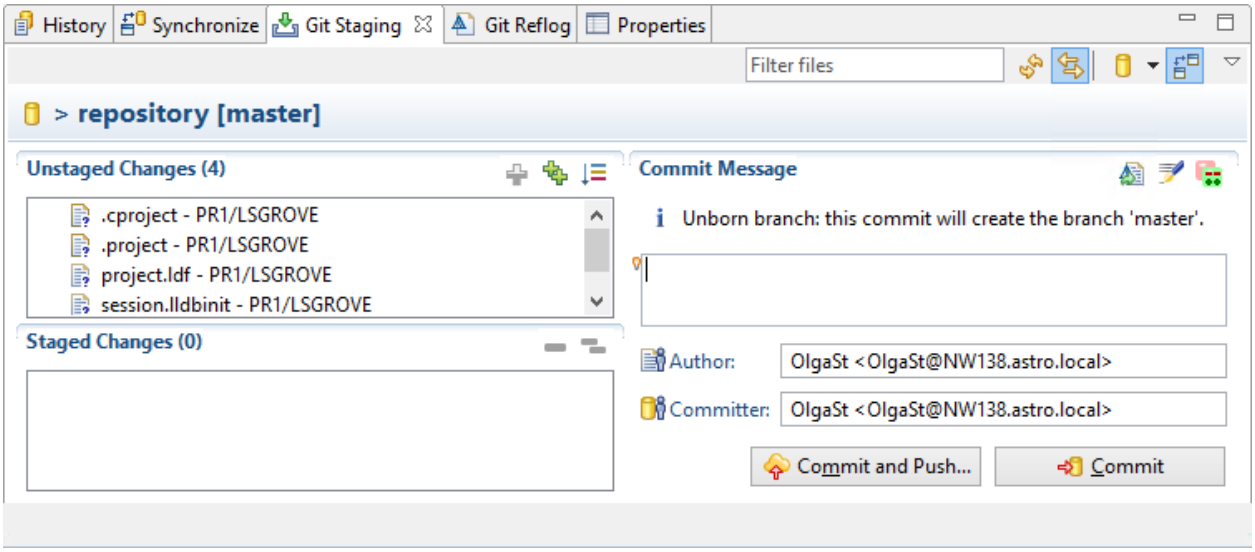


Рисунок 101. Вкладка «Git Staging»

Таблица 46. Кнопки панели инструментов вкладки «Git Staging»

Кнопка	Название	Описание функционирования
	Refresh	Обновить.
	Link with Editor and Selection	Подключиться к редактору и выбранным элементам.

Кнопка	Название	Описание функционирования
	Switch Repository	Подключиться к другому репозиторию.
	Compare Mode	Включить режим сравнения.

На вкладке «**Git Reflog**» открывается журнал ссылок.

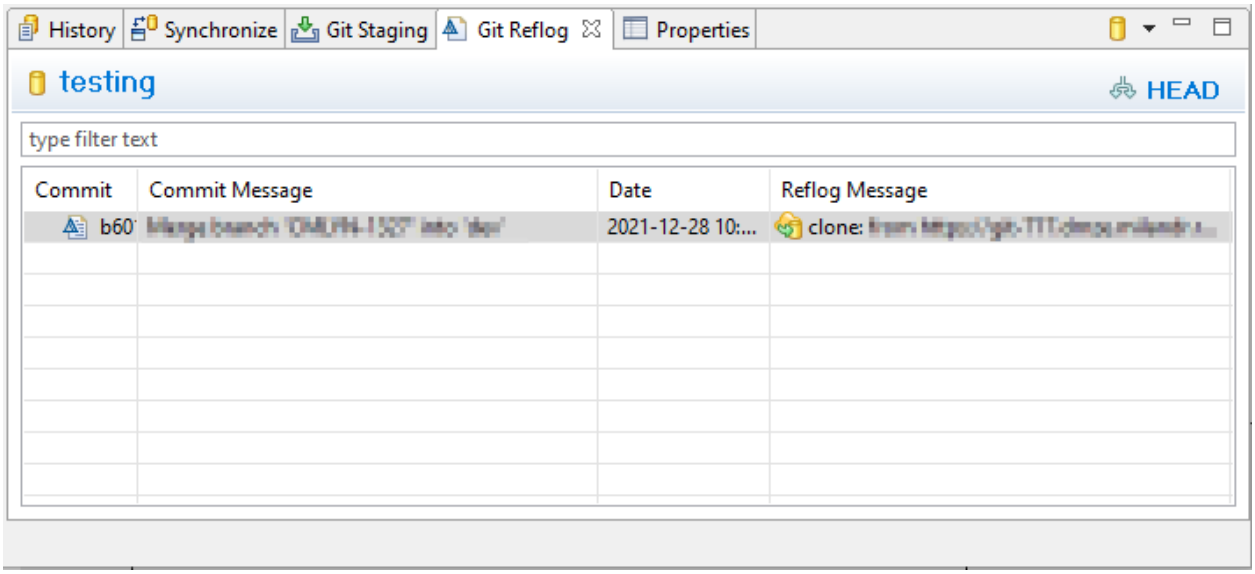


Рисунок 102. Вкладка «Git Reflog»

Таблица 47. Кнопки панели инструментов вкладки «Git Reflog»

Кнопка	Название	Описание функционирования
	Switch Repository	Подключиться к другому репозиторию.
	Choose branch	Выбрать ветку

На вкладке «**Properties**» открываются на просмотр значения репозитория.

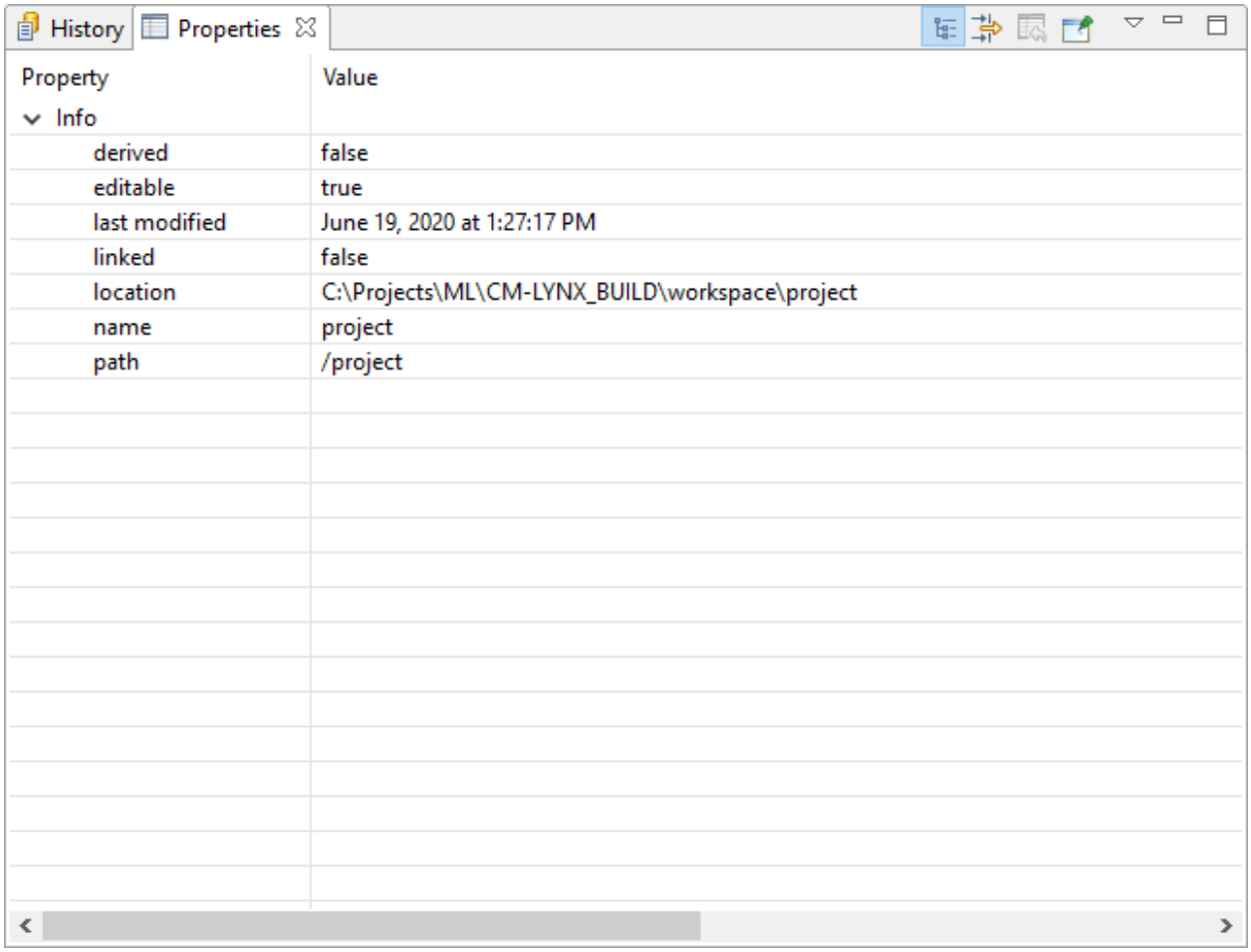


Рисунок 103. Вкладка «Properties»

Таблица 48. Кнопки панели инструментов вкладки «Properties»

Кнопка	Название	Описание функционирования	
	Show Categories	Показать категории.	
	Show Advanced Properties	Показать расширенные настройки.	
	Restore Default Value	Восстановить значения по умолчанию.	
	Pin This Properties View	Закрепить окно за текущим выбранным контекстом.	

Полную справку по работе с данным плагином можно найти в **Help > Help Contents >EGit Documentation**.

6.6. Работа с редакторами

Большинство режимов работы среды состоят из области *редактора* и одной или нескольких вкладок.

Можно проассоциировать разные *редакторы* с разными типами файлов. Например, когда файл открывается для редактирования (двойным щелчком мыши на одной из вкладок работы с ресурсами: например, на панели «**Project Explorer**»), ассоциированный с ним редактор откроет его в рабочей области программы.

Если для ресурса нет ассоциированных редакторов, то при открытии такого файла на редактирование CM-LYNX попытается запустить *внешний редактор* вне среды разработки.

Одновременно может быть открыто любое количество редакторов, но активным является только один. Панель основного меню и панель инструментов для окна программы содержат операции, применимые в активном редакторе.

Вкладки в области редактора отображают названия ресурсов, которые в данный момент открыты для редактирования. Звездочка (*) обозначает, что в редакторе есть несохраненные изменения.

По умолчанию редакторы сгруппированы в *области редакторов*, но может быть выбрано их расположение рядом друг с другом, чтобы одновременно видеть исходные файлы.

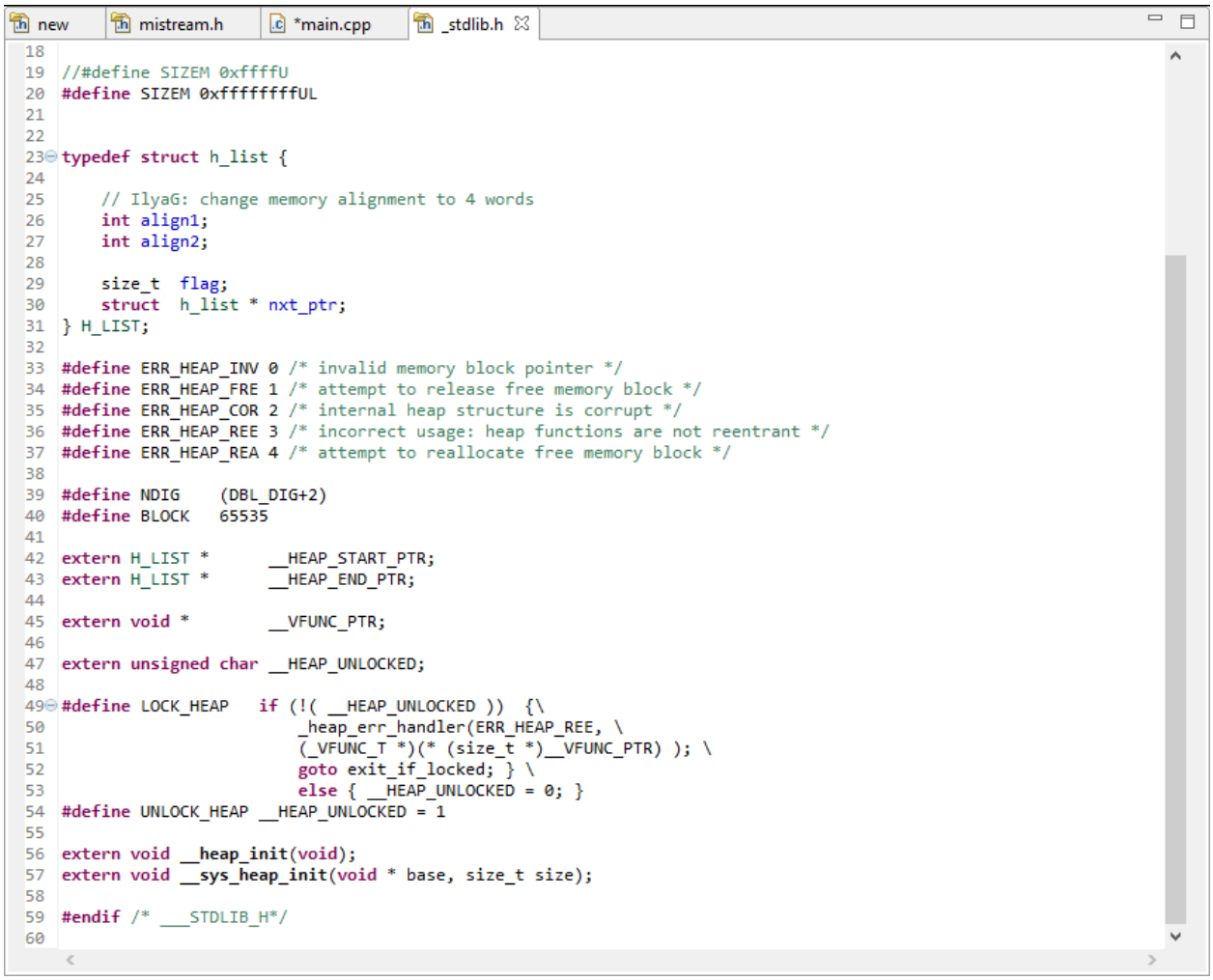


Рисунок 104. Область редакторов

Серая граница на левом краю области редактора может содержать значки, информирующие об ошибках, предупреждениях или проблемам, зафиксированных средой разработки. Значки также отображаются, если были созданы закладки, добавлены точки останова для отладки или настроены задания на вкладке «**Tasks**». Можно просмотреть подробную информацию о любых значках в левом поле редактора, наведя на них курсор мыши.

6.6.1. C/C++ редактор

Редактор C/C++ вызывается автоматически при редактировании исходного C/C++ файла.

Редактор C/C++ выполняет следующие функции:

- выделение синтаксиса;
- помощник кода/содержимого;

- сворачивание кода;
- встроенные функции отладки.

Настройка ряда операций выполняется в окне «**Preferences**» на панели «**C/C++**» (см. п. 6.12.2 Настройки режима работы C/C++).

6.6.2. Внешние редакторы

При необходимости можно использовать *внешнюю программу* для редактирования файлов в CM-LYNX, например, когда в CM-LYNX нет редактора для выбранного типа файлов.

Внешняя программа будет использована по умолчанию, если она зарегистрирована как редактор по умолчанию для файлов данного типа, и никакой другой редактор не зарегистрирован в CM-LYNX для таких файлов. Например, в большинстве систем редактор по умолчанию для файлов JPEG – это приложение для просмотра и редактирования изображений. Если в CM-LYNX нет редактора, ассоциированного с файлами .jpg и/или .jpeg, открытие JPEG из среды разработки приведет к открытию файла вне среды, в системном редакторе по умолчанию.

Открыть *внешнюю программу*, которая не является редактором по умолчанию, можно с помощью контекстного меню, выбрав команду **Open With → Other...** *Внешние программы* могут быть зарегистрированы как редактор для данного типа файлов. Для регистрации редактора используется страница свойств файлов (см. п. 6.6.4. Настройка ассоциации редактора с типом файлов).

6.6.3. Открытие файла на редактирование

Открыть файл на редактирование можно одним из способов:

- Двойным щелчком по файлу на панели «**Project Explorer**» (см. п. 5.3. Панель «Project Explorer»), на вкладке «**C/C++ Projects**» или «**Navigator**»»;
- Выделив файл для редактирования щелчком правой мыши и выбрав пункт «**Open**» в появившемся контекстном меню;
- Двойным щелчком по заметке, ассоциированной с файлом на вкладке «**Bookmarks**»;
- Двойным щелчком по ошибке/предупреждению/заданию, ассоциированному с файлом на вкладке «**Problems**».

Выше указанные способы откроют файл на редактирование с помощью редактора по умолчанию. Для открытия файла с помощью другого редактора необходимо использовать пункт «**Open With**» в контекстном меню.

6.6.4. Настройка ассоциации редактора с типом файлов

Для того чтобы ассоциировать редактор с различными типами файлов, необходимо:

1. В меню «**Window**» выбрать пункт «**Preferences**».
2. В окне «**Preferences**» выбрать «**General -> Editors -> File Associations**».

   Откроется страница «**File Associations»** (Рисунок 105):

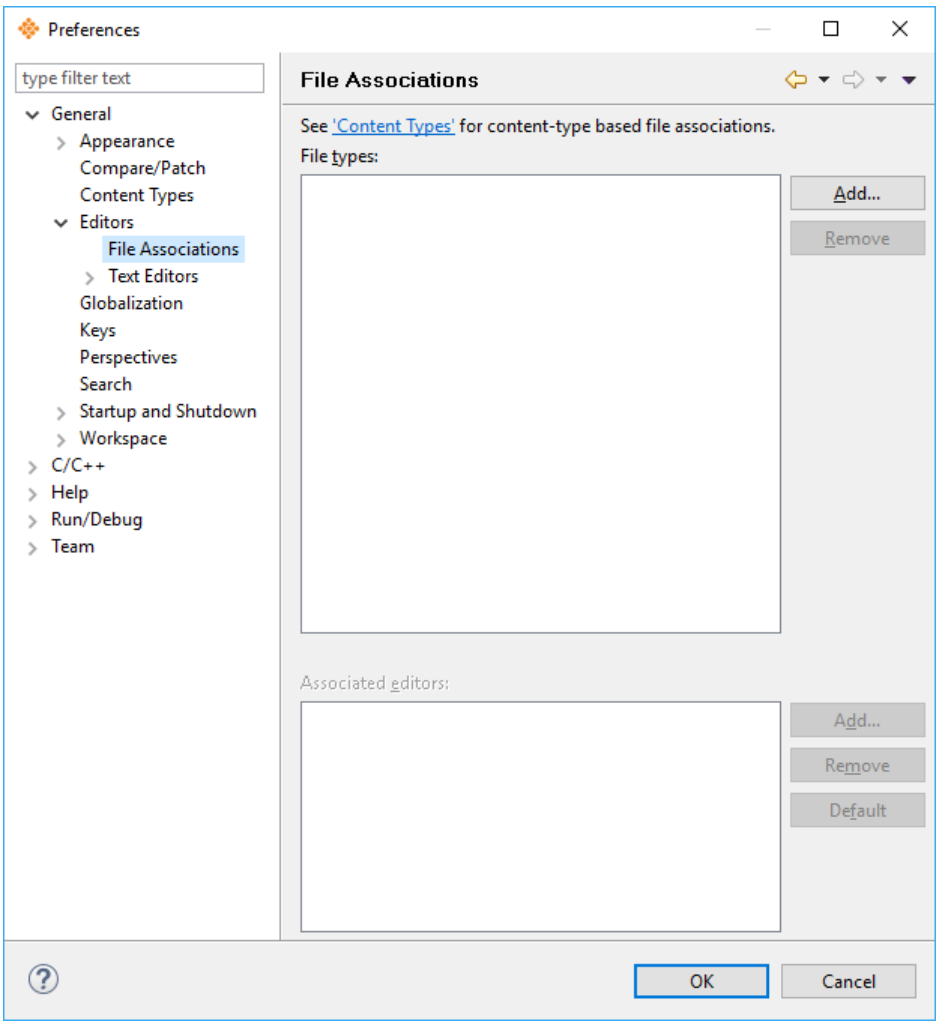


Рисунок 105. Окно «File Associations»

3. Выбрать тип файла из списка «**File types**». Если нужный тип файла отсутствует в списке, его можно добавить. Для этого необходимо:

    1) Нажать на кнопку «**Add...**».

    Откроется окно для добавления типа файла.

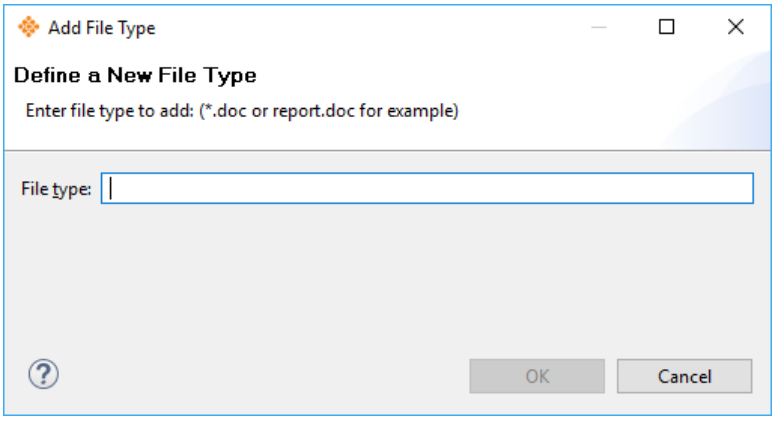


Рисунок 106. Окно «Add File Type»

    2) Ввести нужный тип файла (например, *.ldf).

    3) Нажать на кнопку «**OK**».

4. В списке «**Associated editors**» выбрать редактор, который будет ассоциирован с данным типом файла. Чтобы добавить редактор в список, необходимо:

    1) Нажать на кнопку «**Add...**». Откроется страница (Рисунок 107).

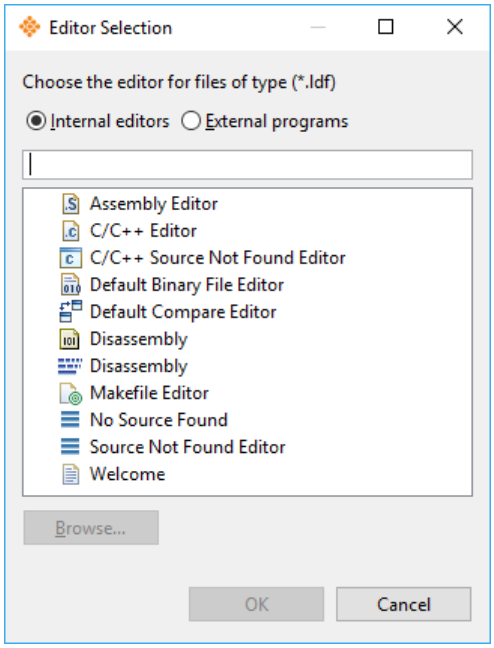


Рисунок 107. Окно «Editor Selection»

    2) Выбрать редактор из предложенного перечня внутренних редакторов.

        При отсутствии подходящего внутреннего редактора или желании использовать внешний редактор для открытия данного типа файла необходимо установить переключатель в положение «**External Programs**» и выбрать из списка одну из внешних программ. Можно воспользоваться кнопкой «**Browse**» для выбора файла из файловой системы.

        Внешний редактор будет открывать файлы указанного типа вне рабочей области среды разработки.

    3) Нажать на кнопку «**OK**» после выбора редактора.

5. Для одного типа файла может быть ассоциировано более одного редактора. Для ассоциации одного типа файла с несколькими редакторами необходимо повторить шаги 3-4 для каждого из редакторов. Если с одним типом файла ассоциировано несколько редакторов, следует указать редактор по умолчанию. Для этого необходимо выбрать редактор в списке «**Associated editors**» и нажать кнопку «**Default**».

6. По завершении настройки нажать на кнопку «**OK**».

6.6.5. Редактирование файлов вне рабочей среды

Для того чтобы отредактировать файл вне рабочей среды, необходимо:

1. Перейти в файловой системе в каталог рабочей области и открыть файл, который необходимо изменить, с помощью внешнего редактора.

2. Отредактировать файл по мере необходимости, сохранить и закрыть его.

3. Щелкнуть правой кнопкой мыши отредактированный файл в среде на вкладке работы с ресурсами (например, на панели «**Project Explorer**») и выбрать пункт «**Refresh**» в контекстном меню.

Среда разработки автоматически обработает изменения, внесенные вне рабочей среды.

При регулярной работе с внешними редакторами рекомендуется включить автоматическое обновление. Для этого необходимо:

1. Перейти в диалоговое окно «**Preferences**» (выбрав пункт «**Preferences**» в меню «**Window**»).
2. Перейти на страницу «**General -> Workspace**» (см. Рисунок 265).
3. Установить флаги «**Refresh using native hooks or polling**» и «**Refresh on acces**s».

6.6.6. Преднастройка редактора C/C++

Пользователь может изменить некоторые настройки редактора C/C++ так, как ему удобно. Настройка этих параметров доступна в окне «**Preferences**» на панели «**C/C++->Editor**» (см. п. 6.12.2.5. Настройки редактора C/C++).

Для того чтобы изменить некоторые параметры работы редактора C/C++, необходимо:

1. В меню «**Window**» выбрать пункт «**Preferences**».
2. Развернуть элемент **C/C++»** и выбрать пункт «**Editor»**.
3. Настроить основные параметры (см. п. 6.12.2.5. Настройки редактора C/C++) и нажать кнопку «**Apply**».
4. Для настройки цветов, используемых в области редактора, выбрать пункт «**Syntax Coloring**», настроить используемые в области редактора цвета (см. п. 6.12.2.5.7. кода) и нажать кнопку «**Apply**».
5. Для настройки параметров работы Content Assist (см. п. 6.12.2.5.1. Настройки автозавершения) выбрать пункт «**Content Assist**», изменить необходимые настройки и нажать кнопку «**Apply**».
6. По завершении настройки нажать на кнопку «**OK**».

6.6.7. Комментирование кода

Можно закомментировать одну или несколько строк кода в окне редактора C/C++. Для того чтобы закомментировать одну или несколько строк кода, следует добавить ведущие символы **//** в начало каждой строки. Также можно закомментировать сразу нескольких строк (фрагмент) кода, используя символы **/******/**.

Для того чтобы добавить комментарий в виде **//** в код, необходимо:

1. В редакторе C/C++ выбрать строку или строки, которые необходимо закомментировать.

   Если строки не выделены, то комментарий будет добавлен в текущую позицию курсора.

2. В контекстном меню, которое вызывается правой кнопкой мыши, выбрать команду « **Source > Toggle Comment**» или использовать комбинацию горячих клавиш **Ctrl+/**.

Для того чтобы добавить комментарий в виде **/* */** в код, необходимо:

1. В редакторе C/C++ выделить фрагмент кода, который необходимо закомментировать.
2. В контекстном меню, которое вызывается правой кнопкой мыши, выбрать команду « **Source > Add Block Comment**» или использовать комбинацию горячих клавиш **Ctrl+Shift+/**.

3. Для того чтобы удалить комментарий фрагмента кода, необходимо в контекстном меню, которое вызывается правой кнопкой мыши, выбрать команду « **Source > Remove Block Comment**» или использовать комбинацию горячих клавиш **Ctrl+Shift+**.

6.6.8. Работа с функцией завершения кода (Content Assist)

Функция завершения кода (Content Assist) – это набор встроенных инструментов, которые позволяют уменьшить количество нажатий клавиш, необходимых для создания кода. Функция завершения кода состоит из нескольких компонентов, которые прогнозируют, что разработчик будет вводить, на основе текущего контекста, области и префикса.

Content assist обеспечивает завершение кода в любом месте документа. Для текущего проекта отображается список элементов, которые начинаются с введенной комбинации букв, а релевантность каждого предложения определяется в следующем порядке:

- Поля (Fields);
- Переменные (Variables);
- Методы (Methods);
- Функции (Functions);
- Классы (Classes);
- Структуры (Structs);
- Объединения (Unions);
- Пространство имен (Namespaces);
- Перечисления (Enumerations).

Можно создавать и сохранять шаблоны для часто используемых секций кода, которые будут вставляться в соответствии с областью действия. Content Assist также обеспечивает быстрый доступ к шаблонам кода. При вызове функции завершения кода отображается список элементов кода и шаблонов, которые начинаются с введенной комбинации букв. Можно выбрать элемент или шаблон из списка и вставить его непосредственно в код.

Функция завершения кода запускается при вводе комбинации клавиш «**Ctrl+пробел**», также она автоматически активируется при вводе «.», «->» или «::».

Если найдено только одно совпадение, оно подставляется в текущую позицию курсора автоматически.

Если совпадения отсутствуют, в статусе будет отображаться сообщение о том, что совпадений не найдено.

6.6.8.1. Использование функции завершения кода

Content Assist можно использовать для вставки элементов C / C++ и шаблонов (для часто используемых фрагментов кода) в код.

Для этого необходимо:

1. В Редакторе C/C++ ввести по крайней мере первую букву шаблона или элемента.

2. Нажать комбинацию клавиш «**Ctrl+пробел**».

   Отобразится список элементов и шаблонов (шаблоны помечаются значком), начинающихся с введенного символа (последовательности символов).

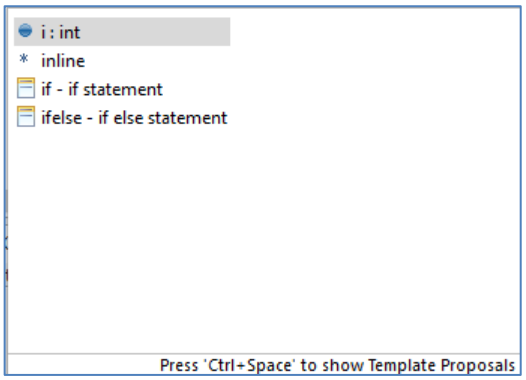


Рисунок 108. Окно «Content Assist»

3. Выполнить одно из действий:
   - продолжить ввод (список будет сокращаться; как только в списке останется один элемент, он будет подставлен в код автоматически);
   - дважды щелкнуть по нужному элементу или шаблону для вставки его в код;
   - нажать на клавишу «**Esc**» для закрытия окна с предложенными вариантами при отсутствии подходящего варианта.

6.6.8.2. Настройка шаблонов

Можно настроить шаблоны, позволяющие быстро подставить в код часто используемые фрагменты кода.

Для того чтобы создать такой шаблон, необходимо:

1. В меню «**Window**» выбрать пункт «**Preferences**».
2. Выбрать **C/C++ → Editor → Templates**.
3. Нажать на кнопку «**New...**».
4. Заполнить поля формы:
   - в поле «**Name**» указать имя шаблона;
   - в поле «**Context**» указать тип шаблона: «**C/C++**» или «**Comment**»;
   - в поле «**Description»** ввести описание шаблона;
   - в поле «**Pattern»** ввести код.
   - нажать кнопку «**Insert Variable**», чтобы добавить переменную из списка в код, введенный в поле «**Pattern».**
5. Нажать кнопку «**OK**».

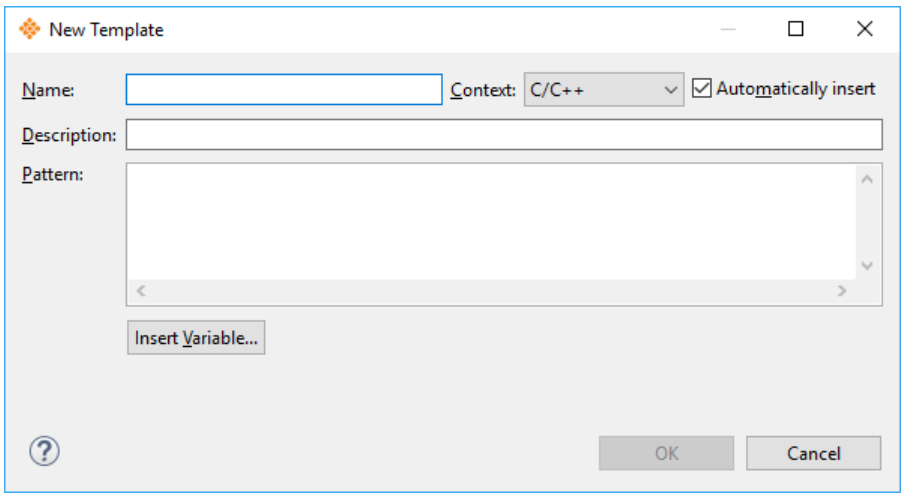


Рисунок 109. Окно «New Template»

Для того чтобы изменить шаблон, необходимо:

1. В меню «**Window**» выбрать пункт «**Preferences**».
2. Выбрать **C/C++ → Editor → Templates**.
3. Выбрать нужный шаблон и нажать кнопку «**Edit...**».
4. Внести правки.
5. Нажать кнопку «**OK**».

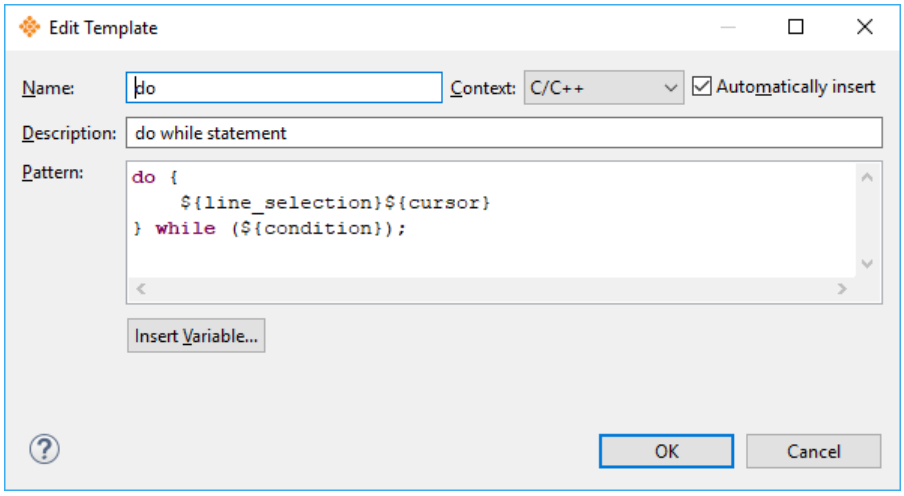


Рисунок 110. Окно «Edit Template»

6.6.8.3. Импорт и экспорт шаблонов

Шаблоны можно импортировать в среду и экспортировать из нее.

Для того чтобы импортировать шаблон в среду, необходимо:

1. В меню «**Window**» выбрать пункт «**Preferences**».
2. Выбрать **C/C++ → Editor → Templates**.
3. Нажать кнопку «**Import...**».

4. Выбрать файл с импортируемым шаблоном.

   Импортируемый шаблон должен быть файлом формате xml, например:

```
<?xml version="1.0" encoding="UTF-8" ?>
<templates>
<template name="author" description="author name" context="C"
enabled="true">author ${user}</template>
</templates>
```

5. Нажать кнопку «**OK**».

Для того чтобы экспортировать шаблон из среды, необходимо:

1. В меню «**Window**» выбрать пункт «**Preferences**».
2. Выбрать **C/C++ → Editor → Templates**.
3. Выбрать шаблон для экспорта и нажать кнопку «**Export...**».
4. Указать путь в файловой системе, куда должен быть сохранен шаблон.
5. Нажать кнопку «**Save**».

6.6.9. Сдвиг строчек кода вправо или влево

Строки кода в редакторе C/C++ можно сдвигать влево или вправо.

Для того чтобы сдвинуть строки кода вправо, необходимо:

1. В редакторе C/C++ выделить все строки, которые требуется переместить.
2. Использовать клавишу «**Tab**».

Для того чтобы сдвинуть строки кода влево, необходимо:

1. В редакторе C/C++ выделите все строки, которые требуется переместить.
2. Использовать комбинацию клавиш «**Shift+Tab**».

6.6.10. Переход к объявлениям C / C++

Можно перейти к объявлению, которое соответствует выбранному элементу в редакторе C/C++. Рекомендуется использовать для поиска объявлений элементов в скомпилированных программах.

Для того чтобы перейти к объявлению C / C++, необходимо:

1. Выделить объект в редакторе C/C++.
2. В контекстном меню выбрать пункт «**Open Declaration**» или нажать клавишу **F3**.

   Откроется окно редактора с объявлением выбранного объекта.

6.6.11. Переименование объектов C / C++

Для переименования переменных, функций, классов, методов, полей или определений типов необходимо:

1. В окне редактора выделить объект, который необходимо переименовать.

2. Контекстном меню выбрать команду «**Refactor > Rename...**» или использовать комбинацию клавиш «**Alt+Shift+R**».

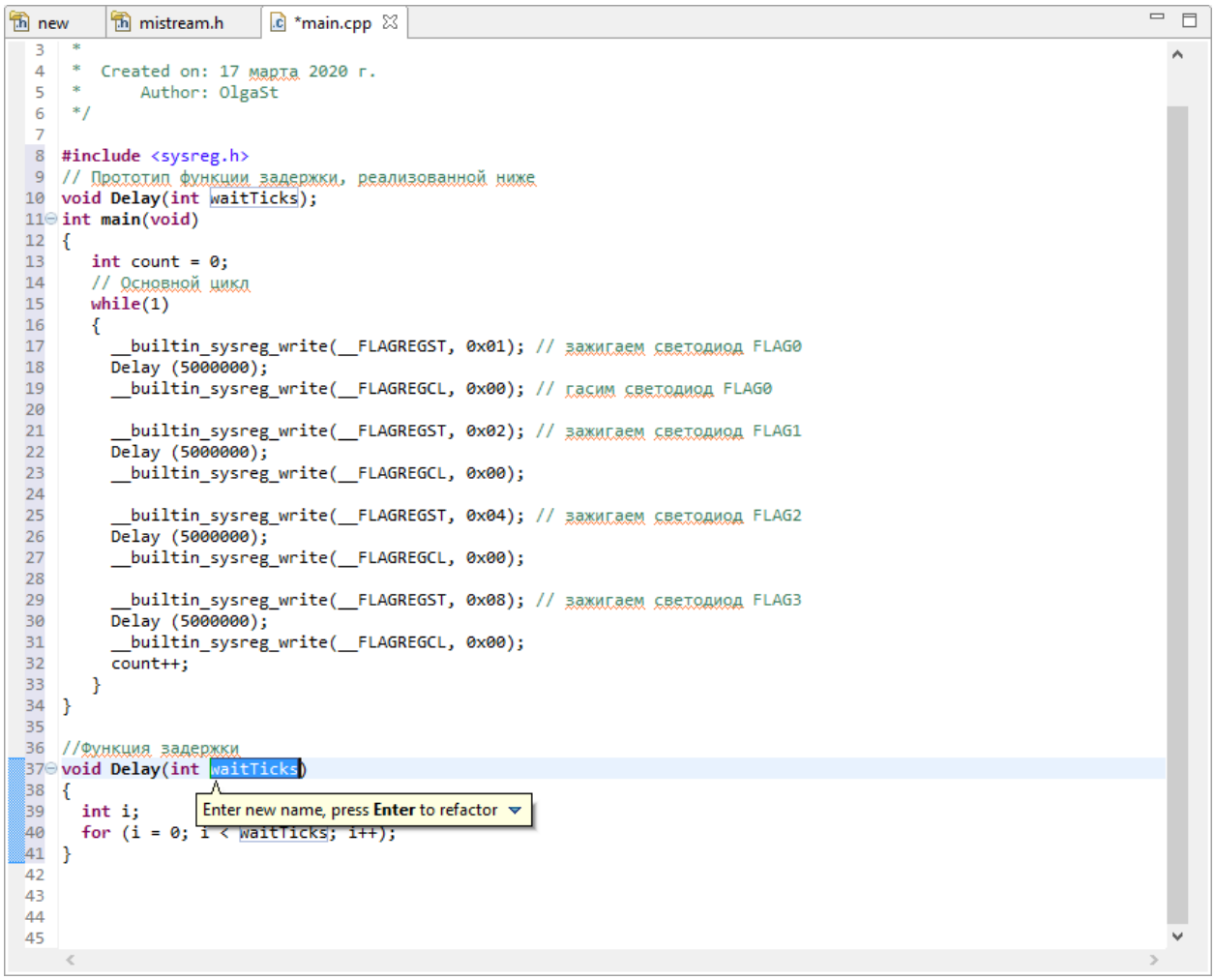


Рисунок 111. Переименование переменной

3. Ввести новое имя.

   Все варианты использования имени выделены; по мере ввода нового имени они обновляются в режиме реального времени. Для предварительного просмотра или изменения каких-либо параметров, можно нажать на треугольник и выбрать пункт «**Open Rename Dialog**» или снова использовать сочетание клавиш «**Alt+Shift+R**».

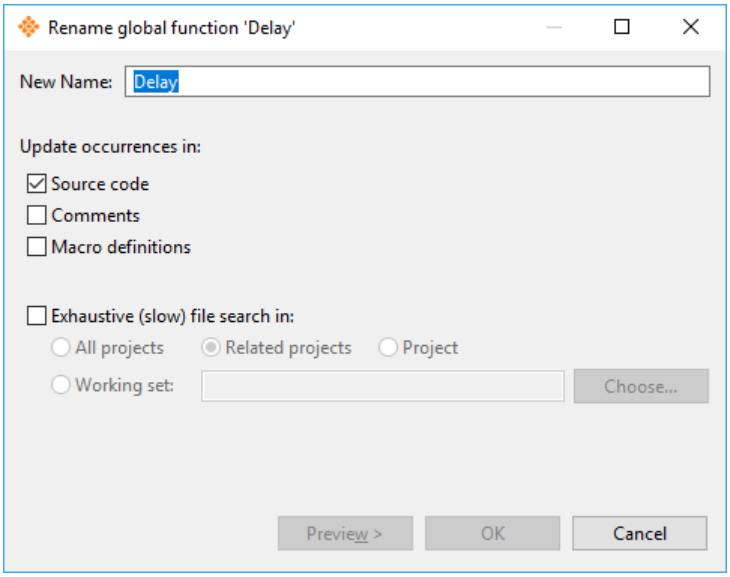


Рисунок 112. Окно «Rename»

С помощью диалогового окна можно указать новое имя и задать различные параметры, влияющие на поиск использований переименованного объекта. С помощью кнопки «**Preview**» доступен предварительный просмотр изменений.

После внесения изменений следует нажать кнопку «**OK**».

4. Нажать клавишу «**Enter**».

   Все объявления, определения и ссылки элемента будут изменены, чтобы использовать новое имя. При необходимости можно воспользоваться стандартной командой отмены изменений.

Примечание: объекты также можно переименовать на вкладке **Project Explorer**.

6.6.12. Переход к определению функции

Для того чтобы перейти к месту, где функция определена, необходимо:

1. В окне редактора C / C++ поместить курсор внутри объявления или определения функции.

2. В контекстном меню выбрать команду «**Refactor > Toggle Function Definition**» или использовать комбинацию клавиш «**Alt+Shift+T**».

   Произойдет переход от определения функции из файла заголовка в файл реализации с соответствующим именем. Если необходимый файл не существует, он будет создан (после диалога подтверждения).

   Также применимо и для методов.

6.6.13. Автоматическое переименование меток в ассемблерных файлах

При разработке ассемблерных файлов (если предполагается использовать механизм исключения неиспользуемого кода компоновщика) существуют требования к расположению и именованию локальных меток. Подробнее см. п. 6.3. документа «Компоновщик MCLINK-TSH. Руководство пользователя».

Для отслеживания корректности именования меток рекомендуется включить флаг Ассемблера «**Pedantic warnings**» (см. п. 6.9.2.4.1.4.1), в результате при нарушении правил именования меток в консоли возможно появление ошибок/предупреждений с кодами A370, A371, A372. Например,

```
[A370] Local label name '.wt_l1' breaks the rules, recommended rename to '.Lwt_l1'
[A371] End label ' HAL MMU WR TLB.END' for function ' HAL MMU WR TLB' not found, recommended to
insert
```

При появлении ошибок/предупреждений с кодами A370, A371 устранить проблемы с именованием меток можно, используя встроенный механизм среды. Для ошибок/предупреждений с кодом A372 возможно только исправление «вручную», непосредственно в тексте файла.

Для того чтобы устранить проблемы с именованием меток в ассемблерных файлах (при появлении ошибок/предупреждений с кодами A370, A371), необходимо:

1. Выбирать по очереди проблемы на вкладке «**Problems**» или сообщения об ошибке/предупреждении на вкладке «**Console**» (двойным кликом мыши на строке в таблице проблем или сообщению).

   В результате курсор переместится в нужную позицию в редакторе.

2. Нажать комбинацию клавиш «**Ctrl+1**».

   Откроется окно выбора действия по исправлению проблемы (см. Рисунок 113, Рисунок 114).

3. Выбрать вариант исправления из предложенных, кликнув по нужном варианту.

   Для сообщений с кодом **A370** поддерживаются два режима исправления (Рисунок 113):

   - **Change all** – переименование метки во всех местах выбранного исходного файла, где она встречается (по умолчанию);
   - **Change selected** – переименование метки только в выбранном месте.

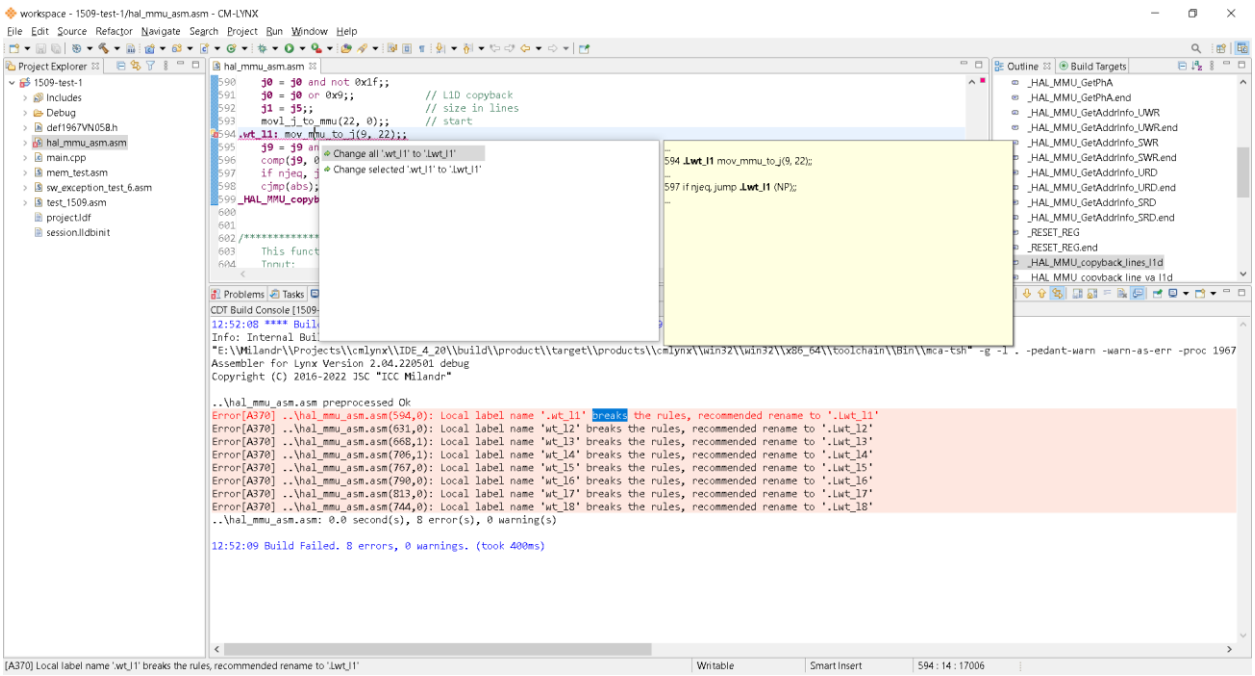


Рисунок 113. Окно автозамены для сообщения с кодом A370

Для сообщений с кодом **A371** можно вставить метку конца функции, выбрав вариант «**Insert**» (Рисунок 114).

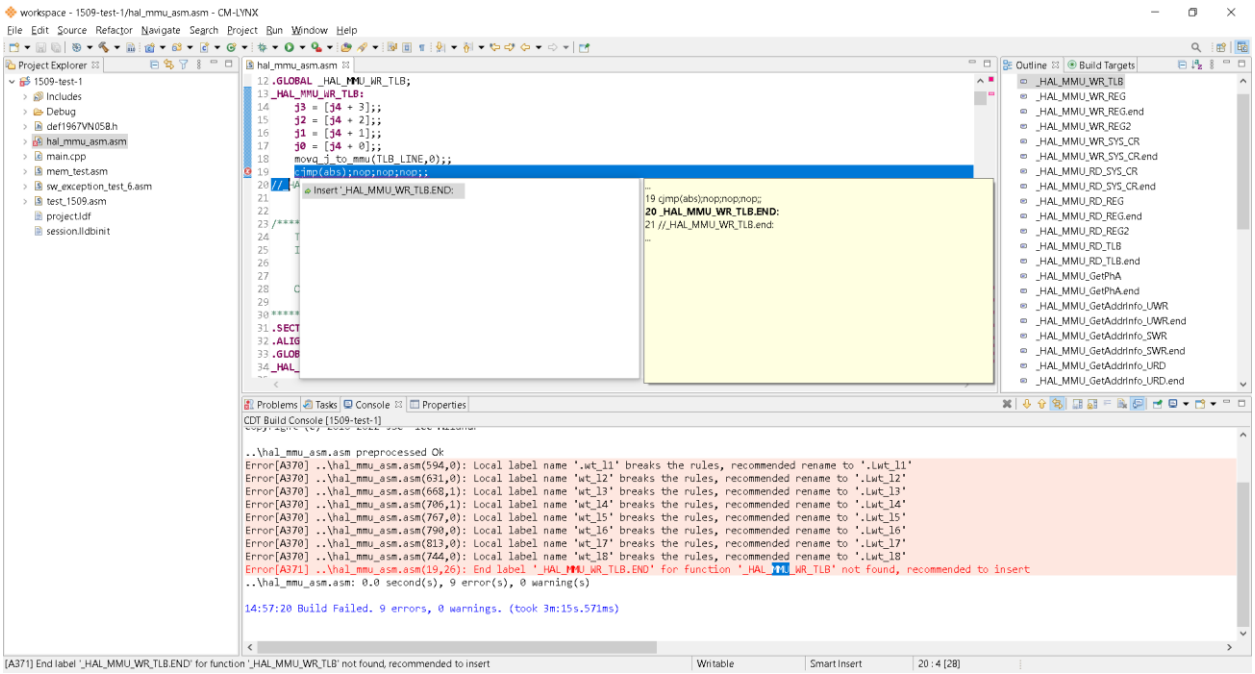


Рисунок 114. Окно автозамены для сообщения с кодом A371

6.6.14. Подсветка синтаксиса в ассемблерных файлах

Для удобства работы с ассбемлерным кодом в CM-LYNX встроена поддержка подсветки синтаксиса ассемблера для 1967BH0XX в части разбиения на линии инструкций.

Подсветка отображается в отдельной полосе слева от исходного кода и номеров строк, и представляет собой отметки рядом с однострочными и многострочными линиями инструкций. Это позволяет лучше видеть линии, которые были ошибочно не завершены и т.п.

По настройке подсветки см. также п. 6.12.1.4.2.1. Цвет линий задаётся настройкой **«Range indicator color»** на панели «**Colors and Fonts**» (см. п. 6.12.1.1.1).

```
320
321 stagen_1_start:
322         xyr11:10 = CB L[j1 += j22];     // r10
323         k17 = k16 + k9 (BR);
324         xyfr2 = r9 + r13, fr3 = r9 - r13;;
325
326         xyr9:8 = CB L[j0 += j22];;       // r8
327         k18 = k17 + k9 (BR);
328
329
330         xfr16 = r0 + r4, fr20 = r0 - r4; // xr1
331         yfr16 = r1 + r7, fr20 = r1 - r7;; // yr
332
333
334         xyr13:12 = CB L[j2 += j22];      // r12
335         k19 = k18 + k9 (BR);
336
337         xfr17 = r2 + r6, fr21 = r2 - r6;// xr17
338         yfr21 = r3 + r5, fr17 = r3 - r5;;// xr1
339
340         L[k6+k16] = xr17:16;;            // stc
341
342         xyr15:14 = CB L[j3 += j22];      // r14
343         k16 = k19 + k9 (BR);;
344
345         L[k6+k18] = yr17:16;             // stc
346         xyfr0 = r8 + r12, fr1 = r8 - r12;;
347
348         L[k6+k17] = xr21:20;             // stc
```

Рисунок 115. Фрагмент ассемблерного кода с подсветкой синтаксиса по-умолчанию (только многострочных линий)

```
321 stagen_1_start:
322         xyr11:10 = CB L[j1 += j22];     // r1
323         k17 = k16 + k9 (BR);
324         xyfr2 = r9 + r13, fr3 = r9 - r13;;
325
326         xyr9:8 = CB L[j0 += j22];;       //
327         k18 = k17 + k9 (BR);
328
329
330         xfr16 = r0 + r4, fr20 = r0 - r4; //
331         yfr16 = r1 + r7, fr20 = r1 - r7;; //
332
333
334         xyr13:12 = CB L[j2 += j22];      //
335         k19 = k18 + k9 (BR);
336
337         xfr17 = r2 + r6, fr21 = r2 - r6;// xr
338         yfr21 = r3 + r5, fr17 = r3 - r5;;//
339
340         L[k6+k16] = xr17:16;;            //
341
342         xyr15:14 = CB L[j3 += j22];      //
343         k16 = k19 + k9 (BR);;
344
345         L[k6+k18] = yr17:16;             //
346         xyfr0 = r8 + r12, fr1 = r8 - r12;;
347
348         L[k6+k17] = xr21:20;             //
```

Рисунок 116. Тот же фрагмент ассемблерного кода с подсветкой синтаксиса однострочных и многострочных линий, без жирных меток.

6.7. Использование локальной истории

Локальный журнал редактирования файла (локальная история) сохраняется при создании или изменении файла. Каждый раз, при редактировании и сохранении файла, также сохраняется копия. Таким образом, можно отменить последние изменения или даже восстановить удаленный файл. Каждая копия файла в локальной истории уникальна по дате и времени сохранения файла. Локальную историю имеют только файлы.

6.7.1. Сравнение текста файла с локальной историей

Для того чтобы сравнить ресурс рабочей среды с состоянием в локальной истории, необходимо:

1. Выбрать ресурс, который требуется сравнить с состоянием локального журнала.
2. В контекстном меню выбрать пункт «**Compare With > Local History**».
3. Выбрать ревизию в списке «**Local History**».

   Откроется редактор сравнения текста (Рисунок 117).

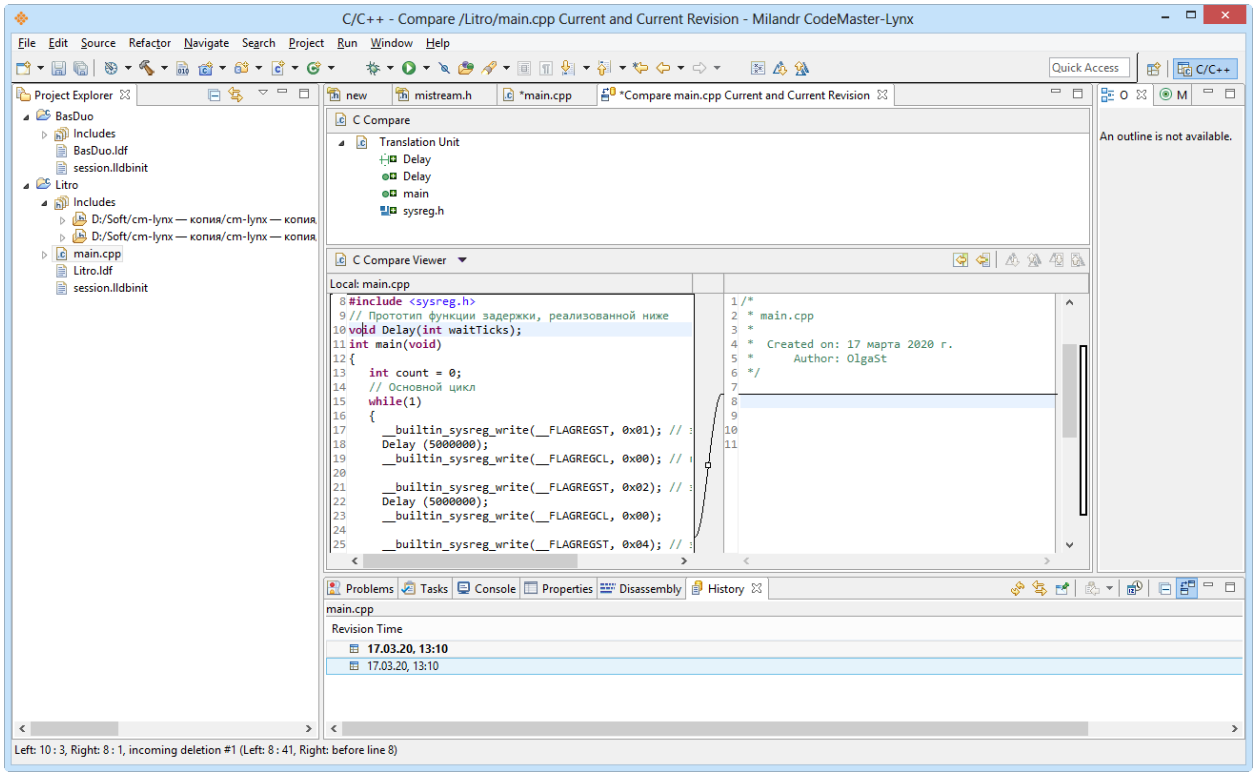


Рисунок 117. Редактор сравнения текста

Слева отображается текущее состояние ресурса; справа — состояние ресурса в соответствии с локальным журналом.

4. Просмотреть внесенные изменения, используя кнопки «**Next Change**» и «**Previous Change**».

   Редактор сравнения текстов имеет следующие элементы управления (кнопки):

	Copy All Non-Conflicting Changes From Right To Left	Скопировать все неконфликтующие изменения из правой части экрана в левую.

	**Copy Current Changes From Right To Left**	Скопировать текущие изменения из правой части в левую.
	**Next Difference**	Показать следующее отличие.
	**Previous Difference**	Показать предыдущее отличие.
	**Next Change**	Показать следующее изменение.
	**Previous Change**	Показать предыдущее изменение.

6.7.2. Замена ресурса локальной историей

Для того чтобы заменить ресурс локальной историей, необходимо:

1. Выбрать ресурс, который требуется заменить его локальной историей.
2. В контекстном меню выбрать пункт «**Replace with -> Local History**».

   Откроется окно сравнения файла с локальной историей.
3. Выбрать ревизию из предложенного списка двойным щелчком.

   Отобразится редактор сравнения (см. Рисунок 118):

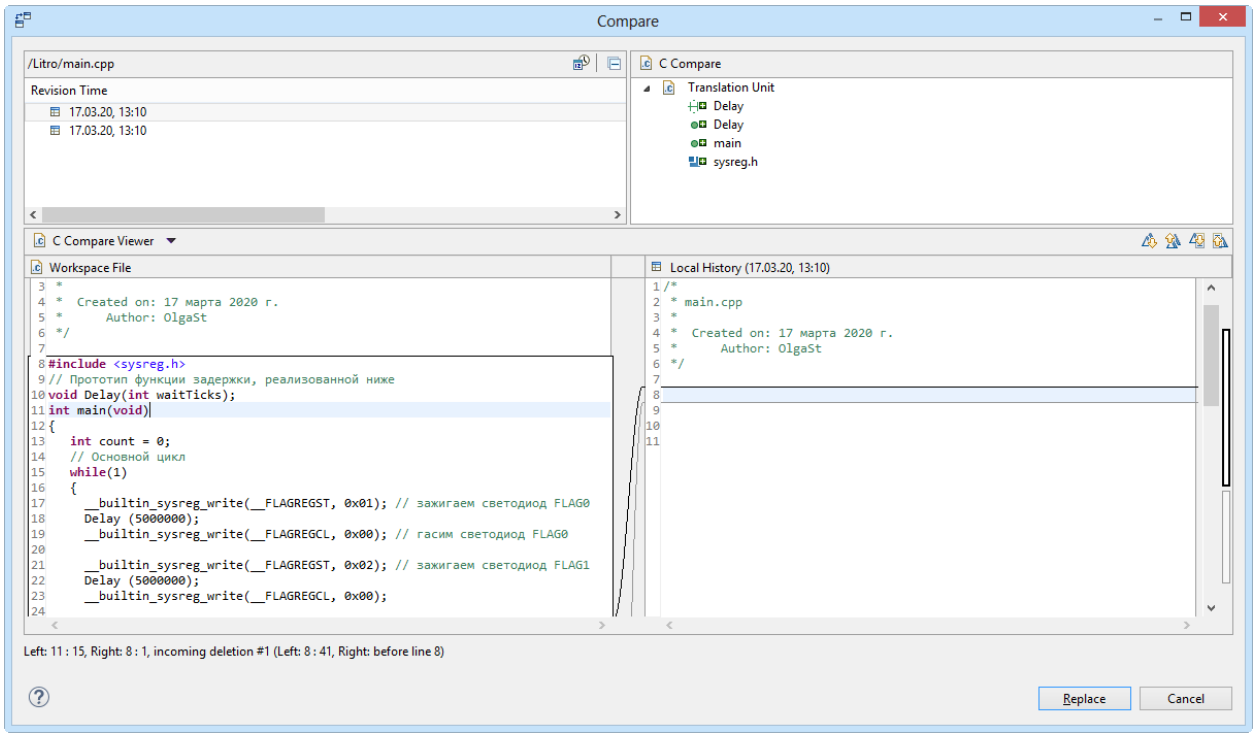


Рисунок 118. Окно «Compare» с выбранной ревизией

Используя кнопки  «**Next Change**» и  «**Previous Change**», можно просмотреть отличия локальной истории от текущего состояния ресурса.

4. Нажать кнопку «**Replace**».

6.7.3. Восстановление ресурса из локальной истории

Для того чтобы восстановить ресурс из локальной истории, необходимо:

1. Выбрать проект или папку, удаленные файлы из которых требуется восстановить из локальной истории.

2. В контекстном меню выбрать пункт «**Restore from Local History…**».

   Откроется окно (Рисунок 119):

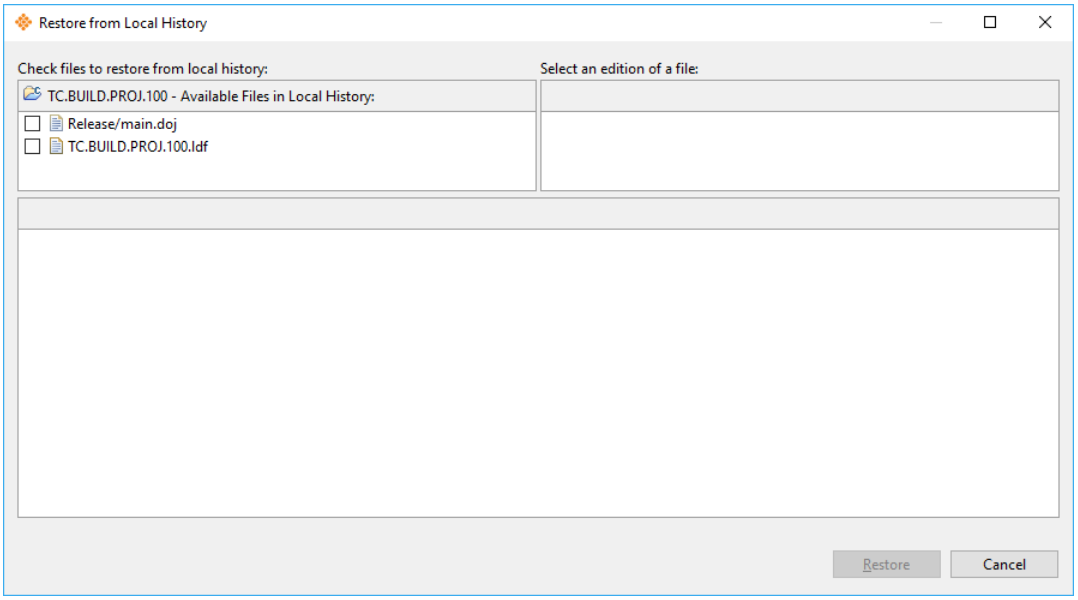


Рисунок 119. Окно «Restore from Local History»

3. Отметить файлы для восстановления из предложенного списка, установив флаги напротив нужных ресурсов.

4. Нажать кнопку «**Restore**».

6.7.4. Настройки локальной истории

По умолчанию размер локальной истории ограничен следующими параметрами:

- Срок хранения файлов - 7 дней;
- Максимальное количество копий файла - 50;
- Максимальный размер файла - 1Мб.

Изменить ограничения размера локальной истории можно в окне «**Preferences->General->Workspace->Local History**» (подробнее см. п. 6.12.1.9.3. Настройки локальной истории).

6.8. Использование меток

Метки – объекты, которые могут быть ассоциированы с ресурсами. Используются следующие виды меток:

- Задания (Tasks);
- Проблемы (Problems);
- Закладки (Bookmarks).

Для работы с метками используются вкладки «**Tasks**», «**Problems**» и «**Bookmarks**» соответственно. Вкладки «**Tasks**» (см. п. 5.4.2.4. Вкладка «Tasks») и «**Problems**» (см. п. 5.4.2.3. Вкладка «Problems») присутствуют в режиме разработки по умолчанию. Вкладку «**Bookmarks**» можно открыть, используя команду «**Show view->Other...->General->Bookmarks**».

6.8.1. Закладки

Закладки (Bookmarks) создаются пользователем и указывают либо на определенную строку ресурса, либо на сам ресурс. Закладки позволяют быстро открыть файл на редактирование. Для этого необходимо дважды щелкнув по соответствующей ему закладке на вкладке «**Bookmarks**».

Для работы с закладками используется вкладка «**Bookmarks**».

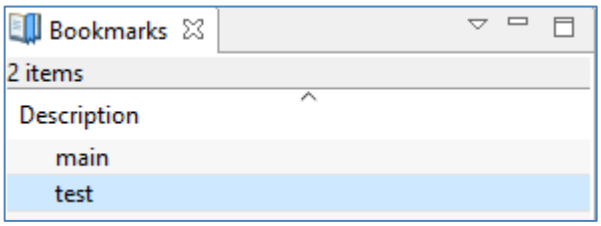


Рисунок 120. Вкладка «Bookmarks»

Вкладку «**Bookmarks**» можно открыть, используя команду «**Show view->Other...->General->Bookmarks**».

6.8.1.1. Добавление закладки на файл

Для того чтобы создать закладку, указывающую на весь файл целиком, необходимо:

1. Выбрать файл, который необходимо добавить в список закладок, на вкладке навигации (например, на вкладке «**Project Explorer**»).
2. В меню «**Edit**» выбрать пункт «**Add Bookmark...**».
3. Указать наименование закладки в поле «**Description**» и нажать на кнопку «**OK**».

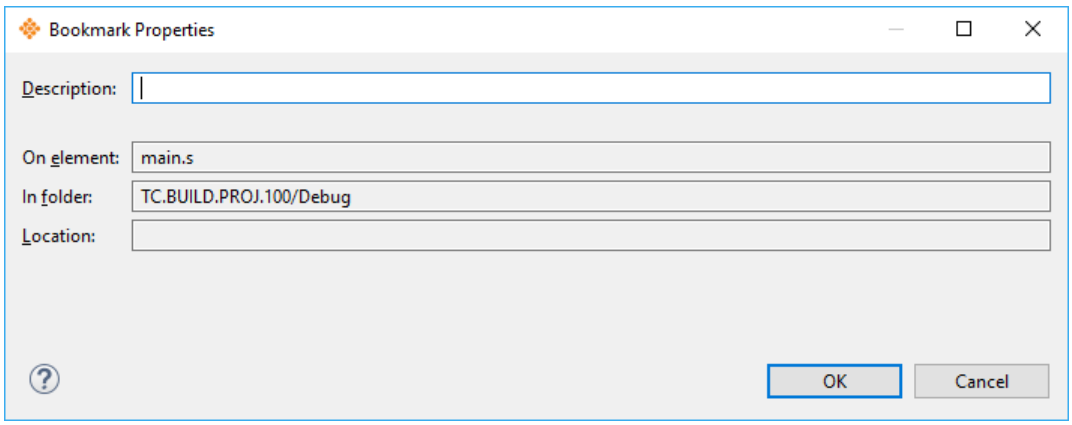


Рисунок 121. Окно «Bookmark Properties»

6.8.1.2. Добавление закладки внутри файла

Для того чтобы создать закладку, указывающую на фрагмент внутри файла, необходимо:

1. Открыть файл на редактирование.

2. Щелкнуть правой клавишей по серой границе в левой части окна редактора рядом со строкой кода или текста, на которые необходимо установить закладку.

3. В контекстном меню выбрать пункт «**Add Bookmark**».

4. Указать наименование закладки в поле «**Description**» и нажать на кнопку «**OK**».

   Закладка будет добавлена на серой границе в левой части окна редактора и на вкладке «**Bookmarks**».

6.8.1.3. Удаление закладки

Закладку можно удалить на вкладке «**Bookmarks**» или из области редактора. После удаления закладки любым из способов, закладка, соответствующая удаленной, будет также удалена.

Для того чтобы удалить закладку на вкладке «**Bookmarks**», необходимо:

1. Щелкнуть правой клавишей по закладке, которую необходимо удалить, на вкладке «**Bookmarks**».

2. В контекстном меню выбрать пункт «**Delete**».

3. Подтвердить операцию, нажав на кнопку «**Yes**».

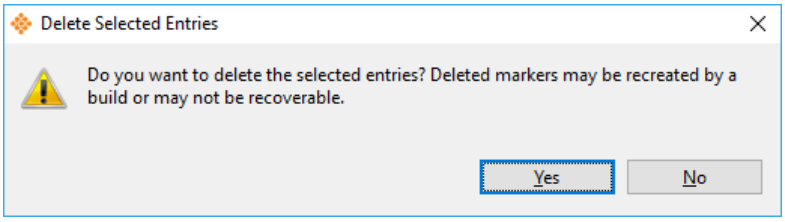


Рисунок 122. Окно «Delete Selected Entries»

Для того чтобы удалить закладку из области редактора, необходимо:

1. Щелкнуть правой клавишей мыши по графическому значку, соответствующему нужной закладке на серой границе в левой части окна редактора.

2. В контекстном меню выбрать пункт «**Remove Bookmark**».

6.8.2. Задания

Задания (Tasks) могут быть двух типов:

- Автоматически генерируемая информация (problems, warnings, errors), ассоциированная с ресурсом;
- Настроенные пользователем задания, которые могут быть ассоциированы или не ассоциированы с ресурсом.

Для работы с заданиями используется вкладка «**Tasks**» (см. п. 5.4.2.4. Вкладка «Tasks»), которая присутствует в режиме разработки по умолчанию.

6.8.2.1. Добавление задания

Для того чтобы добавить задание, необходимо:

1. В контекстном меню на вкладке «**Tasks**» выбрать пункт «**Add Task...**».
2. Задать наименование задания в поле «**Description**».
3. При необходимости, изменить приоритет задания.
4. Нажать на кнопку «**OK**».

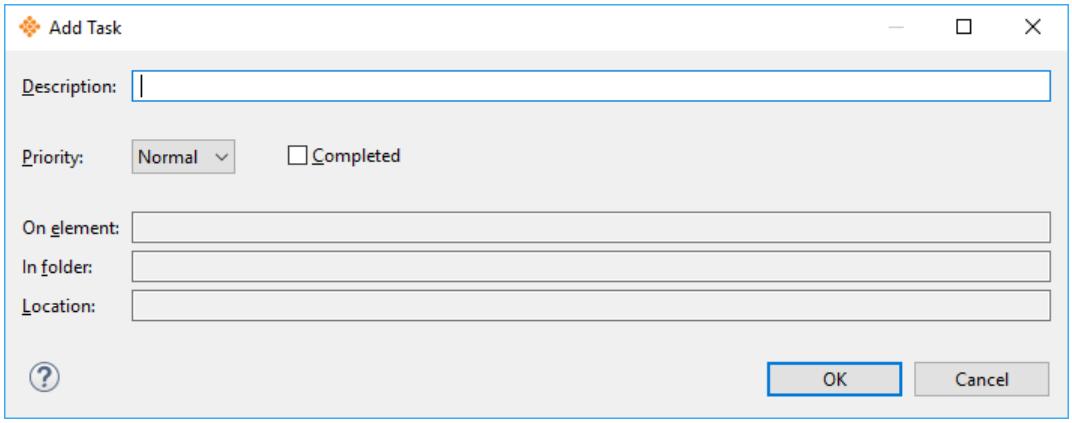


Рисунок 123. Окно «Add Task»

Можно создать задание, автоматически ассоциированное с ресурсом. Например, это может быть удобно для настройки напоминания вернуться к редактированию определенной строчки кода позднее. Для этого необходимо:

1. Открыть нужный ресурс на редактирование.
2. Щелкнуть правой клавишей по серой границе в левой части окна редактора рядом со строкой кода или текста, на которые необходимо создать задание.
3. В контекстном меню выбрать пункт «**Add Task**».
4. Заполнить поля формы.
5. Нажать на кнопку «**OK**».

6.8.2.2. Удаление задания

Задание можно удалить на вкладке «**Tasks**» или из области редактора. После удаления задания любым из способов, задание, соответствующее удаленному, будет также удалено.

Для того чтобы удалить задание на вкладке «**Tasks**», необходимо:

1. Щелкнуть правой клавишей по заданию, которую необходимо удалить.
2. В контекстном меню выбрать пункт «**Delete**».
3. Подтвердить операцию, нажав на кнопку «**Yes**».

Для того чтобы удалить задание из области редактора, необходимо:

1. Щелкнуть правой кнопкой мыши по графическому значку, соответствующему нужному заданию на серой границе в левой части окна редактора.
2. В контекстном меню выбрать пункт «**Remove Task**».

6.9. Настройка свойств проекта

Окно «**Properties**» предназначено для просмотра и настройки свойств проектов C/C++, папок и отдельных файлов, вне зависимости от вида проекта.

Открыть окно «**Properties**» можно одним из способов:

- щелкнув по выбранному ресурсу правой кнопкой мыши и выбрав пункт «**Properties**» в контекстном меню;
- щелкнув по выбранному ресурсу и выбрав пункт «**Properties**» в меню «**Project**».

6.9.1. Настройки ресурсов проекта

На панели «**Resource**» окна «**Properties**» отображается основная информация о выбранном ресурсе.

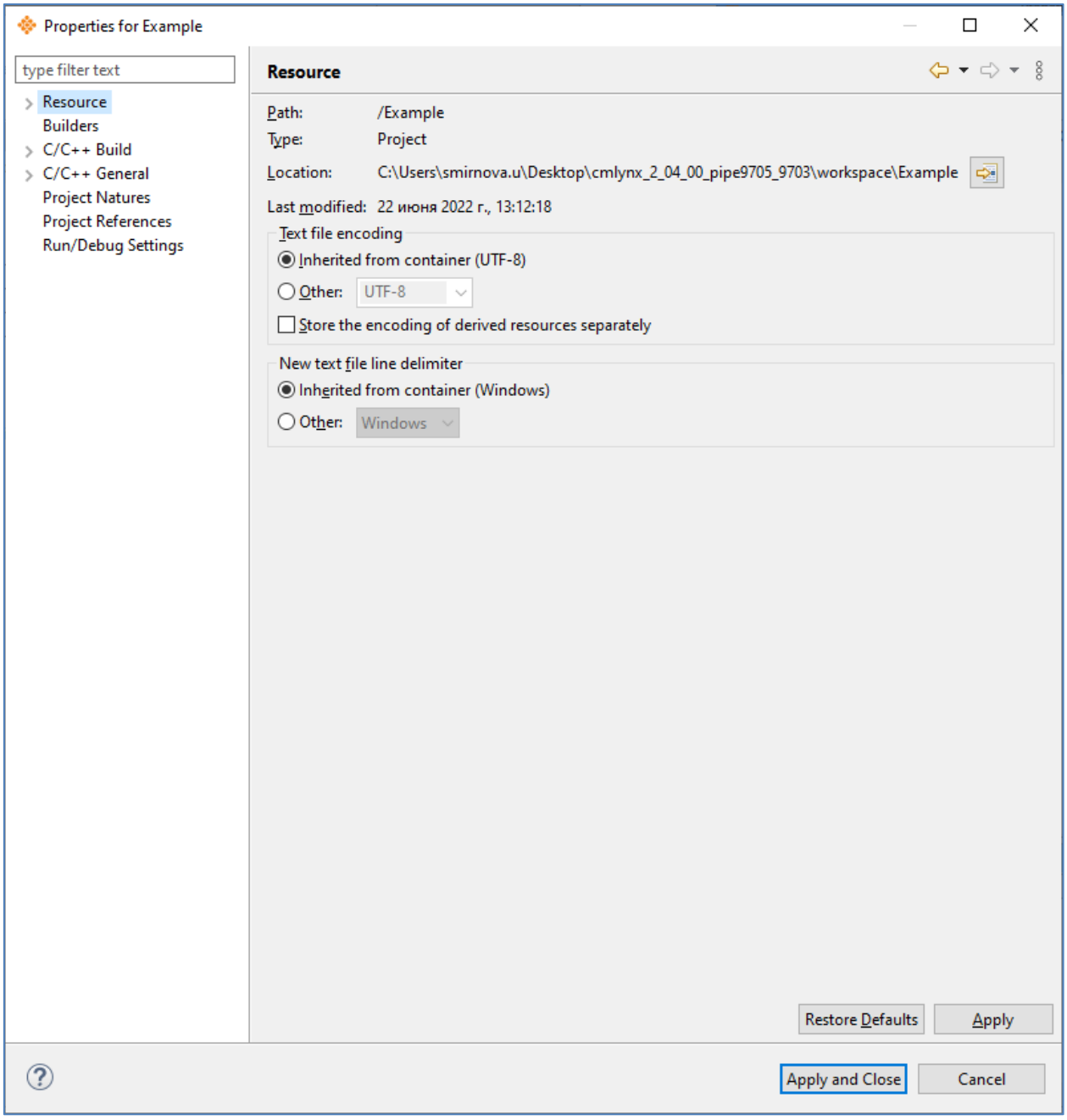


Рисунок 124. Панель «Resource» для объектов типа Project

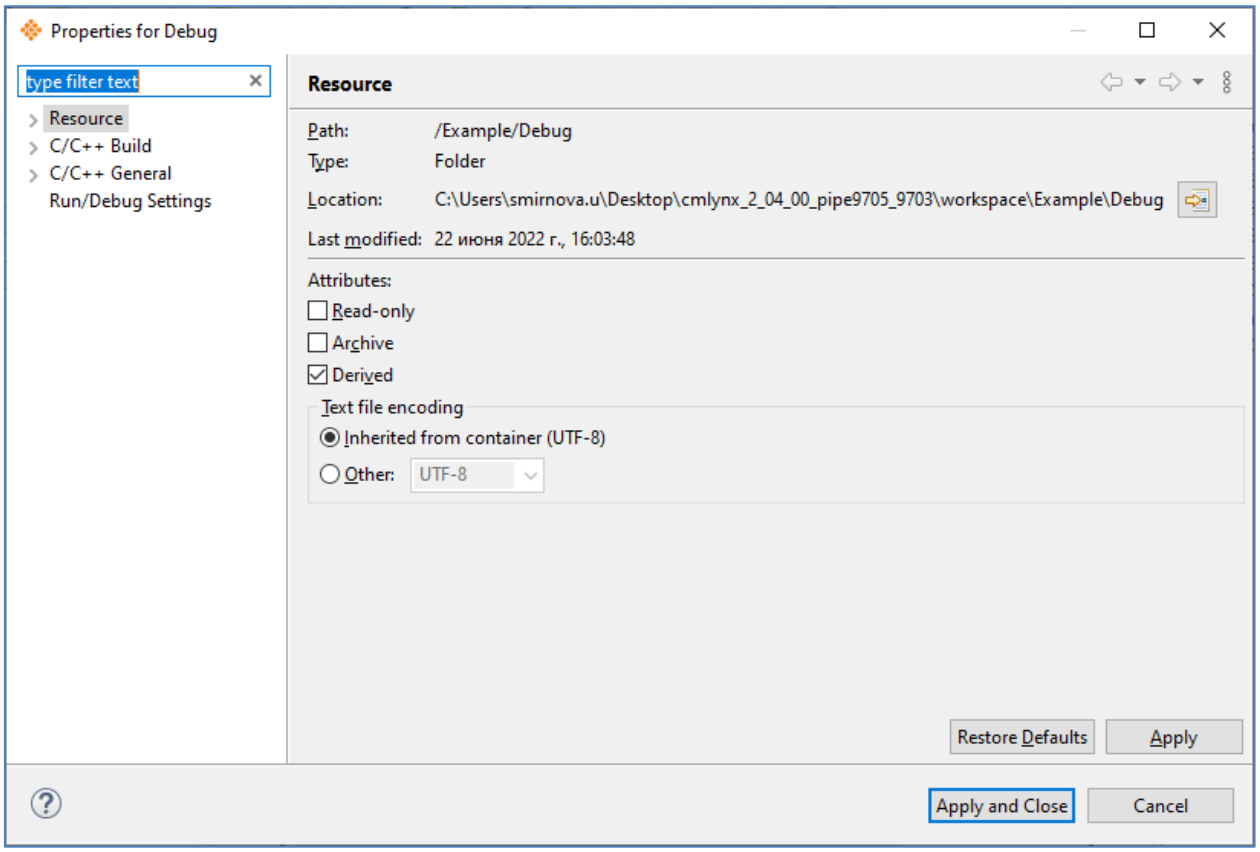


Рисунок 125. Панель «Resource» для объектов типа Folder

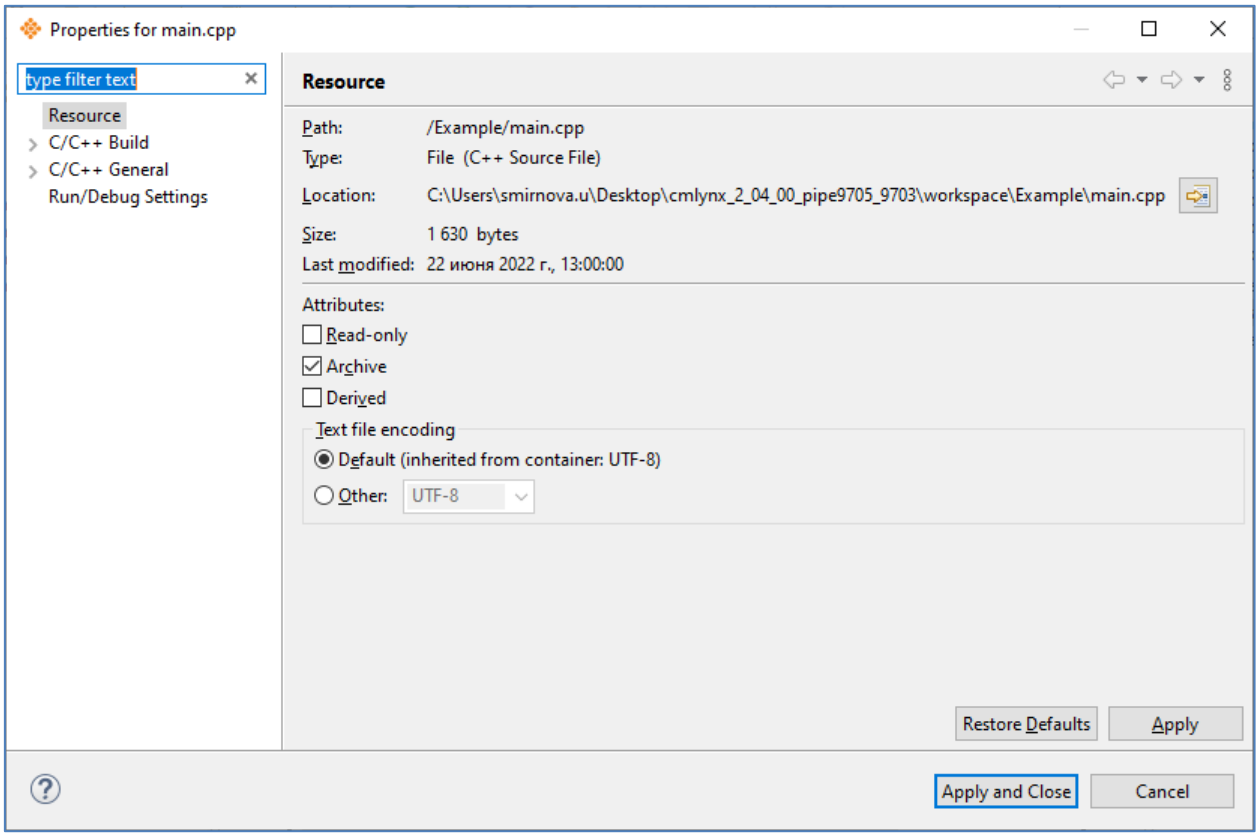


Рисунок 126. Панель «Resource» для объектов типа File

Таблица 49. Элементы управления панелей «Resource»

Название элемента	Описание функционирования	По умолчанию
Path	Расположение выбранного типа ресурса в рабочем пространстве. Например, как и папки, проекты располагаются в директориях в файловой системе.	–
Type	Тип выбранного ресурса: ▪ **Folder** (папка); ▪ **Project** (проект); ▪ **File** (файл).	–
Location	Расположение выбранного ресурса в файловой системе.	–
Size	Размер в байтах. Только для файлов.	
Last modified	Дата последнего изменения для выбранного ресурса.	–
Attributes	Задать атрибуты ресурса (только для папки или файла): ▪ **Read-only** – только для чтения; ▪ **Archive** – архивный; ▪ **Derived** – производный.	
Text file encoding	Задать кодировку текста файла: ▪ **Inherited from container** (только для папок и проектов) – ресурс наследует кодировку текста от контейнера; ▪ **Default** (только для файлов) **–** кодировка по умолчанию; ▪ **Other** – выбор альтернативной текстовой кодировки ресурса, отличной от той, которая установлена для контейнера. При работе с текстовыми файлами из других источников (с кодировкой, отличной от требуемой) следует выбрать из списка необходимую кодировку. Таким образом, можно обмениваться файлами с другими.	Наследует кодировку от контейнера
Store the encoding of derived resources separately	Хранить кодировку производных ресурсов отдельно (только для проектов).	Выкл
New text file line delimiter	Выбрать разделитель для нового текстового файла (только для проектов): ▪ **Inherited from container** – ресурс наследует разделитель текста от контейнера; ▪ **Other** – выбор альтернативного разделителя.	Наследует от контейнера

6.9.1.1. Работа со связанными ресурсами

Связанные ресурсы – это файлы и папки, хранящиеся в файловой системе вне проекта.

На панели «**Linked Resources**» (на вкладке «**Path Variables**») пользователь может добавить, изменить или удалить переменные пути для связанных ресурсов.

Для добавления переменной пути следует:

1. Установить курсор на проект, в главном меню приложения выбрать **File → Properties.**

2. В открывшемся окне перейти к **Resource → Linked Resources→Path Variables.**

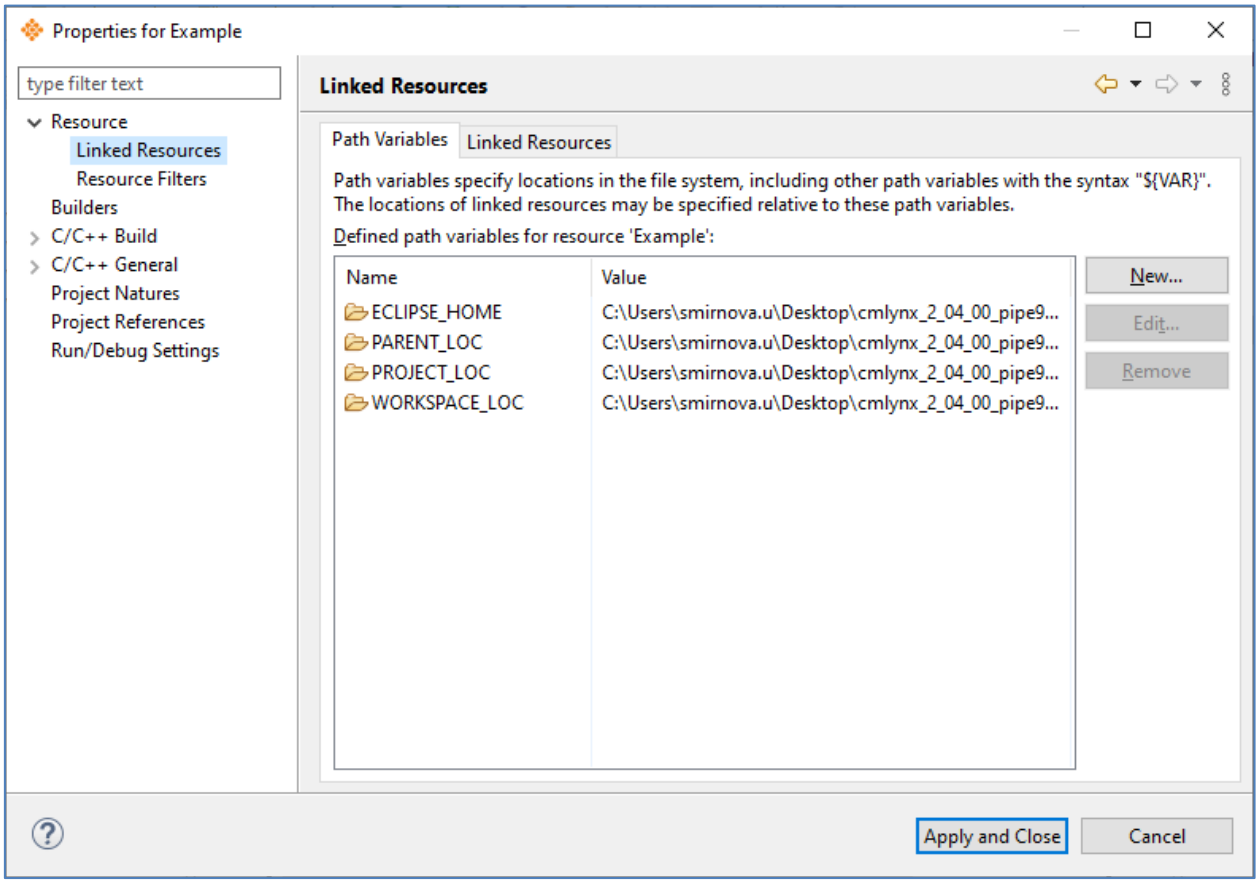


Рисунок 127. Панель «Linked Resources», вкладка «Path Variables»

3. Выбрать в списке путей нужный и нажать кнопку «**New**».

4. В открывшемся диалоговом окне задать параметры переменной.

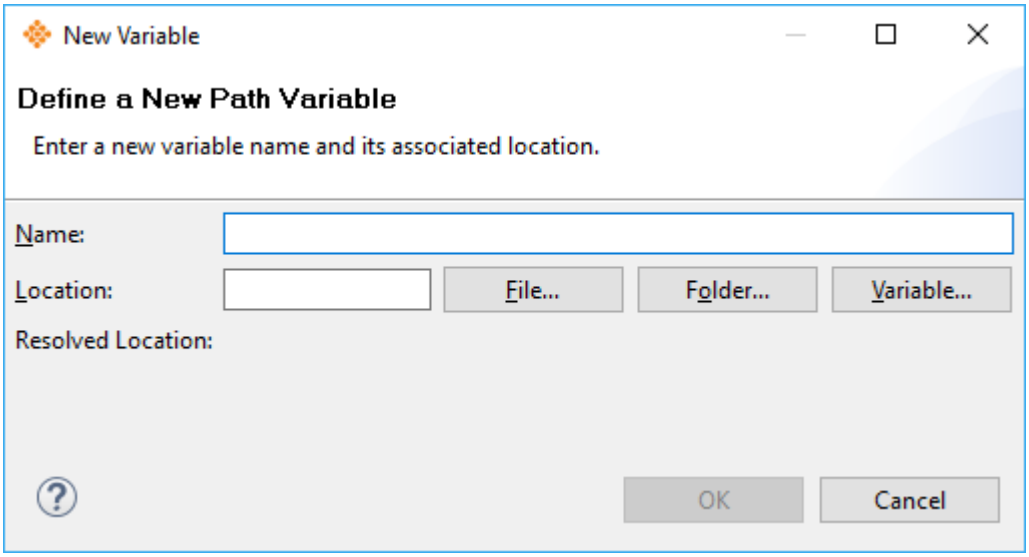


Рисунок 128. Добавление переменной пути

5. Нажать кнопку «**OK**».

Для редактирования переменной пути следует нажать кнопку «**Edit**». Для удаления переменной пути следует нажать кнопку «**Remove**».

Пользователь может просмотреть пути к папкам связанных ресурсов проекта на вкладке «**Linked Resources**».

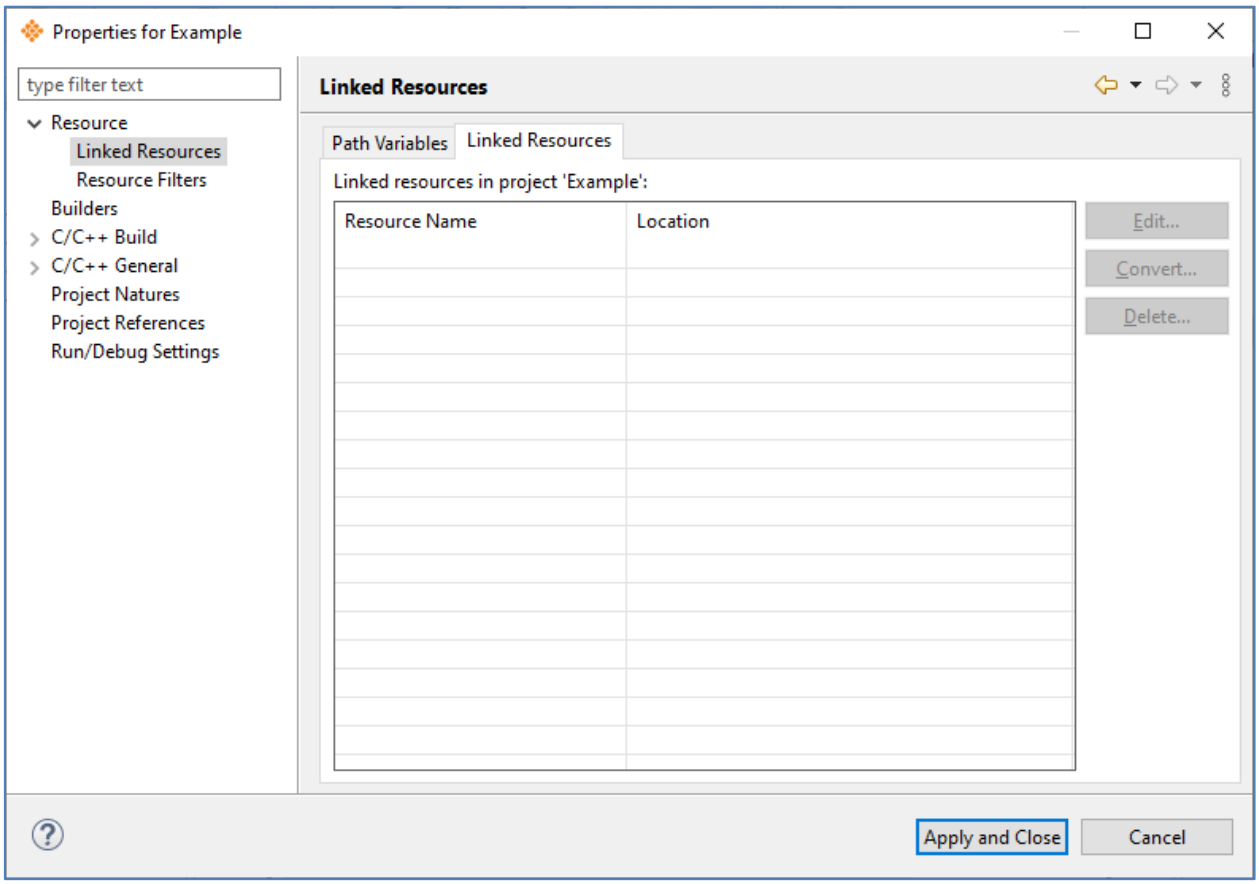


Рисунок 129. Панель «Linked Resources», вкладка «Linked Resources»

6.9.1.2. Настройка фильтров ресурсов

Фильтры ресурсов позволяют настроить файлы и папки, которые при обновлении будут автоматически включены в проект в виде иерархии.

Добавление *фильтров ресурсов* к проекту или папке систематически не дает некоторым записям файловой системы появляться в дереве ресурсов.

Фильтры ресурсов могут быть двух типов:

- «**Включить**» (**Include Only)** - позволяет при обновлении появиться в проекте только тем файлам и/или папкам, которые соответствуют условиям фильтра;
- «**Исключить**» (**Exclude All)** - не дает при обновлении появиться в проекте тем файлам и/или папкам, которые соответствуют условиям фильтра. Если существует несколько таких фильтров, исключенными элементами будут те, которые совпадут с любым существующим фильтром исключения**.**

При одновременном использовании в папке или проекте *фильтров ресурсов* обоих видов «**Включить**» и «**Исключить**» в рабочее пространство будут включены только те файлы и/или папки, которые совпадают с условиями фильтра «**Включить**» и не совпадают с условиями фильтра «**Исключить**».

Фильтры ресурсов применяются только к файлам и/или папкам, которые косвенно включаются в рабочее пространство благодаря операции обновления. На *связанные* файлы и папки *фильтры ресурсов* не влияют, так как они должны быть созданы непосредственно пользователем.

Фильтр ресурсов может применяться рекурсивно ко всем потомкам проекта или папки, где были созданы. В таком случае *фильтр ресурсов* будет применяться к подпапке, будто он сам был создан в этой подпапке, следуя вышеупомянутым правилам.

Панель «**Resource Filters**» предназначена для управления, создания и редактирования фильтров ресурсов.

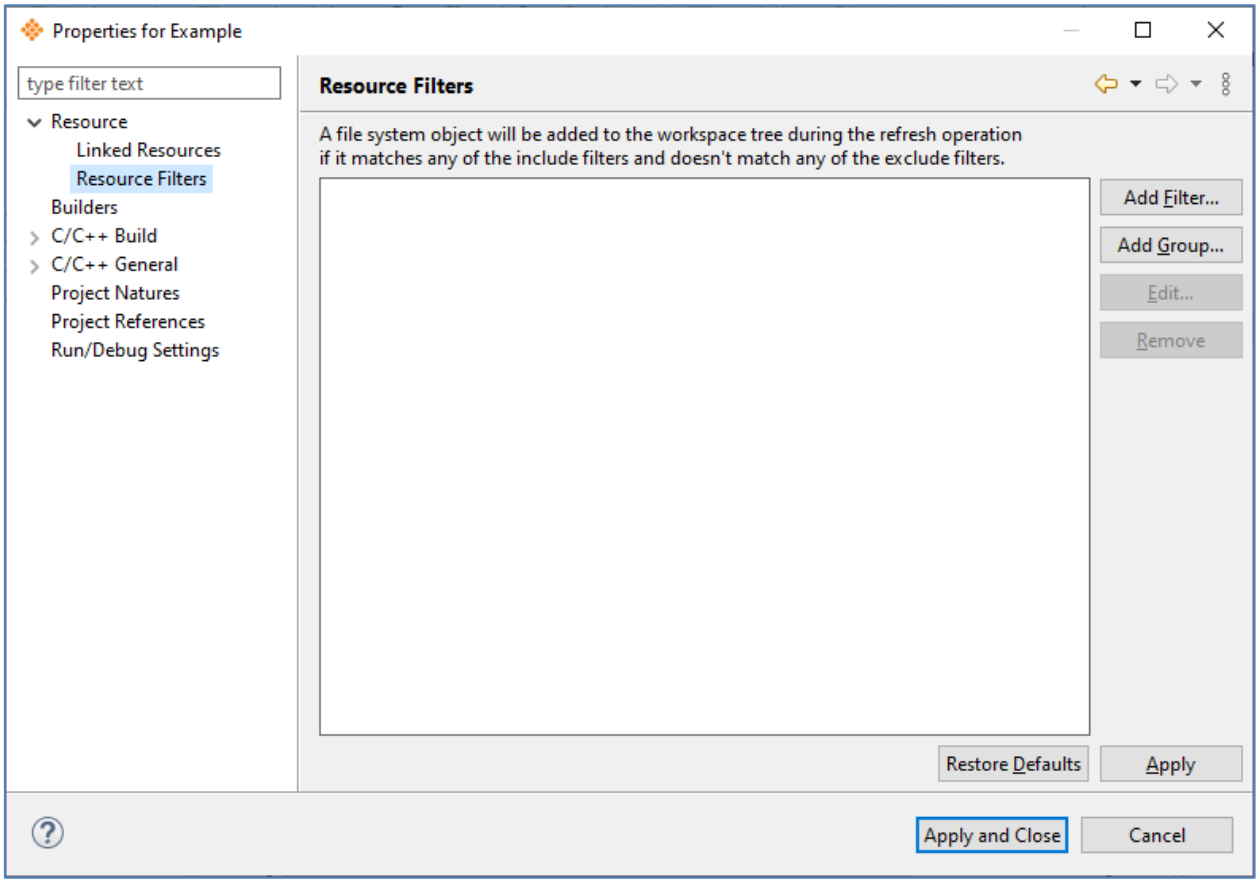


Рисунок 130. Панель «Resource Filters»

Для добавления фильтра следует:

1. Нажать кнопку «**Add Filter**» в правой части окна.
2. В открывшемся диалоговом окне задать параметры фильтра.

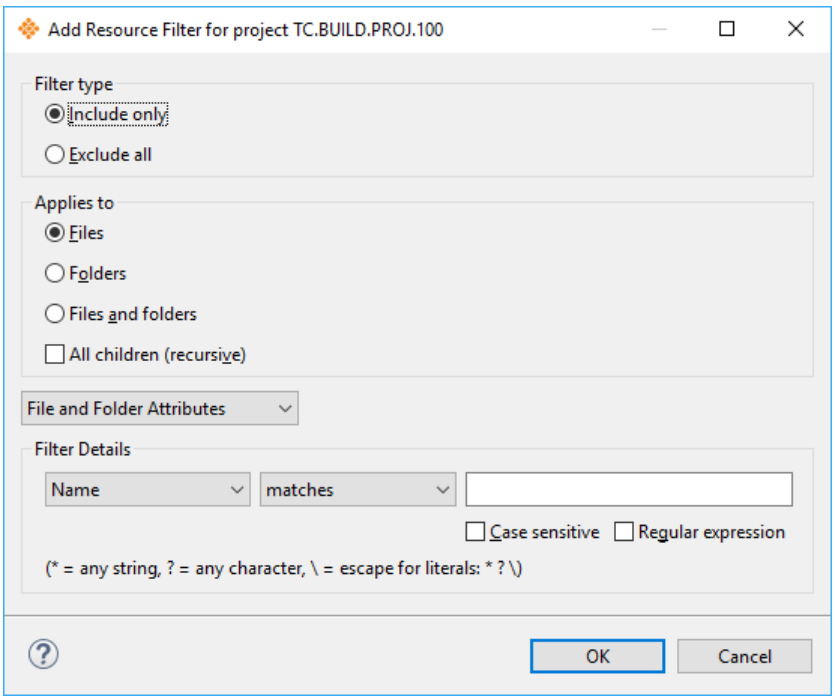


Рисунок 131. Добавление фильтра

3. Нажать кнопку «**OK**» в окне настройки фильтра.

4. Нажать кнопку «**Apply**» в окне свойств.

Для добавления группы фильтров следует нажать кнопку «**Add Group**».

Фильтры ресурсов могут быть также заданы при создании новой папки. Для этого в окне «**New Folder**» следует нажать кнопку «**Resource Filters...**» в разделе «**Advanced**».

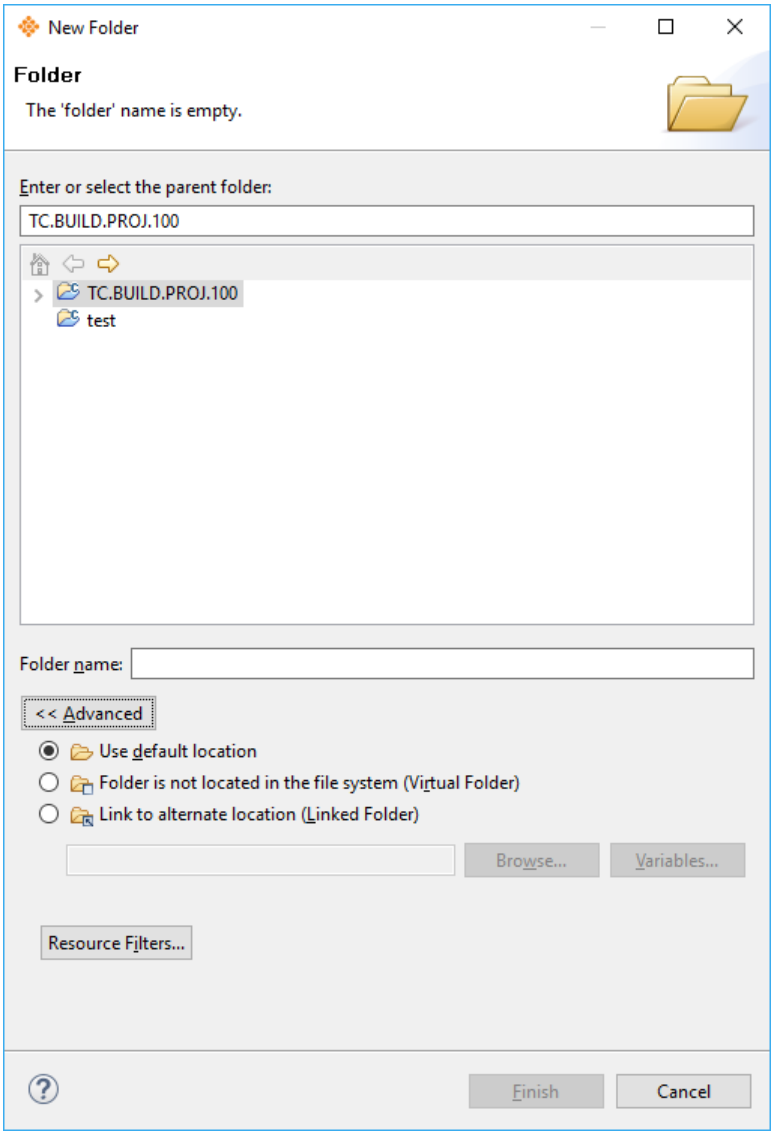


Рисунок 132. Добавление фильтров ресурсов при создании папки

6.9.2. Настройка параметров сборки проекта

Панель «**C/C++ Build**» предназначена для настройки параметров сборки проекта. Для перехода на панель следует выбрать **C/C++ Build**.

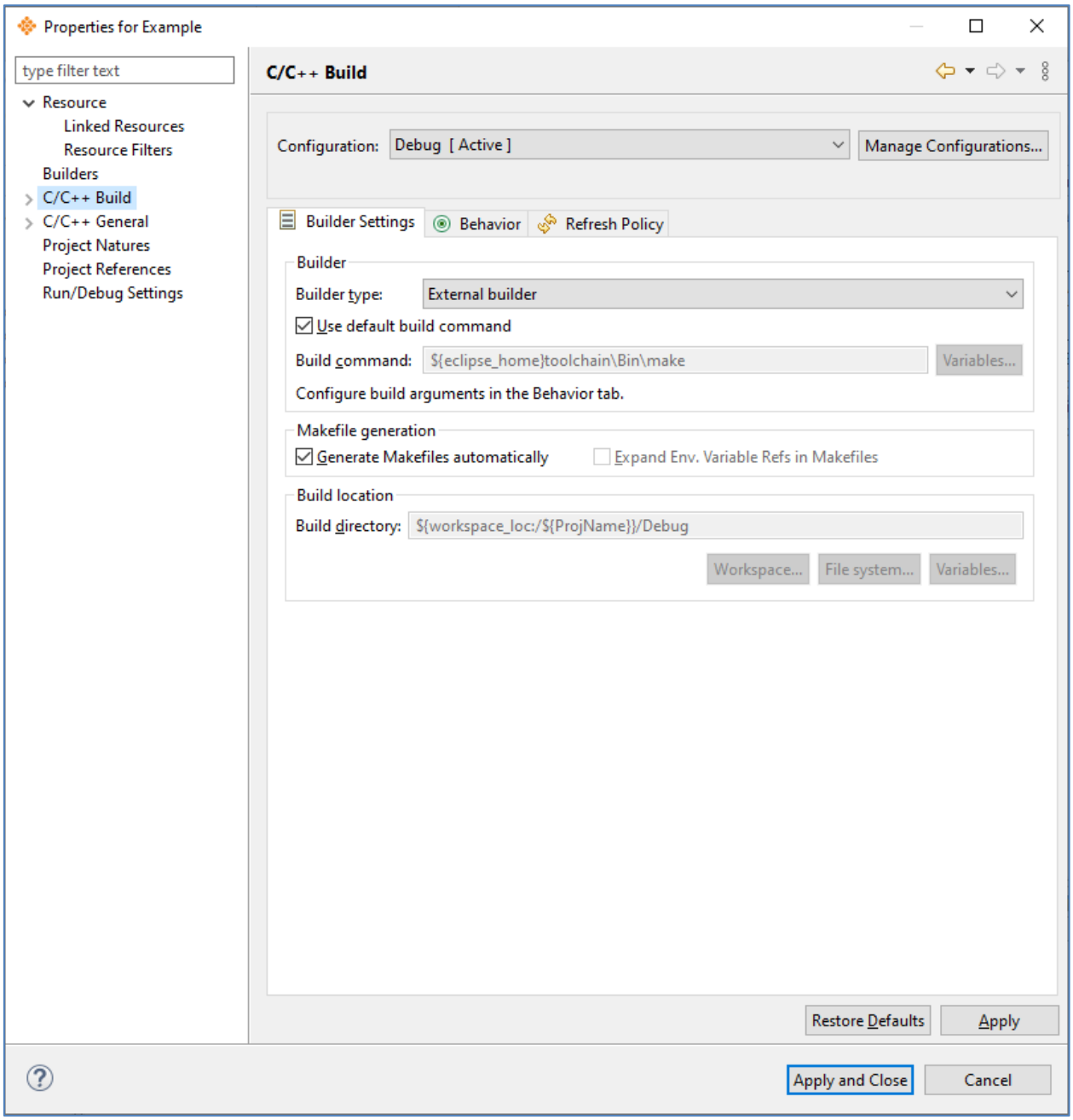


Рисунок 133. Панель «C/C++ Build»

В верхней части панели «**C/C++ Build**» расположены элементы управления конфигурацией ресурса:

- **Configuration** – выбор настраиваемой конфигурации ресурса;
- **Manage Configurations...** – открытие диалогового окна «**Manage Configurations**» для управления конфигурациями ресурса.

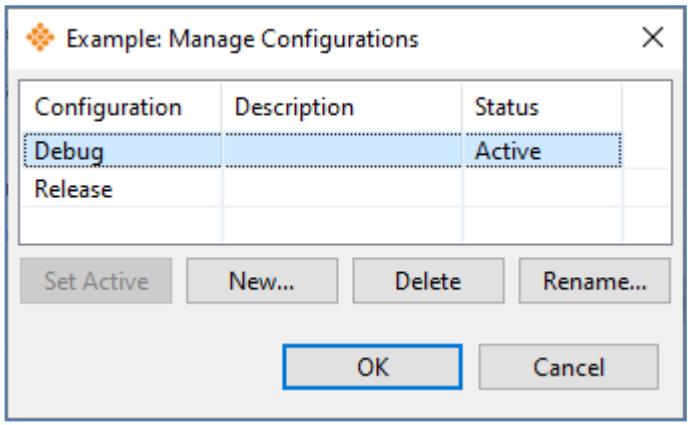


Рисунок 134. Окно «Manage Configurations»

Таблица 50. Элементы управления окна «Manage Configurations»

Название элемента	Описание функционирования
Set Active	Назначить активной выбранную конфигурацию.
New...	Открыть окно для создания новой конфигурации.
Delete	Удалить выбранную конфигурацию.
Rename...	Открыть окно для переименования выбранной конфигурации.

В центральной части панели «**C/C++ Build**» расположены вкладки:

- «**Builder Settings**»;
- «**Behaviour**»;
- «**Refresh Policy**».

Набор вкладок зависит от типа выбранного ресурса.

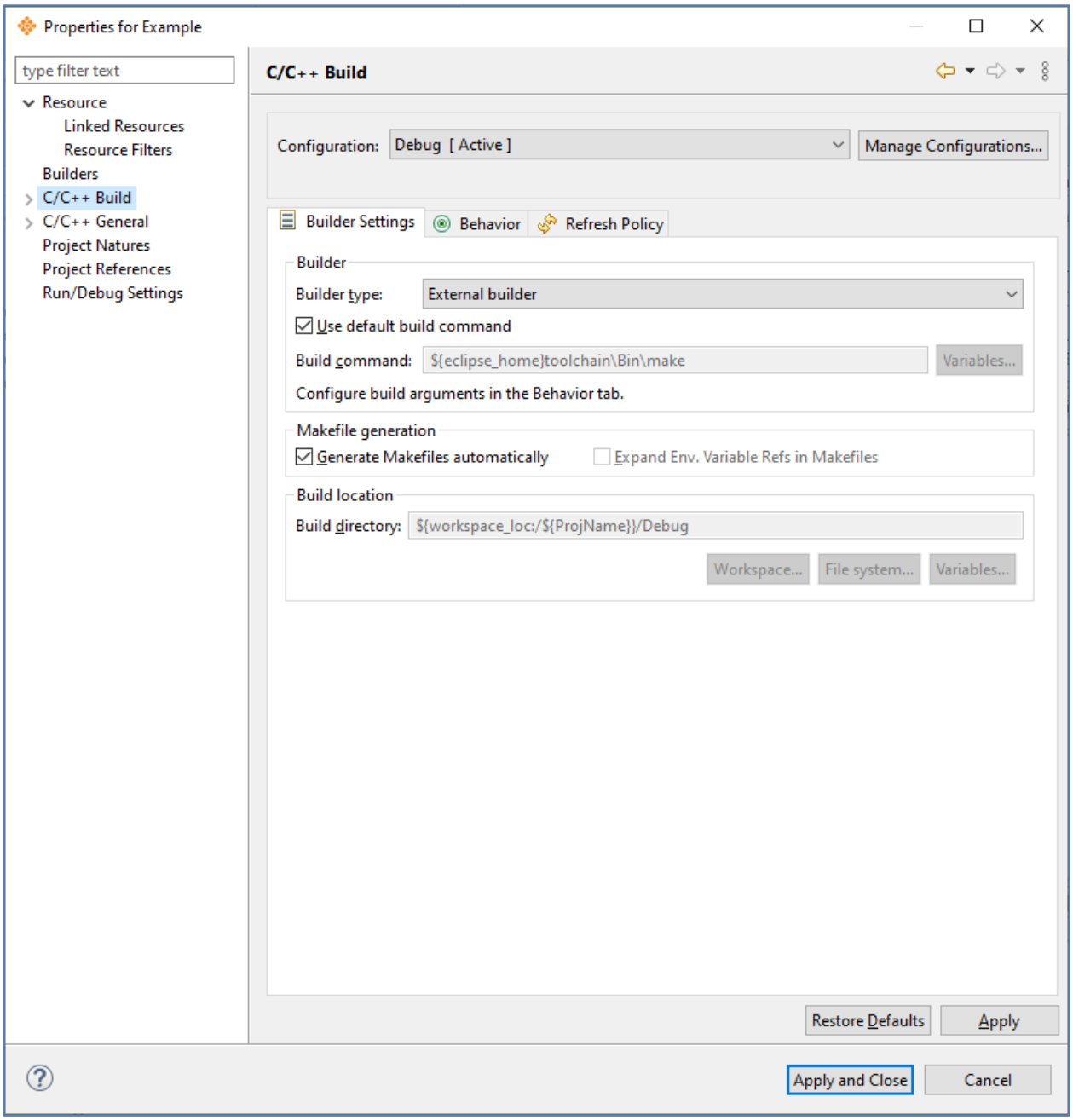


Рисунок 135. Вкладка «Builder Settings»

Таблица 51. Элементы управления вкладки «Builder Settings»

Название элемента	Описание функционирования	По умолчанию
Builder type	Выбрать подсистему сборки проекта из списка.	–
Use default build command	Использовать команды сборки по умолчанию.	Вкл
Build command	Задать команды сборки. Для выбора переменных следует нажать кнопку «**Variables**».	–
Generate Makefiles automatically	Автоматически генерировать Make-файлы.	Вкл

Название элемента	Описание функционирования	По умолчанию
Expand Env. Variable Refs in Makefiles	Расширить ссылки переменных в файлах Make.	–
Build directory	Задать папку сборки.	–

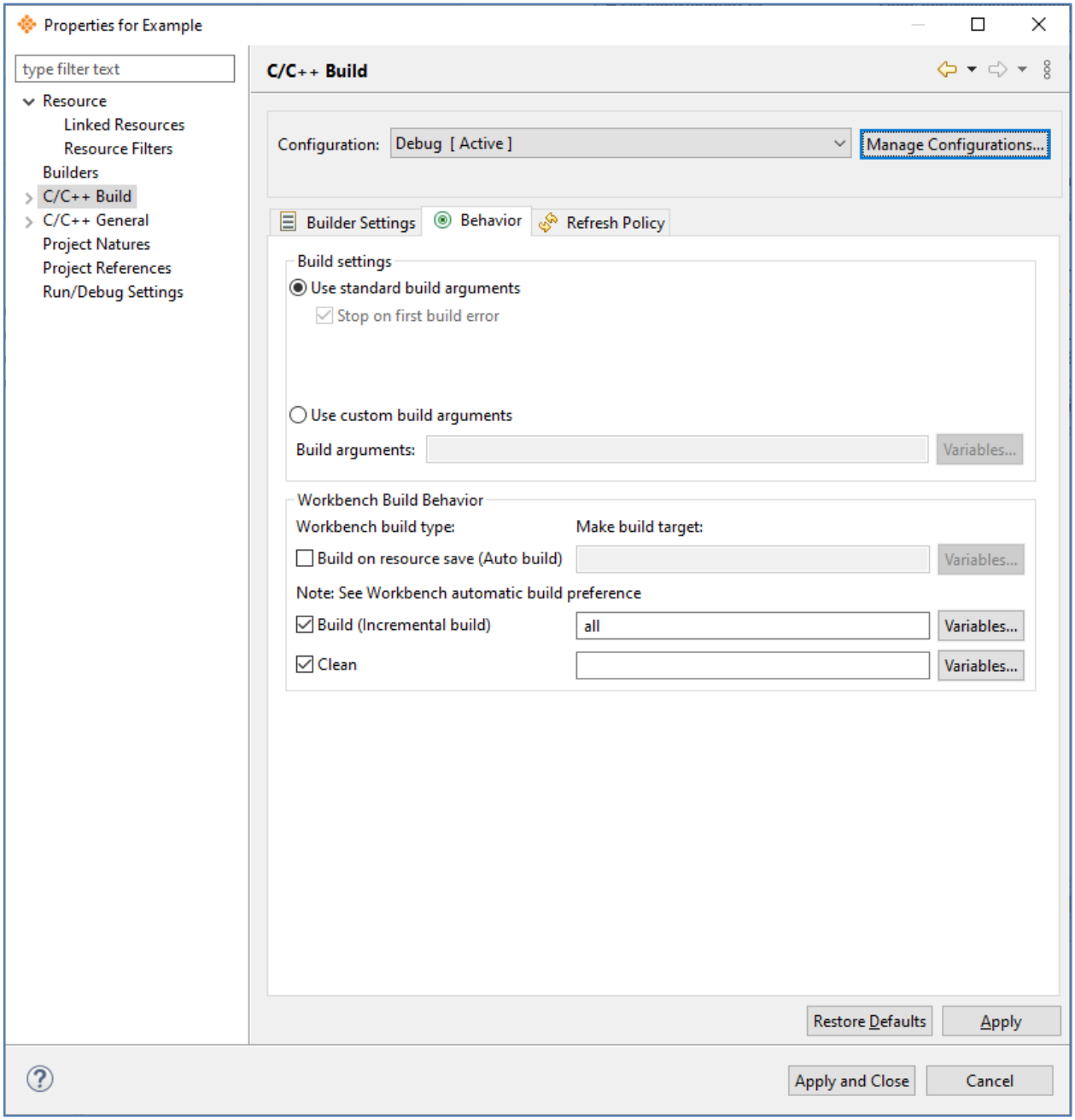


Рисунок 136. Вкладка «Behaviour»

Таблица 52. Элементы управления вкладки «Behaviour»

Название элемента	Описание функционирования	По умолчанию
Use standard buld arguments	Ипользовать стандартные аргументы сборки.	Вкл

Название элемента	Описание функционирования	По умолчанию
Stop on first build error	Остановить сборку проекта при появлении первой ошибки. Включен всегда, отключить нельзя.	Вкл
Use custom build arguments	Использовать произвольные аргументы сборки. Аргументы сборки задаются с помощью кнопки «**Variables**».	Выкл
Build on resource save (Auto build)	Задать параметры вызова стандартной программы сборки при сохранении файла. Для выбора переменных следует нажать кнопку «**Variables**». Не рекомендуется использовать автосборку для проектов C/C++.	Выкл
Build (Incremental Build)	Задать параметры вызова стандартной программы сборки при выполнении инкрементальной сборки. Для выбора переменных следует нажать кнопку «**Variables**».	Вкл
Clean	Задать параметры вызова стандартной программы сборки при выполнении очистки. Для выбора переменных следует нажать кнопку «**Variables**».	Вкл

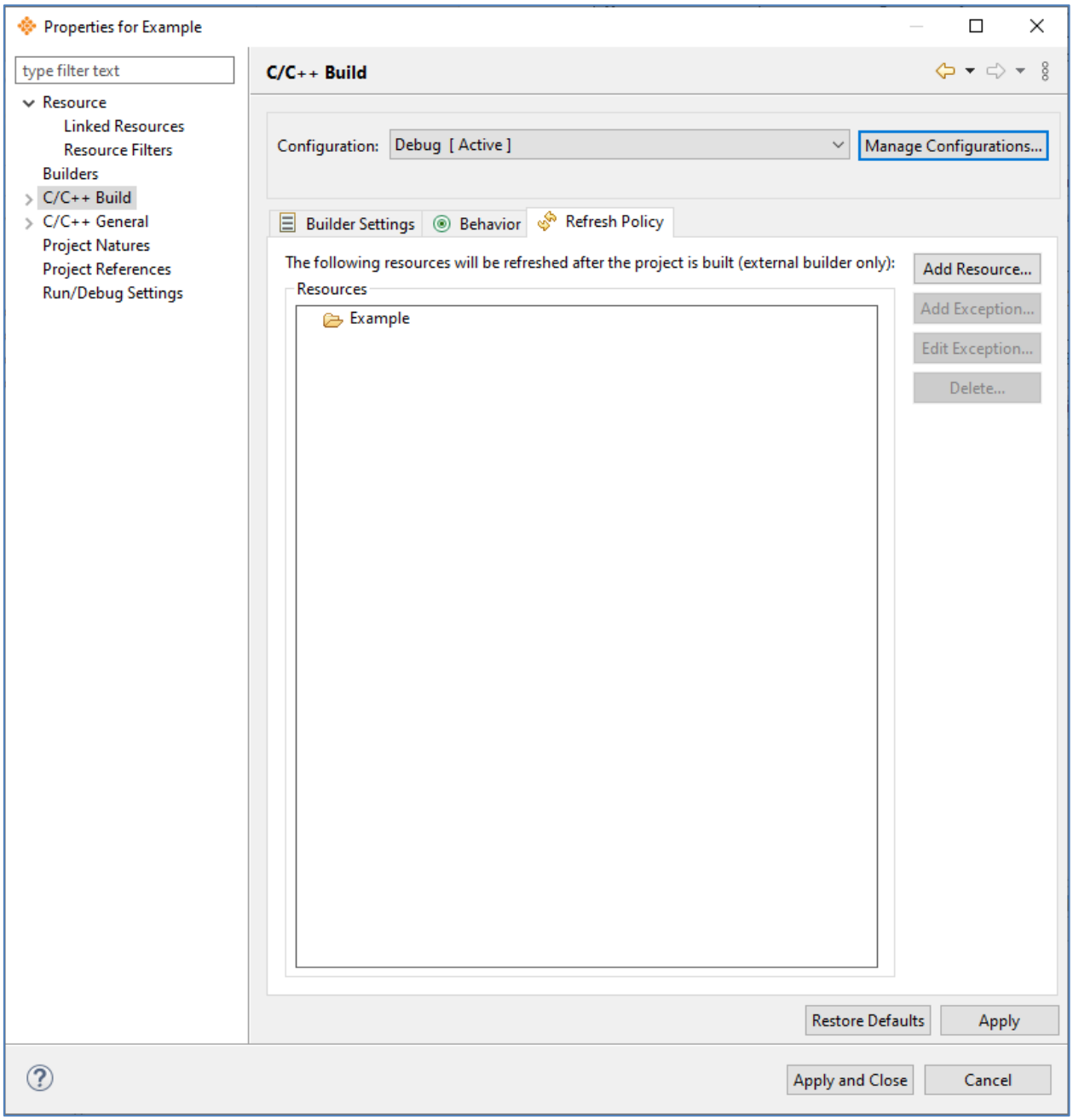


Рисунок 137. Вкладка «Refresh Policy»

Таблица 53. Элементы управления вкладки «Refresh Policy»

Название элемента	Описание функционирования
Add Resource...	Открыть диалоговое окно выбора ресурсов.
Add Exception…	Открыть диалоговое окно выбора ресурсов, которые исключаются из обновления.
Edit Exception...	Открыть диалоговое окно редактирования списка ресурсов, которые исключаются из обновления.
Delete…	Удалить выбранный ресурс из списка.

6.9.2.1. Настройки переменных сборки

Для того чтобы в конфигурациях проекта были доступны переменные путей, задаваемые пользователем, необходимо создать переменные в параметрах проекта на панели «**Build Variables**».

Созданные таким образом переменные могут быть использованы в других параметрах сборки проекта, в частности на вкладке **C/C++ Build->Settings->Tool Settings->Includes** в конструкциях вида **`${VAR}`**.

Для установки параметров сборки проекта также можно использовать системные переменные в конструкциях вида **`${env_var:VAR}`**.

Эти перемененные можно использовать во вкладках «**Include**» настроек компиляции и компоновки проектов с помощью выбранного тулчейна.

Панель «**Build Variables**» предназначена для настройки списка переменных, используемых подсистемой сборки проекта. Панель доступна только для проектов C/C++.

Для перехода на панель «**Build Variables**» следует выбрать **C/C++ Build → Build Variables**.

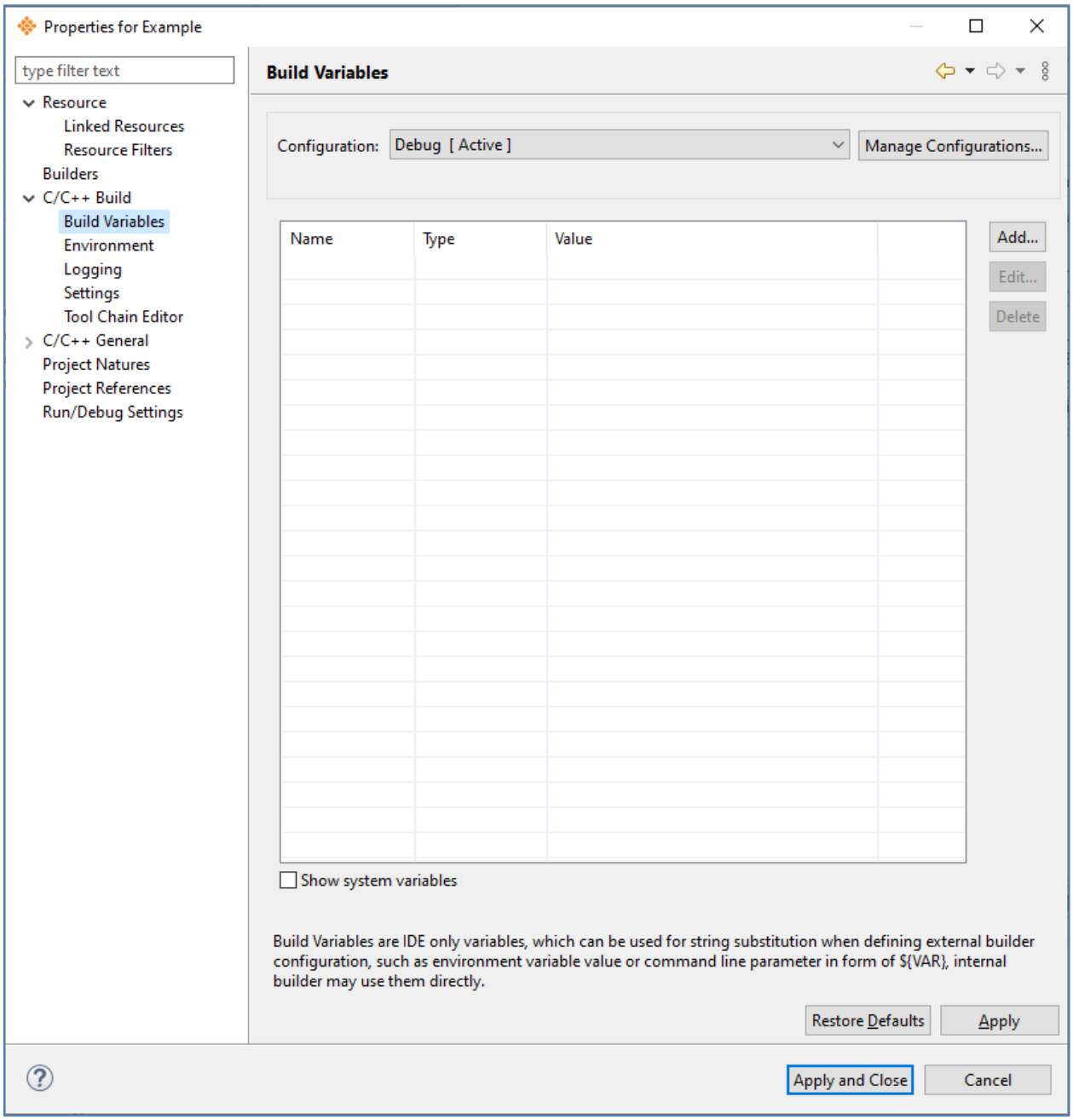


Рисунок 138. Панель «Build Variables»

Таблица 54. Элементы управления панели «Build Variables»

Название элемента	Описание функционирования	По умолчанию
Configuration	Определить тип конфигурации(ий) для выбранного проекта.	Отладка
Manage Configurations	Открыть диалоговое окно «**Manage Configurations**», в котором можно задать активные конфигурации на основе наборов инструментов, заданных для выбранного проекта. Кроме того, можно создавать новые конфигурации, переименовывать существующие или удалять ненужные.	–
Add…	Добавить новую переменную и соответствующее ей значение.	–
Edit…	Изменить имя и значение выбранной переменной.	–

Название элемента	Описание функционирования	По умолчанию
Delete	Удалить выбранные переменные из списка. Примечание: некоторые переменные доступны только для чтения, их нельзя удалить.	–
Show system variables	Включить/исключить системные переменные в/из список(ка) переменных.	Выкл

6.9.2.2. Настройки списка переменных среды

Панель «**Environment**» предназначена для настройки списка переменных среды. Панель доступна только для проектов C/C++.

Для перехода на панель «**Environment**» следует выбрать **C/C++ Build → Environment**.

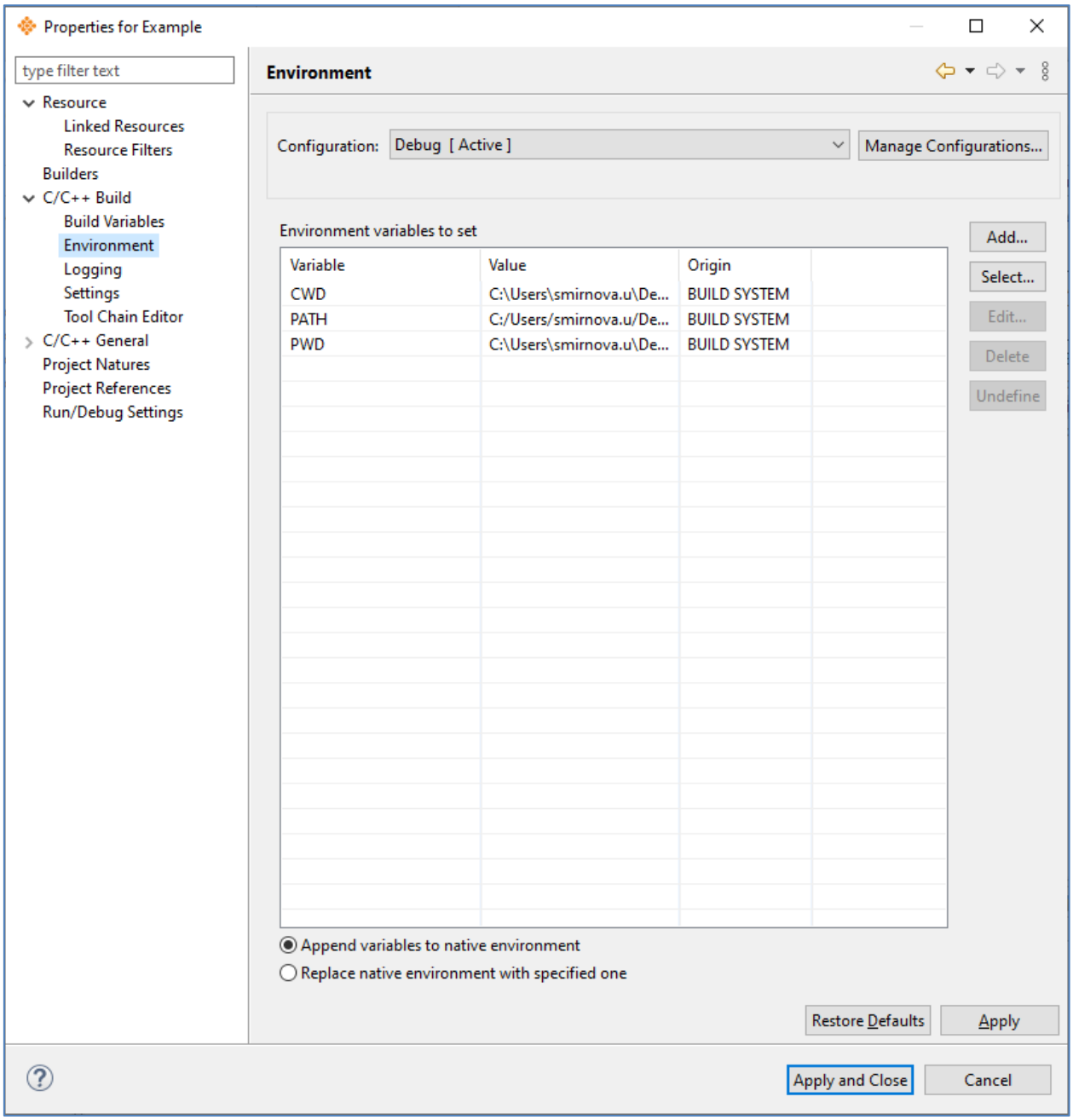


Рисунок 139. Панель «Environment»

Таблица 55. Элементы управления панели «Environment»

Название элемента	Описание функционирования	По умолчанию
Configuration	Определить тип конфигурации(ий) для выбранного проекта.	Отладка
Manage Configurations	Открыть диалоговое окно «**Manage Configurations**», в котором можно задать активные конфигурации на основе наборов инструментов, заданных для выбранного проекта. Кроме того, можно создавать новые конфигурации, переименовывать существующие или удалять ненужные.	–
Environment variables to set	Список переменных среды и соответствующих им значений. Эти значения переменных среды используются при сборке.	–
Variable	Имя переменной среды.	–

Название элемента	Описание функционирования	По умолчанию
Value	Значение переменной среды.	–
Append variables to native environment	Добавить переменные к собственной среде во время выполнения.	Вкл
Replace native environment with specified one	Заменить собственную среду с установленными переменными, затем восстановить собственную среду по завершении.	Выкл
Add...	Открыть диалоговое окно «**New variable**» для создания новой переменной среды и ее значения. Пользовательские переменные среды, созданные пользователем, будут выделены в списке жирным шрифтом.	–
Select...	Открыть диалоговое окно «**Select variables**» для выбора переменной из списка системных.	–
Edit...	Открыть диалоговое окно «**Edit variable**» для изменения имени и значения выбранной переменной среды.	–
Remove	Удалить выбранную переменную среды из списка.	–
Undefine	Удалить определение выбранной переменной, однако у некоторых переменных, таких как **PATH**, нельзя удалить определение.	–

6.9.2.3. Настройки протоколирования

Панель «**Logging**» предназначена для управления процессом протоколирования сборки проекта. Панель доступна только для проектов C/C++.

Для перехода на панель «**Logging**» следует выбрать **C/C++ Build → Logging**.

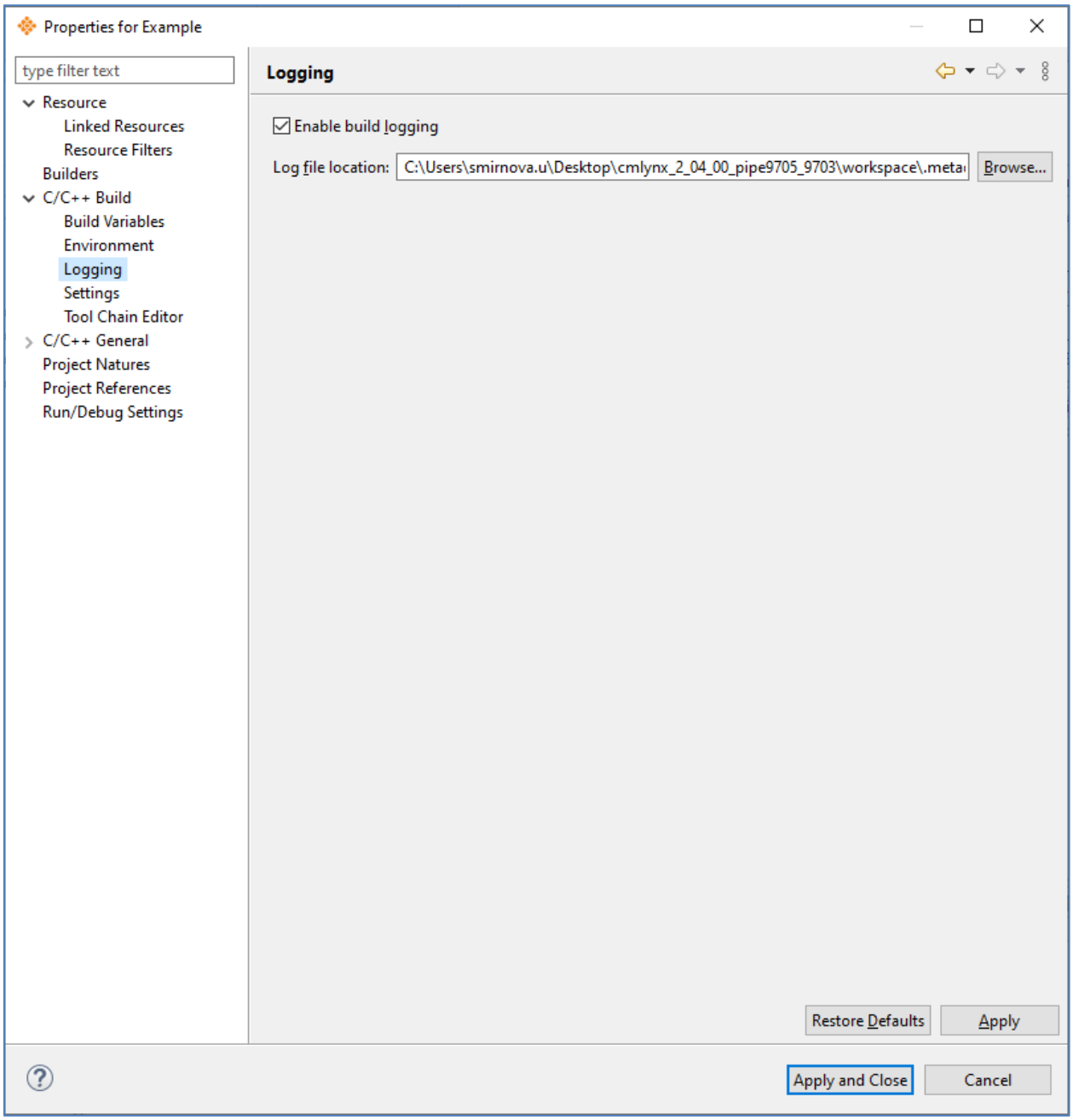


Рисунок 140. Панель «Logging»

Таблица 56. Элементы управления панели «Logging»

Название элемента	Описание функционирования	По умолчанию
Enable build logging	Включить режим протоколирования процесса сборки.	Вкл
Log file location	Задать полный путь к файлу журнала.	–
Browse…	Открыть диалоговое окно «**Choose Log File**» для выбора файла журнала.	–

6.9.2.4. Настройки параметров работы тулчейн

Панель «**Settings**» предназначена для настройки параметров компонентов тулчейн (toolchain). Панель доступна только для проектов C/C++, папок и файлов вне зависимости от типа проекта.

Для перехода на панель «**Settings**» следует выбрать **C/C++ Build → Settings**.

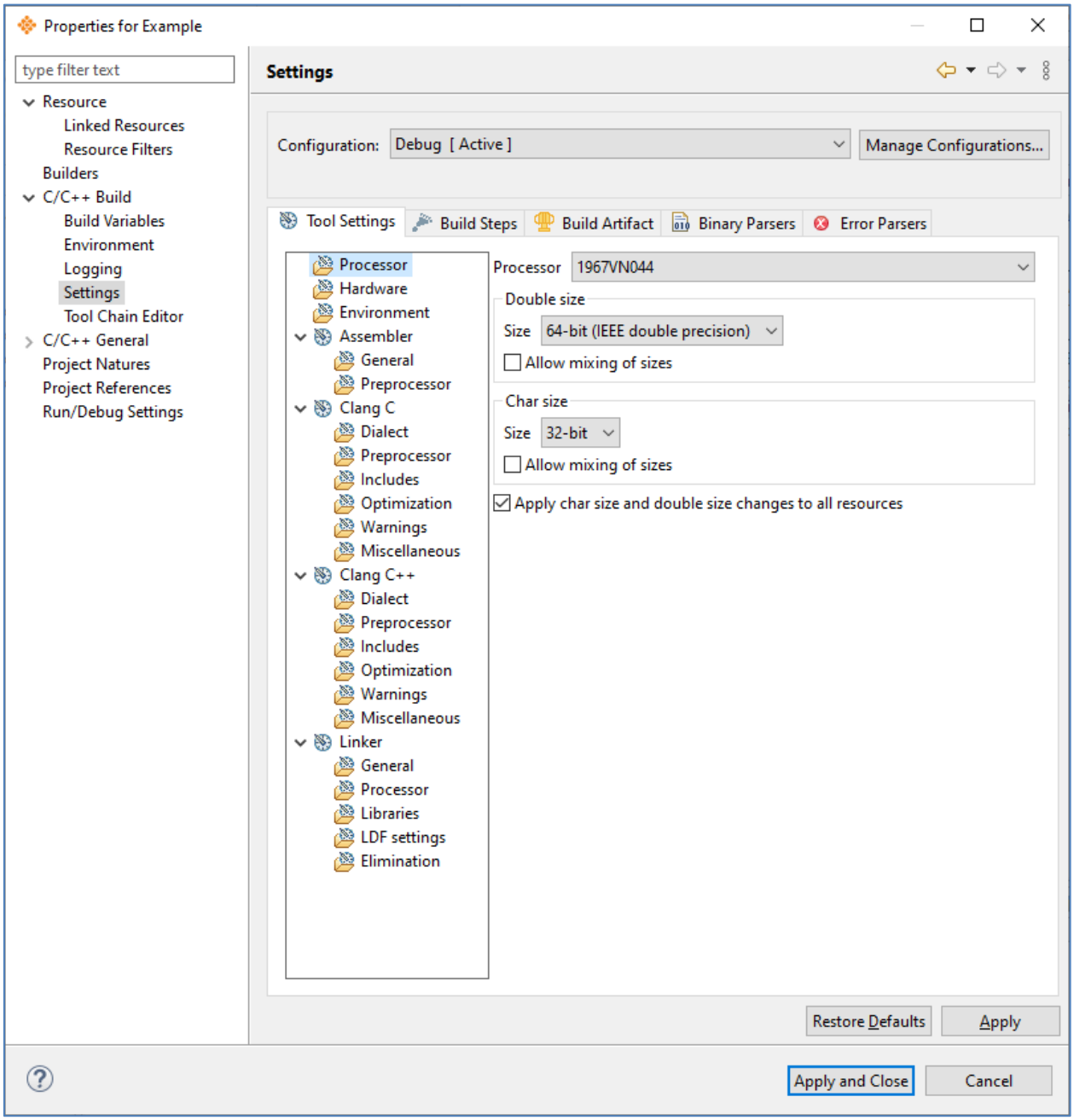


Рисунок 141. Панель «Settings»

В верхней части панели «**Settings**» расположен список существующих конфигураций. Редактирование списка конфигураций (создание, переименование и удаление конфигураций) выполняется в отдельном окне (см. Рисунок 134). Для перехода к этому окну необходимо нажать кнопку «**Manage Configurations**».

Настройки параметров работы тулчейн задаются для конфигурации, т.е. а рамках одного проекта можно настроить несколько конфигураций для разных процессоров и с разными параметрами.

В центральной части панели «**Settings**» расположены вкладки:

- «**Tool Settings**»;
- «**Build Steps**»;
- «**Build Artefact**»;
- «**Binary Parsers**»;

- «**Error Parsers**».

Набор вкладок зависит от типа выбранного ресурса.

6.9.2.4.1. Настройки компонентов тулчейн

На вкладке **«Tool Settings»** находятся настройки компонентов тулчейн (см. Рисунок 141).

Справа на вкладке «**Tool Settings**» приведен список настроек для выбранной категории.

6.9.2.4.1.1 Процессор

На панели **«Processor»** (см. Рисунок 142) задаются тип процессора и размеры типов для текущей конфигурации.

Примечание: для каждой конфигурации сборки (Debug/ Release) тип процессора и размеры типов задаются индивидуально.

Таблица 57. Элементы управления панели «Processor»

Название элемента	Описание функционирования	По умолчанию
Processor	Выбор процессора для текущей конфигурации: ▪ 1967VN028 Rev.2; ▪ 1967VN044; ▪ 1967VN044 Rev.3 (по умолчанию); ▪ 1967VN058; ▪ 1967VN034 (устаревший). **Примечания:** **1.** При изменении типа процессора в конфигурации сборки происходит автоматическая смена процессора, установленного в конфигурации отладки. При этом выводится предупреждающий диалог (при открытии конфигурации или запуске сессии отладки), который можно отключить глобально для всего рабочего пространства. **2**. При переключении на процессор 1967VN058 или с него на другой происходит автоматическое добавление/удаление каталога «toolchain\LibLLVM\1967VN058» в настройках компоновщика (см. п. 6.9.2.4.1.6.5). **3**. Также при переключении на процессор 1967VN058 становится доступна настройка функций управления памятью процессора (флаг «Default Memory management functions», см. Рисунок 143).	Последний использовавшийся
Double size	Выбрать размер типа `double`: **Size** – выбор размера. Можно установить 32 бита (IEEE одинарной точности) или 64 бита (IEEE двойной точности); **Allow mixing of sizes** – при установленном флаге разрешается компоновать результирующий объектный файл с объектными файлами, скомпилированными с различным размером типа `double`. **Примечание:** для проектов на C++ использовать флаг «**Allow mixing of sizes**» настоятельно не рекомендуется.	64 бита (IEEE двойной точности) Выкл

Название элемента	Описание функционирования	По умолчанию
Char size	Выбрать размер типа **`char`**: **Size** – выбор размера. Можно установить 8 или 32 бита; **Allow mixing of sizes** – при установленном флаге разрешается компоновать результирующий объектный файл с объектными файлами, скомпилированными с различным размером типа **`char`**. **Примечание:** для проектов на C++ использовать флаг «Allow mixing of sizes» настоятельно не рекомендуется.	32 бита Выкл
Apply char size and double size changes to all resources	Включение/отключение опции применения текущих настроек для всех ресурсов проекта. Исключение составляют те ресурсы, рядом с которыми стоит значок ключа, т.е. для них заданы отдельные параметры в текущей конфигурации.	Вкл
Default memory management functions	Включение/отключение опции управления памятью (только для процессора 1967ВН058). После устанеовки флага «Default memory management functions» становятся доступны настройки «Number of cores», «Enable caches» и «Memory management mode». В зависимости от значений этих настроек выбирается соответствующая версия LDF и стартового модуля.	Выкл
Number of cores	Использовать конфигурацию памяти для заданного числа внутренних ядер процессора. Определяет выбор шаблонного LDF и распределения памяти L3 для одного или двух внутренних ядер процессора. Значение по умолчанию - одно ядро.	1
Cashes	Использовать конфигурацию памяти для выбранных кешей и включить их при запуске. Определяет выбор шаблонного LDF с учетом занимаемой кэшами памяти, версию специализированного загрузчика, а также работу модуля управления памятью в части использования кешей. Возможные значения: L1, L1+L2. Значение по умолчанию - L1.	L1

Название элемента	Описание функционирования	По умолчанию
Management mode	Включить функции управления памятью MMU в заданном режиме. Определяет выбор версии специализированного загрузчика с включением функций управления памятью в заданном режиме. Возможные значения: ▪ Fixed TLB only - защищенный режим MMU. Доступная память автоматически конфигурируется при старте и соответствует только таблице TLB, с максимальным числом дескрипторов памяти – 32; ▪ Fixed TLB with hardware management-защищенный режим MMU. Доступная память автоматически конфигурируется при старте и соответствует фиксированной части TLB (в текущей версии 24 дескриптора), а также дополняется динамически загружаемой аппаратно из таблиц трансляций уровней L1 и L2 (до 8 дескрипторов); ▪ Fixed TLB with software management- защищенный режим MMU. Доступная память автоматически конфигурируется при старте и соответствует фиксированной части TLB (в текущей версии 24 сегмента), а также дополняется динамически загружаемой программно (до 8 сегментов). Модуль управления памятью из поставляемой стандартной библиотеки делает это также из таблиц трансляций уровней L1 и L2, для совместимости алгоритмов с аппаратным режимом подкачки дескрипторов; ▪ TA_mode – не защищенный, упрощенный режим.	Fixed TLB only

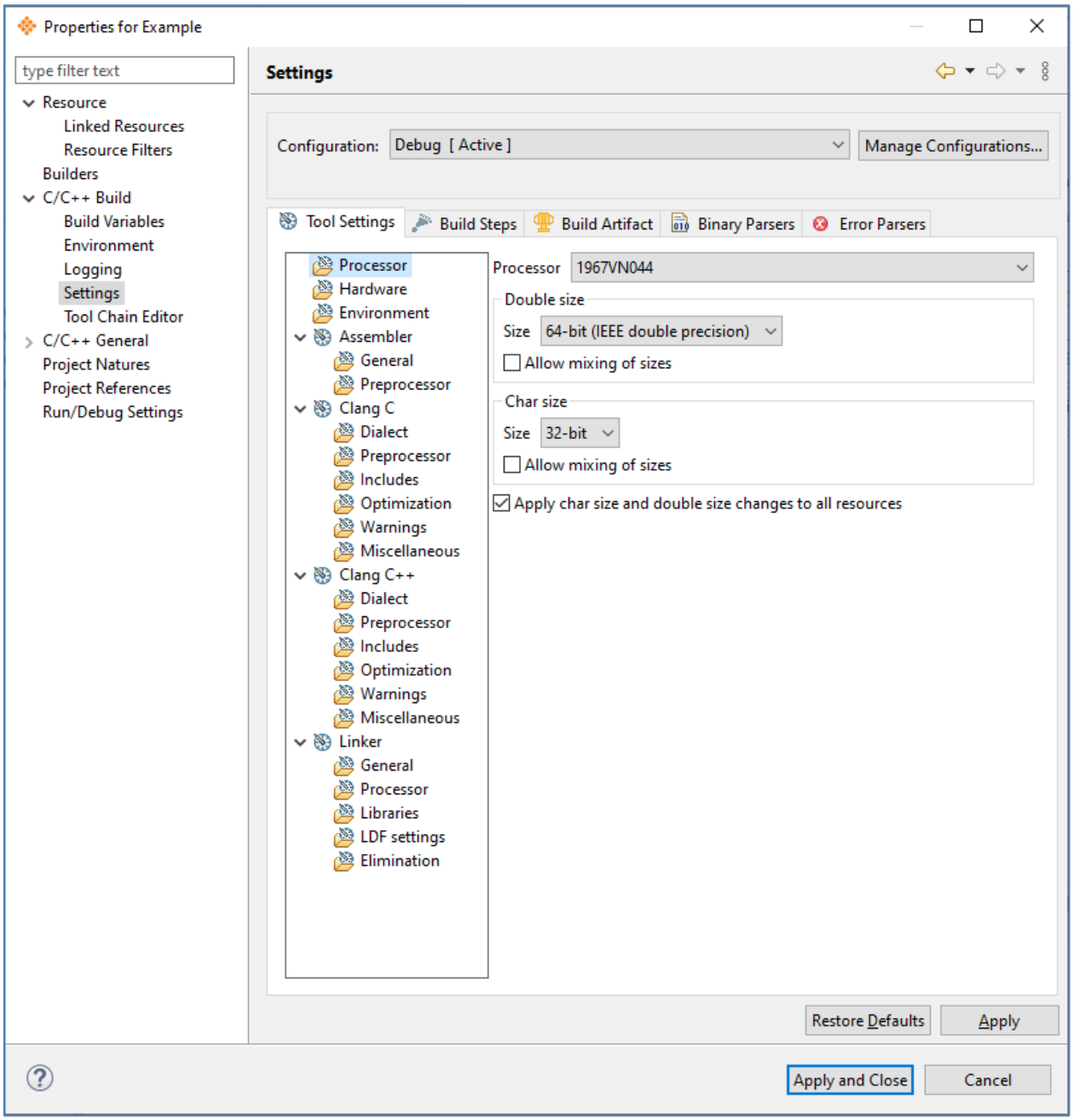


Рисунок 142. Панель «Processor»

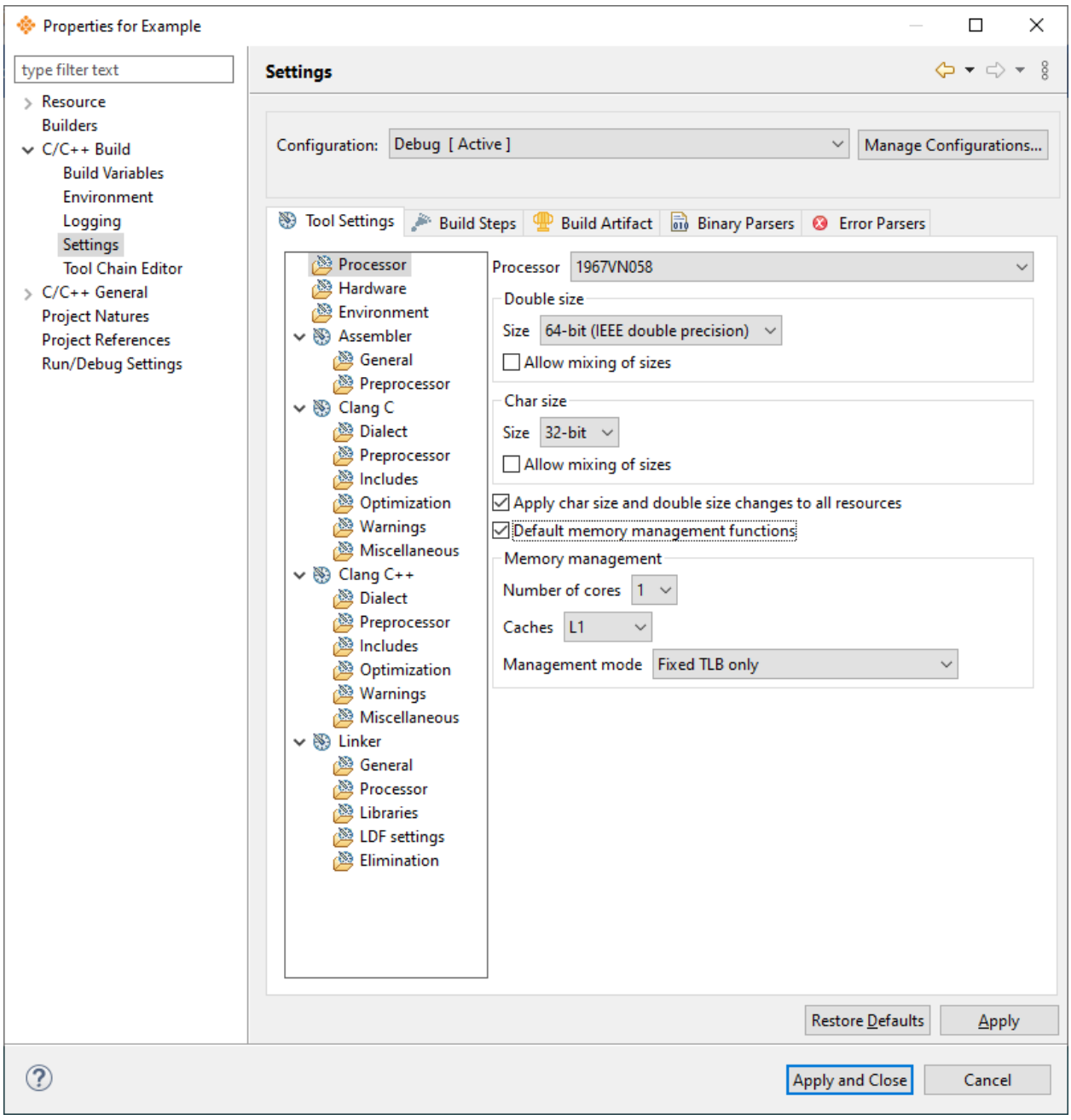


Рисунок 143. Панель «Processor» (для процессора 1967ВН058)

6.9.2.4.1.2 Оборудование

На панели **«Hardware»** (см. Рисунок 144) задаются дополнительные настройки используемого оборудования, в том числе вариант подключения HAL.

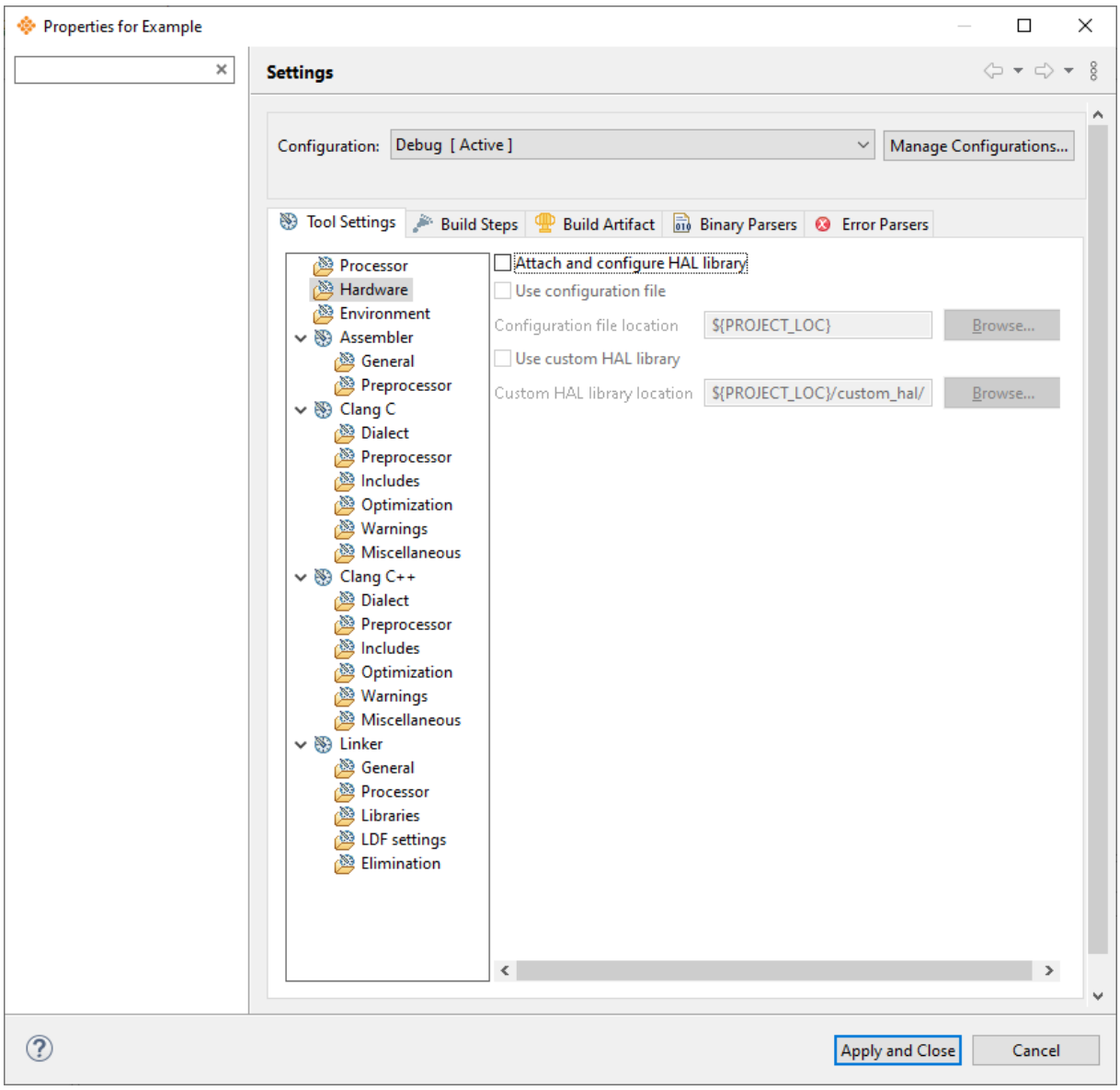


Рисунок 144. Панель «Hardware» с настройками по умолчанию

Таблица 58. Элементы управления панели «Hardware»

Название элемента	Описание функционирования	По умолчанию
Attach and configure HAL library	Подключить библиотеку HAL. При включении данной опции появляется рамка «**Generic hardware settings**» с дополнительными элементами – см. Рисунок 145.	Выкл

Название элемента	Описание функционирования	По умолчанию
Use configuration file	Использовать файл конфигурации. Если флаг «**Use configuration file**» выключен, то параметры HAL задаются настройками в свойствах проекта и передаются в параметры компилятора. В том числе, передаётся макрос HAL_NO_CONFIG. Если флаг «**Use configuration file**» включен, то в параметры компилятора передаётся указанный путь для поиска файла конфигурации. При при включенном флаге «**Use configuration file**» блок настроек HAL в свойствах проекта («**Generic hardware settings**») скрывается.	Выкл
Configuration file location	Путь для поиска файла конфигурации с настройками HAL.	${PROJECT_LOC}
Use custom HAL library	Использовать пользовательскую версию библиотеки HAL. При при включенном флаге «**Use custom HAL library**» блок настроек HAL в свойствах проекта («**Generic hardware settings**») скрывается.	Выкл
Custom HAL library location	Путь до пользовательской версии библиотеки HAL. **Примечание:** При необходимости изменить данный параметр рекомендуется предварительно отключить флаг «**Use custom HAL library**», указать путь к файлу, а затем включить флаг снова.	${PROJECT_LOC}/ custom_hal/
Chip input clock (XTI), KHz	Задаёт внешнюю частоту в килогерцах (макрос «`XTI_KHZ`»).	25000
System bus clock (SCLK), Hz	Задаёт частоту внешней шины в герцах (макрос «`SCLK_HZ`»).	50000000
Core clock (CCLK), Hz	Задаёт частоту ядра процессора в герцах (макрос «`CCLK_HZ`»).	250000000
Assert level	Задает уровень детализации вывода сообщений об ошибках при использовании HAL (только для 1967ВН058). Возможные значения: ▪ 0 – None (сообщение не выводится); ▪ 1 - File and line (указывается имя файла и строка в нем); ▪ 2 - File, line and expression (указывается имя файла, строка в нем и сообщение).	1-File and line

Подключение HAL возможно в двух режимах:

1. Использование HAL, поставляемого вместе с CM-LYNX. Для этого необходимо включить опцию «**Attach and configure HAL library**» и не включать «**Use custom HAL library**». В проект будет добавлена виртуальная ссылка «`_hal`», указывающая на каталог HAL соответствующий процессору, заданному на вкладке «**Processor**».

2. Использование пользовательского HAL. Для этого необходимо включить опцию «**Attach and configure HAL library**», задать путь до пользовательского HAL в поле «**Custom HAL library location**» и включить опцию «**Use custom HAL library**».

Для корректной работы пользовательский HAL должен соответствовать некоторым ограничениям:

- Заголовочные файлы должны храниться в подкаталоге «**inc**» от указанного;
- Необходимый к компиляции исходный код должен храниться в указанном каталоге или его подкаталогах;
- Дополнительные настройки, сверх задаваемых в «**Generic hardware settings**», пользователю придётся задавать вручную, в коде или в настройках конфигурации.

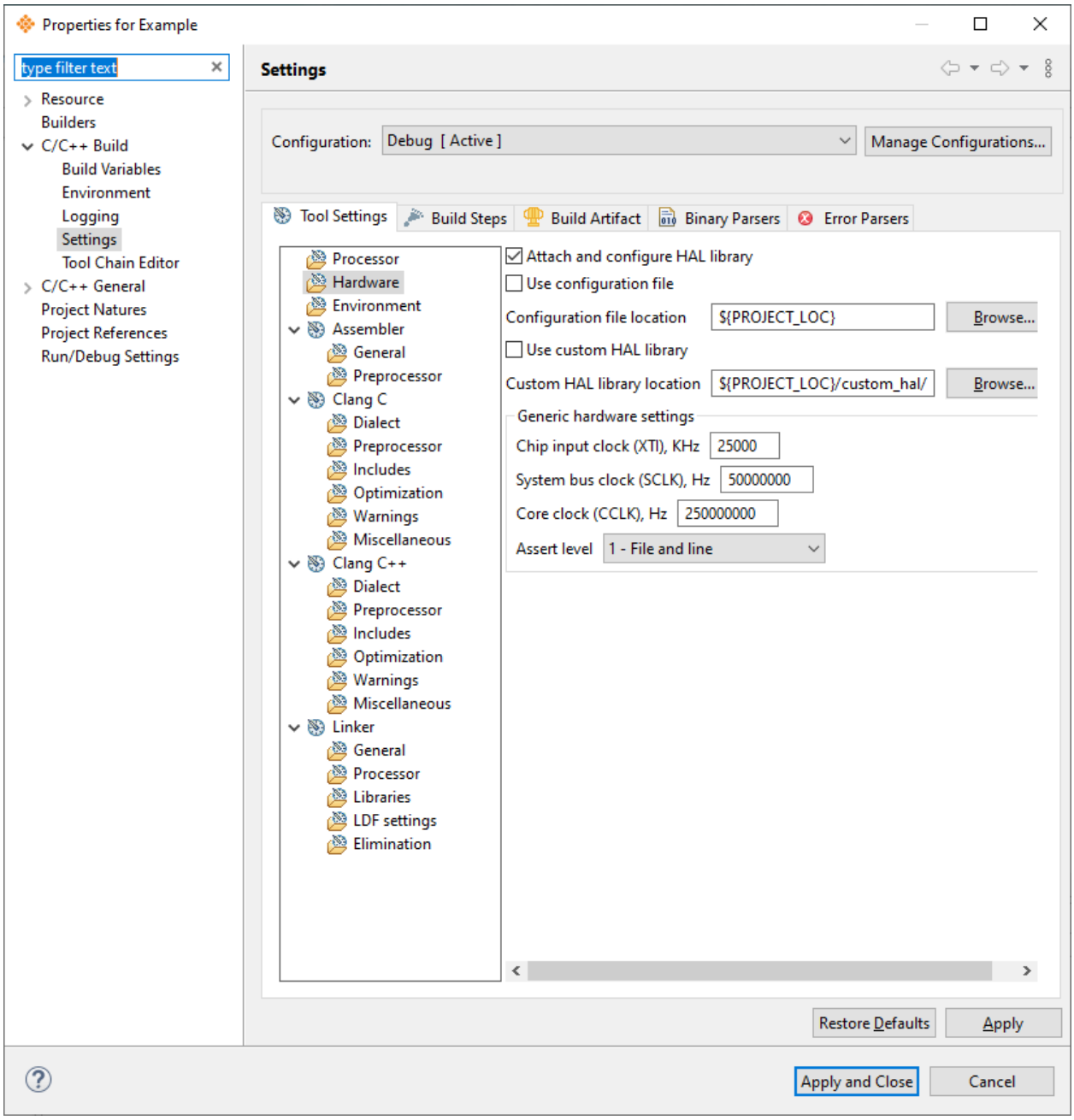


Рисунок 145. Панель «Hardware» с активной панелью «Generic hardware settings»

6.9.2.4.1.3 Окружение

На панели **«Environment»** (см. Рисунок 146) задаются настройки подключения окружения, в том числе операционной системы.

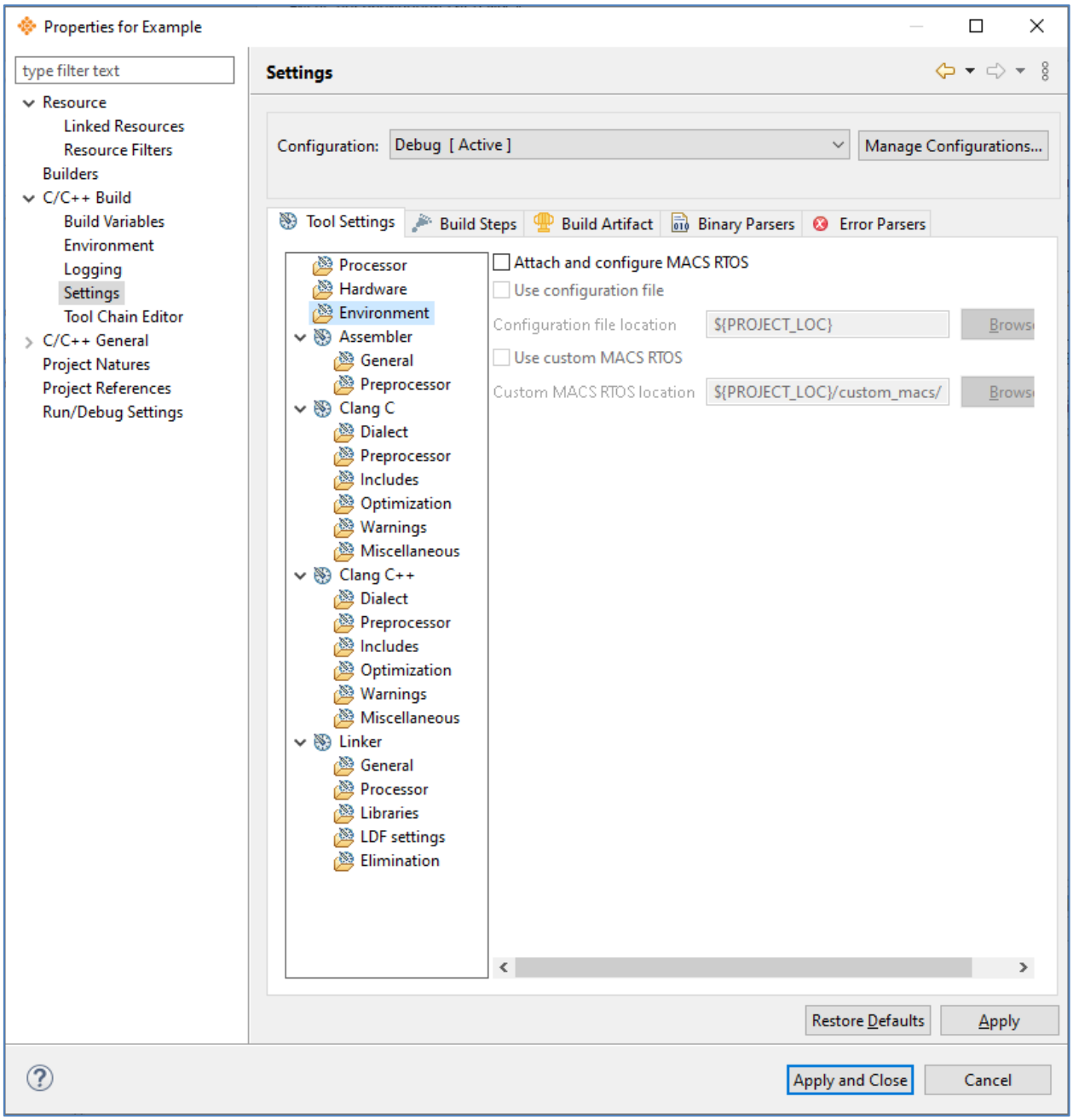


Рисунок 146. Панель «Environment» с настройками по умолчанию

Таблица 59. Элементы управления панели «Environment»

Название элемента	Описание функционирования	По умолчанию
Attach and configure MACS RTOS	Подключить операционную систему ОСРВ МАКС-Lynx. Операционная система описана в документе ML_ОСРВ_МАКС_Lynx_Руководство_пользователя При включении данной опции появляется рамка «**MACS RTOS settings**» с дополнительными элементами – см. Рисунок 147.	Выкл
Use configuration file	Использовать конфигурационный файл «**macs_config.h**».	Выкл

Название элемента	Описание функционирования	По умолчанию
Configuration file location	Путь к файлу конфигурации. **Примечание:** При необходимости изменить данный параметр рекомендуется предварительно отключить флаг «**Use configuration file**», указать путь к файлу, а затем включить флаг снова.	${PROJECT_LOC}
Use custom MACS RTOS	Использовать пользовательскую версию МАКС.	Выкл
Custom MACS RTOS location	Путь до пользовательской версии МАКС. **Примечание:** При необходимости изменить данный параметр рекомендуется предварительно отключить флаг «**Use custom MACS RTOS**», указать путь к файлу, а затем включить флаг снова.	${PROJECT_LOC}/ custom_macs/
OS tick rate, Hz	Начальная частота переключения задач в вытесняющем режиме, Гц (макрос «**MACS_INIT_TICK_RATE_HZ**»).	50
Native task stack size, words	Размер стека по умолчанию для задачи МАКС в словах (макрос «**MACS_TASK_STACK_SIZE**»).	1200
POSIX thread stack size, words	Размер стека по умолчанию для потока pthread в словах (макрос «**MACS_PTHREAD_STACK_SIZE**»).	50000
Stack limit overrun control	Программный контроль переполнения стека (макрос «**MACS_WATCH_STACK**»).	Вкл
Mutex priority inversion resolver	Обработка инверсии приоритетов (макрос «**MACS_MUTEX_PRIORITY_INVERSION**»).	Выкл
Run default POSIX thread	Запуск с выполнением потока pthread по умолчанию (макрос «**MACS_PTHREAD_START**»).	Выкл
Run default native task	Запуск с выполнением задачи МАКС по умолчанию (макрос «**MACS_TASK_START**»).	Вкл
Memory wiping with random number	Очистка памяти случайным числом при освобождении (макрос «**MACS_MEM_WIPE**»).	Вкл
SMP configuration	Конфигурация ОСРВ в многоядерном режиме SMP (макрос «**MACS_SMP**»).	Выкл
Multiprocessor group ID	Идентификатор группы процессорных ядер (макрос «**MACS_CPU_GROUP_ID**»).	0
Debug mode	Режим отладки с дополнительным контролем (макрос «**MACS_DEBUG**»).	Выкл
Trace mode	Трассировка с передачей событий по JTAG или LINK (макрос «**MACS_TRACE**»).	Выкл
Tracing to stdout	Трассировка выводом сообщений в стандартную консоль (макрос «**MACS_PRINTF_ALLOWED**»).	Вкл

Поддерживается подключение операционной системы реальной времени МАКС в 2 режимах:

1. Использование ОСРВ МАКС-Lynx, поставляемого вместе с CM-LYNX. Для этого необходимо включить опцию «**Attach and configure RTOS MACS**» и не включать «**Use custom MACS RTOS**». В проект будет добавлена виртуальная ссылка «`_macs`», указывающая на каталог ОСРВ МАКС-Lynx в CM-LYNX.

2. Использование пользовательской версии ОСРВ МАКС-Lynx. Для этого необходимо включить опцию «**Attach and configure RTOS MACS**», задать путь до пользовательской версии ОСРВ МАКС-Lynx в поле «**Custom MACS RTOS location**» и включить опцию «**Use custom MACS RTOS**».

   Для корректной работы пользовательская версия МАКС должен соответствовать некоторым ограничениям:

   - Заголовочные файлы должны храниться в подкаталоге «**include**» от указанного;
   - Необходимый к компиляции исходный код должен храниться в указанном каталоге или его подкаталогах;
   - Дополнительные настройки, сверх задаваемых в «**MACS RTOS settings**», пользователю придётся задавать вручную, в коде или в настройках конфигурации.

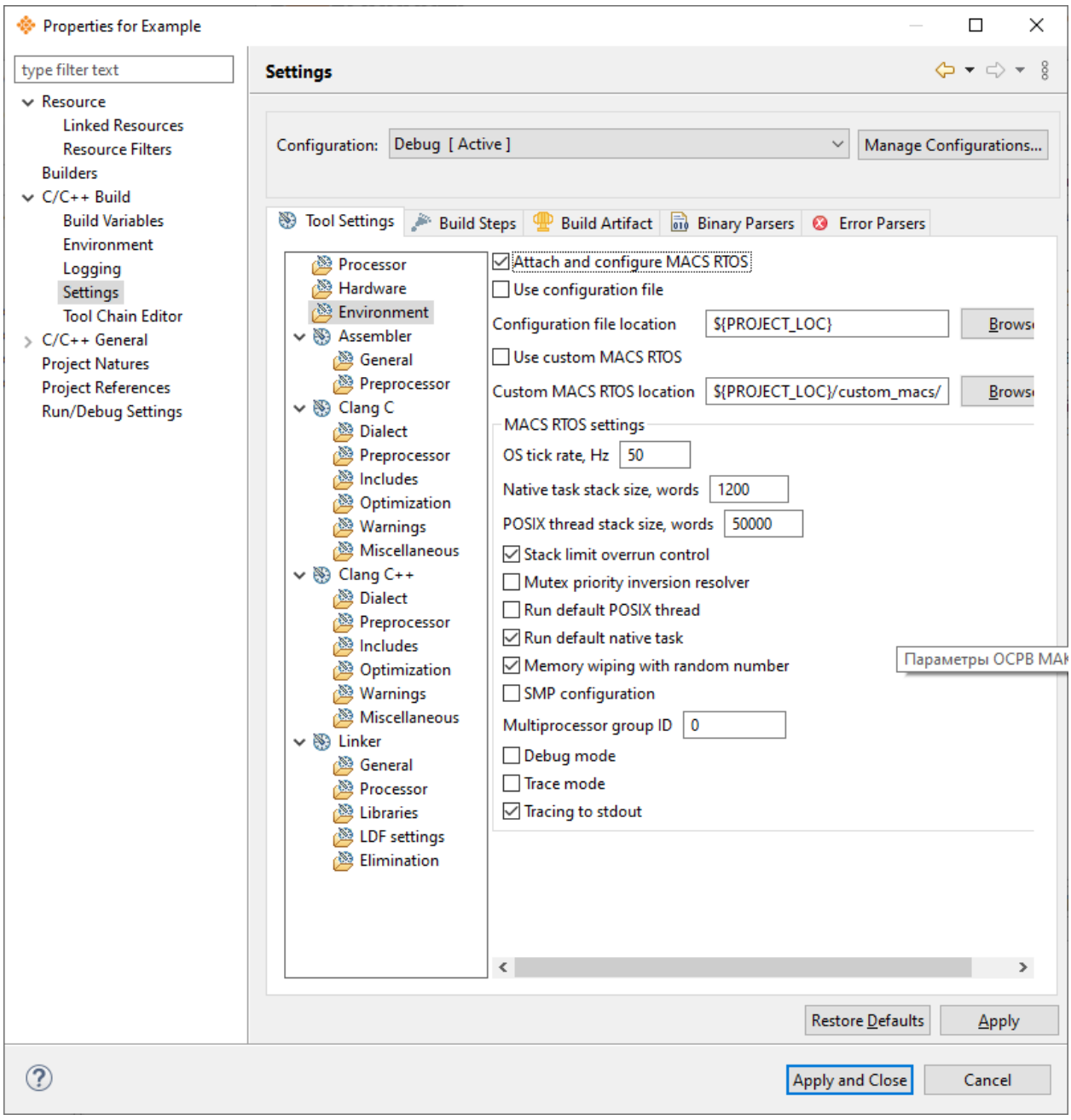


Рисунок 147. Панель «Environment» при подключении ОСРВ МАКС-Lynx

6.9.2.4.1.4 Ассемблер

На панели **«Assembler»** (см. Рисунок 148) доступны для просмотра параметры вызова ассемблера и список, передаваемых ему опций.

Таблица 60. Элементы управления панели «Assembler»

Название элемента	Описание функционирования
Command	Строка вызова ассемблера.
All options	Полный список используемых опций ассемблера.
Command line pattern	Формат строки вызова ассемблера с опциями.

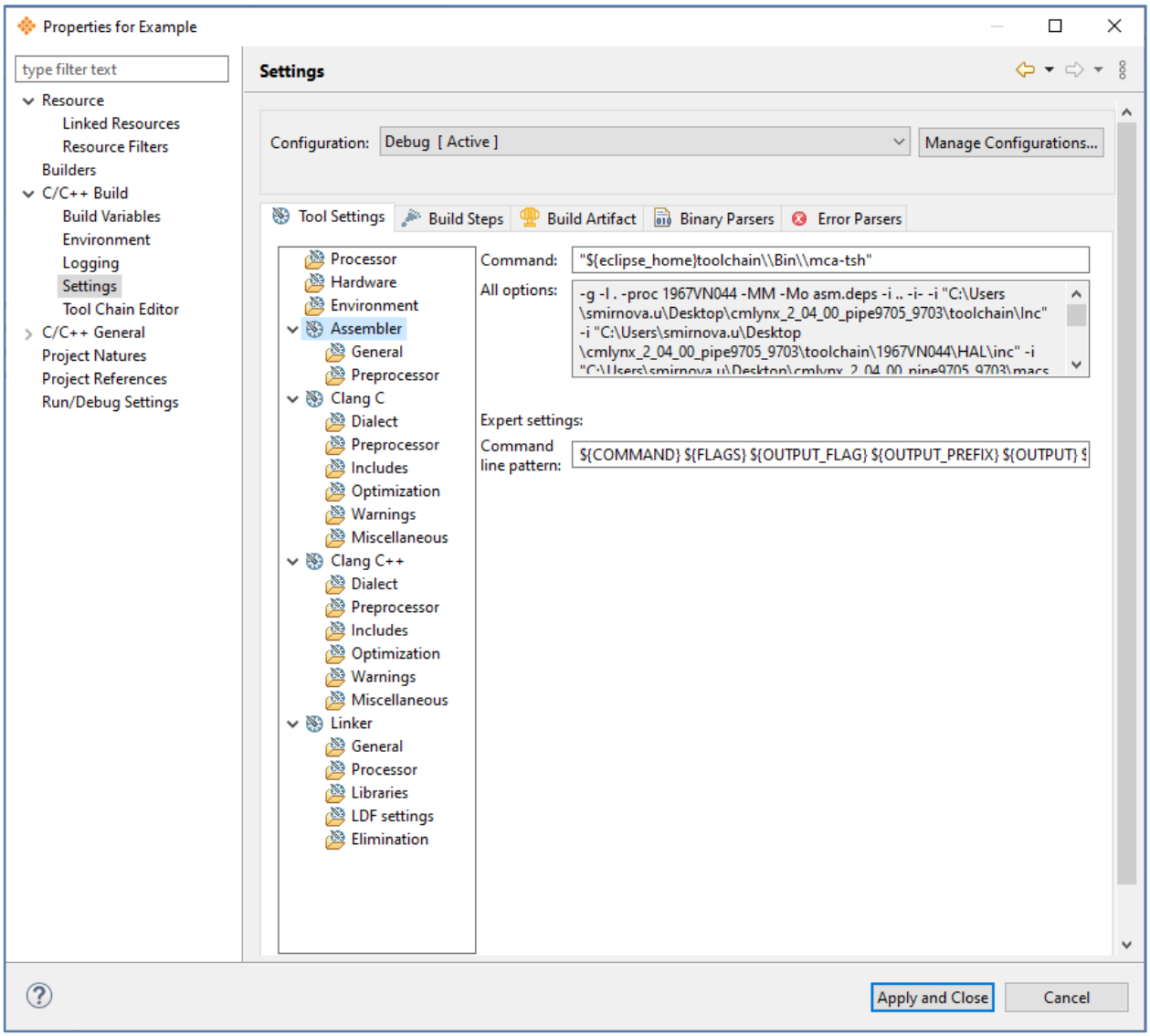


Рисунок 148. Панель «Assembler»

6.9.2.4.1.4.1 Основные настройки

На панели «**Assembler → General**» (см. Рисунок 149) настраиваются основные параметры работы ассемблера (см. Таблица 61).

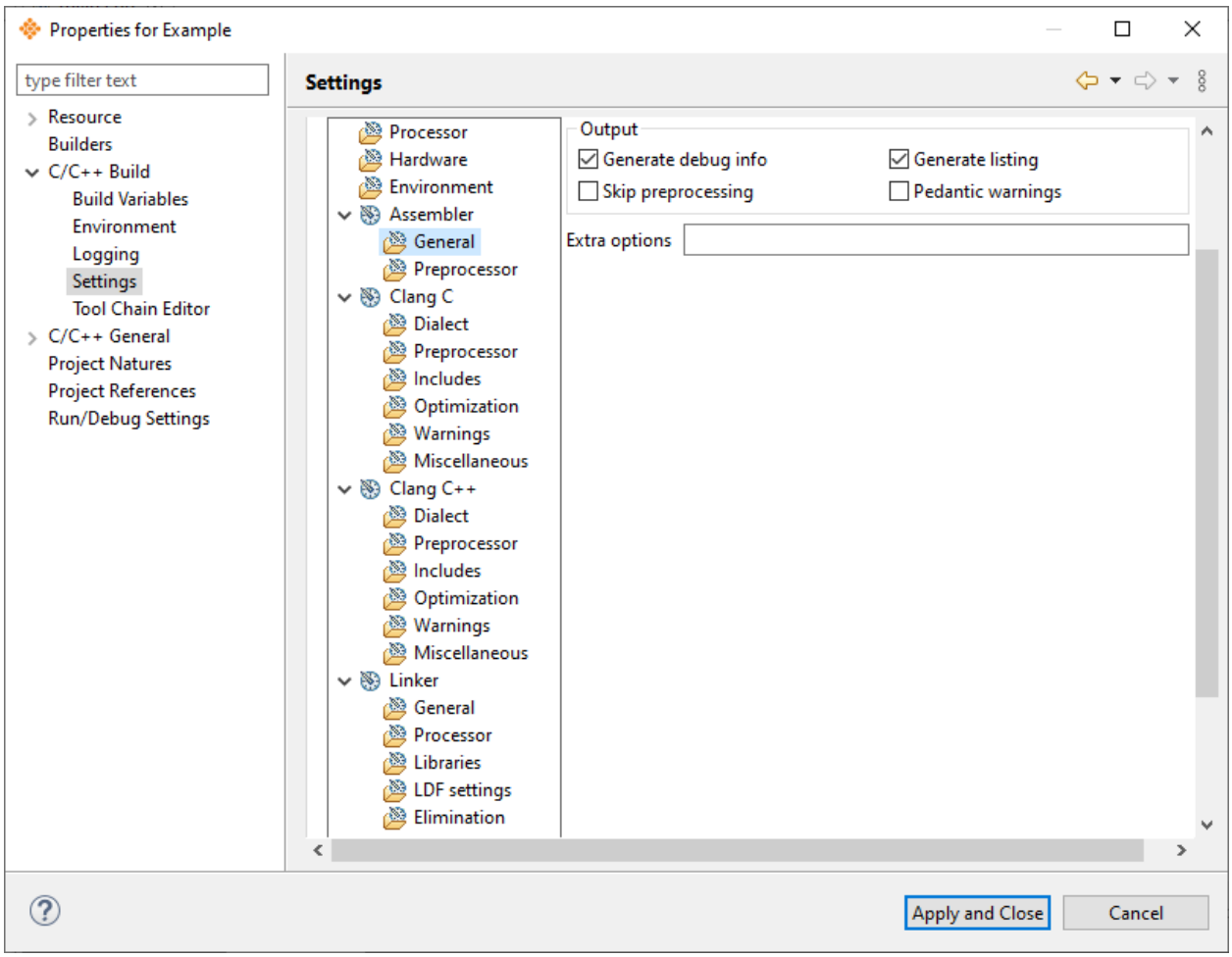


Рисунок 149. Панель «Assembler → General»

Таблица 61. Элементы управления панели «Assembler → General»

Название элемента	Описание функционирования	По умолчанию
Generate debug info	Добавить в выходной объектный файл информацию, необходимую для отладки по исходному тексту программы.	Вкл
Generate listing	Создать файл листинга в форме: **имя_исходного_файла.LST**.	Вкл
Skip preprocessing	При ассемблировании пропустить фазу препроцессинга исходных файлов.	Выкл
Pedantic warnings	Включить выдачу дополнительных предупреждений. При включенном флаге в ассемблер передается ключ «**-pedant-warn**». При включенном флаге **«Pedantic warnings»** выводятся уведомления о метках, нарушающих правила компоновщика по удалению неиспользуемого кода.	Выкл
Extra options	Список дополнительных опций, которые будут передаваться ассемблеру в командной строке. Опции ассемблера описаны в документе ML_MCA-TSH Руководство пользователя.	–

6.9.2.4.1.4.2 Настройки препроцессора

Настройки препроцессора (см. Таблица 62) задаются на панели «**Assembler →Preprocessor**» (см. Рисунок 150).

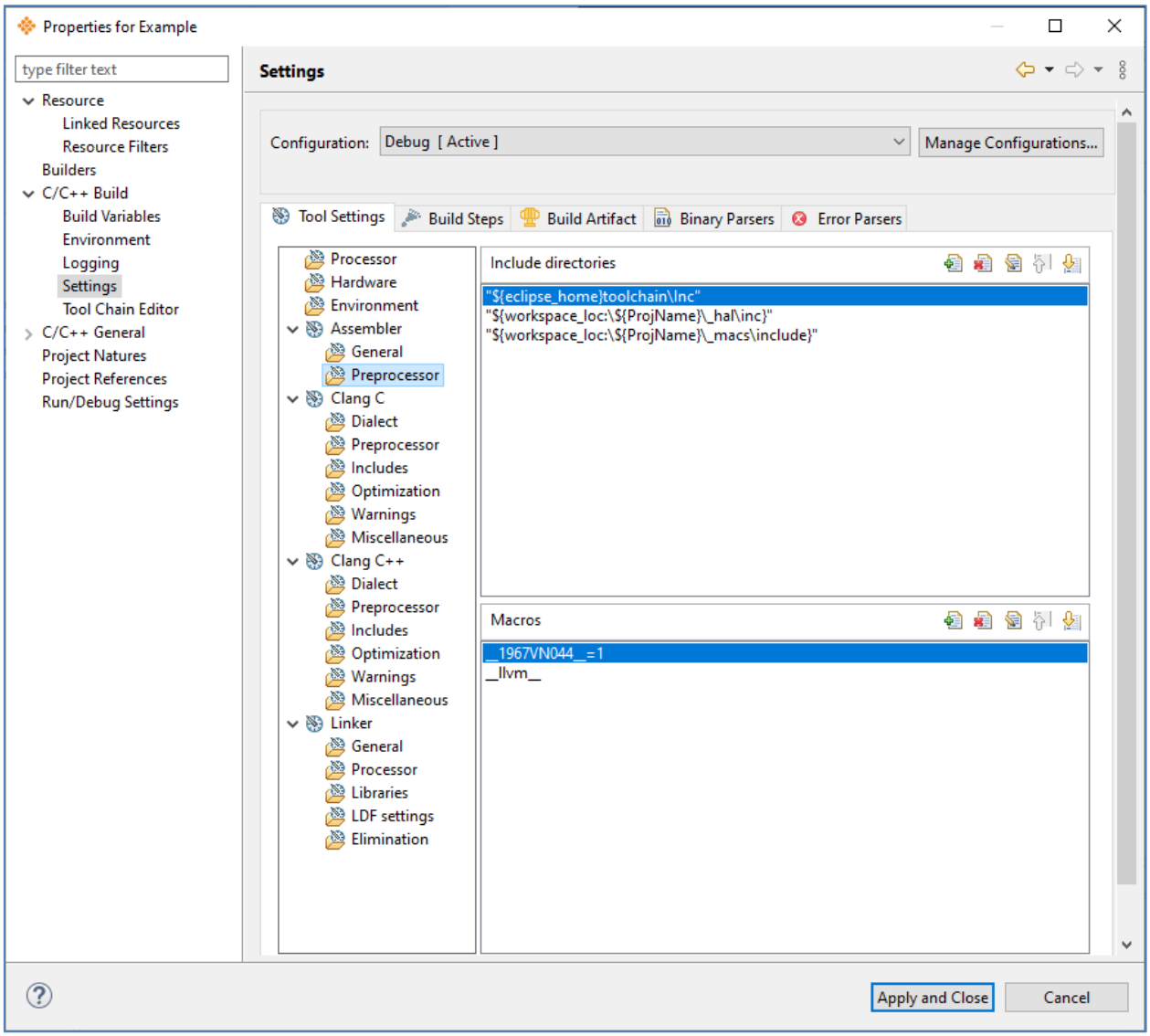


Рисунок 150. Панель «Assembler → Preprocessor»

Таблица 62. Элементы управления панели «Assembler →Preprocessor»

Название элемента	Описание функционирования
Include directories	Список каталогов, в которых будет производиться поиск include-файлов.
Macros	Список макросов, которые будут передаваться ассемблеру. Поддерживаемые макросы ассемблера описаны в документе ML_MCA-TSH Руководство пользователя.

6.9.2.4.1.5 Компилятор C и C++

На панелях **«Clang C»** и «**Clang C++**» (см. Рисунок 151) доступны для просмотра параметры вызова компилятора и список, передаваемых ему опций.

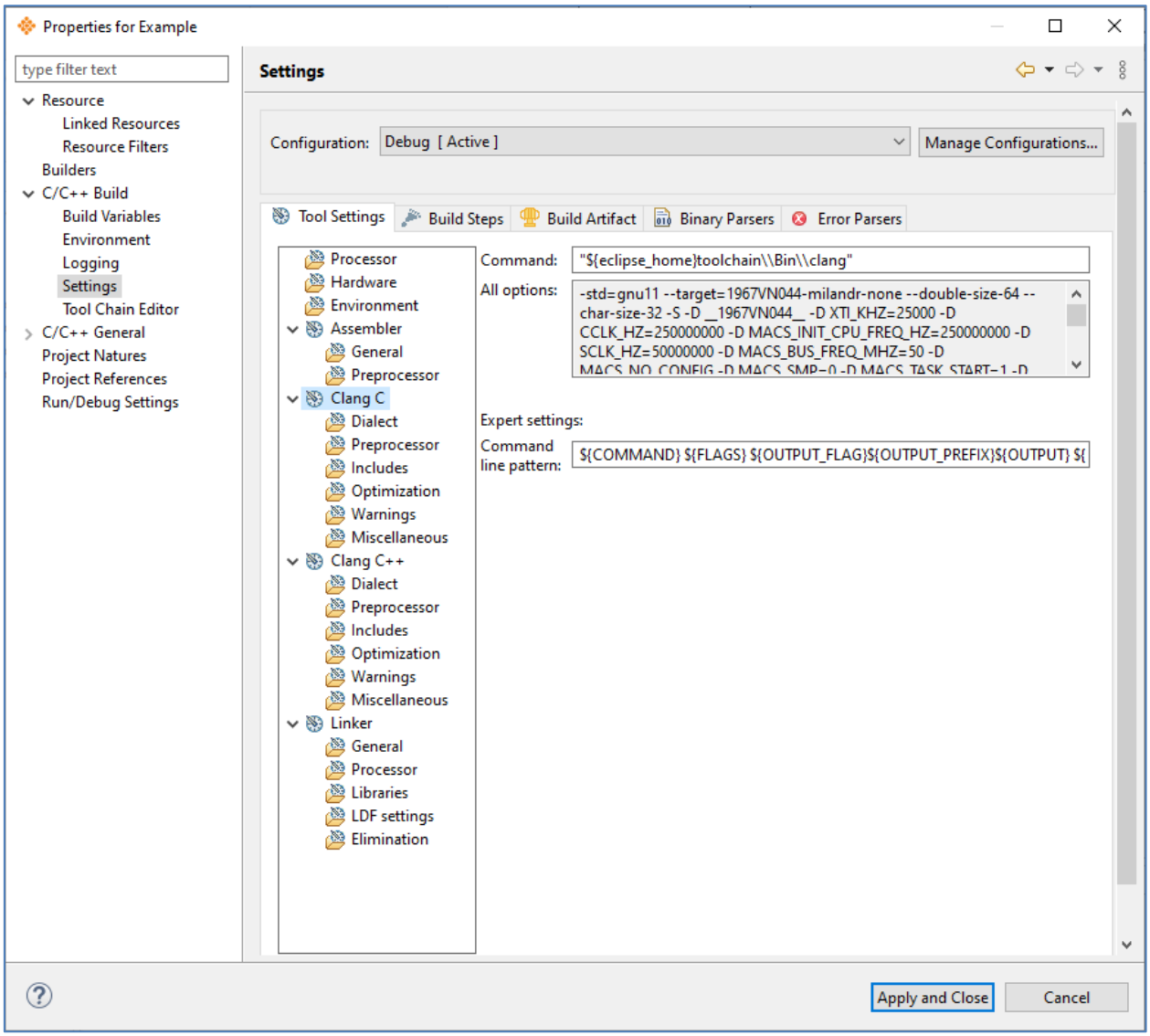


Рисунок 151. Панель «Clang C»

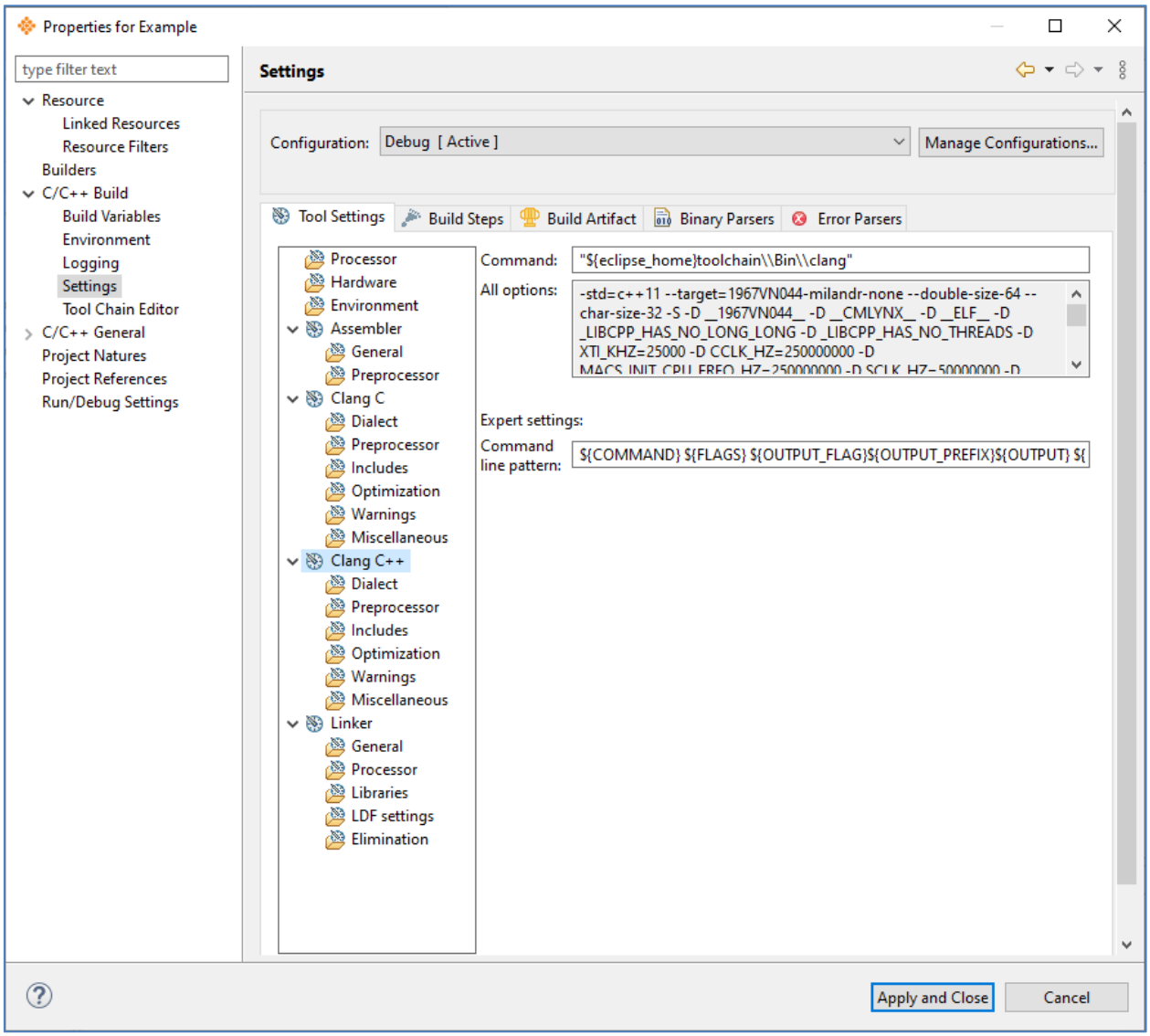


Рисунок 152. Панель «Clang C++»

Таблица 63. Элементы управления панели «Clang C» и «Clang C++»

Название элемента	Описание функционирования
Command	Строка вызова компилятора.
All options	Полный список используемых опций компилятора. Поддерживаемые опции компилятора описаны в документе ML_CLANG Руководство пользователя.
Command line pattern	Формат строки вызова компилятора с опциями.

6.9.2.4.1.5.1 Опции, специфичные для языка программирования

Опции, специфичные для языка программирования, задаются на панели «**Clang C→ Dialect**» и «**Clang C++→ Dialect**».

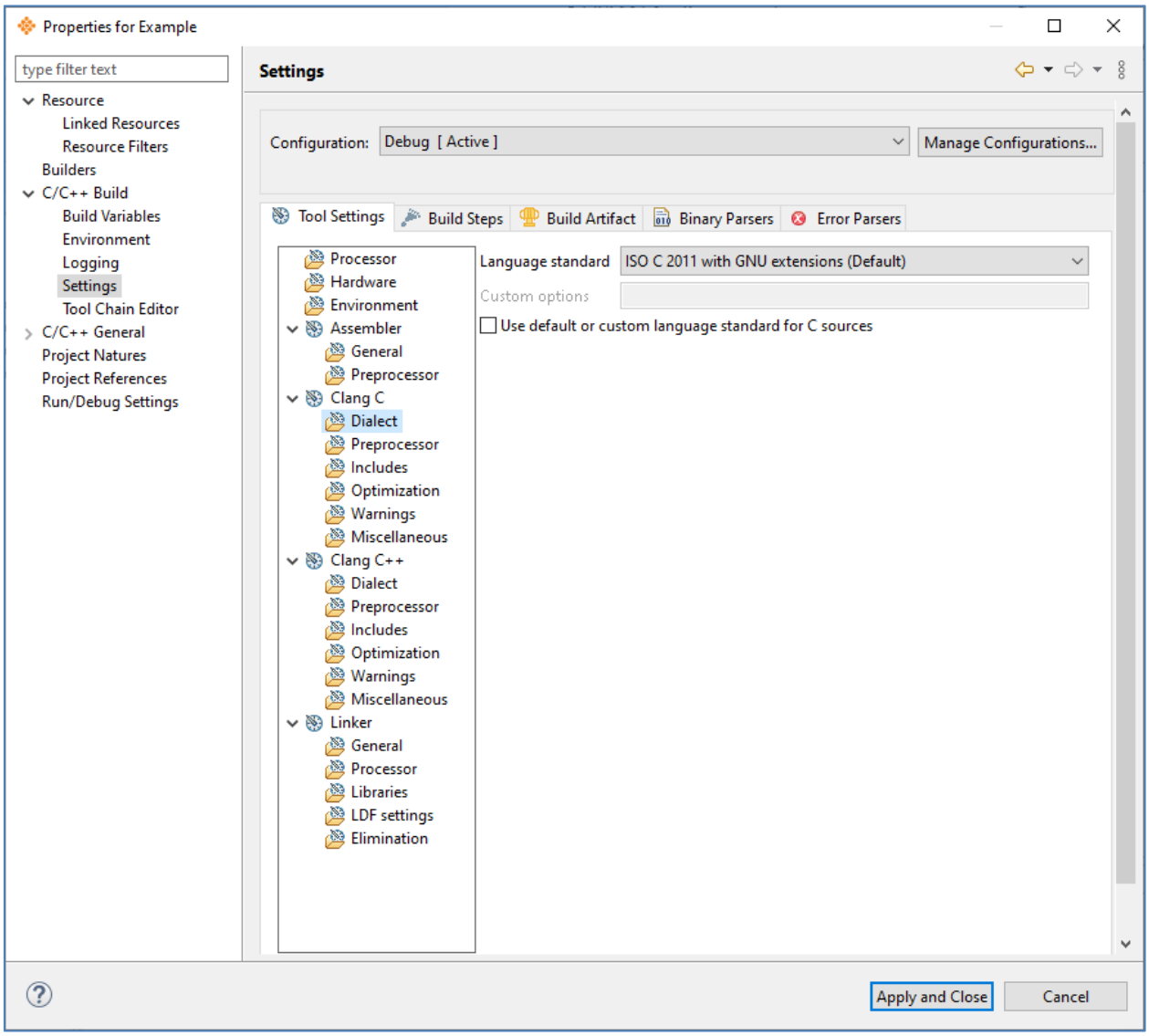


Рисунок 153. Панель «Clang C→ Dialect»

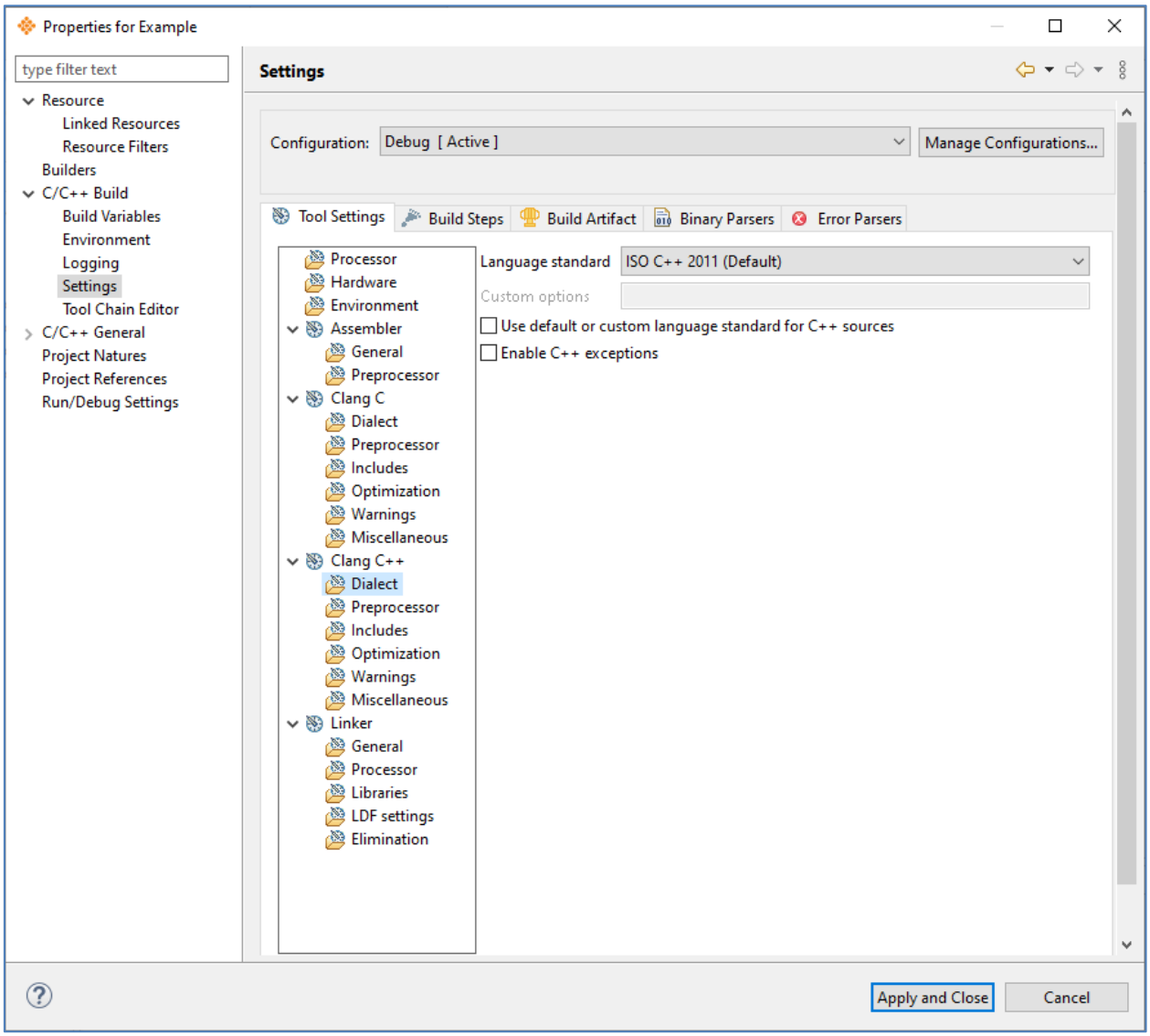


Рисунок 154. Панель «Clang C++→ Dialect»

Таблица 64. Элементы управления панели «Clang C→ Dialect» и «Clang C++→ Dialect»

Название элемента	Описание функционирования	По умолчанию
Language standard	Выбор языкового стандарта.	ISO C 2011 с расширениями GNU (для C) ISO C++ 2011 (для C++)
Custom options	Активна только при включённой опции «**Use default or custom language standard for C sources**»/ «**Use default or custom language standard for C++ sources».** Строка для передачи опций, специфичных для языка программирования (используемый язык, его версия, запрет использования механизма исключений и т.д.), компилятору Clang. Несколько опций в поле следует указывать через пробел.	Пустая

Название элемента	Описание функционирования	По умолчанию
Use default or custom language standard for C sources/ Use default or custom language standard for C++ sources	Включение/отключение применения настраиваемого стандарта или стандарта по умолчанию. При включенной опции становится доступна строка «**Custom options**».	Выкл
Enable C++ exceptions *(только для C++)*	Включить/отключить обработку исключений C++. При включении обработки исключений флаг «-fno-exceptions» не передается и подключаются библиотеки поддержки исключений libcxxabi.dlb, libunwind.dlb. Данный флаг также влияет на порядок подключения каталогов «**toolchain\LibLLVM**» и «**toolchain\LibLLVM\NoExcp**» в настройках компоновщика (см. п. 6.9.2.4.1.6.3).	Выкл

6.9.2.4.1.5.2 Настройки препроцессора компилятора

Настройки препроцессора компилятора задаются на панели «**Clang C → Preprocessor**» и на панели «**Clang C++ → Preprocessor**».

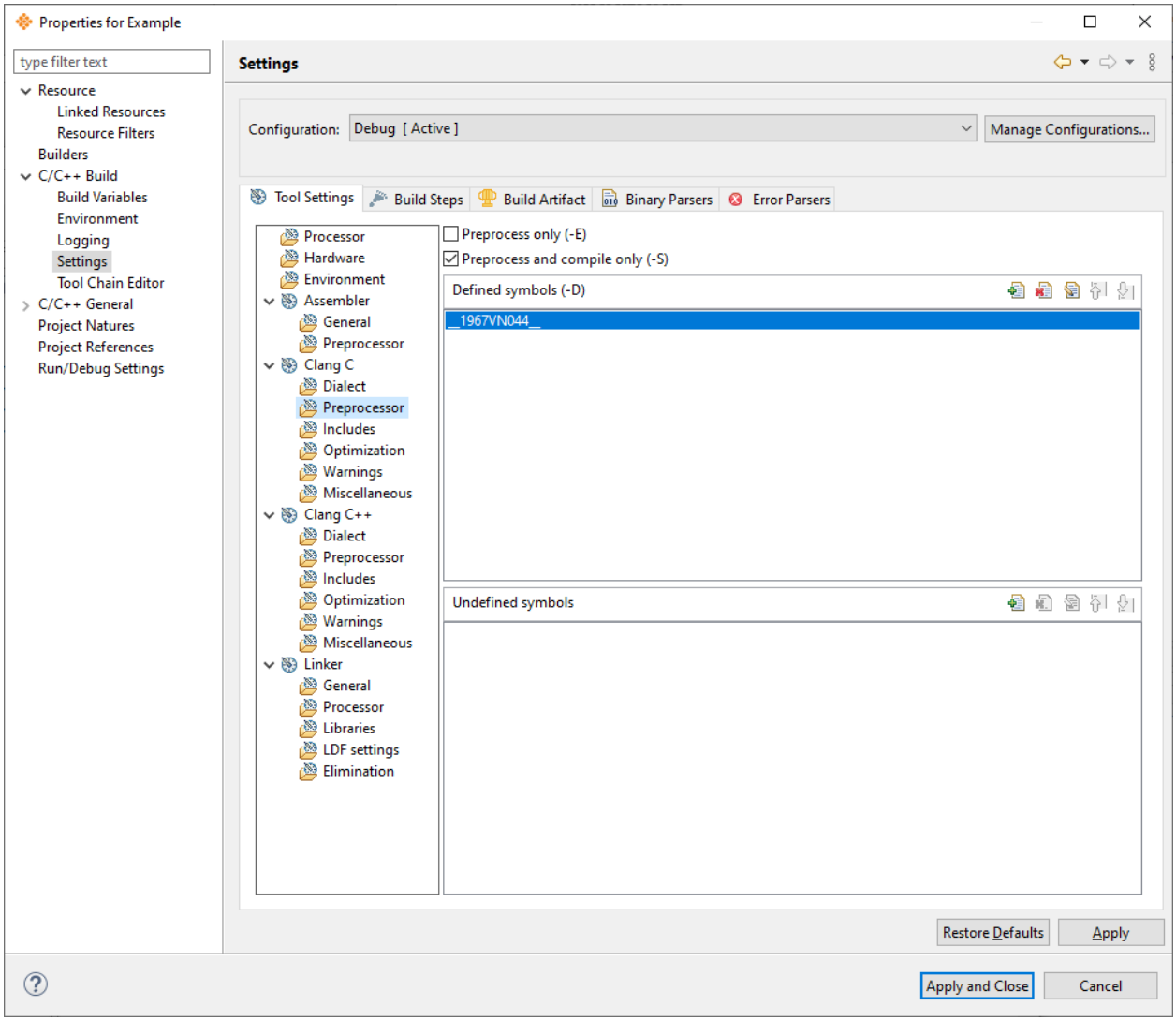


Рисунок 155. Панель «Clang C → Preprocessor»

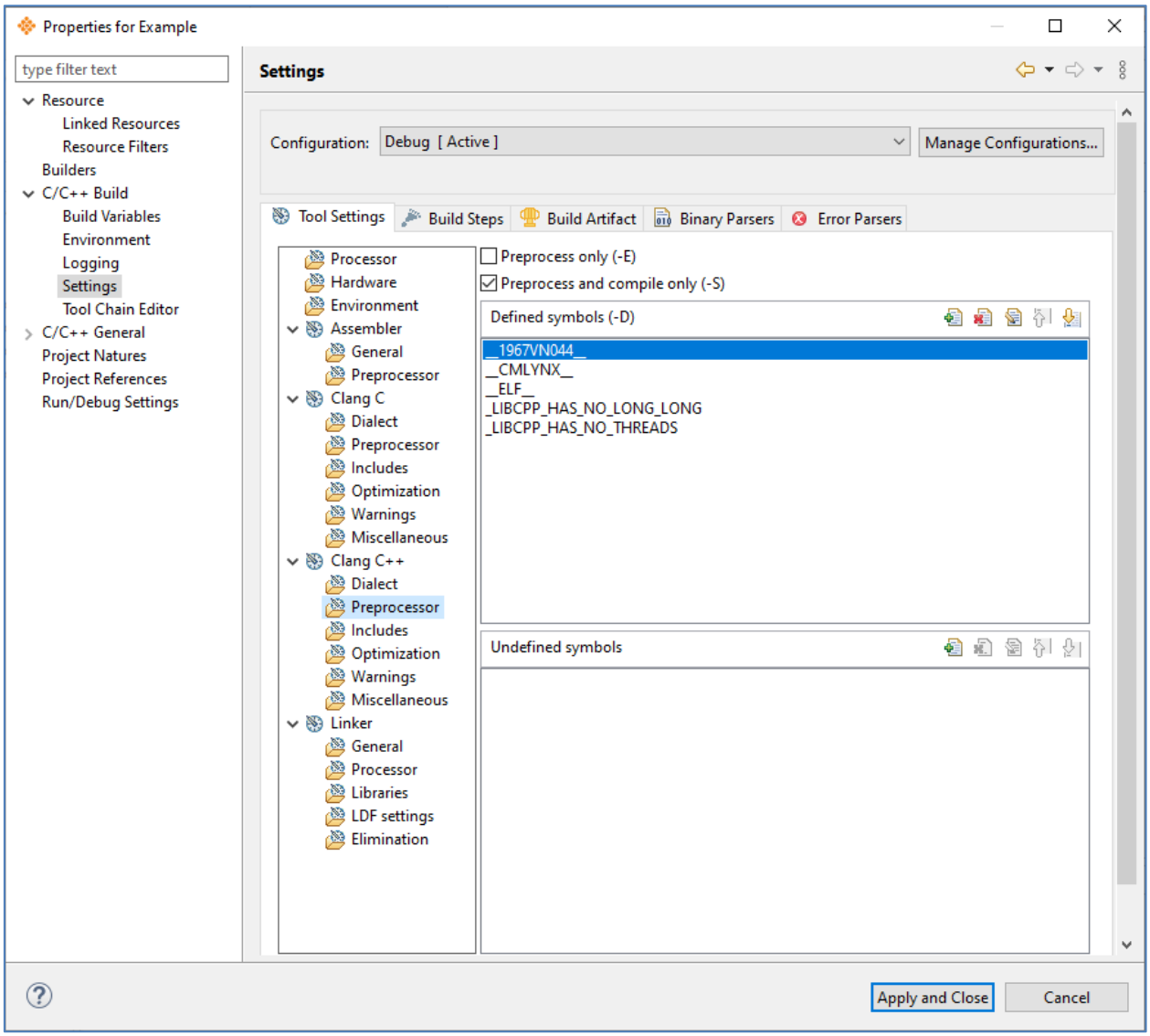


Рисунок 156. Панель «Clang C ++→ Preprocessor»

Таблица 65. Элементы управления панели «Clang C → Preprocessor» и «Clang C++→ Preprocessor»

Название элемента	Описание функционирования	По умолчанию
Preprocess only (-E)	Флаг включения режима препроцессинга (без компоновки).	Откл
Preprocess and compile only (-S)	Флаг включения режима препроцессинга и компиляции (без компоновки).	Вкл
Defined symbols (-D)	Список макросов с заданными значениями, который будет передан компилятору.	`__CMLYNX__`, `__ELF__`, `_LIBCPP_HAS_NO_LONG_LONG`, `_LIBCPP_HAS_NO_THREADS` (для C++)
Undefined symbols	Список файлов, которые будут подключены в начале транслируемого модуля до его синтаксического разбора.	Пустое

6.9.2.4.1.5.3 Список каталогов поиска include- файлов

Список каталогов поиска include- файлов задается на панелях «**Clang C → Includes**» и «**Clang C++ → Includes**».

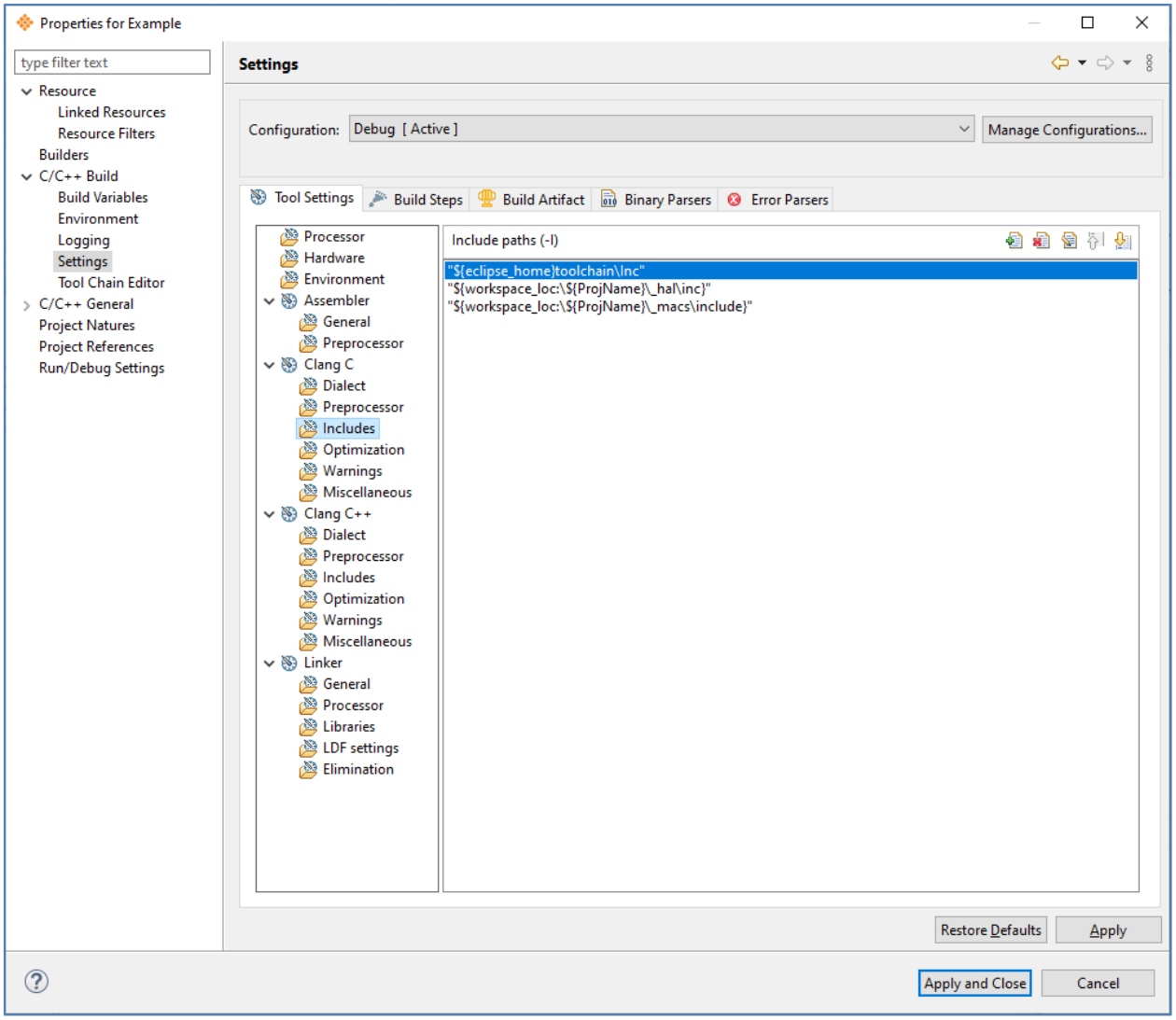


Рисунок 157. Панель « Clang C →Includes»

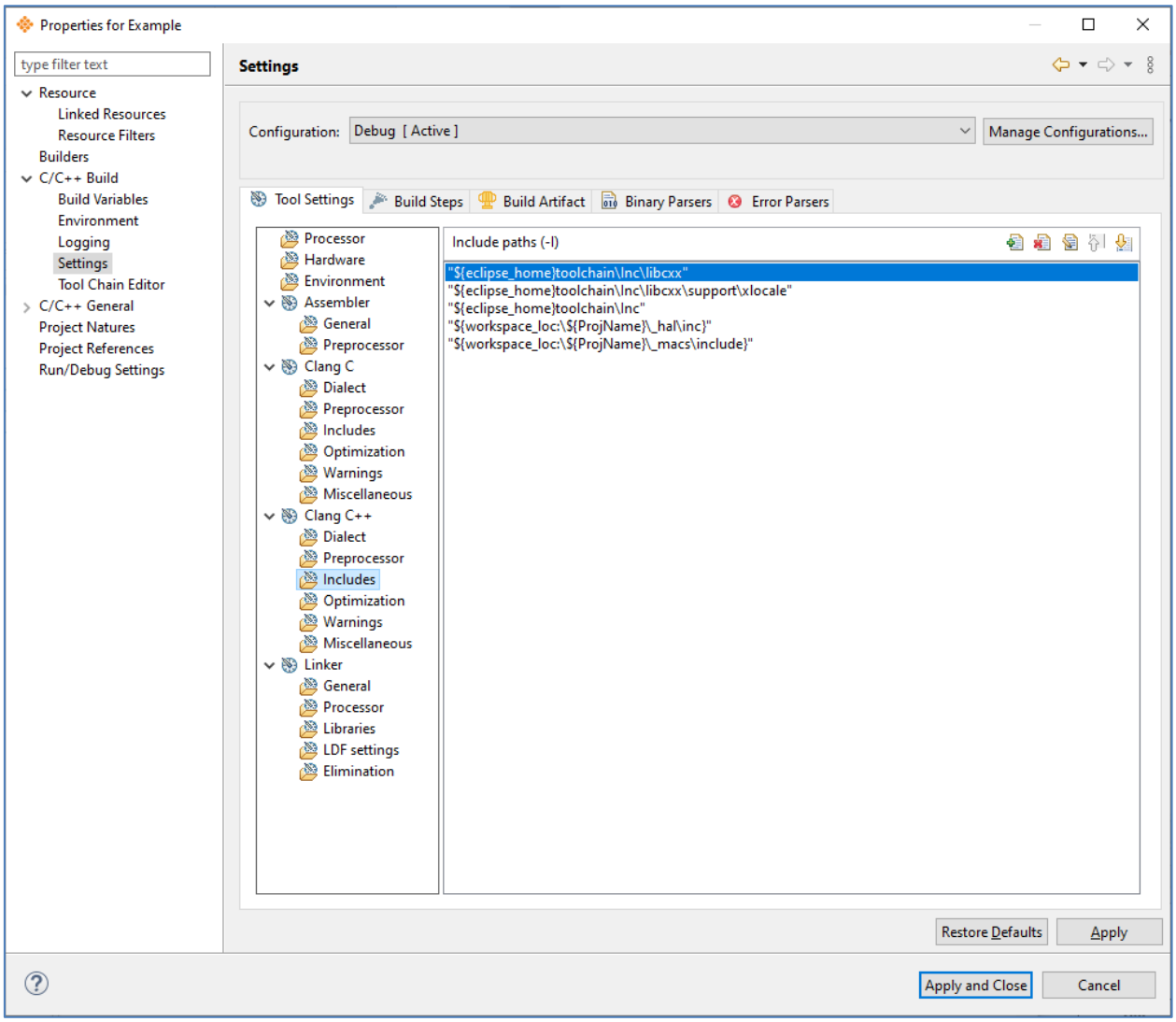


Рисунок 158. Панель « Clang C++ →Includes»

Таблица 66. Элементы управления панели «Clang C →Includes» и «Clang C++→Includes»

Название элемента	Описание функционирования	По умолчанию
Include paths (-I)	Список каталогов, в которых будет производиться поиск include-файлов.	-

6.9.2.4.1.5.4 Настройки оптимизации

Настройки оптимизации задаются на панелях «**Clang C →Optimization**» и «**Clang C++ → Optimization**».

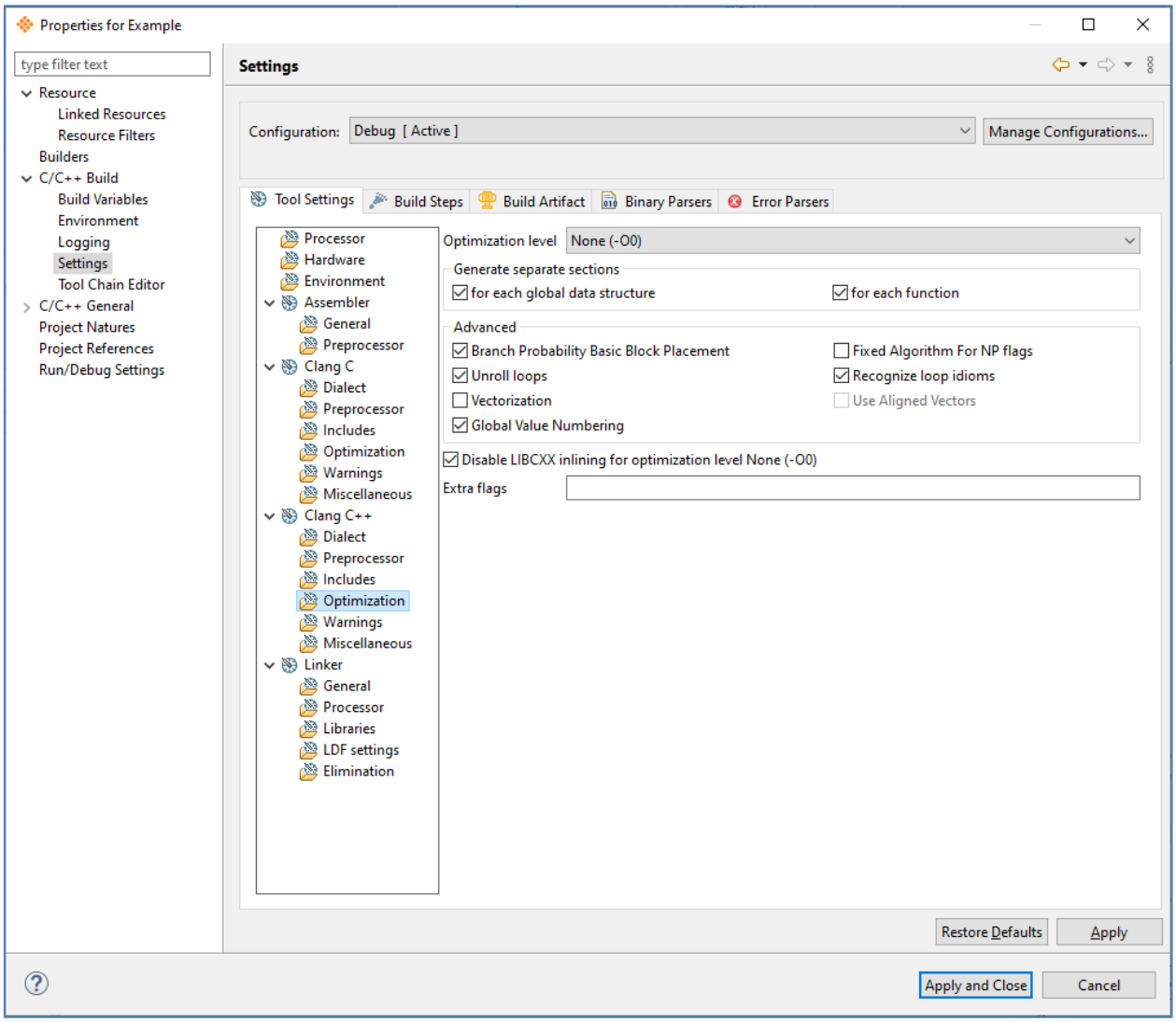


Рисунок 159. Панель «Clang C+ → Optimization»

Таблица 67. Элементы управления панели «Clang C → Optimization» и «Clang C++ → Optimization»

Название элемента	Описание функционирования	По умолчанию
Optimization level	Задает уровень оптимизации. Возможные значения: ▪ **None (-O0)** – без оптимизаций; ▪ **Optimize (-O1)** – уровень оптимизации O1 (минимальный уровень с сохранением отладочной информации); ▪ **Optimize more (-O2)** – уровень оптимизации O2 (оптимальный с акцентом на быстродействие); ▪ **Optimize most (-O3)** – уровень оптимизации O3 (с максимальным быстродействием); ▪ **Optimize size (-Os)** – уровень оптимизации -Os (оптимальный с акцентом на уменьшение размера кода); ▪ **Optimize size most (-Oz)** – уровень оптимизации -Oz (с максимальным уменьшением размера кода). Примечание: Уровень оптимизации может быть задан в том числе для отдельных файлов, см. п. 6.9.2.4.1.8	None (-O0)
Generate separate section for each function	Флаг для включения/отключения режима генерации отдельной секции для каждой функции. Настройка соответствует опции компилятора `-ffunction-sections`	Включен
Generate separate section for each global data structure	Флаг для включения/отключения режима генерации отдельной секции для каждой глобальной структуры данных. Настройка соответствует опции компилятора `-fdata-sections`.	Включен
Advanced		
Branch Probability Basic Block Placement	(для опытных пользователей) Флаг для включения/отключения сценария, когда базовые блоки, которые с наибольшей вероятностью будут выполняться последовательно, располагаются рядом.	Включен
Fixed Algorithm For NP flags	(для опытных пользователей) Флаг для включения/отключения сценария, когда выполнятся расстановка флагов NP у инструкций перехода без предсказания вероятностей.	Выключен
Unroll loops	(для опытных пользователей) Флаг для включения/отключения сценария, когда выполняется развёртывание циклов в линейную последовательность инструкций	Включен
Recognize loop idioms	(для опытных пользователей) Флаг для включения/отключения сценария, когда распознаются идиомы для преобразования простых циклов в ациклическую форму	Включен

Название элемента	Описание функционирования	По умолчанию
Vectorization	(для опытных пользователей) Флаг для включения/отключения сценария векторизации, т.е. при операции над данными в циклах используются команды для групп регистров вместо одиночных.	Выключен
Use Aligned Vectors	(для опытных пользователей) Флаг для включения/отключения сценария выравнивания массивов данных для более эффективной векторизации. Активна только при включенном флаге Vectorization.	Выключен
Global Value Numbering	(для опытных пользователей) Флаг для включения/отключения сценария нумерации глобальных значений, чтобы избежать избыточных инструкций.	Включен
Disable LIBCXX inlining for optimization level None (-O0)	(для опытных пользователей) Флаг для включения/отключения встраивания inline-функций библиотеки libcxx, если выбрано значение уровня оптимизации «*None (-O0)*».	Включен
Extra flags	Поле для задания прочих опций компилятора (полный список опций компилятора представлен в руководстве пользователя на Clang).	Пустое

Примечания:

1. В некоторых случаях опция -Os способна давать более компактный код, чем -Oz. Поэтому для получения минимально возможного размера кода пользователю следует попробовать обе эти опции, определив оптимальную экспериментальным путём.

2. Текущие настройки категории «Advanced» по умолчанию являются наиболее оптимальными. Однако, в ряде случаев, текущие настройки могут негативно сказываться на размере или быстродействии кода. В этом случае можно попробовать изменить текущие настройки, подобрав оптимальное сочетание экспериментальным путём.

6.9.2.4.1.5.5 Настройки предупреждений

Настройки предупреждений задаются на панелях «**Clang C → Warnings**» и «**Clang C++ → Warnings**».

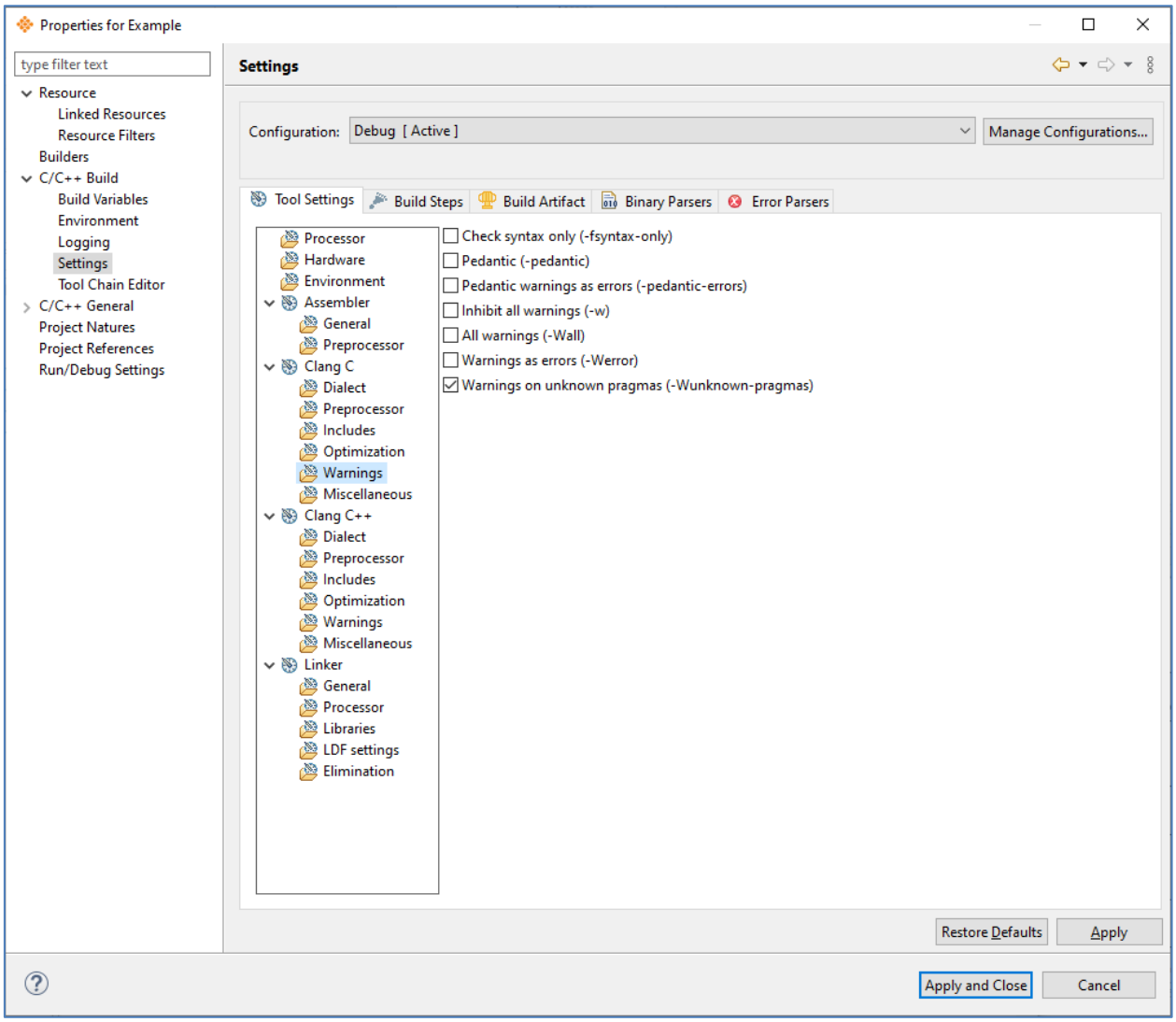


Рисунок 160. Панель «Clang C → Warnings»

Таблица 68. Элементы управления панели «Clang C → Warnings» и «Clang C++ → Warnings»

Название элемента	Описание функционирования	По умолчанию
Check syntax only (-fsyntax-only)	Флаг включения режима «fsyntax-only» (предупреждения выдаются только на программный код, но не выдаются на выявленные на этапе оптимизации ошибки).	Выкл
Pedantic (-pedantic)	Флаг включения выдачи предупреждений на любые отклонения от стандарта ANSI.	Выкл
Pedantic warnings as errors (-pedantic-errors)	Флаг включения выдачи ошибок на любые отклонения от стандарта ANSI.	Выкл
Inhibit all warnings (-w)	Флаг отключения диагностики (предупреждения не выдаются).	Выкл
All warnings (-Wall)	Флаг включения вывода всех предупреждений.	Выкл
Warnings as errors (-Werror)	Флаг включения предупреждений в ошибки (предупреждения выводятся как ошибки).	Выкл

Название элемента	Описание функционирования	По умолчанию
Warnings on unknown pragmas (-Wunknown-pragmas)	Флаг включения выдачи предупреждений на не поддерживаемые компилятором pragma.	Вкл

6.9.2.4.1.5.6 Прочие настройки

Прочие настройки задаются на панелях «**Clang C → Miscellaneous**» и «**Clang C++ → Miscellaneous**».

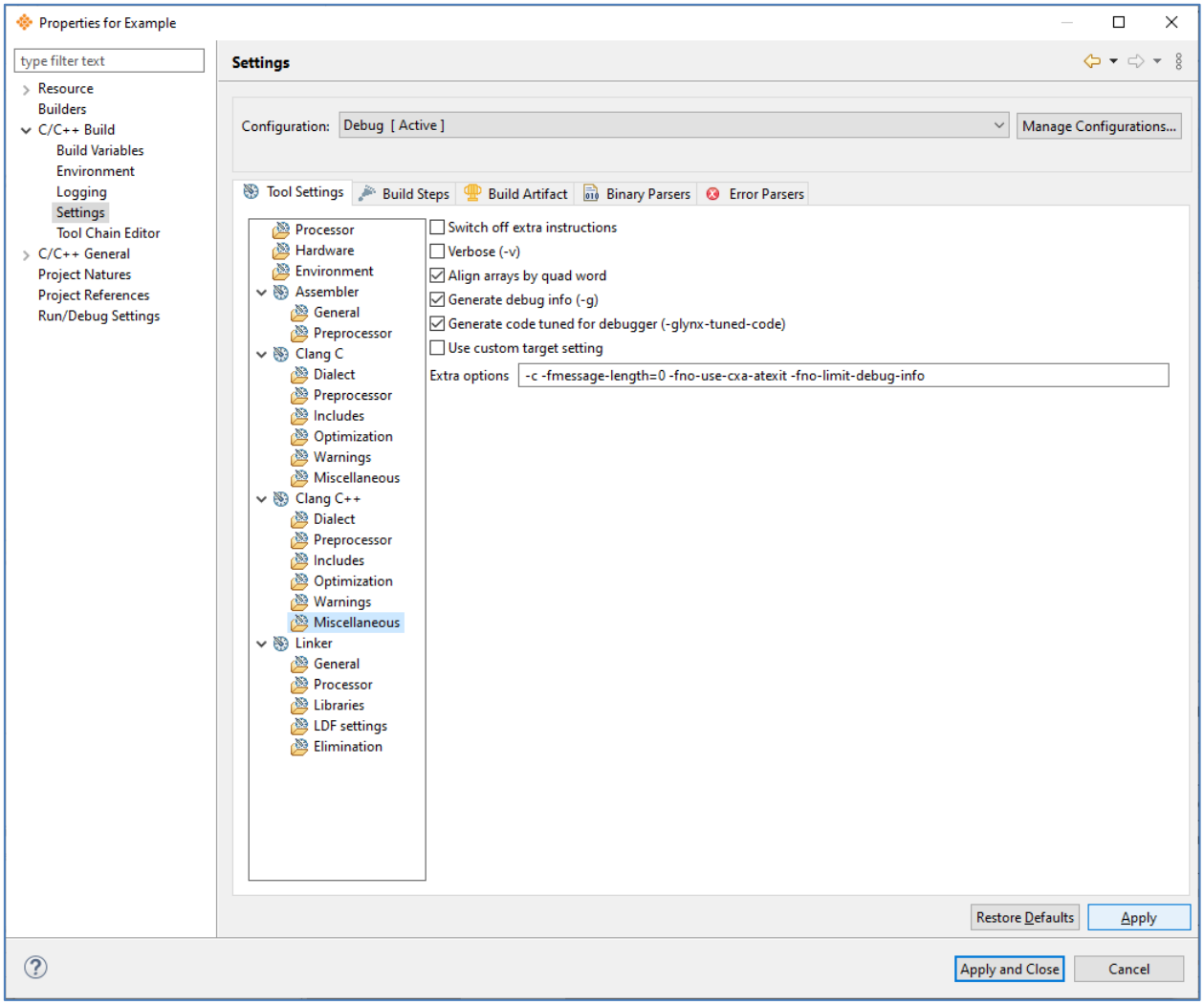


Рисунок 161. Панель « Clang C ++ → Miscellaneous»

Таблица 69. Элементы управления панели «Clang C → Miscellaneous» и «Clang C++ → Miscellaneous»

Название элемента	Описание функционирования	По умолчанию
Switch off extra instructions	Флаг отключения всех инструкций категории extra.	Выкл
Verbose (-v)	Флаг установки режима печати подробной информации во время работы.	Выкл

Название элемента	Описание функционирования	По умолчанию
Align arrays by quad word	Флаг включения режима выравнивания массивов по границе квадрослова.	Вкл
Generate debug info (-g)	Флаг включения формирования отладочной информации, необходимой для отладки по исходному тексту программы.	Вкл
Generate code tuned for debugger (-glynx-tuned-code)	Флаг включения формирования кода, специализированного для отладки, но менее оптимального. Имеет смысл только при формировании отладочной информации.	Выкл.
Use custom target settings	Флаг отключения автоматической генерации параметра командной строки компилятора «`--target=`» на основании выбранного процессора. Пользовательскую установку «`--target=`» можно задать в поле «**Extra options**».	Выкл
Extra options	Строка для ввода прочих ключей компилятора. Ключи указываются в строку через пробел. Поддерживаемые ключи компилятора описаны в документе ML_CLANG Руководство пользователя.	-

6.9.2.4.1.6 Компоновщик

На панели **«Linker»** (см. Рисунок 162) доступны для просмотра параметры вызова компоновщика и список, передаваемых ему опций.

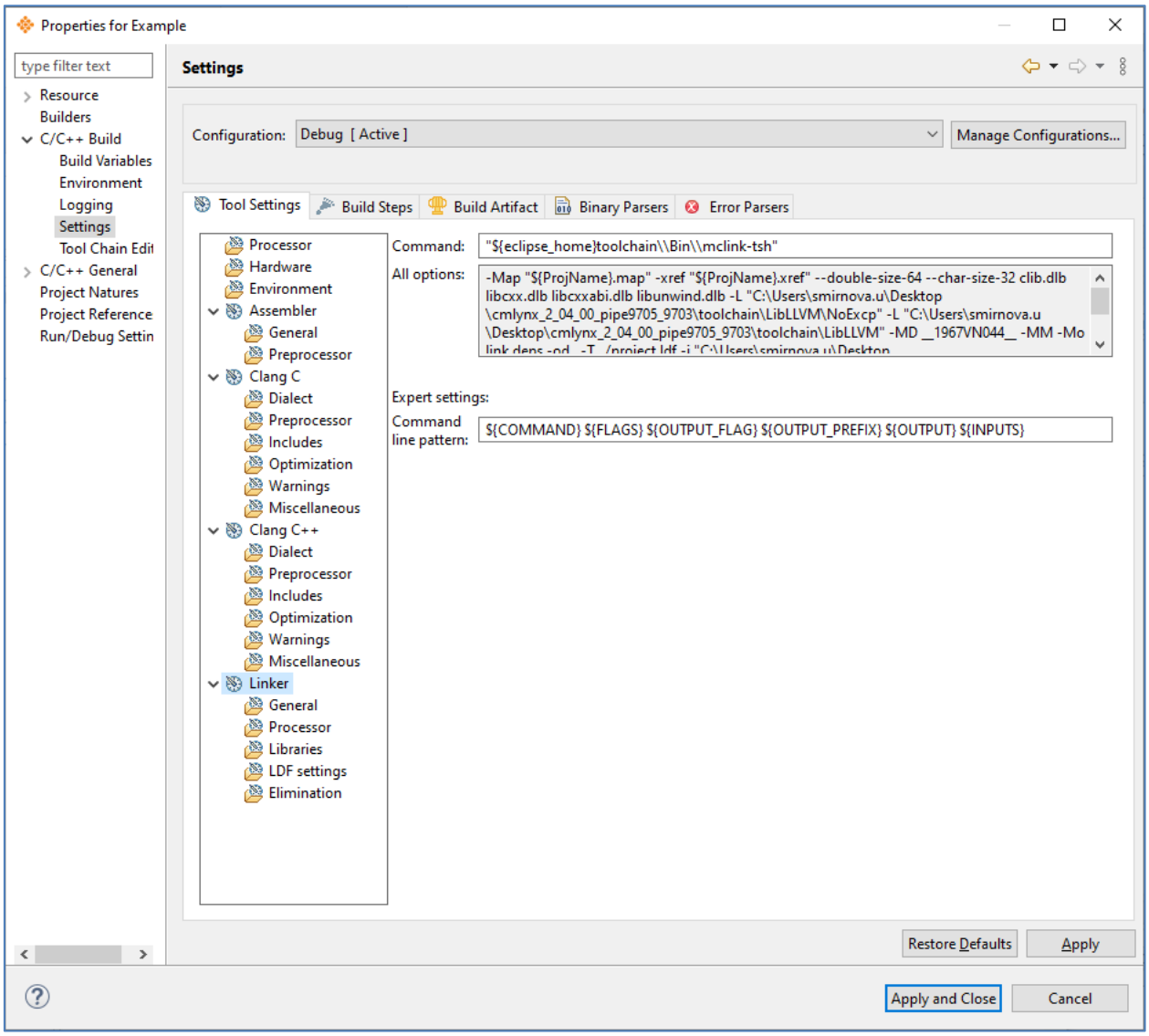


Рисунок 162. Панель «Linker»

Таблица 70. Элементы управления панели «Linker»

Название элемента	Описание функционирования
Command	Строка вызова компоновщика.
All options	Полный список используемых опций компоновщика. Поддерживаемые опции компоновщика описаны в документе ML_MCLINK-TSH Руководство пользователя.
Command line pattern	Формат строки вызова компоновщика с опциями.

6.9.2.4.1.6.1 Основные настройки

На панели «**Linker → General**» (см. Рисунок 163) настраиваются основные параметры компоновщика (см. Таблица 71).

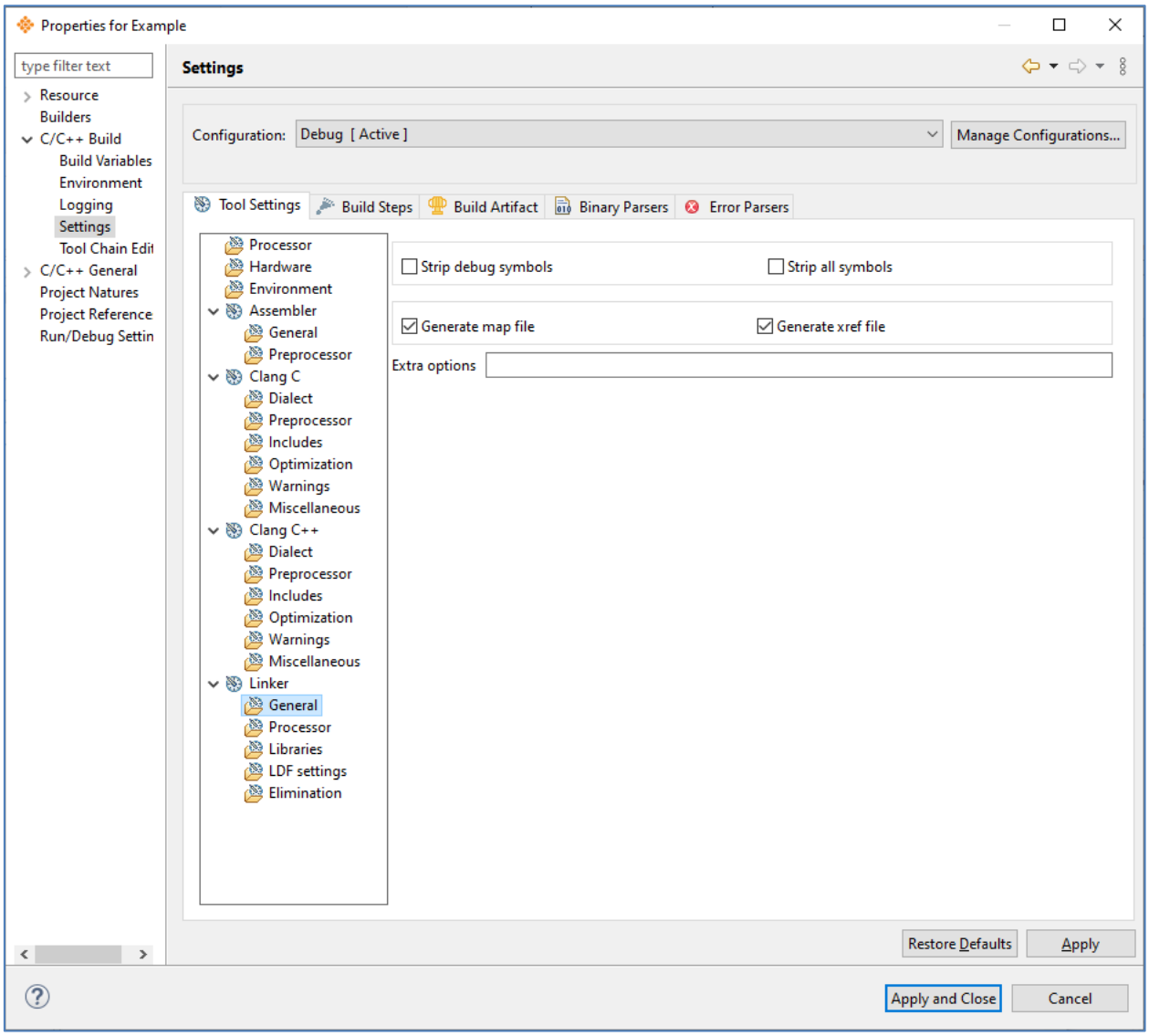


Рисунок 163. Панель «Linker → General»

Таблица 71. Элементы управления панели «Linker → General»

Название элемента	Описание функционирования	По умолчанию
Strip debug symbols	Не помещать отладочную информацию в генерируемый исполняемый файл.	Выкл
Strip all symbols	Не помещать символьную информацию в генерируемый исполняемый файл.	Выкл
Generate map file	Создать MAP файл (отчёт компоновщика).	Вкл
Generate xref file	Создать файл перекрёстных ссылок.	Вкл
Extra options	Указать дополнительные, не поддерживаемые на уровне графического интерфейса, опции компоновщика. Поддерживаемые опции компоновщика описаны в документе ML_MCLINK-TSH Руководство пользователя.	–

6.9.2.4.1.6.2 Настройки размеров типов данных

На панели «**Linker → Processor**» (см. Рисунок 164) можно задать размер типа данных, отличный от используемого при компиляции (см. Таблица 73).

Примечание: по умолчанию настройки размеров типов данных компоновщика совпадают с настройками размеров типов данных компилятора. Изменять эти настройки без особой надобности не рекомендуется. Однако настройки размеров типов данных компоновщика позволяет более гибко управлять процессом компоновки файлов с несовпадающими типами данных. Например, при компоновке проекта, содержащего собранную в режиме «Any» библиотеку, можно, задав режим char size=char 8 при компоновке, ограничить ее использование только реализацией совместимых с режимом char size=char 8 функций.

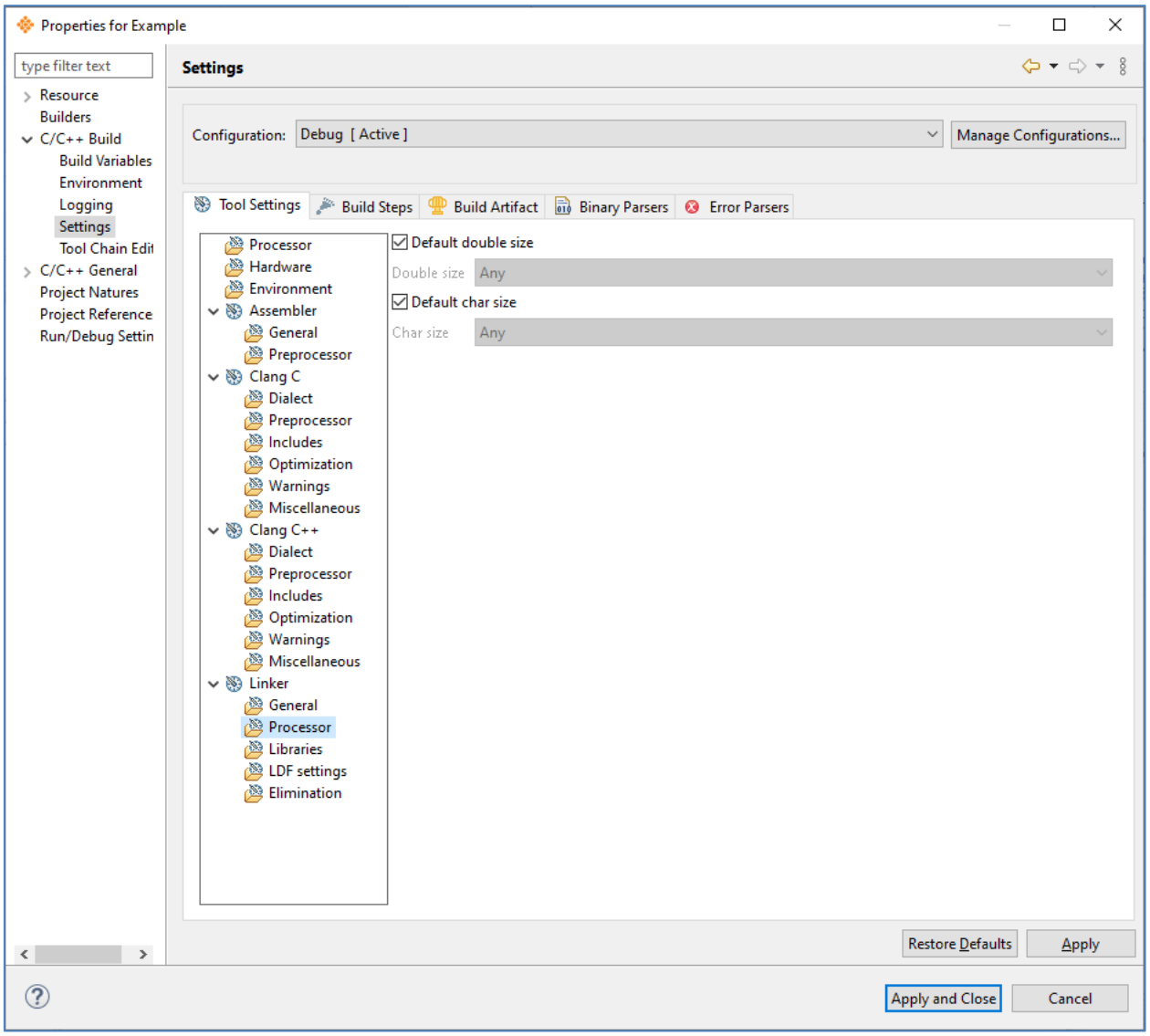


Рисунок 164. Панель «Linker → Processor»

Таблица 72. Элементы управления панели «Linker → Processor»

Название элемента	Описание функционирования
Default double size	Использовать размер типа `double`, заданный по умолчанию для всей конфигурации или выбрать другой в списке «**Double size**».

Название элемента	Описание функционирования
Double size	Выбрать размер типа **`double`**. Доступные варианты: ▪ **Any;** ▪ **32 бита** ▪ **64 бита**.
Default char size	Использовать размер типа **`char`**, заданный по умолчанию для всей конфигурации или выбрать другой в списке «**Char size**».
Char size	Выбрать размер типа **`char`**. Доступные варианты: ▪ **Any;** ▪ **32 бита** ▪ **64 бита**.

6.9.2.4.1.6.3 Подключаемые библиотеки

На панели «**Linker → Libraries**» (см. Рисунок 165) настраиваются подключаемые библиотеки и пути к ним (см. Таблица 73).

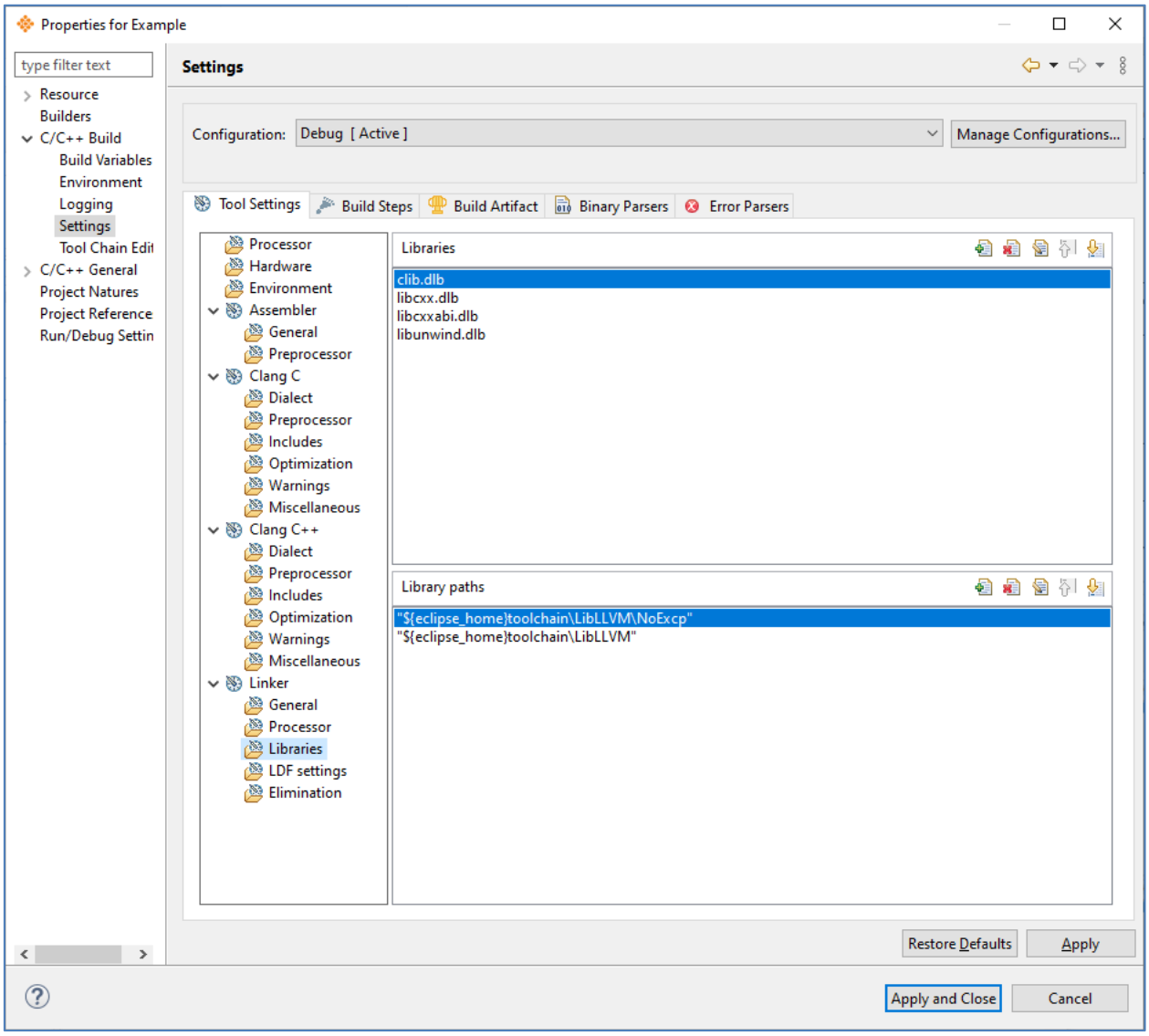


Рисунок 165. Панель «Linker → Libraries»

Таблица 73. Элементы управления панели «Linker → Libraries»

Название элемента	Описание функционирования
Libraries	Список подключаемых к проекту библиотечных файлов.
Library paths	Список каталогов, в которых будет производиться поиск библиотечных файлов в указанном порядке (сверху вниз).

6.9.2.4.1.6.4 Настройки LDF

На панели «**Linker → LDF settings**» задается путь к LDF-файлу, список каталогов с include-файлов и список макросов.

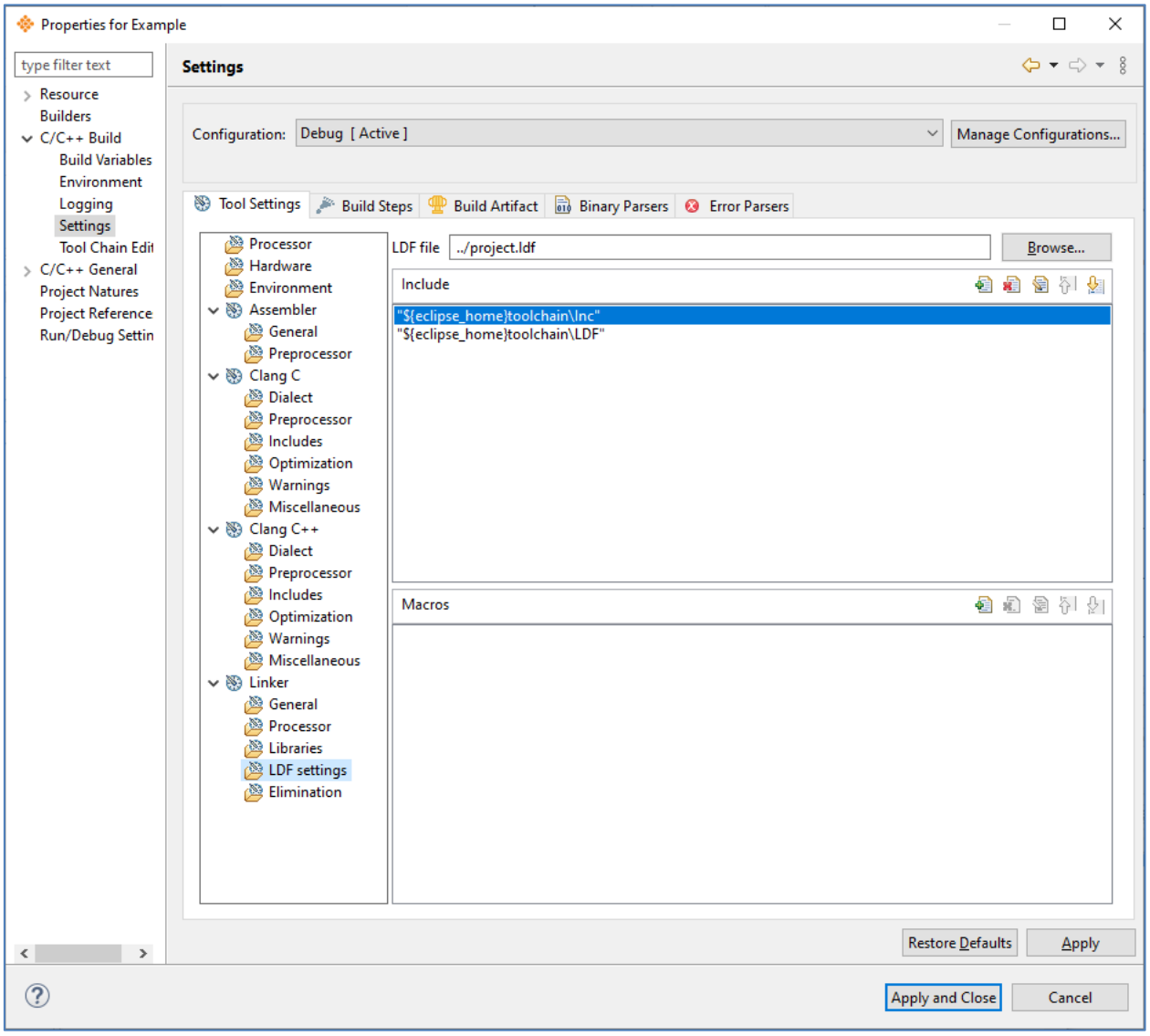


Рисунок 166. Панель «Linker → LDF settings»

Таблица 74. Элементы управления панели «Linker → LDF settings»

Название элемента	Описание функционирования
LDF file	Путь к LDF-файлу. **Примечание.** Если выбран процессор 1967ВН058 и заданы функции управления памятью процессора (включен флаг «Default Memory management functions»), то выбор пути к LDF-файлу недоступен, подставляется LDF-файл в завимости от заданных настроек.
Include	Список каталогов, в которых будет производиться поиск include-файлов.
Macros	Список макросов, которые будут передаваться компоновщику в командной строке.

6.9.2.4.1.6.5 Настройки исключения неиспользуемого кода

На панели «**Linker → Elimination**» задаются настройки исключения неиспользуемого кода.

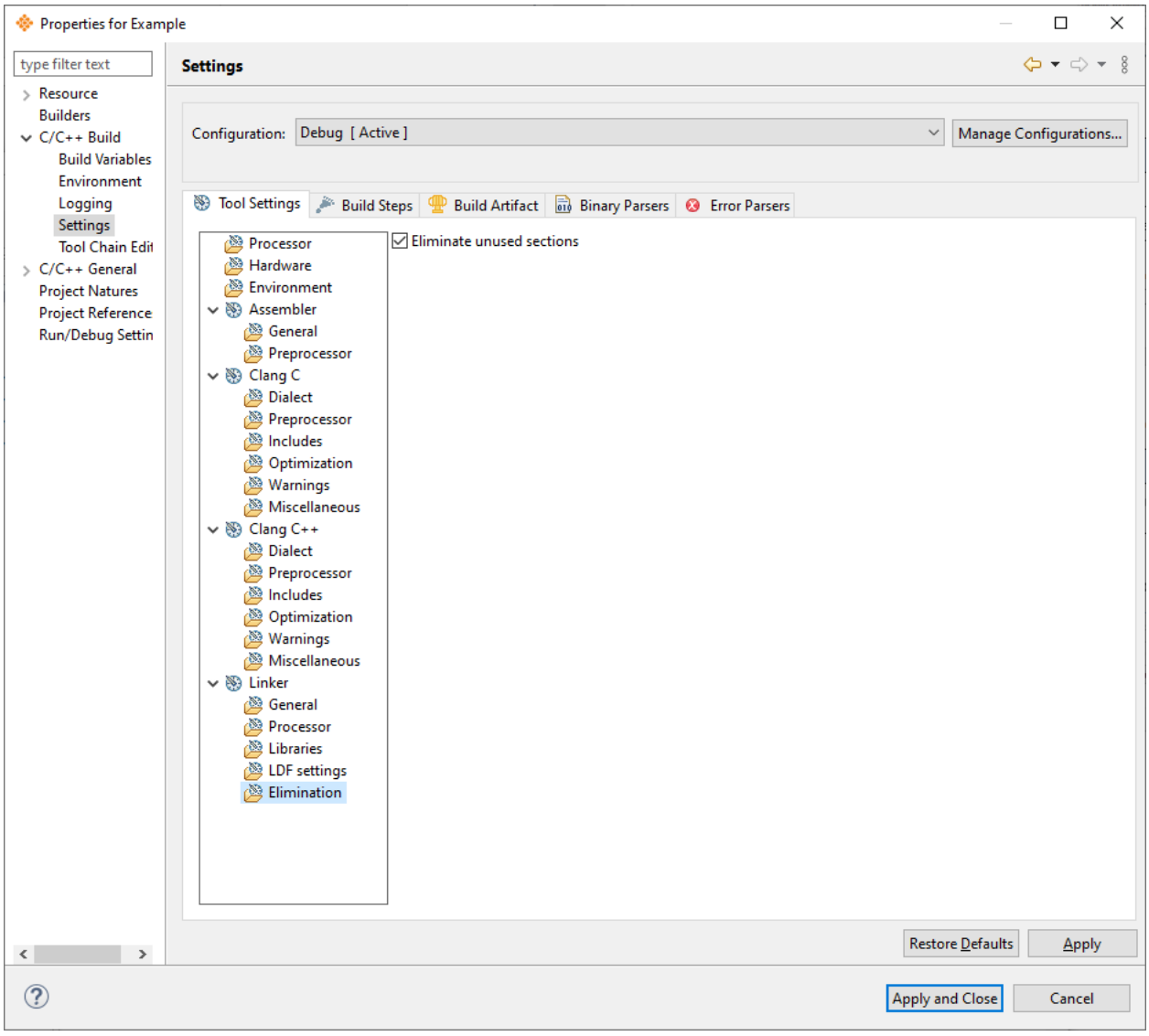


Рисунок 167. Панель «Linker → Elimination»

Таблица 75. Элементы управления панели «Linker → Elimination»

Название элемента	Описание функционирования	По умолчанию
Eliminate unused section	Флаг для включения/отключения режима исключения неиспользуемых секций. Соответствует флагу компоновщика `--gc-sections`. При совместном использовании с настройками компилятора «**Generate separate section for each function**» и «**Generate separate section for each global data structure**» обеспечивает исключение неиспользуемого кода при компоновке.	Включен

6.9.2.4.1.7 Компоновщик библиотек

На панели **«Librarian»** (см. Рисунок 168) доступны для просмотра параметры вызова компоновщика библиотек и список передаваемых ему опций. Панель доступна только для проектов статических библиотек вместо панели **«Linker»**.

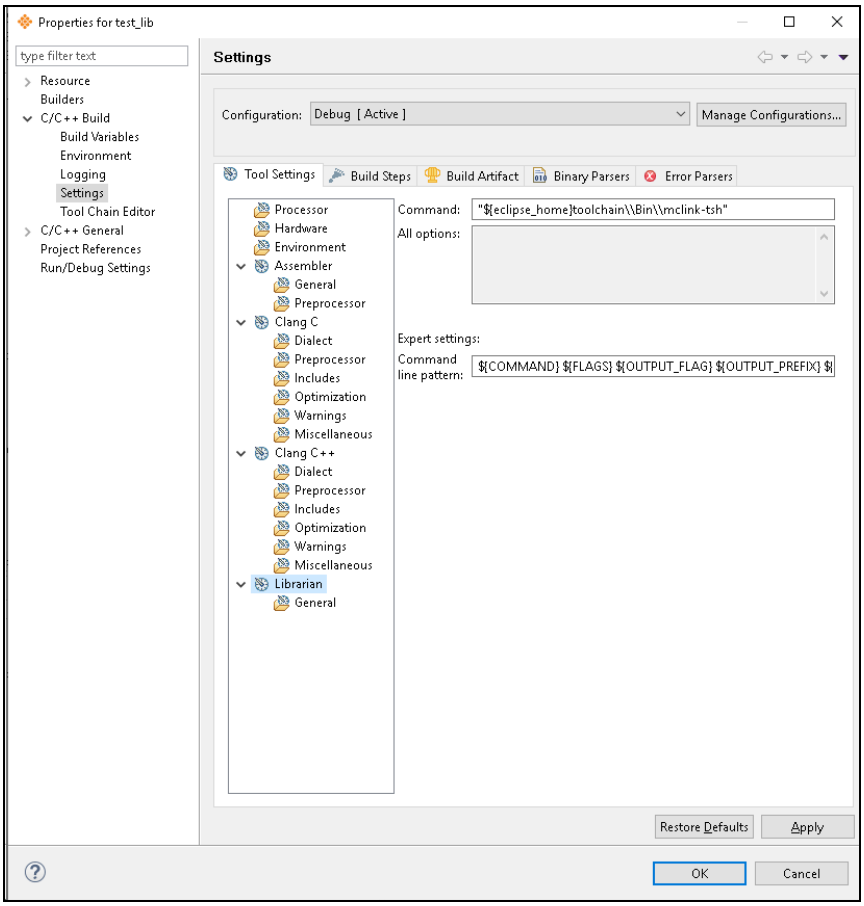


Рисунок 168. Панель «Librarian»

Таблица 76. Элементы управления панели «Librarian»

Название элемента	Описание функционирования
Command	Строка вызова компоновщика библиотек.
All options	Полный список используемых опций компоновщика библиотек. Поддерживаемые опции компоновщика описаны в документе ML_MCLINK-TSH Руководство пользователя.
Command line pattern	Формат строки вызова компоновщика библиотек с опциями.

6.9.2.4.1.7.1 Основные настройки

На панели «**Librarian → General**» (см. Рисунок 169) настраиваются основные параметры компоновщика (см. Таблица 77).

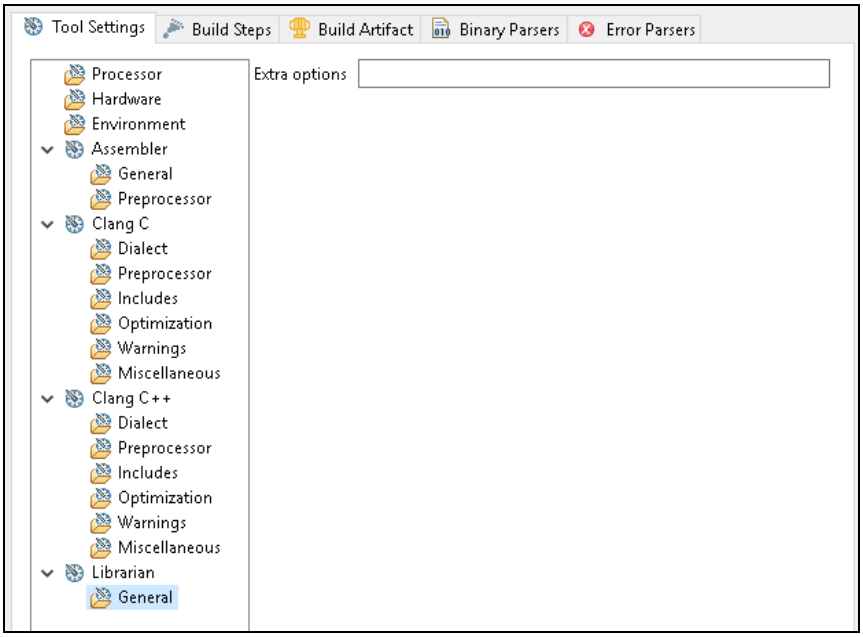


Рисунок 169. Панель «Librarian → General»

Таблица 77. Элементы управления панели «Librarian → General»

Название элемента	Описание функционирования	По умолчанию
Extra options	Дополнительные опции компоновщика библиотек. Поддерживаемые опции компоновщика описаны в документе ML_MCLINK-TSH Руководство пользователя.	–

6.9.2.4.1.8 Настройки компонентов тулчейн на уровне отдельных ресурсов

Пользователь может задать индивидуальные настройки для отдельной папки или файла, отличные от общих настроек проекта. При сборке или отладке данного ресурса будут использованы параметры, заданные отдельно для него. Например, можно задать размеры типов char и double или уровень оптимизации, отличные от общих настроек проекта.

Для этого необходимо:

1. Выбрать нужную папку или файл в дереве проекта и в контекстном меню выбрать пункт «**Properties**».

2. В окне свойств (см. Рисунок 170, Рисунок 171) выбрать нужную конфигурацию.

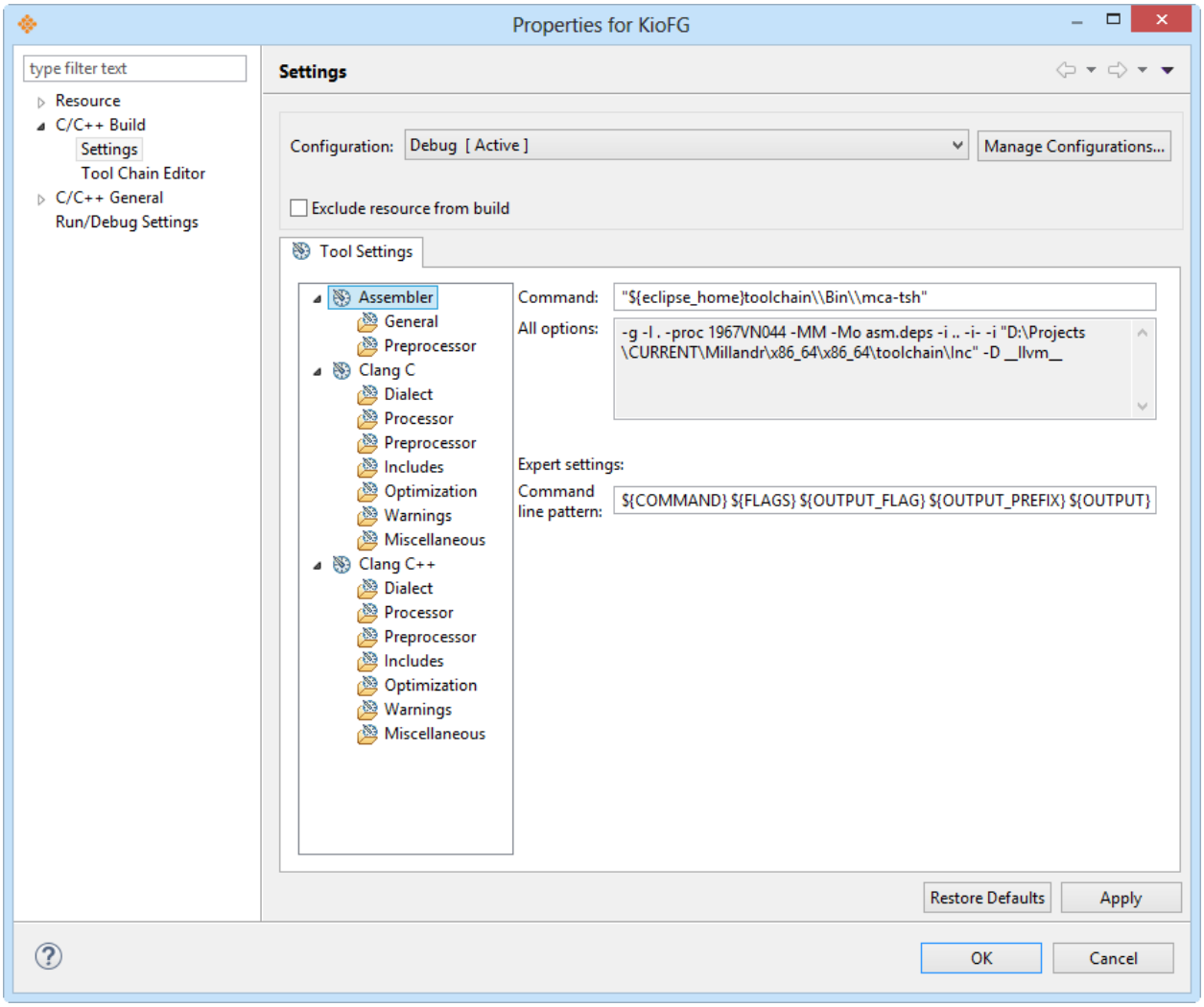


Рисунок 170. Окно свойств для папки

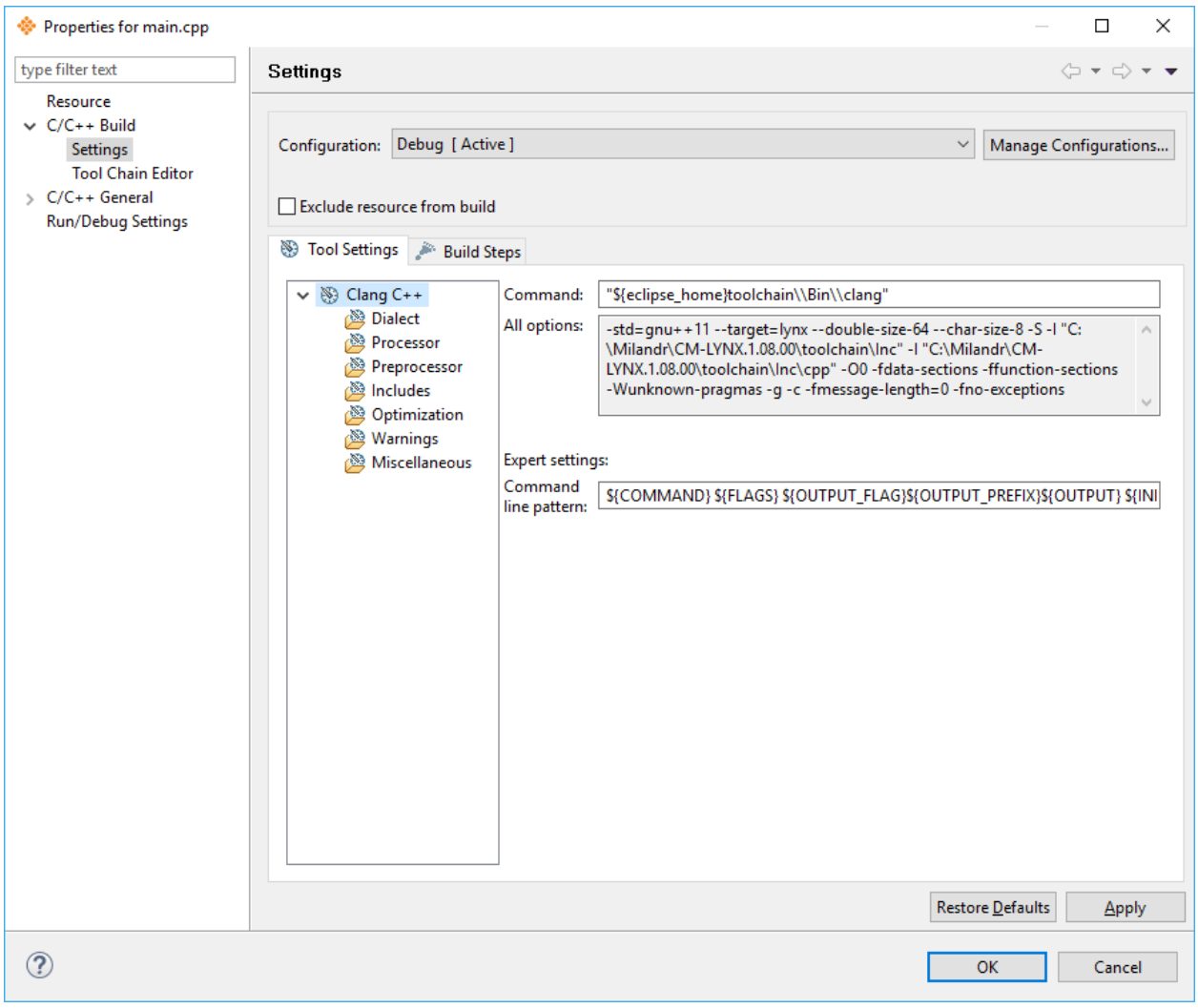


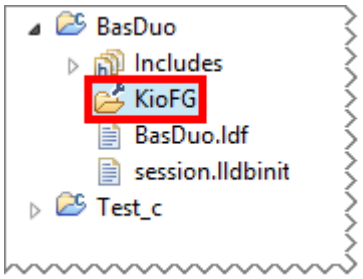
Рисунок 171. Окно свойств для файла *.cpp

3. Изменить необходимые настройки по аналогии с настройками проекта. Подробнее см. п. 6.9.2.4.1 Настройки компонентов тулчейн.

   Для изменения размеров типов char и double необходимо внести изменения на вкладке «**Clang C++->Processor**»/ «**Clang C++->Processor**» и нажать кнопку «**Apply**».

4. Нажать кнопку «**OK**».

   После сохранения настроек около ресурса в дереве проекта появится значок ключа:



Для сброса параметров необходимо воспользоваться одним из способов:

- в окне свойств ресурса нажать «**Restore Defaults**»;
- в контекстном меню выбрать «**Resource Configuration->Reset to Default**», в открывшемся окне выбрать конфигурацию и нажать кнопку «**OK**».

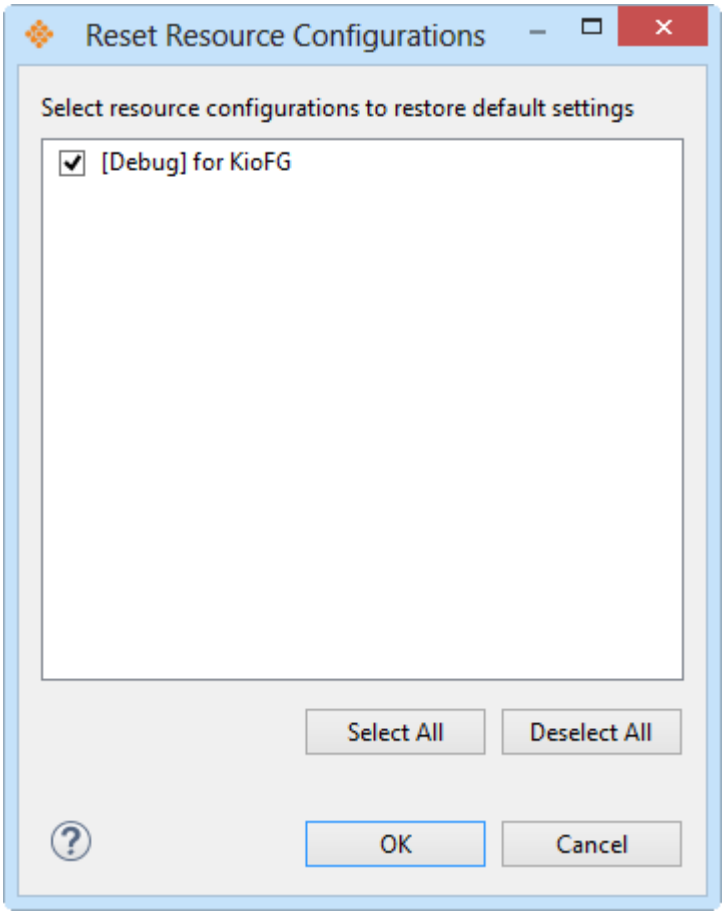


Рисунок 172. Сброс параметров конфигурации для ресурса

6.9.2.4.2. Настройки шагов сборки

На вкладке **«Build Steps»** находятся настройки шагов сборки.

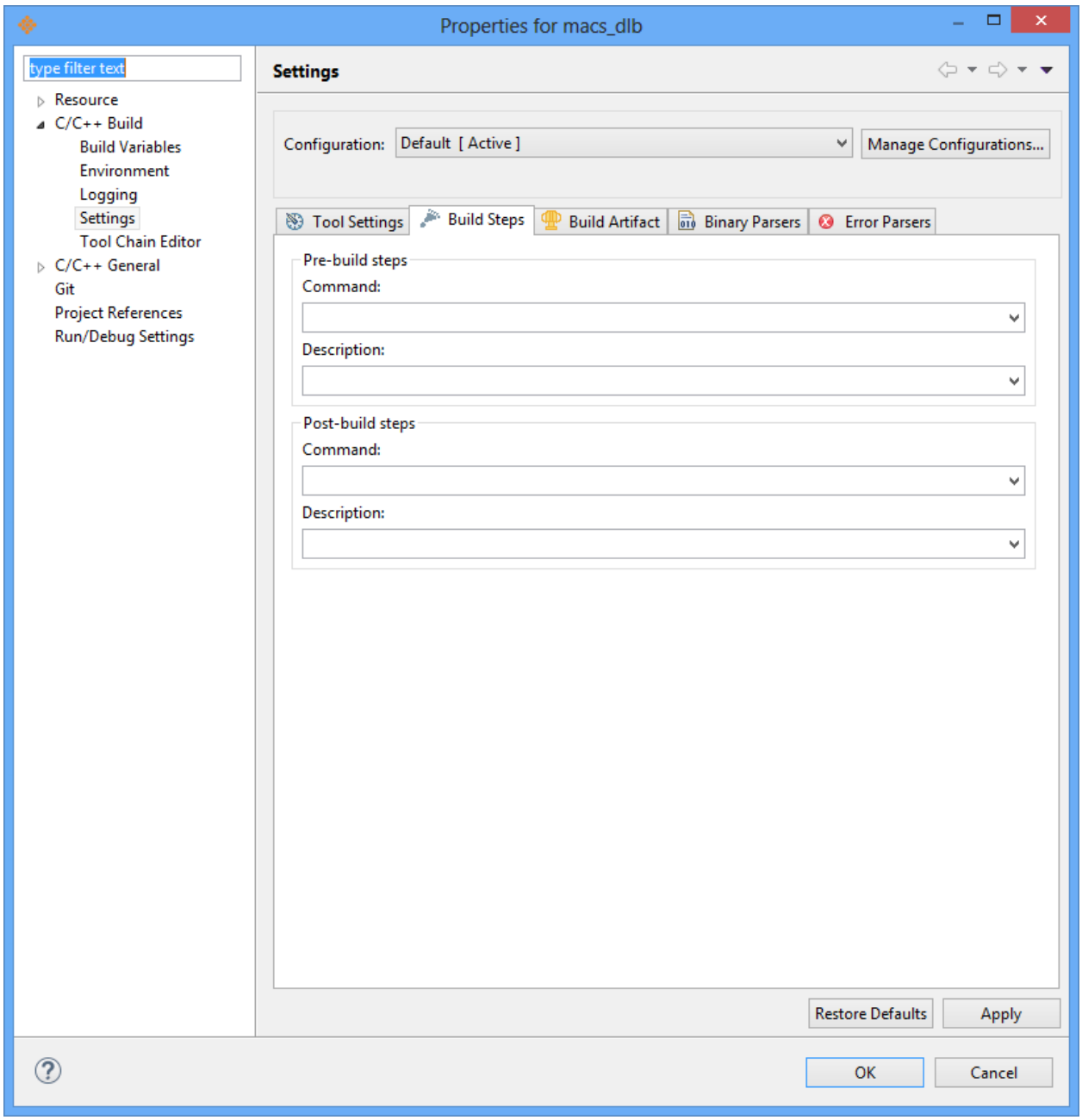


Рисунок 173. Вкладка «Build Steps»

Таблица 78. Элементы управления вкладки «Build Steps»

Название элемента	Описание функционирования
Pre-build Steps	Определить все шаги, которые необходимо выполнить до завершения сборки: ▪ **Command** – указать одну или более команд для немедленного выполнения перед сборкой. Для разделения нескольких команд следует использовать точку с запятой; ▪ **Description** – указать опциональный описательный текст, связанный с шагами до сборки, который отображается в выходных данных сборки до начала выполнения команд перед сборкой.

Название элемента	Описание функционирования
Post-build steps	Определить любые шаги, выполняемые после начала сборки: ▪ **Command** – указать одну или более команд для немедленного выполнения после выполнения сборки. Для разделения нескольких команд следует использовать точку с запятой; ▪ **Description** – указать опциональный описательный текст, связанный с шагами после сборки, который отображается в выходных данных сборки после начала выполнения команд после сборки.

6.9.2.4.3. Настройки артефактов сборки

На вкладке «**Build Artefact**» находятся настройки артефактов сборки.

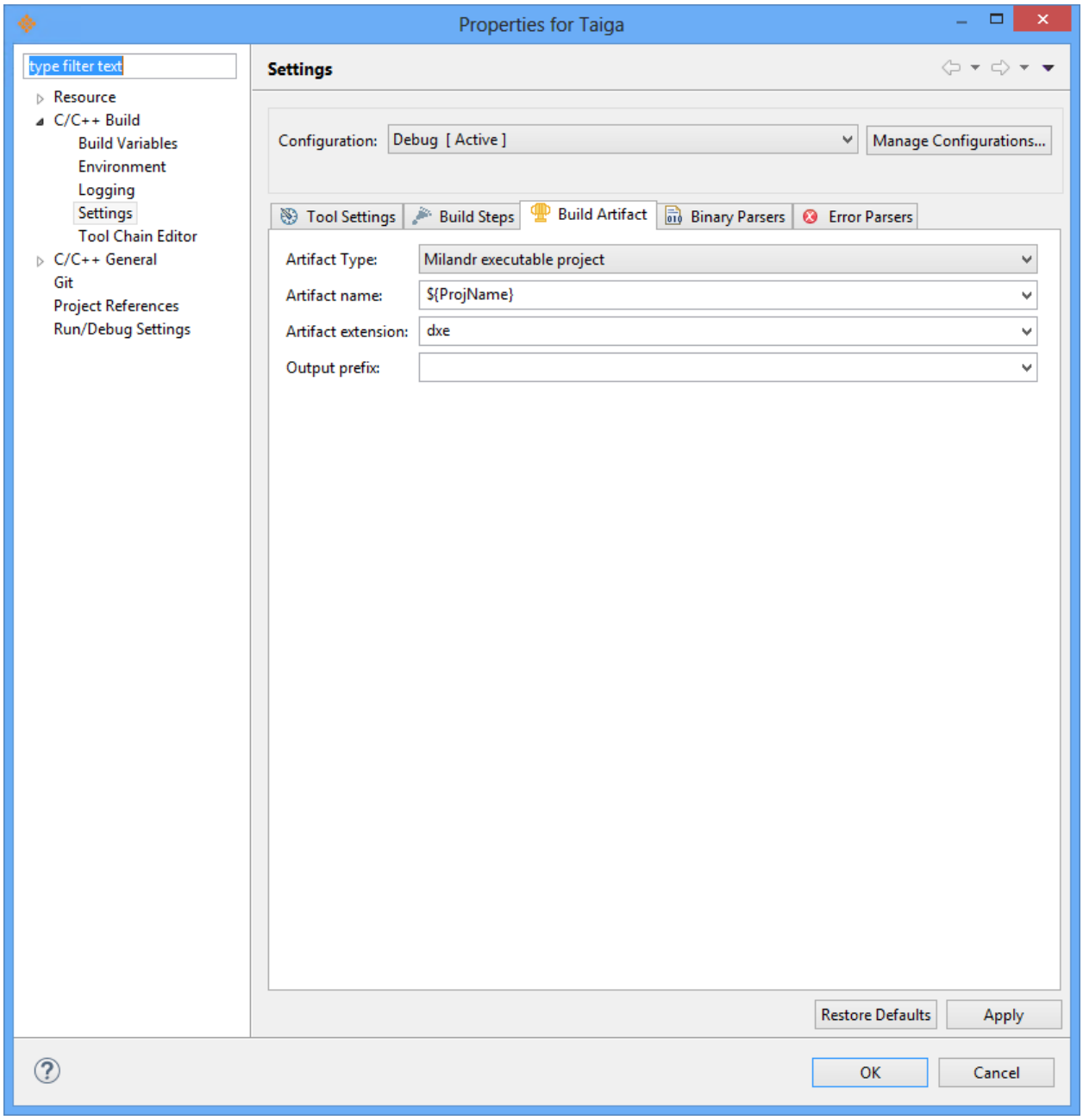


Рисунок 174. Вкладка «Build Artefact»

Таблица 79. Элементы управления вкладки «Build Artefact»

Название элемента	Описание функционирования
Artifact Type	Указать тип выбранного артефакта. Выбрать тип артефакта, который будет участвовать в текущей конфигурации сборки (например, **Executable** и **Static library**).
Artifact name	Указать имя артефакта. По умолчанию его имя и имя проекта совпадают.
Artifact extension	Указать расширение файла для определенного типа артефактов.
Output prefix	Указать префикс, который необходимо добавить к началу выходных параметров.

6.9.2.4.4. Настройки парсеров бинарных файлов

На вкладке «**Binary Parsers**» находятся настройки парсеров бинарных файлов.

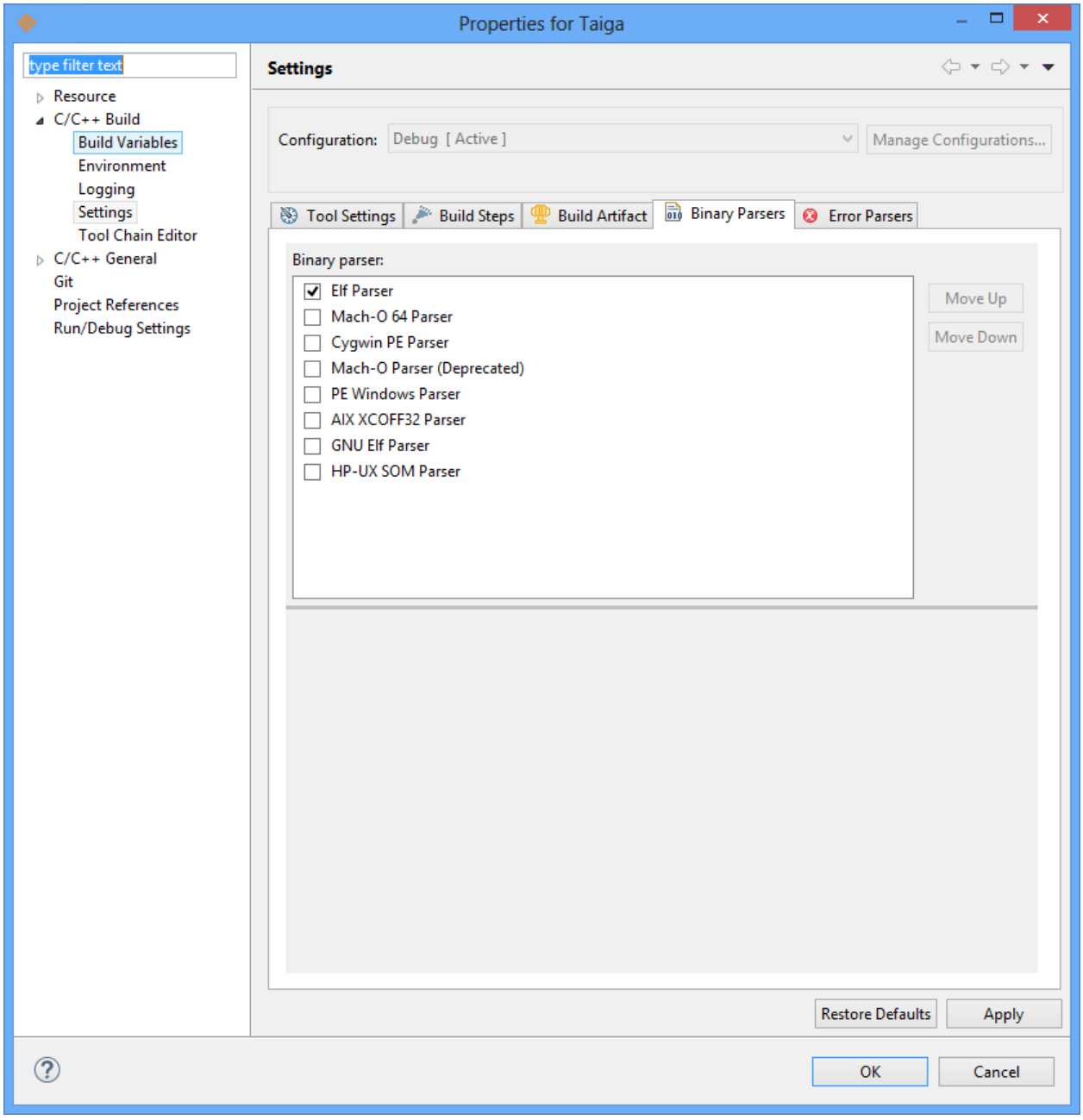


Рисунок 175. Вкладка «Binary Parsers»

Таблица 80. Элементы управления вкладки «Binary Parsers»

Название элемента	Описание функционирования
Binary Parser:	Список доступных парсеров бинарных файлов. Флажком отмечены парсеры, используемые в данный момент.
Move Up	Переместить выбранный парсер вверх по списку.
Move Down	Переместить выбранный парсер вниз по списку.

6.9.2.4.5. Настройки парсера сообщений об ошибках или предупреждений

На вкладке «**Error Parsers**» находятся настройки парсера сообщений об ошибках или предупреждений.

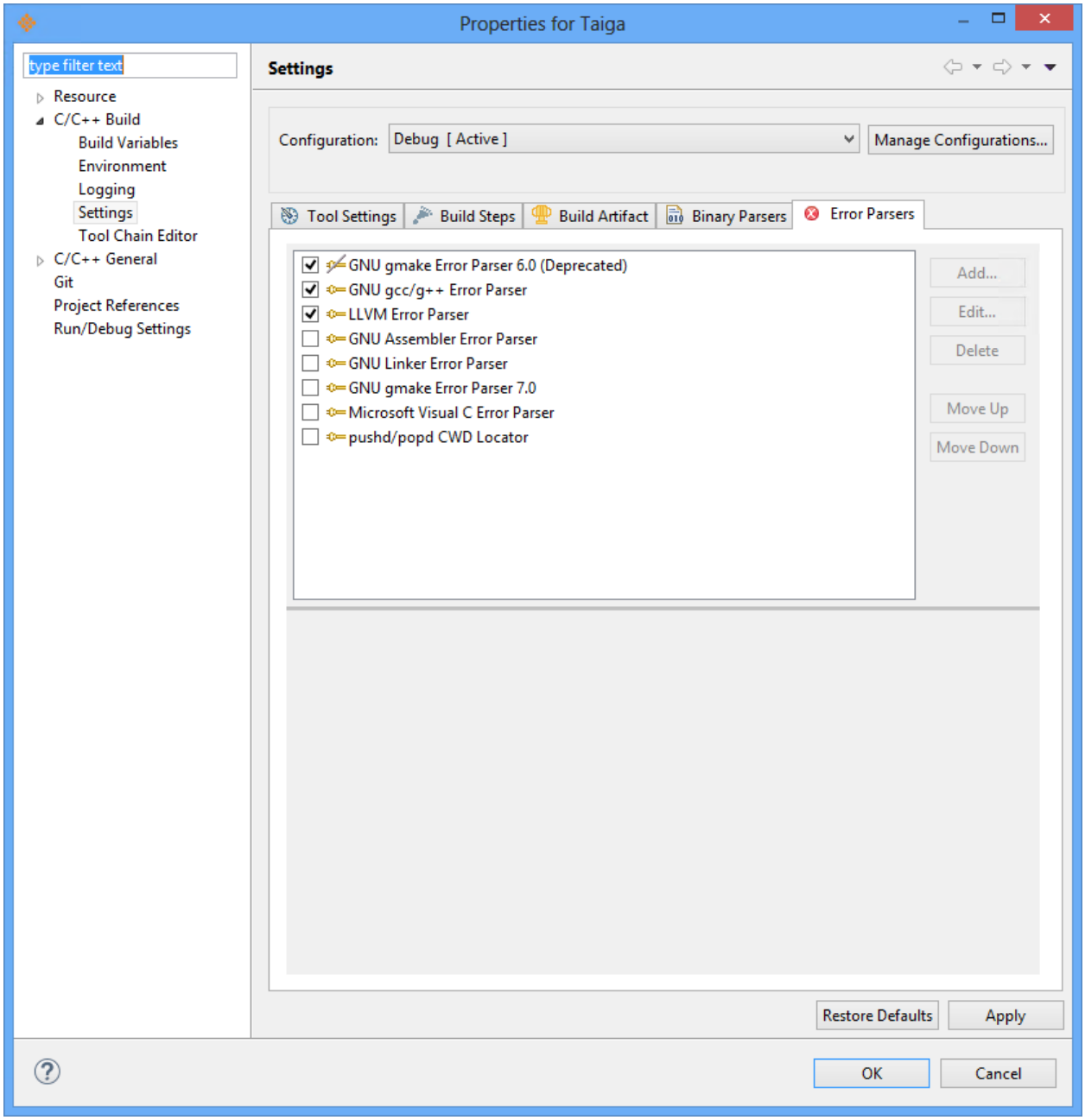


Рисунок 176. Вкладка» Error Parsers»

Вкладка содержит список доступных парсеров сообщений об ошибках и предупреждений. Флагом отмечены парсеры, используемые в данный момент.

Таблица 81. Элементы управления вкладки «Error Parsers»

Название элемента	Описание функционирования
Add…	Создать новый парсер сообщений об ошибках или предупреждений.
Edit…	Изменить уже существующий парсер сообщений об ошибках или предупреждений.
Delete	Удалить выделенный парсер сообщений об ошибках или предупреждений.
Move Up	Переместить выбранный парсер вверх по списку.
Move Down	Переместить выбранный парсер вниз по списку.

6.9.3. Настройки общих свойств проекта C/C++

Панель «**C/C++ General**» предназначена для установки общих свойств проекта. Панель доступна для проектов C/C++.

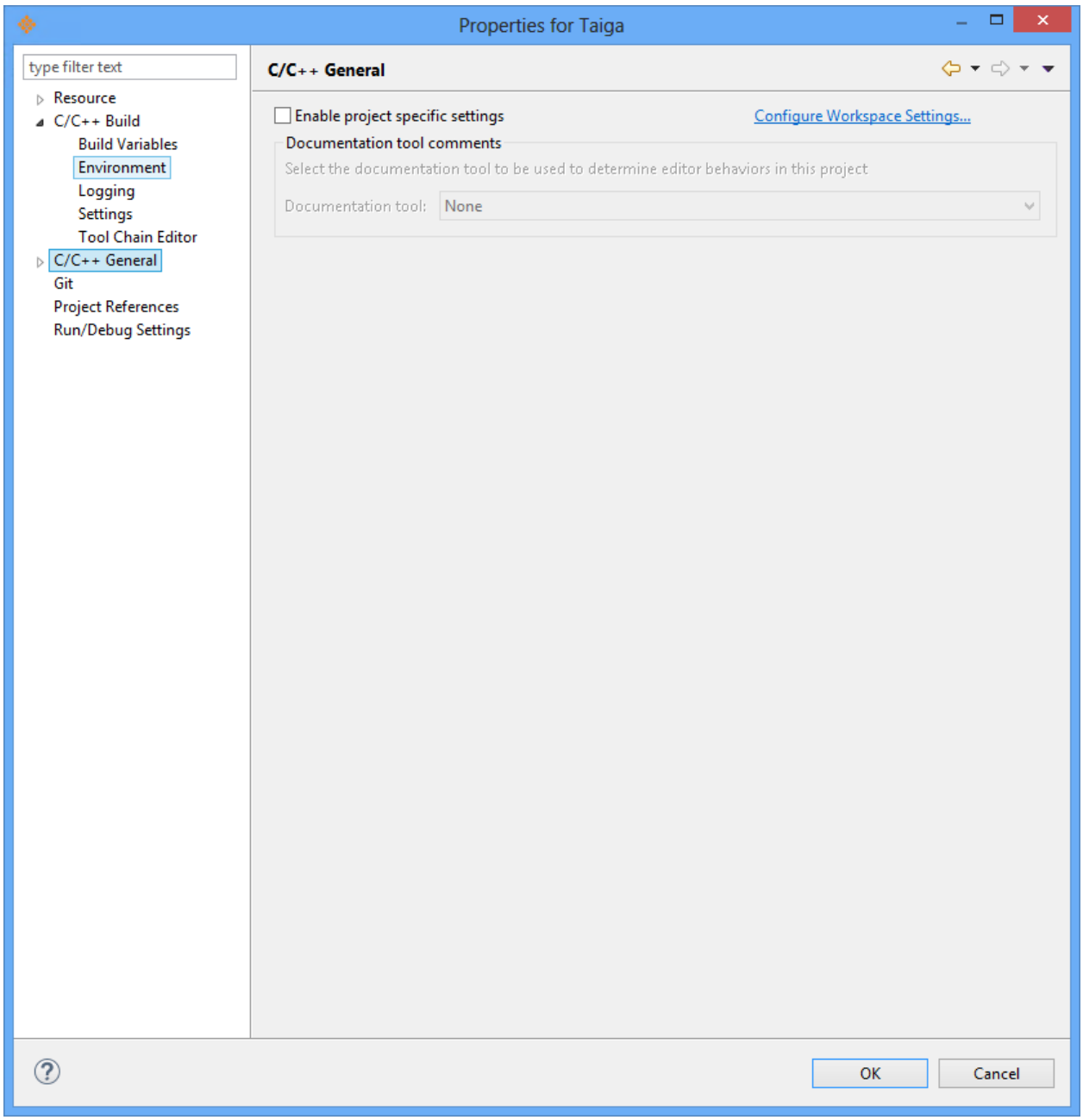


Рисунок 177. Панель «C/C++ General»

Таблица 82. Элементы управления панели «C/C++ General»

Название элемента	Описание функционирования	По умолчанию
Enable project specific settings	Включить/отключить возможность выбора инструмента.	Выкл
Documentation tool comments	Указать, какой инструмент документации использовать для определения вида и поведения редактора. Это позволяет использовать такие функции редактора как автозавершение, создание комментариев к документации и выделение синтаксиса. Выбранный инструмент будет использован для файлов этого проекта и переопределит параметры уровня рабочего пространства.	Нет

6.9.3.1. Типы файлов

Панель «**File Types**» предназначена для просмотра типов файлов. Панель доступна только для проектов C/C++.

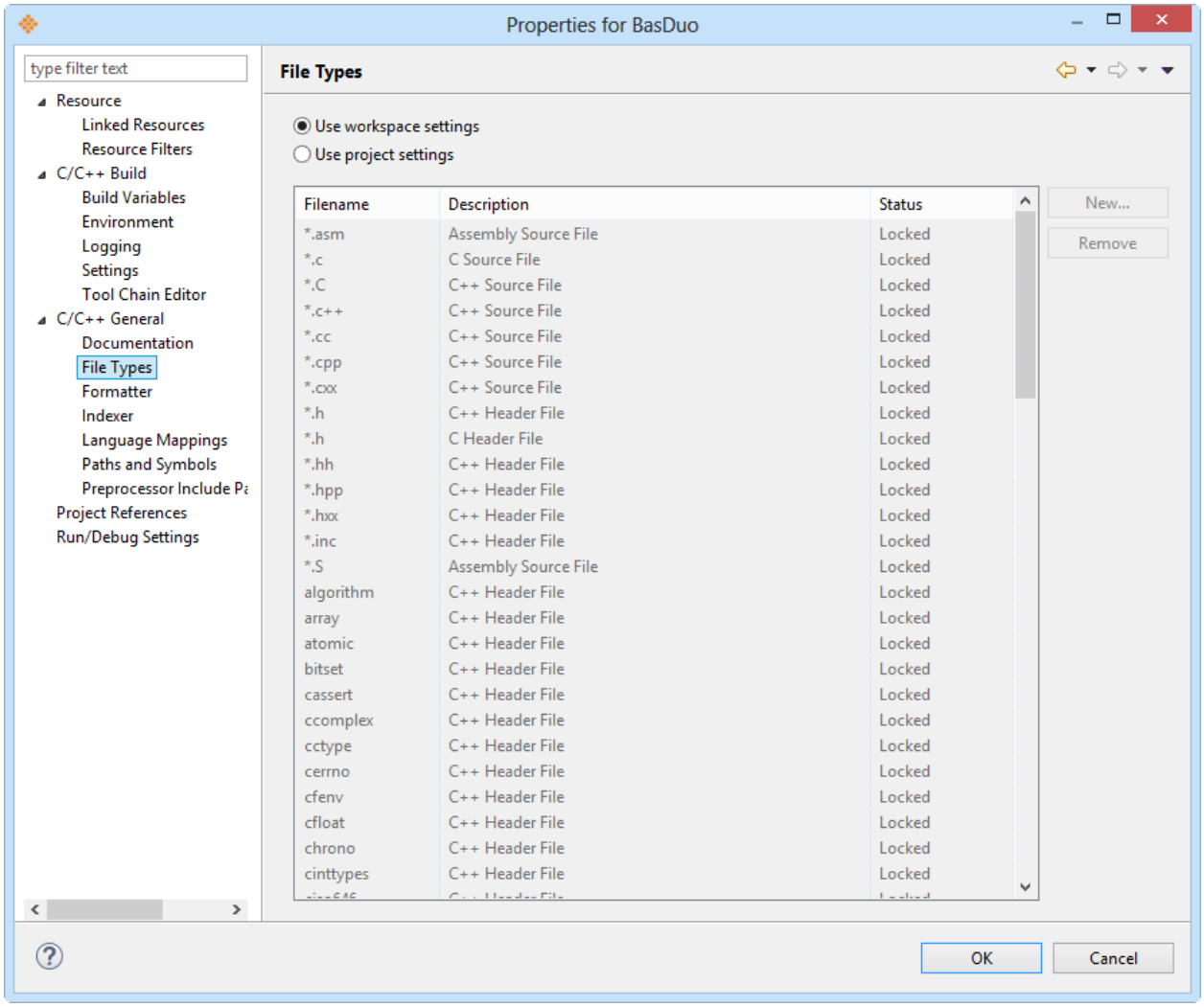


Рисунок 178. Панель «File Types»

Таблица 83. Элементы управления панели «File Types»

Название элемента	Описание функционирования	По умолчанию
Use workspace settings	Использовать настройки управляемой рабочей области.	Вкл
Use Project Settings	Использовать настройки проекта или добавления, или удаления определенных типов файлов.	Выкл
New...	Добавить новый тип файла.	–
Remove	Удалить тип файла.	–

6.9.3.2. Форматирование

Панель «**Formatter**» предназначена для настройки стилей кода проекта для функций «умного» ввода, таких как автовыравнивание и форматирование. Панель доступна только для проектов C/C++.

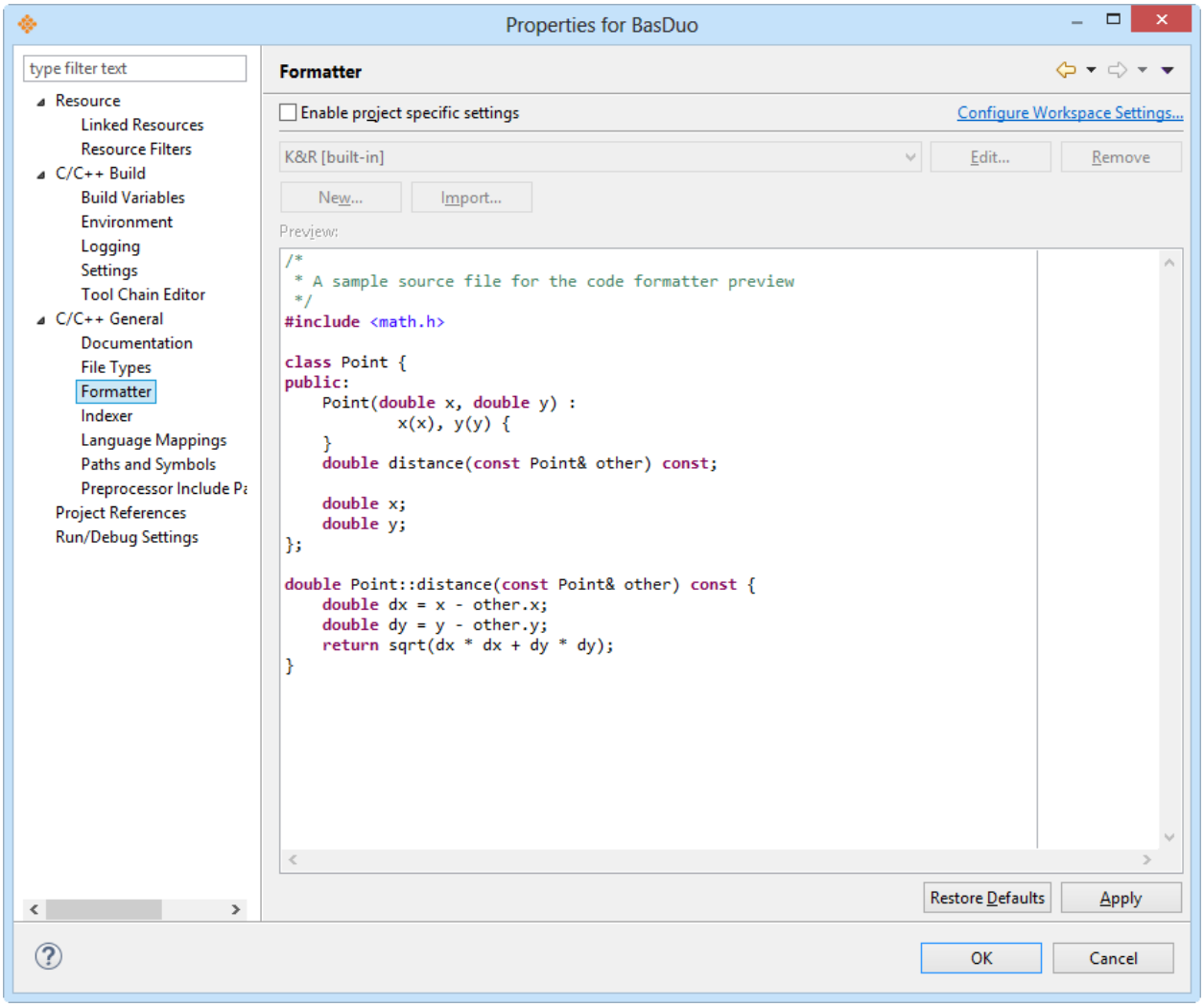


Рисунок 179. Панель «Formatter»

Таблица 84. Элементы управления панели «Formatter»

Название элемента	Описание функционирования	По умолчанию
Enable project specific settings	Включить/отключить возможность изменять настройки.	Выкл
Edit...	Открыть диалоговое окно, которое показывает настройки, хранящиеся в активном профиле. Можно изменить только пользовательские профили.	–
Remove	Удалить выбранный профиль. Это действие доступно только для пользовательских профилей.	–
New...	Открыть диалоговое окно для создания нового профиля. В окне указывается имя нового профиля, а также встроенный или уже существующий пользовательский профиль, на основе которого будет создаваться собственный.	–

Название элемента	Описание функционирования	По умолчанию
Import...	Импортировать профили из файловой системы.	–

6.9.3.3. Индексатор

Панель «**Indexer**» предназначена для выбора индексатора C/C++, который будет использоваться в проекте. Он необходим для поиска и связанных с ним функций, таких как автозавершение. Панель доступна только для проектов C/C++.

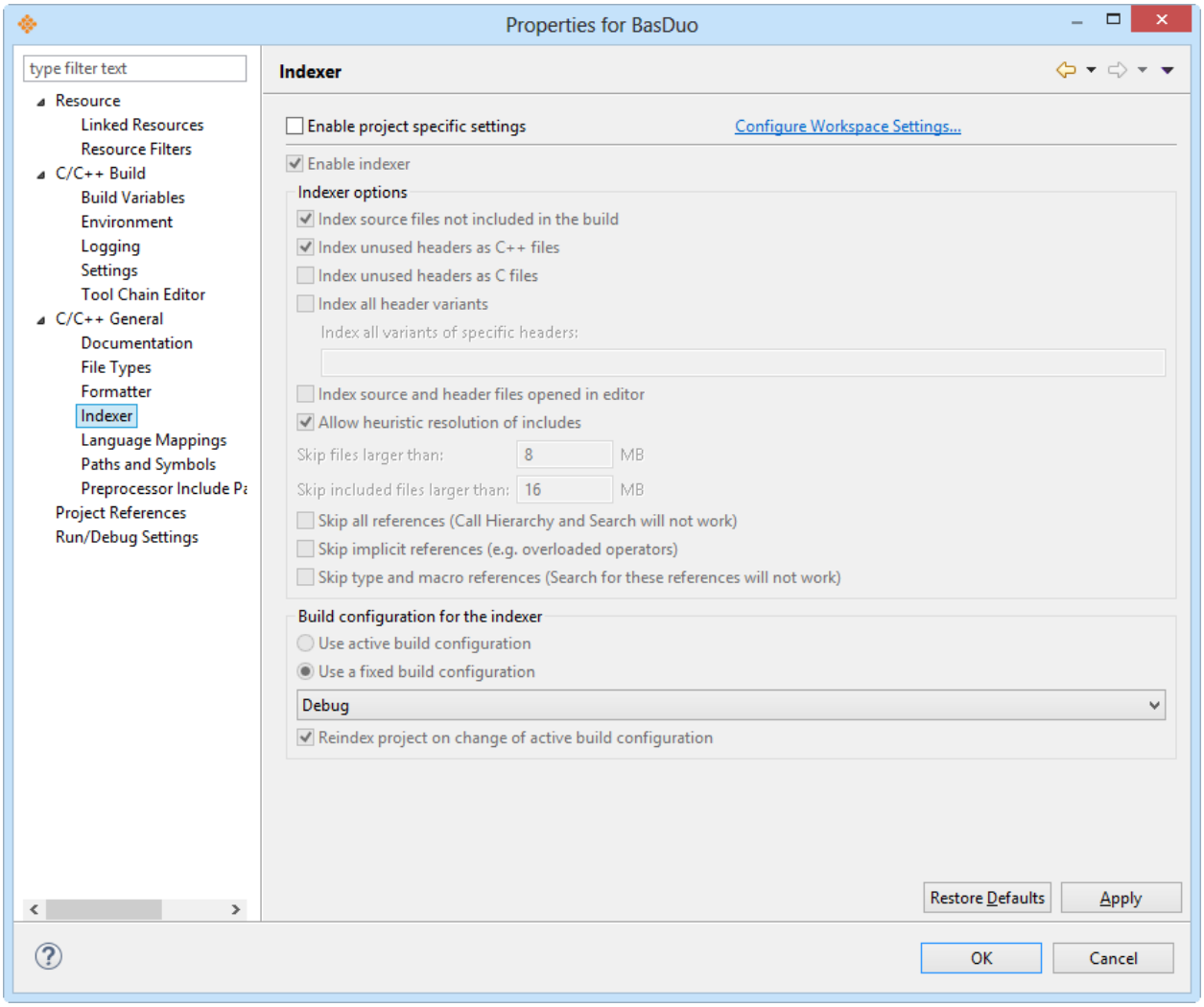


Рисунок 180. Панель «Indexer»

Таблица 85. Элементы управления панели «Indexer»

Название элемента	Описание функционирования	По умолчанию
Enable project specific settings	Включить/отключить возможность изменять настройки.	Выкл
Enable indexer	Включить/отключить индексацию.	Вкл
Index source files not included in the build	Включить индексацию всех исходных файлов, используемых в проекте.	Вкл

Название элемента	Описание функционирования	По умолчанию
Index unused headers	Включить индексацию неиспользуемых заголовочных файлов.	Вкл
Index all header variants	Включить индексацию всех вариаций заголовочных файлов. Для специфичных заголовочных файлов индексируемые варианты можно задать в отдельном поле.	Выкл
Index source and header files opened in editor	Включить индексацию всех исходных и заголовочных файлов, открытых в редакторе.	Выкл
Allow heuristic resolution of includes	Разрешить индексатору пропускать повторяющиеся вложенные файлы.	Вкл
Skip files larger than	Указать максимальный размер файла, пропускаемого при индексации.	8 Мб
Skip included files larger than	Указать максимальный размер используемых в проекте файлов, пропускаемых при индексации.	16 Мб
Skip all references (Call Hierarchy and Search will not work)	Отключить индексацию ссылок для экономии места и повышения производительности. Это отвлекает некоторые ресурсы от работы, такие как «**Call Hierarchy**» и поиск ссылок.	Выкл
Skip implicit references (e.g. overloaded operators)	Отключить индексацию скрытых ссылок.	Выкл
Skip type and macro references (Search for these references will not work)	Отключить индексацию макро или типовые ссылок. Это отвлекает некоторые ресурсы от работы, такие как «**Search**».	Выкл
Use active build configuration	Использовать настройки активной конфигурации индексатора сборки, чтобы выполнить сборку индекса.	Вкл
Use a fixed build configuration	Использовать настройки заданной конфигурации индексатора сборки. Конфигурация выбирается из выпадающего списка.	Выкл
Reindex project on change of active build configuration	Провести повторную индексации при изменении активной конфигурации сборки.	Вкл

6.9.3.4. Списки путей к заголовочным файлам

Вкладка «**Includes**» панели «**Paths and Symbols**» предназначена для работы со списком путей к заголовочным файлам, а также для установки последовательности, в которой на них ссылаются.

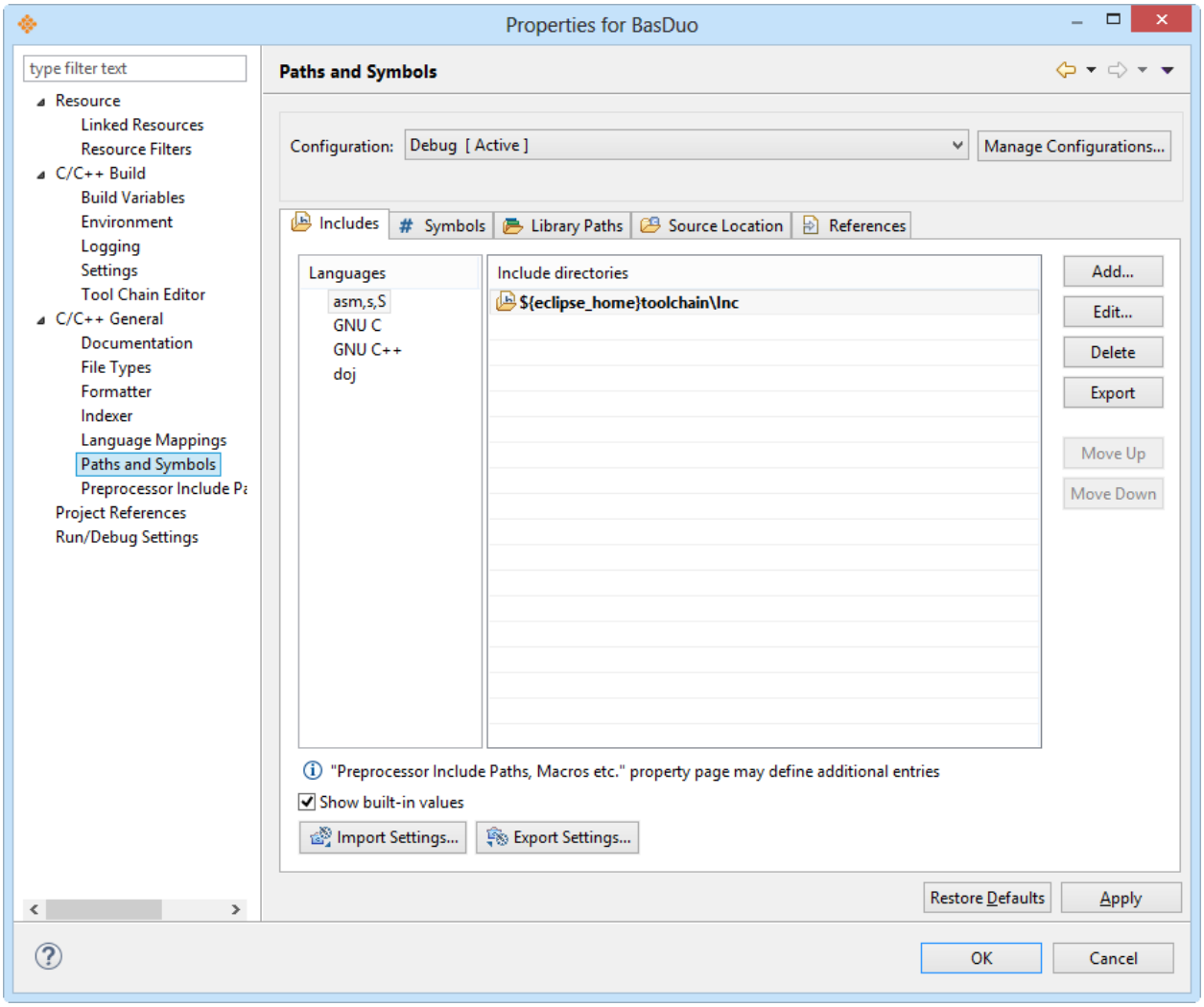


Рисунок 181. Вкладка «Includes»

Таблица 86. Элементы управления вкладки «Includes»

Название элемента	Описание функционирования	По умолчанию
Languages (левая панель)	Список языков, связанных с этим файлом.	–
Include directories (правая панель)	Список путей к заголовочным файлам, связанных с данным языком.	–
Add...	Добавить новый путь к заголовочным файлам.	–
Edit...	Изменить выбранный путь к заголовочным файлам.	–
Delete	Удалить выбранный путь к заголовочным файлам.	–
Export	Экспортировать выбранный путь.	–
Move Up	Переместить выбранный элемент вверх в списке.	–
Move Down	Переместить выбранный элемент вниз в списке.	–
Show built-in values	Показать/скрыть системные (встроенные) значения.	Вкл

Название элемента	Описание функционирования	По умолчанию
Import Settings...	Импортировать настройки символа.	–
Export Settings...	Экспортировать настройки символа.	–

6.9.3.5. Списки путей к библиотекам

Вклада «**Library path**» панели «**Paths and Symbols**» предназначена для работы со списком путей к библиотекам, а также для изменения последовательности, в которой на них ссылаются.

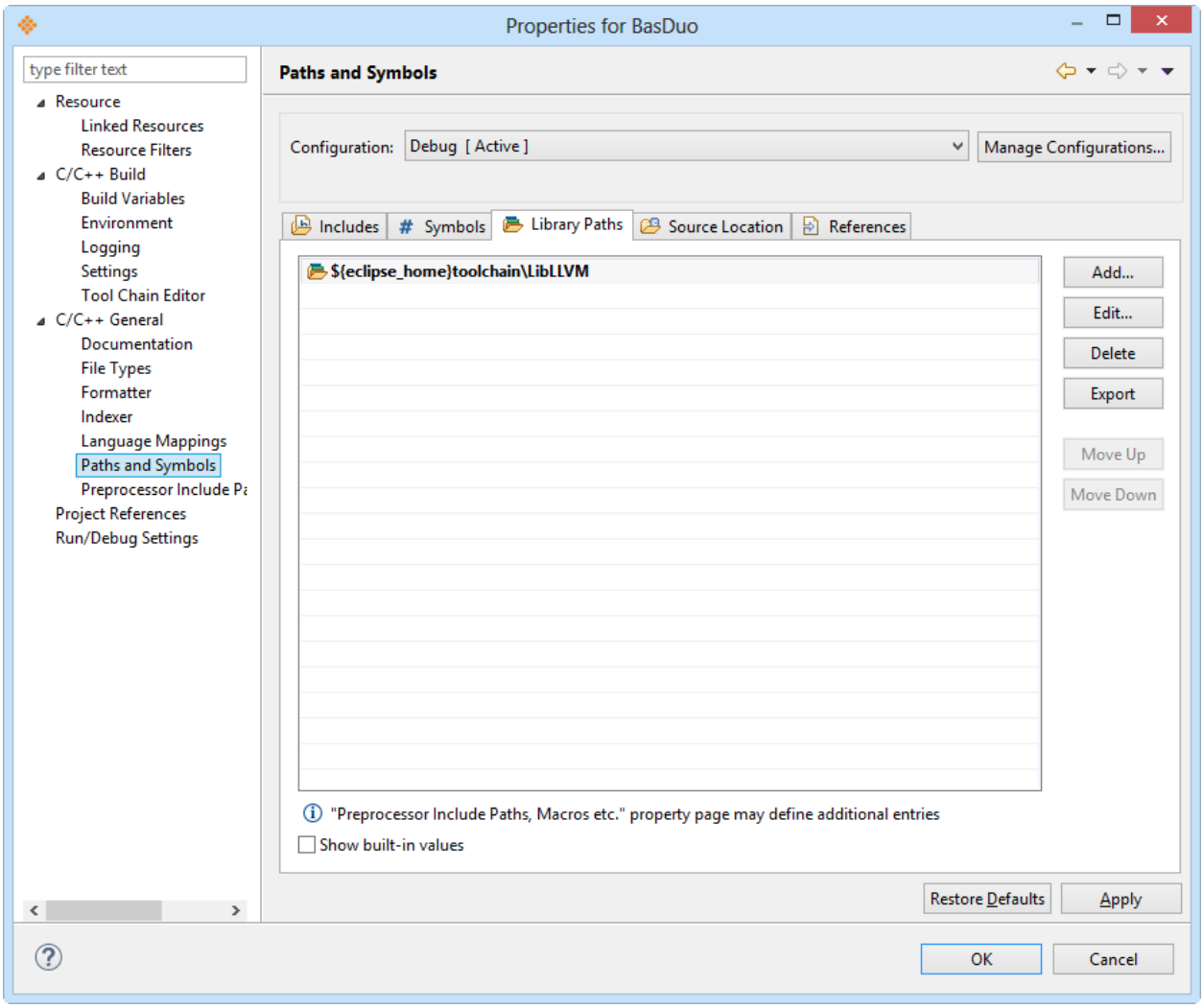


Рисунок 182. Вкладка «Library path»

Таблица 87. Элементы управления вкладки «Library path»

Название элемента	Описание функционирования	По умолчанию
Add...	Добавить новый путь к библиотекам.	–
Edit...	Изменить выбранный путь к библиотекам.	–
Delete	Удалить выбранный путь к библиотекам.	–

Название элемента	Описание функционирования	По умолчанию
Export	Переключить выбранный путь, экспортированный или нет.	–
Move Up	Переместить выбранный элемент вверх в списке.	–
Move Down	Переместить выбранный элемент вниз в списке.	–
Show built-in values	Показать/скрыть системные (встроенные) пути к библиотекам.	Выкл

6.9.3.6. Списки исходных данных проекта

Вкладка «**Source location**» панели «**Paths and Symbols**» предназначена для работы со списком исходных данных проекта, использованных в данной конфигурации.

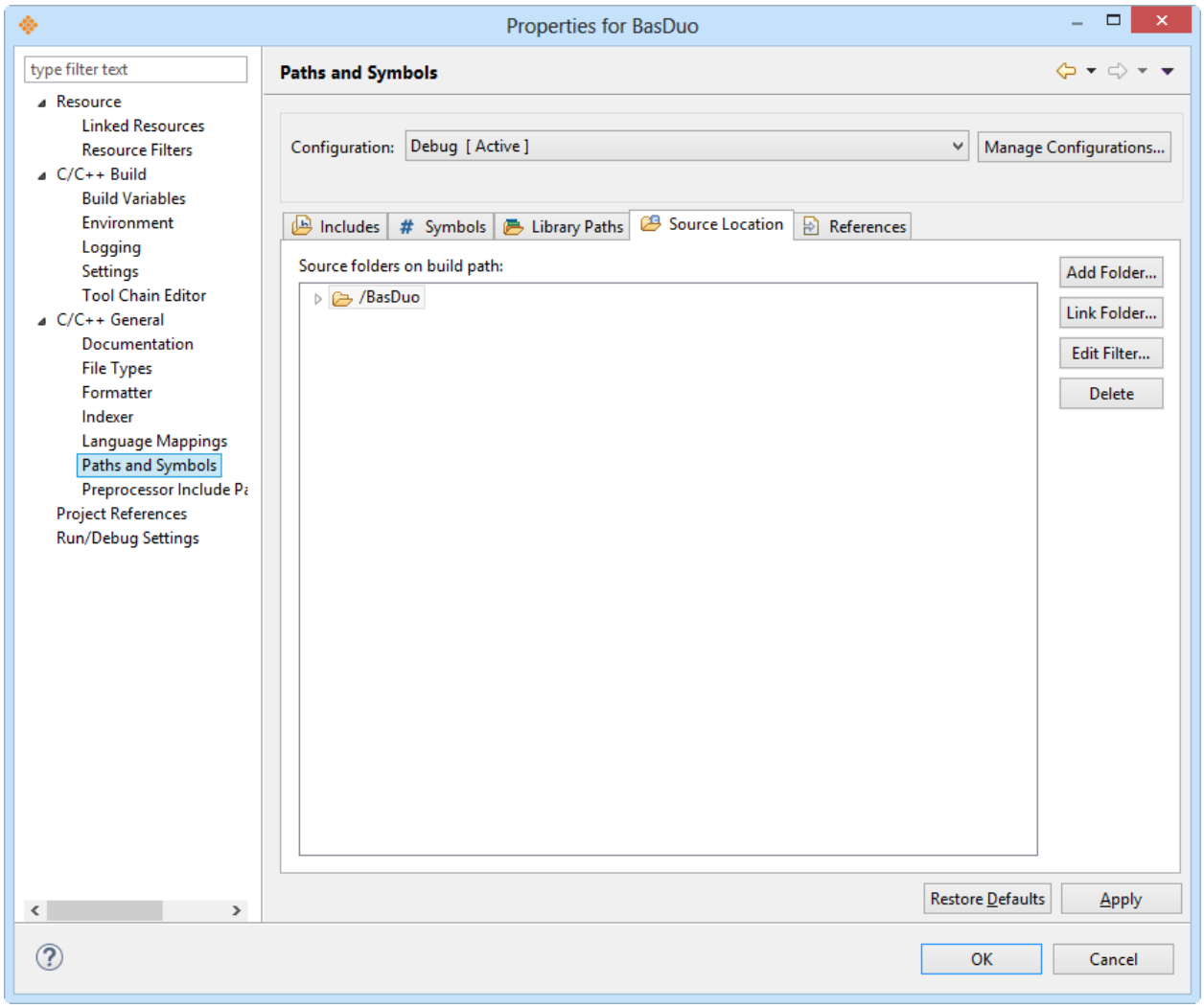


Рисунок 183. Вкладка «Source location»

Таблица 88. Элементы управления вкладки «Source location»

Название элемента	Описание функционирования
Add Folder...	Назначить существующую папку исходной папкой.
Link Folder...	Назначить директорию вне проекта исходной папкой.

Название элемента	Описание функционирования
Edit Filter…	Редактировать фильтр для выбранной исходной папки.
Delete	Удалить исходную папку.

6.9.3.7. Зависимости

Вкладка «**References**» панели «**Paths and Symbols**» предназначена для установления зависимостей между конфигурациями сборки проектов.

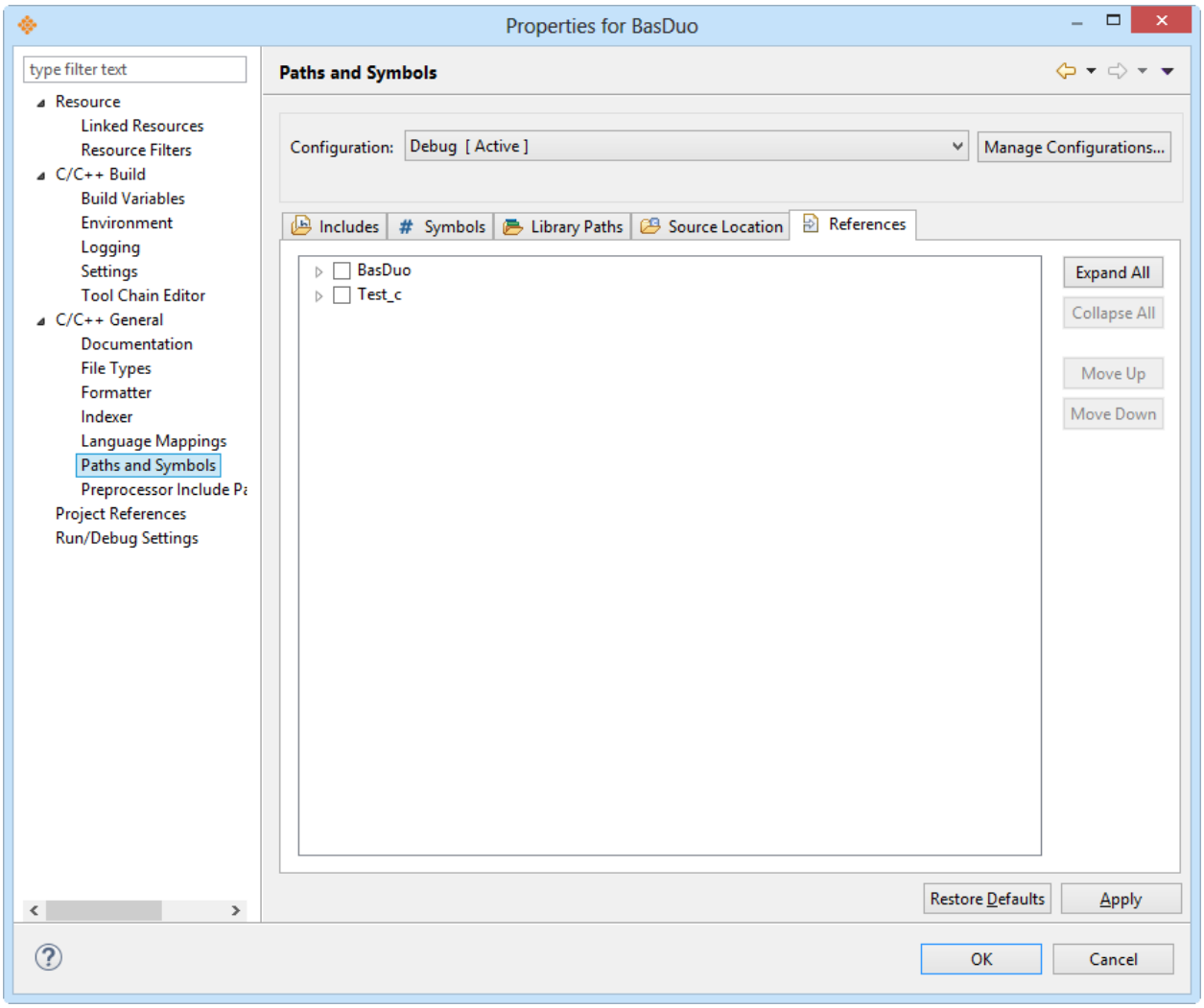


Рисунок 184. Вкладка «References»

Таблица 89. Элементы управления вкладки «References»

Название элемента	Описание функционирования
Expand All	Раскрыть все элементы дерева.
Collapse All	Свернуть все элементы дерева.
Move Up	Переместить проект вверх в списке ссылок.
Move Down	Переместить проект вниз в списке ссылок.

6.9.4. Настройки Git

Панель «**Git**» предназначена для просмотра свойств репозитория Git. Панель доступна для проектов с поддержкой работы в репозитории git.

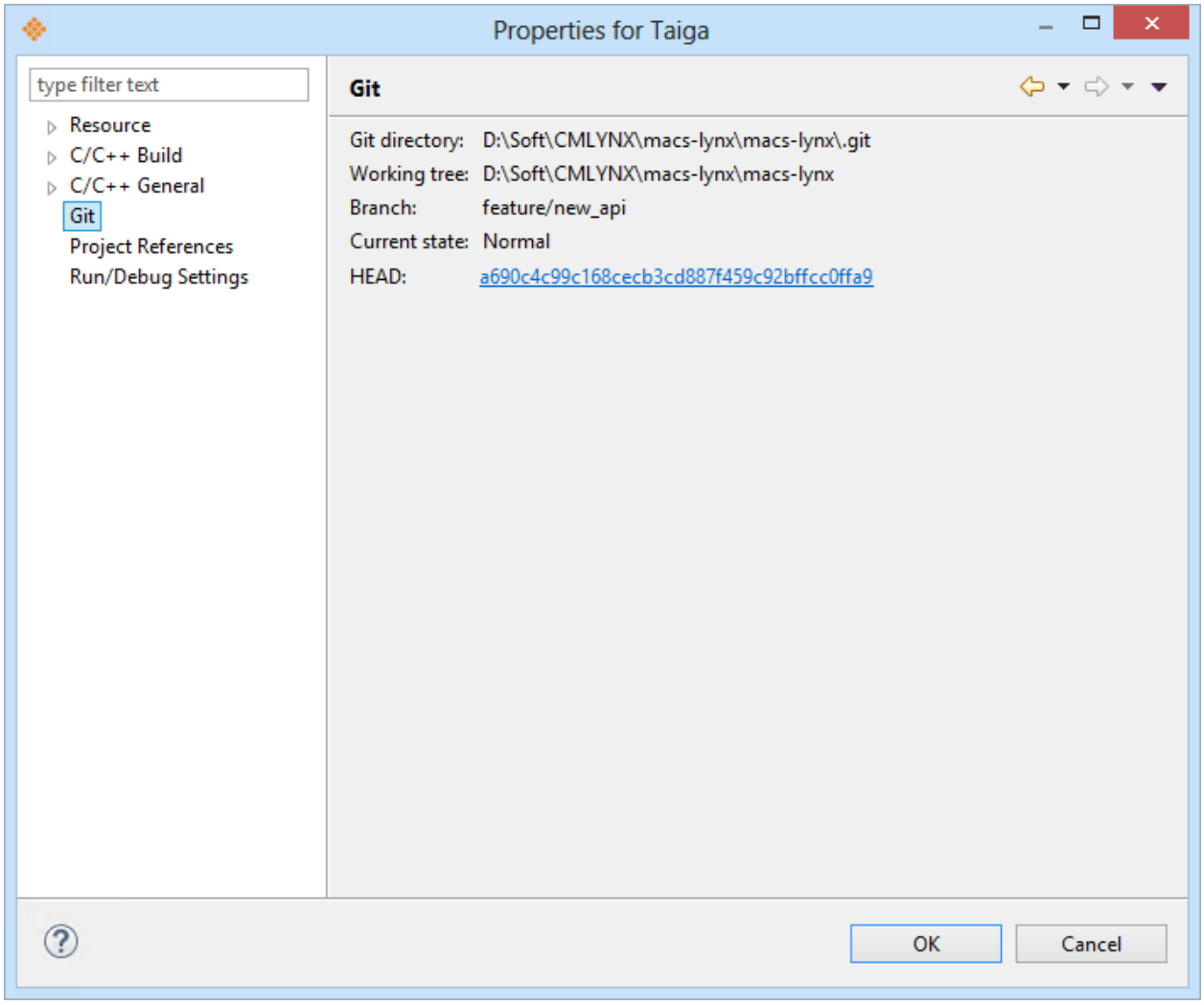


Рисунок 185. Панель «Git»

Таблица 90. Элементы управления панели «Git»

Название элемента	Описание функционирования
Git directory	Путь к директории Git.
Working tree	Путь к рабочей папке репозитория для данного проекта.
Branch	Ссылка на коммит.
Current state	Статус подключения к репозиторию
HEAD	Указатель на текущий коммит

6.10. Сборка проекта

В CM-LYNX доступны следующие варианты сборки проекта:

- Автосборка (если автосборка включена, то сборка будет происходить автоматически при сохранении измененных ресурсов);
- Ручная сборка.

6.10.1. Управление автосборкой

Для того чтобы включить автосборку, необходимо выполнить одну из команд:

- Выбрать в меню «**Project**» пункт «**Build Automatically**» (будет установлена отметка);
- На панели «**Workspace**» окна «**Preferences**» (для перехода используйте команду **Window->Preferences->General->Workspace**) установить флаг «**Build automatically**».

Для того чтобы отключить автосборку, необходимо выполнить одну из команд:

- Выбрать в меню «**Project**» пункт «**Build Automatically**» (отметка будет снята);
- На панели «**Workspace**» окна «**Preferences**» (для перехода используйте команду **Window->Preferences->General->Workspace**) снять флаг «**Build automatically**».

Примечание: обратите внимание, что если расширение файла указано как «*.C», т.е. большими буквами, то данный файл будет интерпретироваться как C++, что может привести к ошибкам при сборке.

6.10.2. Запуск ручной сборки

Таблица 91. Варианты ручной сборки

Операция	Команда
Собрать все проекты	В меню «**Project**» выбрать пункт «**Build All**» или использовать комбинацию горячих клавиш **Ctrl+B**.
Собрать отдельный проект	1 способ: 1. Выбрать проект. 2. Выбрать пункт «**Build Project**» в меню «**Project**». 2 способ: 1. Выбрать проект. 2. Выбрать пункт «**Build Project**» в контекстном меню или нажать на кнопку  на панели инструментов. 3 способ: 1. Выбрать проект. 2. Нажать на кнопку  на панели инструментов.
Очистить и пересобрать с нуля все проекты	1. В меню «**Project**» выбрать пункт «**Clean…**». 2. Выбрать вариант «**Clean all projects**» и нажать кнопку «**OK**».
Очистить и пересобрать отдельные проекты	1 способ: 1. Выбрать проект. 2. В меню «**Project**» выбрать пункт «**Clean…**».

Операция	Команда
	3. Нажать кнопку «**OK**». 2 способ: 1. В меню «**Project**» выбрать пункт «**Clean…**». 2. Выбрать вариант «**Clean project selected below**». 3. Выбрать проекты и нажать кнопку «**OK**».
Собрать отдельные файлы проекта	1. Выбрать файл в проекте или выделить несколько файлов. 2. Правой кнопкой мыши вызвать контекстное меню. 3. Выбрать в контекстном меню пункт «**Build Selected File(s)**». Примечания: 1) Результаты сборки отдельных файлов игнорируются при обычной сборке проекта, т.е. эти файлы будут пересобраны. 2) Сборка отдельных файлов может не срабатывать для файлов, которые были только что добавленны в проект.

6.10.3. Обработка ошибок сборки

После завершения сборки результат сборки отображается на вкладке «**Console**» (см. п. 5.4.2.5 Вкладка «Console»).

Проблемы при сборке классифицируются следующим образом:

- ошибки (Errors);
- предупреждения (Warnings);
- информация (Info).

При наличии ошибок при сборки информация о них доступна:

- на вкладке «**Console**» – отмечены строки, содержащие ошибки;
- на вкладке «**Project Explorer**» - помечены файлы, их родительские папки и соответствующий проект;
- в окне редактора – иконки на границе редактора и проблемный код отмечен волнистой линией;
- на вкладке «**Problems**» (см. п. 5.4.2.3. Вкладка «Problems») – список выявленных проблем.

Для того чтобы перейти к источнику ошибки, необходимо выполнить одно из следующих действий:

- на вкладке «**Console**» дважды щелкнуть по выделенной строке;
- на вкладке «**Problems**» дважды щелкнуть по строке ошибки в таблице.

Если на вкладке «**Problems**» для ошибки в колонке «**Location**» указано значение «**External Location**», можно открыть файл с проблемой. Для этого следует щелкнуть правой кнопкой мыши по такой проблеме и в контекстном меню выбрать пункт «**Open external location**». Если файл найден и определена строка кода с проблемой, то файл откроется в окне редактора C/C++ и курсор установится на строке с ошибкой.

Для перемещения между ошибками следует использовать:

- для перемещения в окне редактора - кнопки  «**Next Annotation**» и  «**Previous Annotation**» панели инструментов;

- для перемещения на вкладке «**Console**» - кнопки  «**Next Error**» и  «**Previous Error**».

6.11. Отладка

Перед запуском отладки необходимо выбрать и настроить конфигурацию отладки. CM-LYNX, начиная с версии 2.08.00, автоматически создаёт отладочную конфигурацию с параметрами по-умолчанию при создании и импорте проекта (см. также п. 6.12.2.4.2).

Конфигурация отладки содержит настроенные параметры отладки: тип отладчика (симулятор или аппаратный отладчик JEM), конфигурация целевой платформы, указание на отлаживаемую программу и другие дополнительные параметры.

Доступна отладка на следующих конфигурациях целевой платформы:

- отдельные процессоры серии 1967ВН0XX;
- группа процессоров серий 1967ВН028 с общим интерфейсом отладки JTAG, объединенная в симметричную многопроцессорную систему SMP (такая система характеризуется тем, что один и тот же исполняемый файл загружается на все процессоры);
- группа процессоров серий 1967ВН с общим интерфейсом отладки JTAG, объединенная в асимметричную многопроцессорную систему AMP (такая система характеризуется тем, что на каждый из процессоров могут быть загружены разные исполняемые файлы);
- а также их различные комбинации.

При необходимости, в CM-LYNX может быть настроено несколько конфигураций отладки. Однако одновременно может быть запущен только один сеанс отладки. При необходимости изменить конфигурацию отладки необходимо остановить текущую сессию отладки и запустить новую сессию с интересующей конфигурацией.

Для обмена настройками отладочной сессии можно воспользоваться механизмом экспорта / импорта настроек (команды **File-> Export...**/**File->Import...**).

6.11.1. Настройка конфигурации и запуск отладки

Примечание: перед настройкой конфигурации и запуском отладки необходимо собрать проект в Debug конфигурации.

Для настройки конфигурации и запуска отладки следует:

1. В меню «**Debug**»  (см. Рисунок 186) выбрать пункт «**Debug Configurations...**».

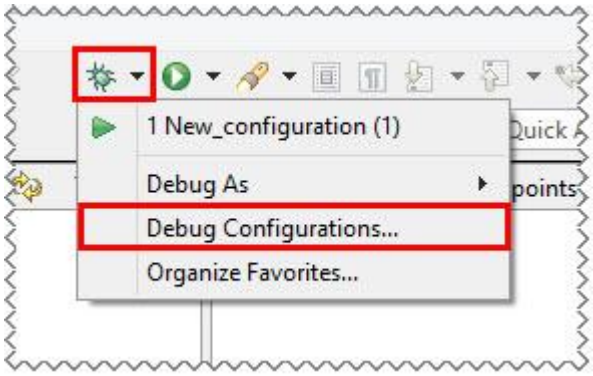


Рисунок 186. Меню «Debug»

2. В открывшемся окне (см. Рисунок 187) выбрать автоматически созданную или создать новую конфигурацию отладки, дважды щелкнув по наименованию конфигурации отладки **Milandr DSP1967 LLDB**.

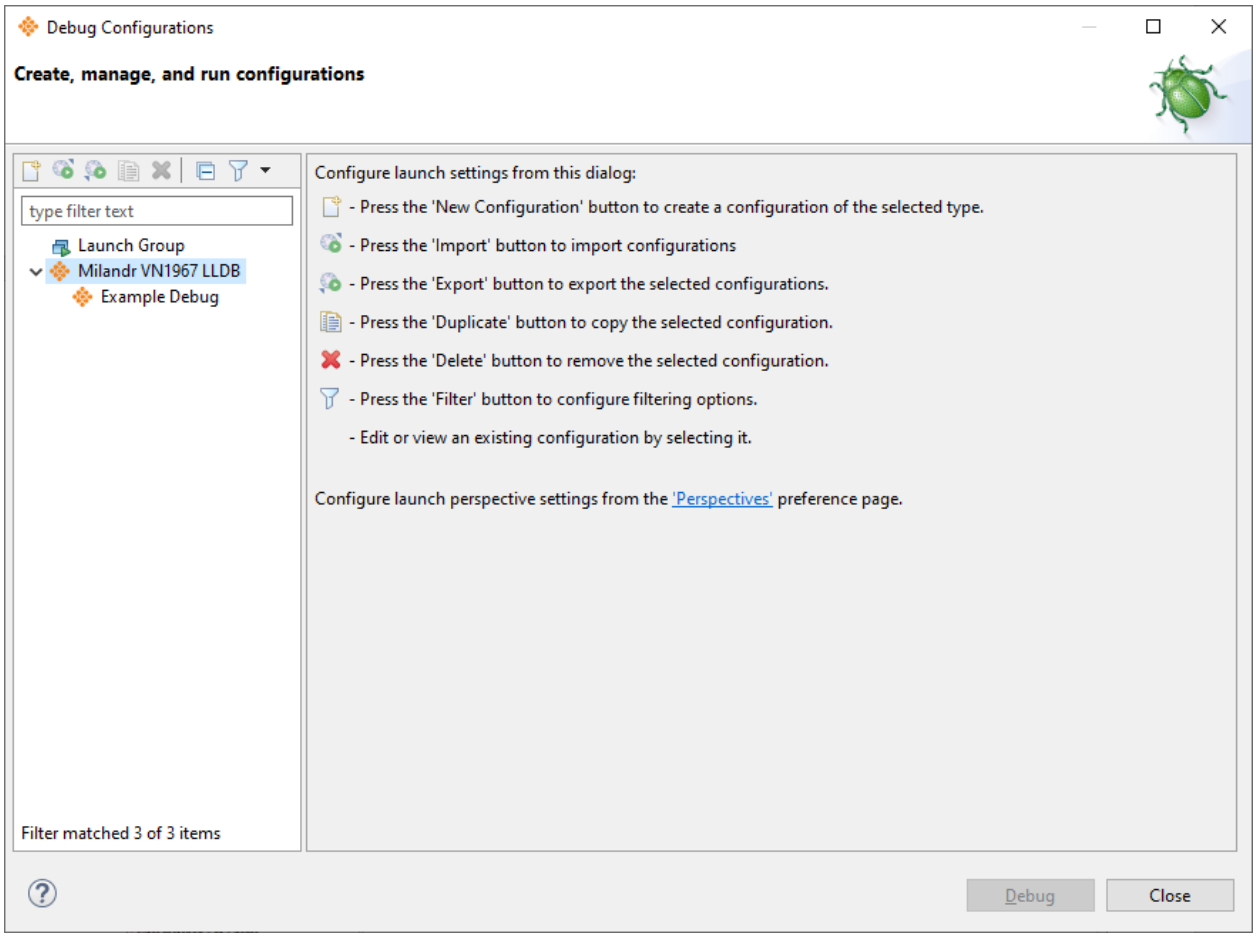


Рисунок 187. Окно конфигураций отладки

3. Задать (изменить) параметры на вкладке «**Main**» (см. Рисунок 188).

   Для однопроцессорной конфигурации:

   1) В поле «**Name**» измените наименование отладочной конфигурации (при необходимости);
   2) В поле «**Run**» выберите тип связи с исполняемым файлом и исполняемый файл. Если выбран тип связи «**Project**», укажите проект. Если выбран тип связи «**File**», укажите путь к файлу.

Для многопроцессорной конфигурации:

1) В поле «**Name**» измените наименование отладочной конфигурации (при необходимости);
2) В поле «**DSP cores**» укажите количество процессоров в цепочке JTAG. Можно воспользоваться кнопкой «**Autodetect**» для автоматического определения числа процессоров в плате.
3) При наличии многопроцессорной конфигурации SMP систем объедините процессоры в группы с помощью счетчика «**Cores in group**». Группа SMP может создаваться с любого номера на любое количество процессоров (см. Рисунок 189).
4) Для каждого процессора или группы SMP укажите тип связи с исполняемым файлом и исполняемый файл. Если выбран тип связи «**Project**», укажите проект. Если выбран тип связи «**File**», укажите путь к файлу.

Подробнее о параметрах настройки на вкладке «**Main**» см. Таблица 92.

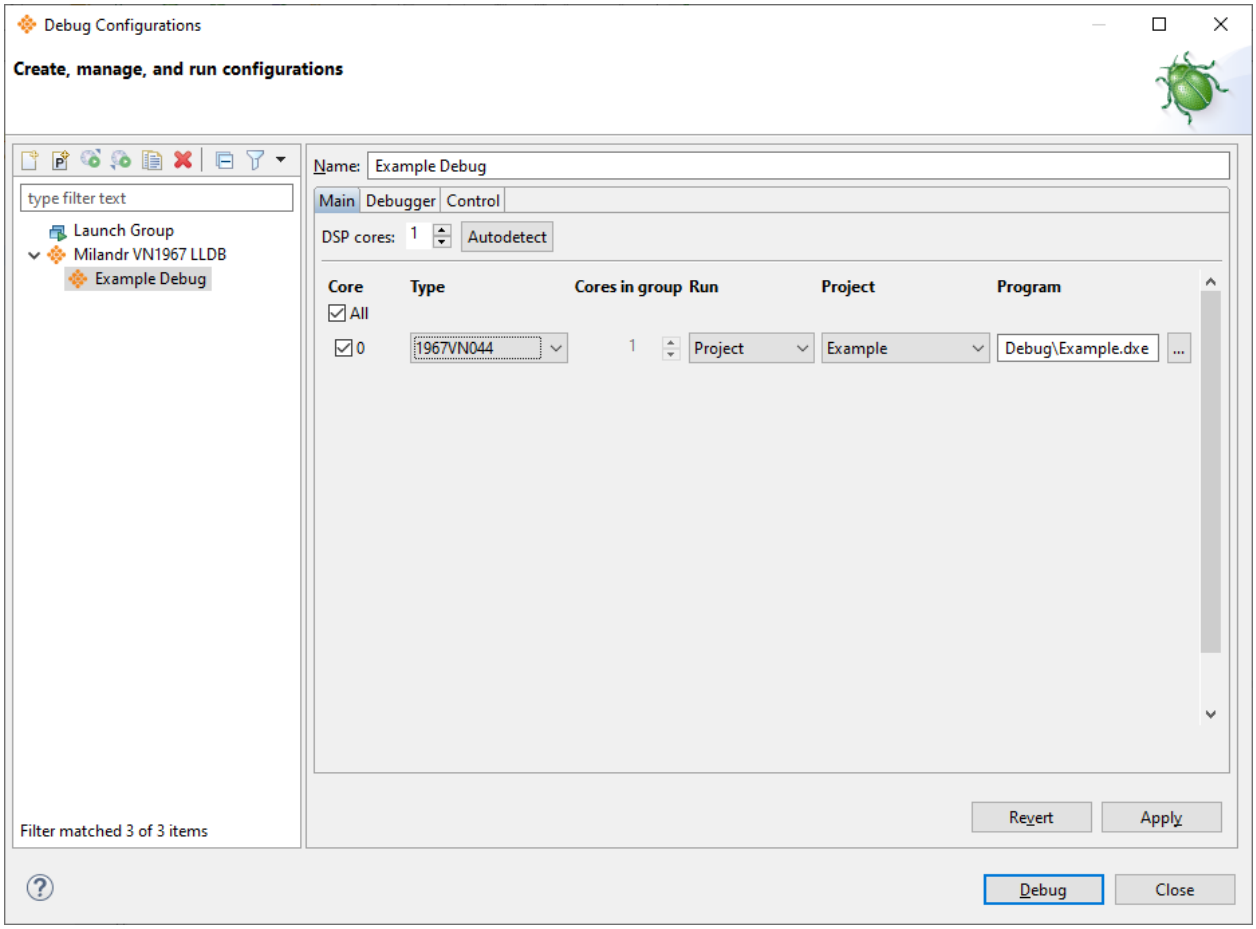


Рисунок 188. Окно «Debug Configurations», вкладка «Main»

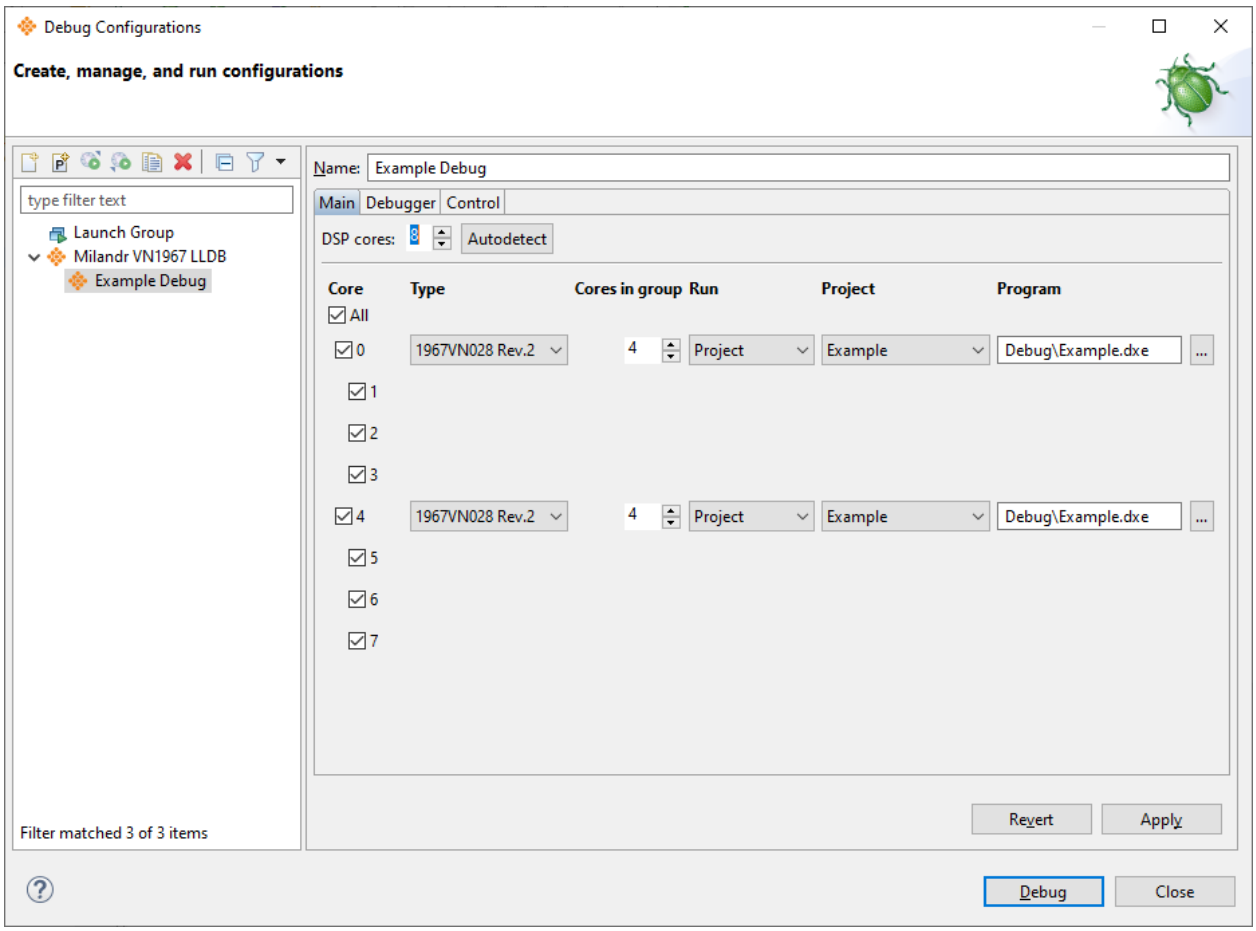


Рисунок 189. Окно «Debug Configurations», вкладка «Main» (настройка многопроцессорной конфигурации)

Таблица 92. Элементы управления вкладки «Main»

Название элемента	Описание функционирования
Name	Наименование конфигурации отладки проекта. По умолчанию: совпадает с наименованием проекта.
DSP cores	Счетчик, который позволяет добавить/удалить процессоры в конфигурацию. По умолчанию: 1. Для многопроцессорной конфигурации следует установить счетчик в значение, равное количеству процессоров в конфигурации.
Autodetect	Кнопка «**Autodetect**» позволяет получить число процессоров цепочке JTAG. Полученное значение автоматически будет установлено в поле «**DSP cores**». **Внимание**: для корректной работы кнопки «**Autodetect**» необходимо подключение к отладочной плате через аппаратный отладчик JEM.
Core	В колонке «**Core**» перечислены номера процессоров, для которых создается конфигурация отладки. Установленный флаг напротив номера процессора включает процессор в конфигурацию отладки. Для исключения какого-то процессора из конфигурации отладки следует снять флаг напротив номера этого процессора. Флаг «**All**» включает все процессоры в конфигурацию отладки.
Type	Поле для выбора типа процессора в конфигурации.

Название элемента	Описание функционирования
Cores in group	Счетчик, который позволяет объединить несколько процессоров в общую отладочную сессию симметричной отладки SMP.
Run	Поле для выбора связи с исполняемым файлом. Возможные значения: ▪ Project (проект); ▪ File (файл); ▪ Disassembly (режим, позволяющий подключиться к уже запущенному процессору в режиме дизассемблера).
Project	Поле для выбора наименования проекта. Выбор доступен из перечня открытых в среде проектов. Поле отображается, только если в списке «**Run**» выбрано значение «**Project**».
Program	Поле для задания пути к результирующему (исполняемому) файлу проекта. Поле отображается, только если в списке «**Run**» выбраны значения «**Project**» или «**File**». При выборе в списке «**Run**» значения «**File**» следует указать файл с компьютера вручную с помощью кнопки «**...**».

4. Задать параметры на вкладке «**Debugger»** (см. Рисунок 190).

   Для отладки с использованием симулятора выберите в поле «**Debugger**» значение «**SIM**». При необходимости, включите расширенный вывод отладочной информации в консоль, уставив флаг «**Verbose logging mode**».

   Для локальной отладки с использованием аппаратного отладчика JEM используйте настройки параметров по умолчанию. При необходимости, измените настройки частоты в поле «**TCK frequency**» и включите расширенный вывод отладочной информации в консоль, уставив флаг «**Verbose logging mode**».

   Для удаленной отладки с использованием аппаратного отладчика JEM:

   1) Установите переключатель «**Connection**» в режим «**Remote**».
   2) Укажите IP-адрес и порт для подключения.
   3) При необходимости, измените настройки частоты в поле «**TCK frequency**» и включите расширенный вывод отладочной информации в консоль, уставив флаг «**Verbose logging mode**».

   Подробнее о параметрах настройки на вкладке «**Debugger**» см. Таблица 93.

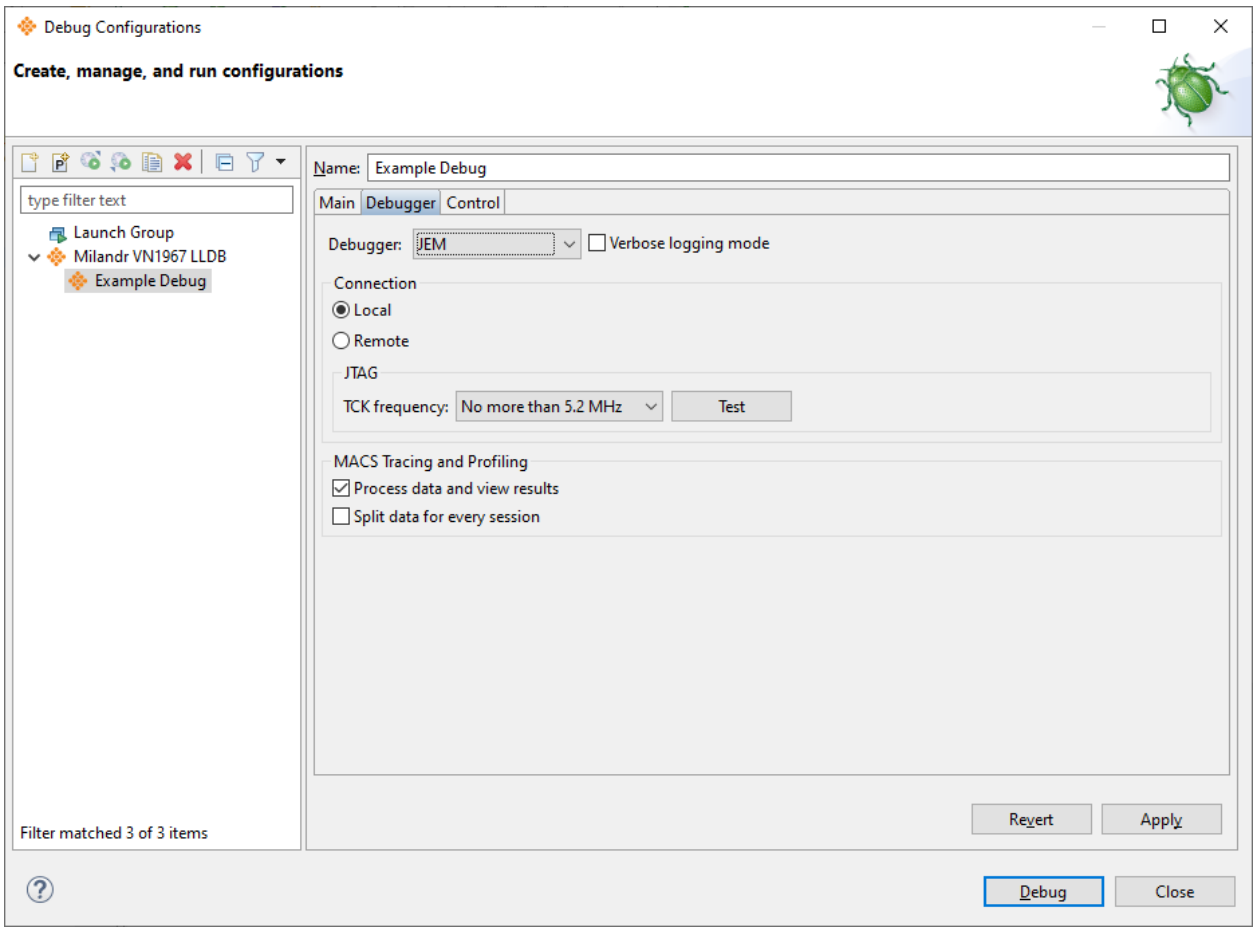


Рисунок 190. Окно «Debug Configurations», вкладка «Debugger»

Таблица 93. Элементы управления вкладки «Debugger»

Название элемента	Описание функционирования
Debugger	Поле выбора типа отладчика. Возможные значения: ▪ **JEM** (отладка с помощью аппаратного отладчика JEM); ▪ **SIM** (отладка с использованием программного симулятора). По умолчанию, установлено значение «**JEM**». **Примечание:** на текущий момент механизм отладки с использованием симулятора имеет следующие ограничения: реализована поддержка только вычислительного ядра процессоров ВН1967028/ ВН1967044/ ВН1967058. Работа с периферией (DMA, LINK-порты и т.д.) не доступна
Verbose logging mode	Флаг включения вывода расширенной отладочной информации в консоль. По умолчанию: отключен.
Connection	Выбор типа отладки. Возможные значения: ▪ **Local** (локальная отладка – отладочная плата подключена к ПК с CM-LYNX); ▪ **Remote** (удаленная отладка – отладочная плата не подключена к ПК с CM-LYNX). При отладке в режиме «**SIM**» настройка не отображается.

Название элемента	Описание функционирования
Host	IP-адрес для удаленной отладки. Поле активно, если выбран тип отладки «**Remote**»
Port	Порт для удаленной отладки. Поле активно, если выбран тип отладки «**Remote**»
TCK frequency	Выбор частоты TCK из списка. При отладке в режиме «**SIM**» настройка не отображается.
Test	Кнопка запускает тестирование JTAG-цепочки для определения числа узлов. **Внимание:** для корректной работы кнопки «**Test**» необходимо подключение к отладочной плате через аппаратный отладчик JEM. При отладке в режиме «**SIM**» кнопка не отображается.
MAKS Tracing и Profiling	В панели «**MAKS Tracing и Profiling**» задаются настройки работы приложения «МАКС Профилирование. Многопроцессорные системы» **(MACSProf**). Подробная информация о работе данного приложения описана в документе «МАКС Профилирование. Многопроцессорные системы. Руководство пользователя». Панель не отображается, если настроена отладка с помощью симулятора или удаленная отладка. Флаг «**Process Data and view result**» включает сбор резуальтатов профилирования и просмотр результатов в приложении **MACSProf**. По умолчанию флаг активен. Флаг «**Split data for every session**» позволяет для каждой сессии отладки, в которой принимаются данные профилирования МАКС, просматривать в отдельной копии приложения **MACSProf**. Если флаг «**Split data for every session**» не активен, то все сессии отладки, в которых принимаются данные профилирования МАКС, будут совмещены в одной копии приложения **MACSProf**. По умолчанию флаг неактивен.

5. Задать параметры на вкладке «**Control»** (Рисунок 191):

   - Задайте (измените) конфигурацию RESET выхода (сбросить или сохранить) при запуске и завершении отладки;
   - Для каждого процессора, SMP-группы задайте последовательность действий по загрузке приложения (остановить/оставить без изменения процессор, загрузить/не загружать приложение, точка останова, запустить/не запускать приложение) и последующие действия (остановить/не изменять/запустить выполнение) в зависимости от решаемых задач.

   Например, для подключения к процессору с запущенным исполняемым файлом в режиме «**Disassembly**» требуется снять (выключить) флаги «**Startup Hardware reset**» и «**Halt**».

   Подробнее о параметрах настройки на вкладке «**Control**» см. Таблица 94.

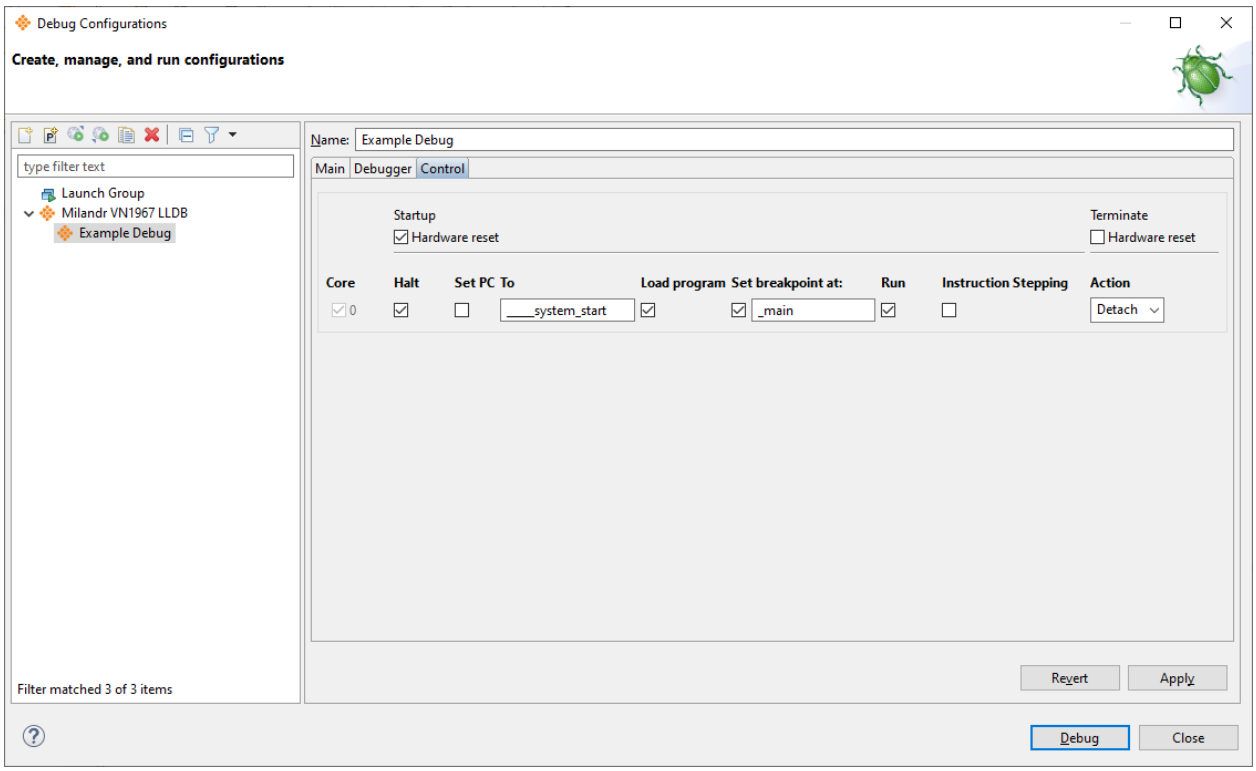


Рисунок 191. Окно «Debug Configurations», вкладка «Control»

Таблица 94. Элементы управления вкладки «Control»

Название элемента	Описание функционирования
Startup hardware reset	Флаг включения аппаратного сброса при запуске отладки. По умолчанию флаг включен. При отладке в режиме «**SIM**» настройка не отображается.
Terminate hardware reset	Флаг включения аппаратного сброса при завершении отладки. По умолчанию флаг выключен. При отладке в режиме «**SIM**» настройка не отображается.
Halt	Флаг активации остановки процессора. При включенном флаге происходит подключение и останов; при выключенном – подключение без изменения состояния.
Set PC	Флаг установки регистра **PC** при запуске отладки в значение, заданное в поле **«To»**. По умолчанию для отладочных конфигураций процессоров 1967ВН028/1967ВН044 флаг не установлен; для 1967ВН058 – установлен. **Примечание:** регистр **PC** также может быть задан с помощью скрипта инициализации session.lldbinint. Если регистр PC задан и с помощью настройки, и с помощью скрипта инициализации, то регистр PC будет выставлен в значение, заданное в срипте инициализации. Подробнее о скрипте инициализации session.lldbinint см. документ «Отладчик LLDB. Руководство пользователя».

Название элемента	Описание функционирования
To	Значение, в которое должен быть выставлен регистр **PC** перед стартом основной программы. В качестве значений допустимы: ▪ абсолютный адрес (в этом случае перед ним должна быть указано «*», например, «*0x00000000»); ▪ «_____system_start»; ▪ имя пользовательской метки. По умолчанию для всех отладочных конфигураций, кроме конфигурации типа «**Disassembly**», в поле задано значение «_____system_start». Для отладочных конфигураций типа «**Disassembly**» в поле «**To**» по умолчанию задано «*0x00000000» для 1967ВН028/1967ВН044 и значение «*0x00100000» для 1967ВН058.
Load program	Флаг активации загрузки программы. По умолчанию флаг включен. В режиме «**Disassembly**» флаг не отображается.
Set breakpoint at	Имя точки останова, на которой должен остановиться отладчик после старта программы. По умолчанию установлено значение «**main**». Для установки точки останова по абсолютному адресу нужно добавить перед ним звездочку (например, *0x2).
Run	Флаг активации/деактивации запуска выполнения программы. По умолчанию флаг включен. Для остановки по нулевому адресу следует убрать флаг «Run».
Instruction Stepping	Флаг активации/деактивации пошагового режима выполнения инструкций в дизассемблированном коде (см. также п. 6.11.5). По умолчанию флаг выключен.
Action	Выбор режима завершения сессии. Возможные значения: ▪ **Halt** (остановить); ▪ **Detach** (не изменять); ▪ **Run** (запустить выполнение). По умолчанию: **Detach** (не изменять). При отладке в режиме «**SIM**» настройка не отображается.

6. Принять изменения, нажав кнопку «**Apply**». Кнопка «**Revert**» позволяет отменить последние изменения.

7. Для запуска отладки нажать кнопку **«Debug»**.

8. Нажать на кнопку «**OK**» для переключения в режим отладки (Рисунок 192).

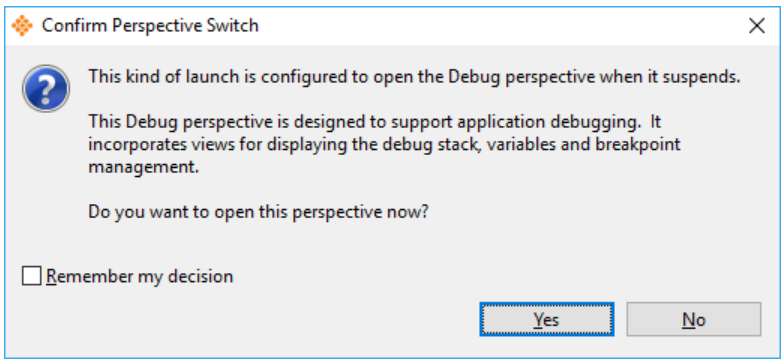


Рисунок 192. Окно «Confirm Perspective Switch»

6.11.1.1. Редактирование конфигурации отладки

При необходимости, в CM-LYNX может быть настроено более одной конфигураций отладки. Однако одновременно может быть запущен только один сеанс отладки. При необходимости изменить конфигурацию отладки необходимо остановить текущую сессию отладки и запустить новую сессию с интересующей конфигурацией. Для этого необходимо:

1. В меню «**Debug**» (см. Рисунок 186) выбрать пункт «**Debug Configurations...**».
2. Выбрать необходимую конфигурацию отладки, щелкнув по ней.
3. При необходимости, внести изменения.
4. Нажать на кнопку «**Debug**».

Информация о последних сеансах отладки отображается в меню, вызываемом кнопкой . Можно запустить сессию отладки, щелкнув по нужной конфигурации в этом списке.

6.11.1.2. Обмен настройками отладочной сессии

Для обмена настройками отладочной сессии можно воспользоваться механизмом экспорта / импорта конфигураций отладки (команды: **File-> Export.../File->Import...,->Run/Debug ->Launch Configuration**). Также можно воспользоваться новыми кнопками «**Import launch configurations**» и «**Export launch configurations**» в диалоге создания/редактирования конфигурации отладки

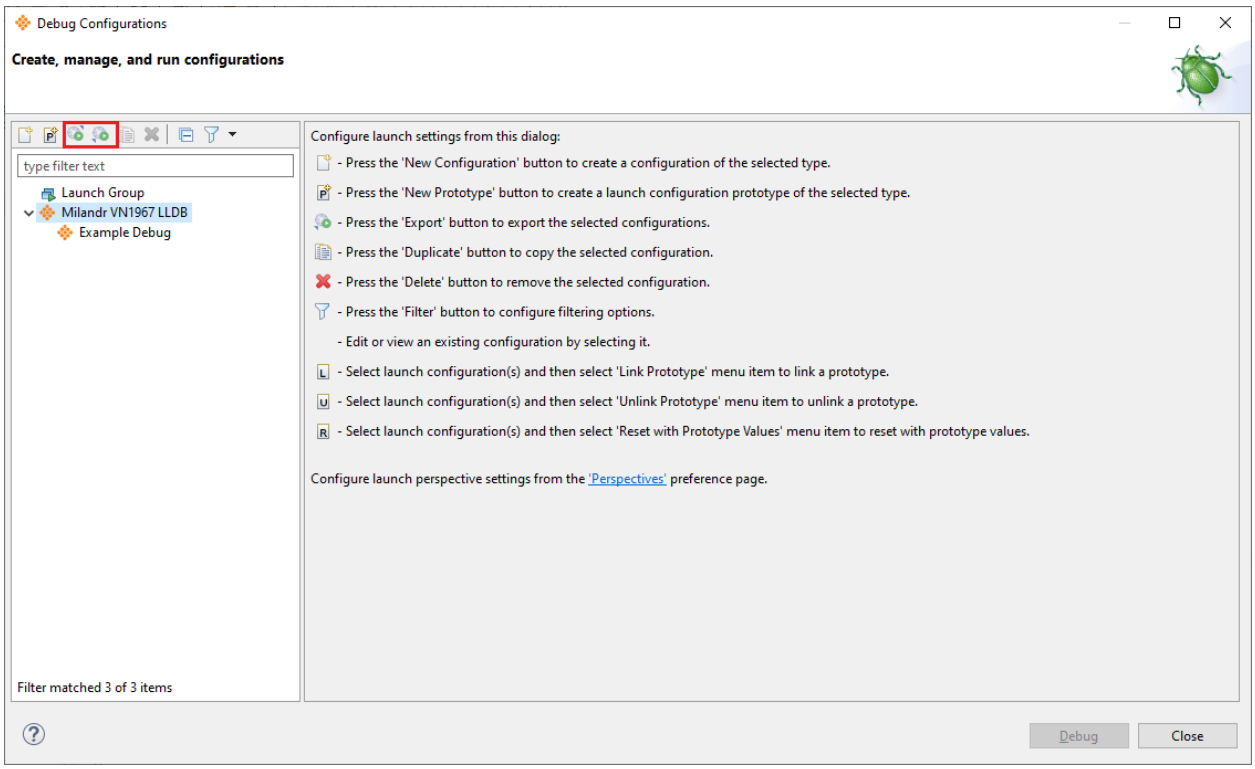


Рисунок 193. Кнопки экспорта/импорта конфигурации отладки

6.11.1.3. Включение параллельного запуска сессий отладки для многопроцессорных систем

При отладке многопроцессорной системы формируется несколько сессий отладки (их количество равно числу процессоров в многопроцессорной системе). По умолчанию, эти сессии отладки запускаются последовательно. Для повышения скорости отладки мультисессии можно включить режим параллельного запуска сессий отладки.

Для того чтобы включить параллельного запуска сессий отладки для многопроцессорных систем, необходимо:

1. Открыть окно «**Preferences**», используя команду «**Window - >Preferences**».
2. Перейти на страницу «**LLDB**» (**C/C++->Debug->LLDB**).
3. Установить флаг «**Launch sessions in parallel**» и нажать кнопку «**Apply**».

При включении режима параллельного запуска сессий отладки может потребоваться корректировка таймаутов выполнения команд отладки (особенно, при большом объеме исполняемых файлов, числе процессоров, или низкой частоте JTAG). Подробнее см. п. 6.12.2.4.4. Настройка таймаутов выполнения операций отладки.

6.11.1.4. Соответствие между конфигурациями отладки и проектами

Конфигурация отладки ассоциируется с проектом, из которого она запускает исполняемый файл. В случае многопроцессорной отладки, конфигурация ассоциируется со всеми проектами, из которых она запускает исполняемые файлы. Конфигурации, использующие только режим «**Disassembly**», не ассоциируется ни с одним проектом.

Если в настройках рабочего пространства включить запуск конфигурации отладки, ассоциированной с текущим проектом (см. п. 6.12.3.1), то при нажатии на кнопку будет запускаться последняя запущенная конфигурация только из перечня, ассоциированных с текущим проектом.

Если для текущего проекта нет асссоциированных конфигураций, то будет запускаться предыдущая из общего списка (если выбрано «If not launchable: Launch the previously launched application») или выводиться сообщение об отсутствии ассоциированных конфигураций (если выбрано «If not launchable: Launch the associated project»).

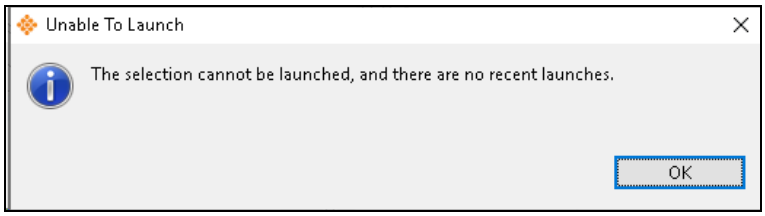


Рисунок 194. Сообщение об отсутствии ассоциированных конфигураций с текущим проектом

Данная настройка может быть полезна для отладки, требующей частых переключений между нескольких проектами. Название конфигурации, которая будет запущена, можно увидеть, если навести курсор мыши на кнопку . Настройка не влияет на перечень конфигураций, выпадающих из кнопки по нажатию на , а также на перечень, находящийся в пункте «**Debug History**» в меню «**Run**».

Перечень конфигураций, ассоциированных с данным проектом, можно увидеть на панели «**Run/Debug Settings**» в свойствах проекта.

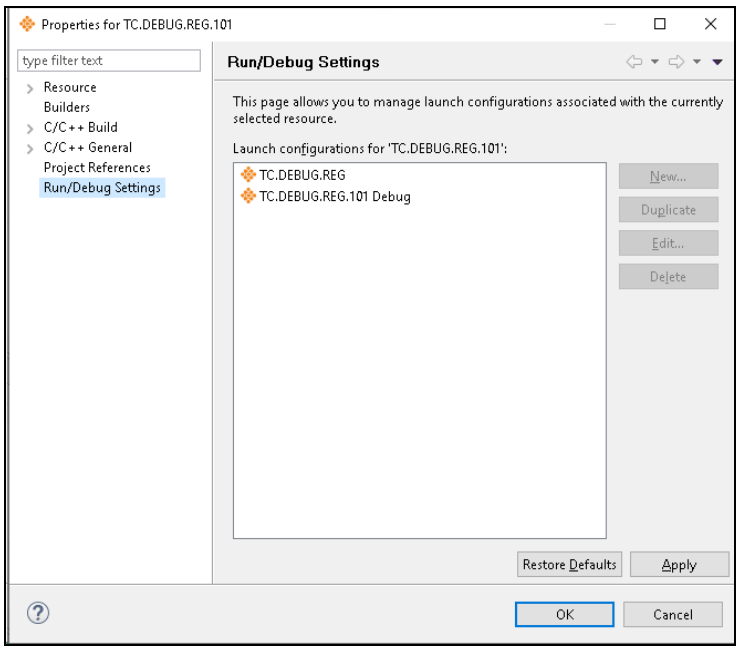


Рисунок 195. Панель «Run/Debug Settings» отображает перечень ассоциированных конфигураций

6.11.2. Операции при отладке

В процессе отладки пользователю доступны следующие действия:

1. Управление сессией отладки (запуск/останов программы и т.д.), просмотр стека вызовов. Выполняется на вкладке **«Debug»**. Подробнее см. п. 6.11.3. Управление сессией отладки.

2. Установка/снятие точек останова. Подробнее см. п. 6.11.7. Установка/снятие точек останова.

3. Установка точек наблюдения. Подробнее см. п. 6.11.8 Установка точки наблюдения.

4. Пошаговое выполнение инструкций программы. Подробнее см. п. 6.11.5. Пошаговое выполнение инструкций программы.

5. Выгрузка данных дизассемблера в текстовый файл. Подробнее см. п. 6.11.6. Выгрузка данных дисассемблера в текстовый файл.

6. Просмотр и редактирование значений регистров. Подробнее см. п. 6.11.9. Просмотр и редактирование значений регистров.

7. Просмотр и редактирование значений локальных и глобальных переменных. Подробнее см. п. 6.11.10. Просмотр и редактирование значений переменных.

8. Задание выражений и просмотр их значений. Подробнее см. п. 6.11.11. Просмотр и редактирование значений выражений.

9. Просмотр и редактирование содержимого ячеек памяти целевого процессора. Подробнее см. п. 6.11.12. Просмотр и редактирование содержимого памяти целевого процессора.

10. Одновременное управление всеми процессорами при многопроцессорной отладке. Подробнее см. п. 6.11.13. Управление процессорами при многопроцессорной отладке.

11. Просмотр результатов профилирования МАКС для проектов, разработанных с применением ОСРВ МАКС-Lynx и включенной трассировкой событий ОСРВ МАКС-Lynx. Подбробнее см. п. 6.11.15. Просмотр результатов профилирования МАКС.

6.11.3. Управление сессией отладки

Таблица 95. Основные операции по управлению сессией отладки

Операция	Команда
Завершить отладку	Одним из способов: ▪ Нажать на кнопку  на панели инструментов; ▪ Нажать на кнопку  на панели инструментов вкладки «**Debug**»; ▪ Выбрать пункт «**Terminate**» в меню «**Run**»; ▪ Выбрать пункт «**Terminate**» в контекстном меню на вкладке «**Debug**»; ▪ Нажать комбинацию клавиш «**Ctrl+F2**».
Приостановить отладку	Одним из способов: ▪ Нажать на кнопку  на панели инструментов; ▪ Нажать на кнопку  на панели инструментов вкладки «**Debug**»; ▪ Выбрать пункт «**Suspend**» в меню «**Run**»; ▪ Выбрать пункт «**Suspend**» в контекстном меню на вкладке «**Debug**».
Возобновить отладку	Одним из способов: ▪ Нажать на кнопку  на панели инструментов;

Операция	Команда
	▪ Нажать на кнопку на панели инструментов вкладки «**Debug**»; ▪ Выбрать пункт «**Resume**» в меню «**Run**»; ▪ Выбрать пункт «**Resume**» в контекстном меню на вкладке «**Debug**»; ▪ Нажать горячую клавишу «**F8**».
Запустить отладку	Одним из способов: ▪ Нажать на кнопку на панели инструментов; ▪ Выбрать пункт «**Debug**» в меню «**Run**»; ▪ Выбрать пункт «**Debug**» в контекстном меню на вкладке «**Debug**»; ▪ Нажать горячую клавишу «**F11**».
Удалить сессию отладки	Одним из способов: ▪ Нажать горячую клавишу «**Delete**».
Перезапустить отладку	Одним из способов: ▪ Удерживая клавишу «**Shift**», нажать на кнопку на панели инструментов (доступно начиная с версии 2.03.00); ▪ Выбрать пункт «**Terminate and Relaunch**» в контекстном меню на вкладке «**Debug**»;

6.11.4. Просмотр стека вызовов

Просмотр стека вызовов доступен на вкладке «**Debug**» (подробнее см. п. 5.4.3.1. Вкладка «Debug»), расположенной на левой верхней панели окна.

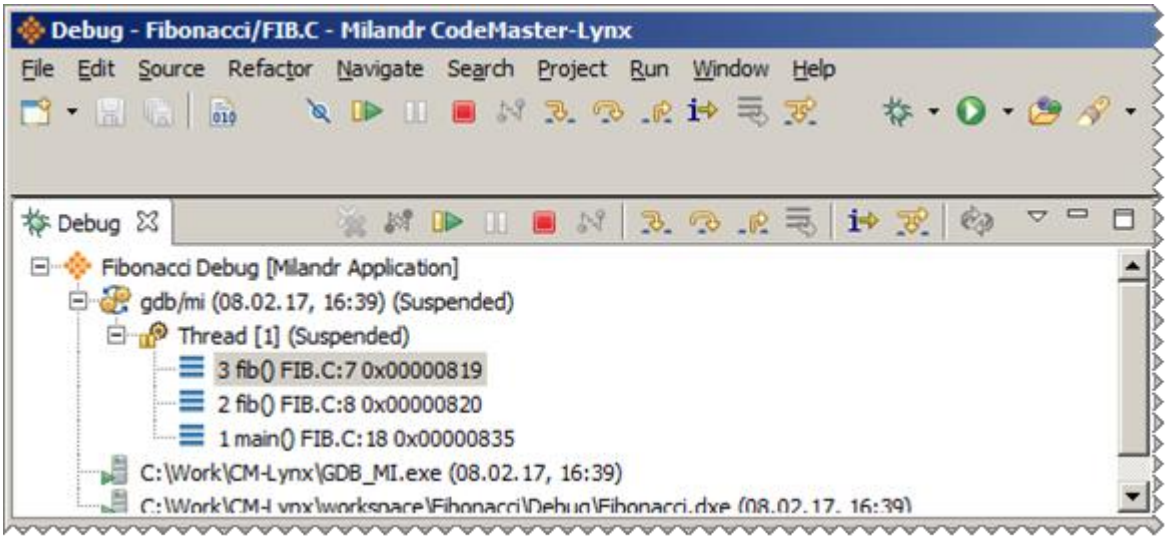


Рисунок 196. Вкладка «Debug»

Вкладка «**Debug**» отображает состояние потоков сеанса отладки:

- Running (выполняется);
- Suspended (приостановлен).

Для приостановленного потока сеанса отладки отображается стек вызовов, при этом последний шаг выполнения подсвечен серым фоном. При приостановленной сессии отладки можно:

- просматривать значения переменных на вкладке «**Variables**» выбранного шага стека вызовов. При необходимости можно перемещаться по шагам стека вызовов и просматривать соответствующие этому вызову переменные и их значения. Подробнее о просмотре переменных на вкладке «**Variables**» см. п. 6.11.10. Просмотр и редактирование значений переменных. После возобновления отладки стек вызовов и вкладка «**Variables**» будут очищены;
- просматривать значения регистров на вкладках «**Registers (Core)**» и «**Registers (Peripherals)**»;
- выполнять пошаговую отладку программы;
- просматривать содержимое ячеек памяти.

В случае разрушения стека вызовов можно попробовать воспользоваться буфером трассировки – см. п. 6.11.17**.**

6.11.5. Пошаговое выполнение инструкций программы

Контроль выполнения инструкций программы доступен в окне редактора и на вкладке «**Disassembly.**

Вкладка «**Disassembly**», расположенная на центральной правой панели окна, отображает загруженную программу в виде инструкций ассемблера вместе с исходным кодом для сравнения. Выполняемая строка отмечена стрелкой и выделена цветом. Подробнее см. п. 5.4.3.6. Вкладка «Disassembly».

Можно настроить удобное для пользователя отображение информации на вкладке «**Disassembly**»:

- включить/отключить отображение адреса, соответствующего инструкции, установив/отключив флаг «**Show Addresses**»;
- включить/отключить отображение опкодов, установив/отключив флаг «**Show Opcodes**»;
- включить/отключить отображения смещения функции, установив/отключив флаг «**Show Function Offsets**».

Для прокрутки уже загруженного диапазона адресов можно использовать колёсико мыши.

Для догрузки адресов дизассемблирования можно использовать клавиши:

вверх	Догружает предыдущий адрес
вниз	Подгружает следующий адрес
PgUp	Подгружает предыдущую страницу
PgDn	Подгружает следующую страницу

Также доступна подгрузка адресов дизассемблирования с помощью с помощью колеса мыши и кнопок "вверх"/ «вниз» вертикальной полосы прокрутки. Прокрутка вверх с помощью колеса мыши или кнопки "вверх" вертикальной полосы прокрутки обновляет содержимое окна с уменьшением начального адреса. Прокрутка вниз с помощью колеса мыши или кнопки "вниз" вертикальной полосы прокрутки обновляет содержимое окна с увеличением конечного адреса.

Для обновления содержимого представления окна «**Disassembly**» также предусмотрена кнопка «**Refresh View**».

Пошаговое выполнение используется для автономного тестирования и отладки небольших программ. При пошаговом выполнении программы код выполняется строчка за строчкой. В CM-LYNX возможны следующие команды пошагового выполнения:

- **Step Into** – если выполняемая строчка кода содержит вызов функции, процедуры, то происходит вызов, и программа останавливается на первой строчке вызываемой функции, процедуры;
- **Step Over** - если выполняемая строчка кода содержит вызов функции, процедуры или метода, то происходит вызов и выполнение всей функции и программа останавливается на первой строчке после вызываемой функции;
- **Step Return** - предназначена для выхода из функции в вызывающую функцию. Эта команда продолжит выполнение функции и остановит выполнение на первой строчке после вызываемой функции.

Таблица 96. Основные операции по контролю выполнения инструкций программы

Операция	Команда
Выполнение программы до курсора	Одним из способов: ▪ нажать на кнопку на панели инструментов; ▪ выбрать пункт «**Run to line**» в меню «**Run**»; ▪ выбрать пункт «**Run to line**» в контекстном меню, вызываемом щелчком правой клавишей мыши, вкладки «**Disassembly**» или окна редактора; ▪ с помощью комбинации клавиш **Ctrl-R**.
Включение/отключение режима пошагового выполнения программы	Одним из способов: ▪ нажать на кнопку на панели инструментов; ▪ выбрать пункт «**Instruction Stepping Mode**» в меню «**Run**»; ▪ включить флаг «**Instruction Stepping**» в используемой конфигурации отладки.
Шаг с заходом в функцию (**Step Into**)	Одним из способов: ▪ выбрать пункт «**Step Into**» в меню «**Run**»; ▪ нажать на кнопку на панели инструментов вкладки «**Debug**»; ▪ выбрать пункт «**Step Into**» в контекстном меню вкладки «**Debug**»; ▪ нажать клавишу **F5**.
Шаг без захода в функцию (**Step Over**)	Одним из способов: ▪ выбрать пункт «**Step Over**» в меню «**Run**»; ▪ нажать на кнопку на панели инструментов вкладки «**Debug**»; ▪ выбрать пункт «**Step Over**» в контекстном меню вкладки «**Debug**»; ▪ нажать клавишу **F6**.
Возврат в вызывающую функцию (**Step Return**)	Одним из способов: ▪ выбрать пункт «**Step Return**» в меню «**Run**»; ▪ нажать на кнопку на панели инструментов вкладки «**Debug**»;

Операция	Команда
	▪ выбрать пункт «**Step Return**» в контекстном меню вкладки «**Debug**»; ▪ нажать клавишу **F7**.

Примечание. В некоторых случаях, из-за особенностей архитектуры и отладчика, работа функции «Step Return», а также других команд пошагового выполнения, может происходить не совсем корректно. Если данные эффекты наблюдаются, рекомендуется включить опцию «Generate code tuned for debugger» (см. п. 6.9.2.4.1.5.6) на время отладки для соответствующего языка или файла. Она приводит к формированию менее оптимального кода, однако нормализует работу команд пошагового выполнения.

Примечание. Команды, описанные ниже, предназначены для опытных разработчиков, так как допускают перенос выполнения на произвольный адрес, в том числе в коде другой функции. При некорректном использовании это может привести к искажению или разрушению стека вызовов.

Таблица 97. Продвинутые операции по контролю выполнения программы

Операция	Команда
Перенести выполнение на указанную строку (**Move to Line**)	Одним из способов: ▪ нажать на кнопку  на панели инструментов; ▪ выбрать пункт «**Move to line**» в меню «**Run**»; ▪ выбрать пункт «**Move to line**» в контекстном меню, вызываемом щелчком правой клавишей мыши, вкладки «**Disassembly**» или окна редактора.
Запустить выполнение с указанной строки (**Resume at Line**)	Одним из способов: ▪ нажать на кнопку  на панели инструментов; ▪ выбрать пункт «**Resume at line**» в меню «**Run**»; ▪ выбрать пункт «**Resume at line**» в контекстном меню, вызываемом щелчком правой клавишей мыши, вкладки «**Disassembly**» или окна редактора.

6.11.6. Выгрузка данных дисассемблера в текстовый файл

Для того чтобы экспортировать содержимое дисассемблера в текстовой файл, необходимо:

1. Нажать на кнопку .
2. Задать диапазон экспортируемого участка с помощью параметров:

Start address	Адрес, начиная с которого будут выгружены данные. Автоматически проставляется значение, на котором находится курсор в окне «**Disassembly**»
End address	Адрес, по который будут выгружены данные. Если задать значение поля «**Length**» поле будет вычислено автоматически.
Length	Длина экспортируемого участка. Если задано значение поля «**End address**» поле заполняется автоматически.

3. Указать путь к файлу, в который будут выгружены данные, в поле «**File name**».

4. Выбрать, какие именно данные будут выгружены в файл, проставив нужные флаги:

Address	Адреса
Opcodes	Опкоды
Offsets	Смещения
Symbols	Символы
Sources	Источники

5. Нажать на кнопку «**OK**».

Примечание: данные выгружаются непосредственно из дисассемблера, поэтому содержимое вкладки «**Disassembly**» и результирующего текстового файла может не совпадать.

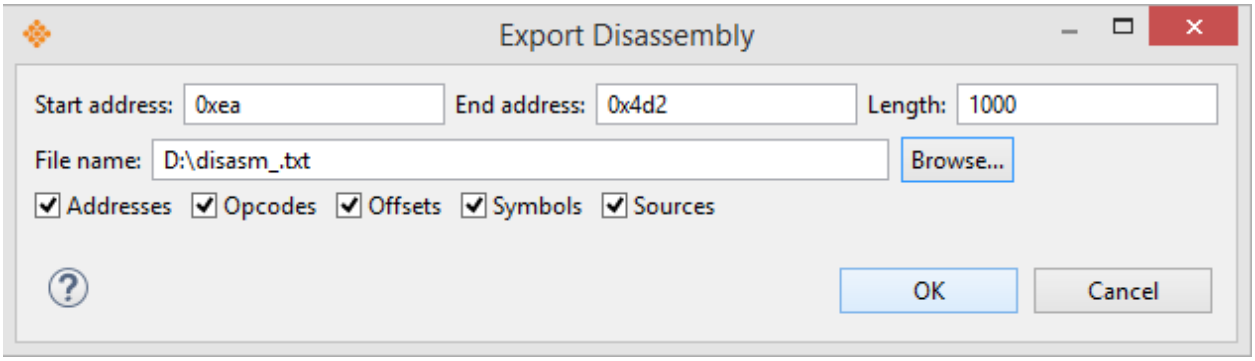


Рисунок 197. Выгрузка данных дисассемблера в текстовой файл

6.11.7. Установка/снятие точек останова

Точка останова (breakpoint, контрольная точка, точка прерывания) – точка программы, при достижении которой выполнение отлаживаемой программы временно приостанавливается.

Точки останова позволяют приостанавливать выполнение программы именно в том месте и в то время, когда это необходимо. Вместо того чтобы перемещаться по коду от строки к строке или от инструкции к инструкции, можно разрешить выполнение программы до тех пор, пока она не достигнет точки остановки, а затем начать поиск и устранение ошибок.

Когда выполнение в точке останова останавливается, отлаживаемая программа переходит в режим приостановки. В этом режиме все элементы, например, функции, переменные и объекты, сохраняются в памяти, но их перемещения и активность приостанавливаются. Во время режима приостановки выполнения можно произвести поиск ошибок и нарушений целостности данных, проверив положения элементов и их состояние.

Можно одновременно указать несколько точек останова в разных местах кода, в том числе и в разных модулях программы. Приложение будет выполняться до тех пор, пока управление не перейдет к первой встретившейся в программе точке останова.

Работа с точками останова доступна:

- на вкладке «**Breakpoints**» (см. п. 5.4.3.4. Вкладка «Breakpoints»);
- на вкладке «**Disassembly**» (см. п. 5.4.3.6. Вкладка «Disassembly»);
- в окне редактора.

Для установки точки останова необходимо дважды щелкнуть клавишей мыши на нужной строке в рабочей области редактора или на вкладке «**Disassembly**». Удаление установленной точки останова осуществляется аналогично.

Если отладочная сессия запущена с проектом или загруженный файл dxe имеет полноценный блок отладочной информации (секция «debug_info»), то на вкладке «**Disassembly**» можно поставить точки останова двух типов:

- Address Breakpoint (точка останова по адресу);
- Line Breakpoint (точка останова на строке).

Примечание: для установки *Line Breakpoint* необходимо включить режим «**Show Source**».

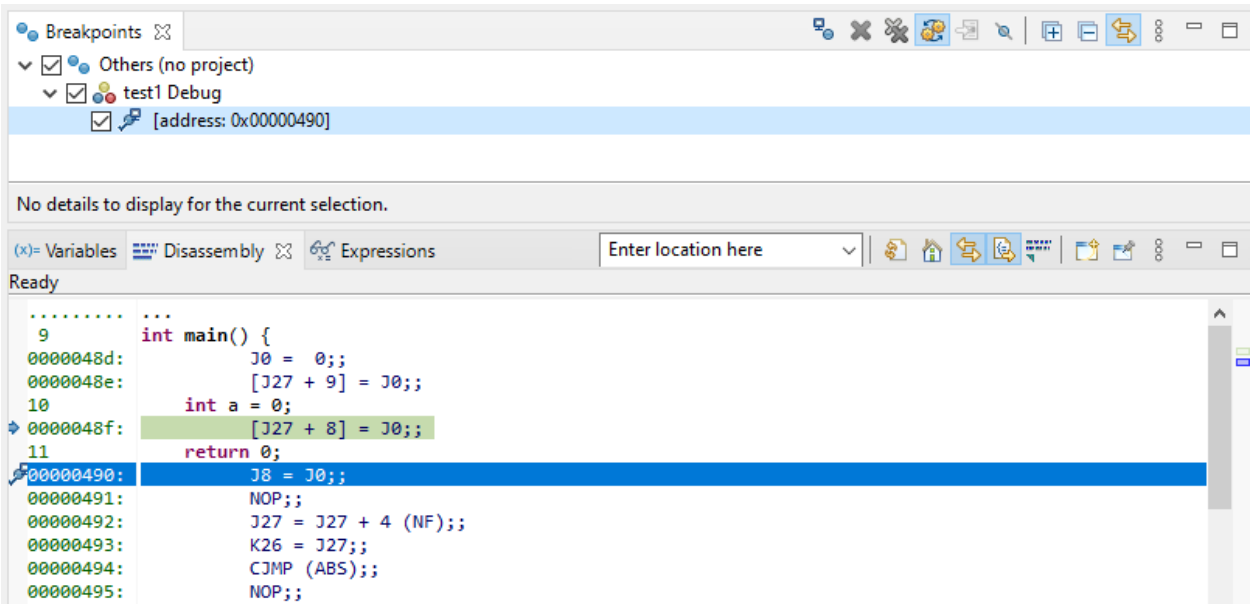


Рисунок 198. Установка точки останова типа «Address Breakpoint»

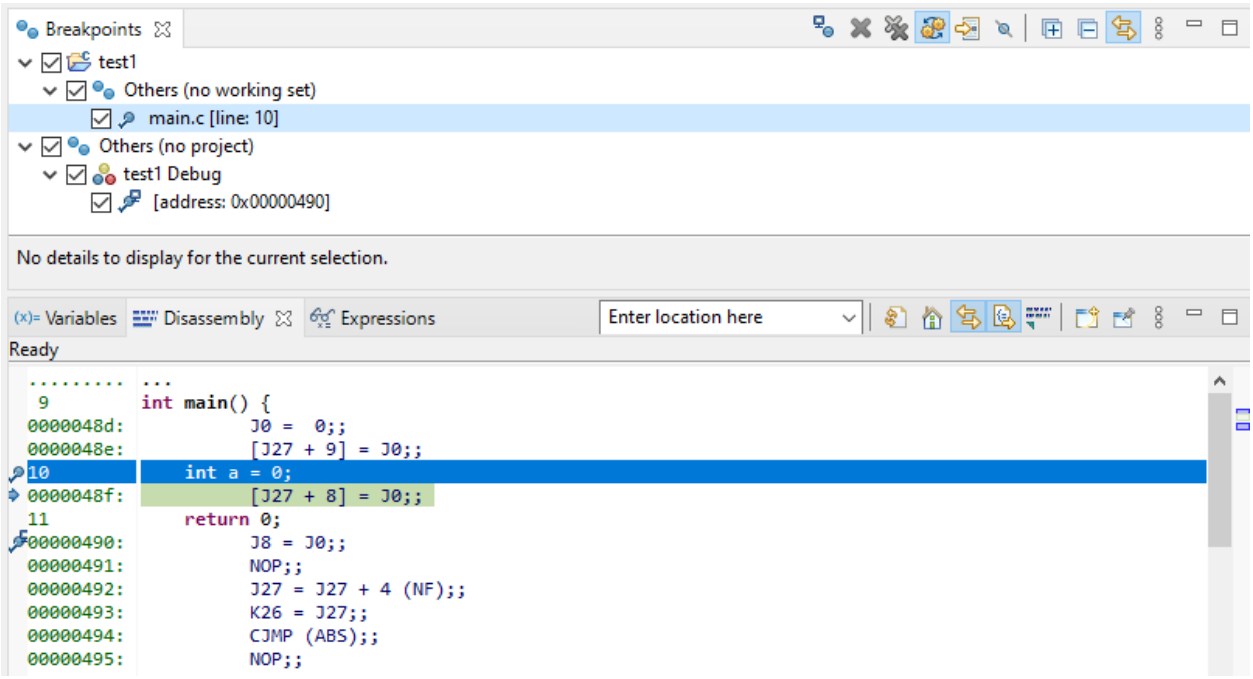


Рисунок 199. Установка точки останова типа «Line Breakpoint»

Если отладочная сессия запущена с файлом (без полноценной отладочной информации) или в режиме «Disassembly», то на вкладке «**Disassembly**» можно поставить только точку останова типа *Address Breakpoint*.

Начиная с версии CM-LYNX 2.08.00, точки останова типа **«Line Breakpoint»** по-умолчанию привязаны к проекту, а адресные – к отладочной конфигурации.

Привязка означает, что работать такие точки останова будут только при отладке соответствующего проекта или отладочной конфигурации, а в других – не будут. Привязанные к проекту – будут работать только при отладке отладочной конфигурации, связанной с данным проектом. Привязанные к отладочной конфигурации – только при отладке данной конфигурации.

До версии 2.08.00 все расставленные в разных проектах и сессиях точки останова были общими в рамках рабочего пространства.

Адресную точку останова можно также установить по кнопке «**Add address breakpoint**» в окне «**Breakpoints**». При запущенном сеансе отладки, для остановленной программы, автоматически подставляется текущее значение адреса останова, также такая точка останова будет привязана к текущей отладочной конфигурации. Если же создавать адресную точку останова при не запущенном сеансе отладки, то такая точка останова не будет привязана к отладочной конфигурации и будет работать во всех.

Точку останова типа **«Line Breakpoint»** можно установить в окне редактора.

Установленные точки останова отображаются на вкладке «**Breakpoints**». При включенной кнопке **«Show Breakpoints Supported by Selected Target»** отображаются только те точки останова, которые будут срабатывать в текущей отладочной конфигурации. При запущенной отладке точки останова, которые были фактически активированы (т.е. переданы процессору), снабжаются графическим флажком «✔». Например, на Рисунок 200 приведён вид вкладки «**Breakpoints**» при отладке конфигурации **«test1 Debug»**, активированы точки останова на строке 10 и на адресе 0x49d, а остальные – нет.

Начиная с версии 2.09.00 пункт меню **«Sort By»** в окне **«Breakpoints»** полностью функционален. По умолчанию выбран режим «**Creation Time»**; точки останова отображаются сверху вниз в порядке их создания (самые новые — выше). Выбранная сортировка работает вне зависимости от режима **«Group By»** и от того, запущена ли сессия отладки. Сортировка влияет только на порядок отображения в IDE и не изменяет фактический порядок регистрации точек останова в целевом процессоре.

Двойным щелчком мыши в строке с выбранной точкой останова можно найти ее положение в редакторе (если применимо). Также можно включить или выключить точки останова, удалить их, экспортировать или импортировать точки останова, задать свойства для точек останова.

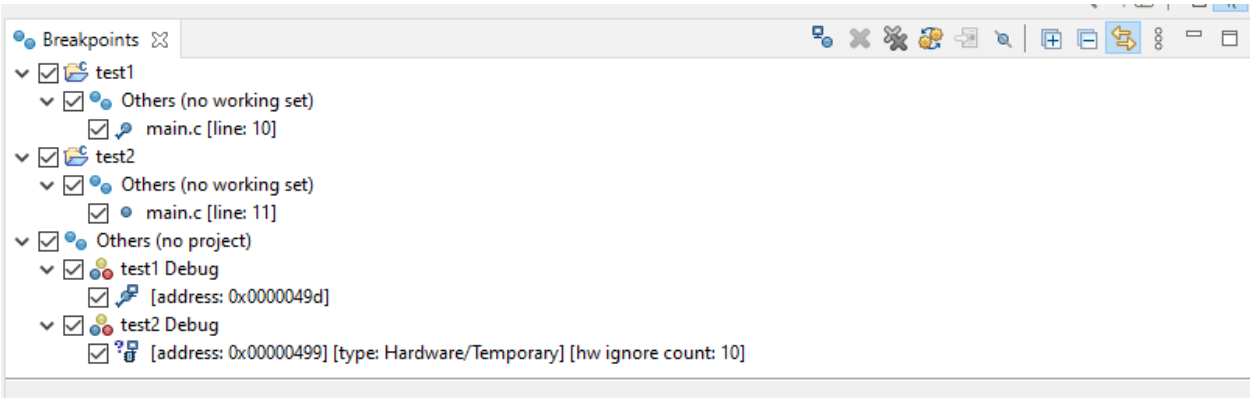


Рисунок 200. Окно просмотра точек останова

Примечания:

1. Из-за особенностей работы процессоров серии 1967ВН0ХХ программные точки останова реализованы с помощью модификации программного кода (в точке останова первый опкод в линии

инструкций заменяется на инструкцию "ловушка эмулятора"). Однако в окнах «**Disassembly**», «**Memory**» и т.д. отображается исходное содержимое памяти. Данную особенность необходимо учитывать при отладке программ, выполняющих контроль целостности исполняемого кода, или производящих чтение или модификацию исполняемого кода "на лету" для других целей.

2. По этой же причине адресные точки останова срабатывают, только если оказываются на первом опкоде в линии инструкций. Установка адресной точки останова на второй, третий или четвёртый опкод может также привести к неправильному выполнению линии инструкций. При попытке установки адресной точки останова на вкладке «**Disassembly**» по двойному щелчку или с помощью контекстного меню «**Toggle Breakpoint**» на не первый опкод в линии пользователю будет выведено диалоговое окно с уточняющим вопросом (см. Рисунок 201). В случае ответа «**No**», точка останова добавлена не будет. В случае включения флага «**Do not ask me again**», точка останова не будет добавляться без диалогового окна.

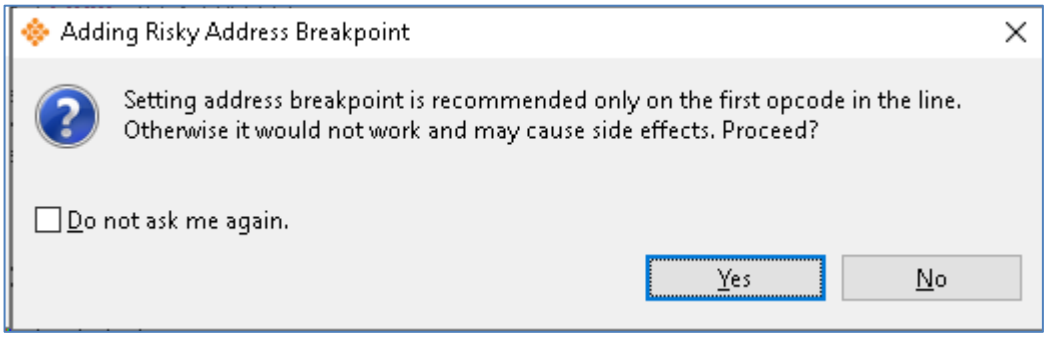


Рисунок 201. Диалоговое окно «Adding Risky Address Breakpoint»

Восстановить появление диалогового окна можно с помощью настройки «**Prompt for confirmation when adding risky address breakpoint**», см. п. 6.12.3.

Установка адресных точек останова через диалоговое окно редактирования точки останова выполняется на любой опкод в линии.

3. Адресная точка останова может оказаться на не первом опкоде после сборки модифицированного проекта из-за иного расположения инструкций в новом коде. В связи с чем адресные точки останова необходимо применять с осторожностью при отладке, сопровождающейся изменениями в коде.

Все адресные точки останова, привязанные к отладочным конфигурациям проекта, по умолчанию отключаются при сборке данного проекта (если сборка сопровождается генерацией новых файлов), а не привязанные – при сборке любого проекта. За данное поведение отвечает настройка «**Disable address breakpoints and watchpoints on rebuild**», см. п. 6.12.3.

6.11.7.1. Задание свойств для точки останова

Для того чтобы просмотреть и отредактировать свойства точки останова, необходимо:

1. Выбрать точки останова и в контекстном меню на вкладке «**Breakpoints**» выбрать пункт «**Breakpoint Properties...**».

2. Изменить необходимые поля на вкладке «**Common**». Например, можно изменить тип точки останова в поле «**Type**».

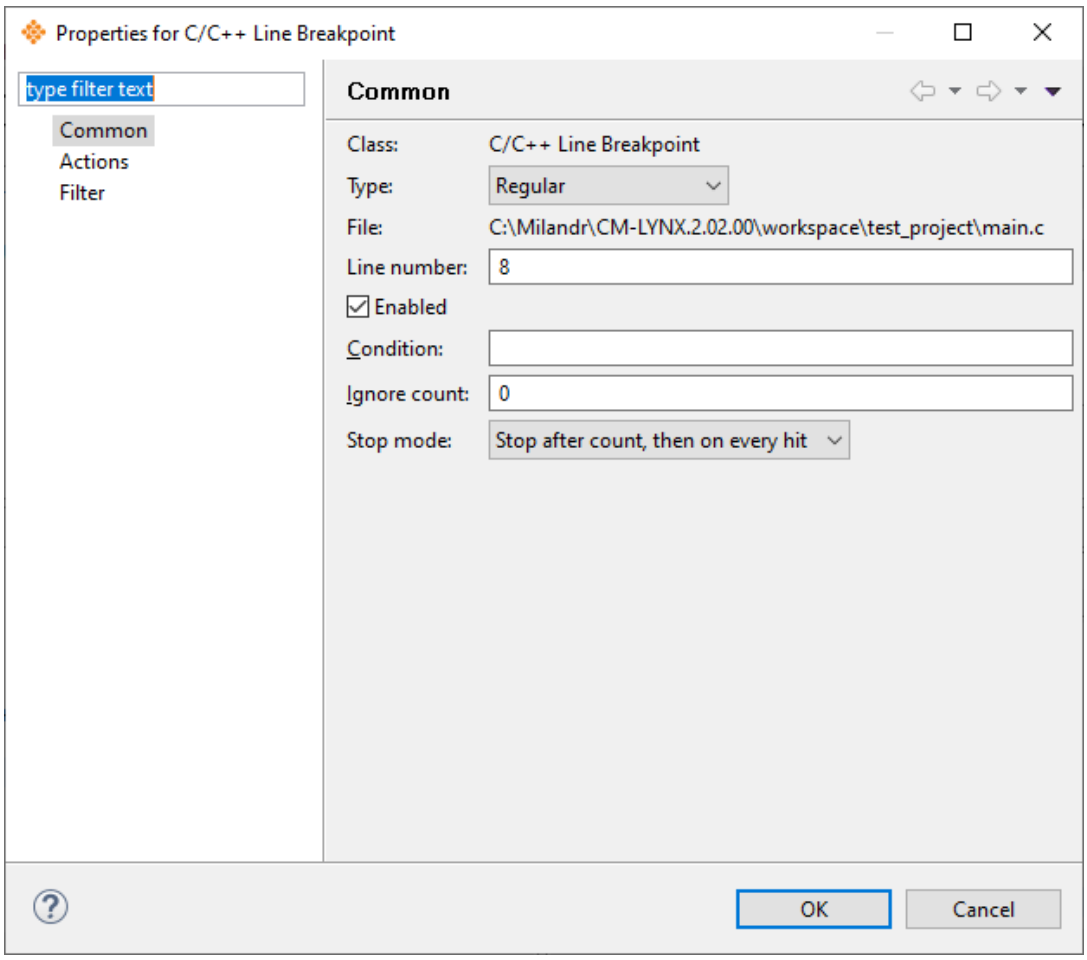


Рисунок 202. Окно «Properties for C/C++ Line Breakpoint». Вкладка «Common»

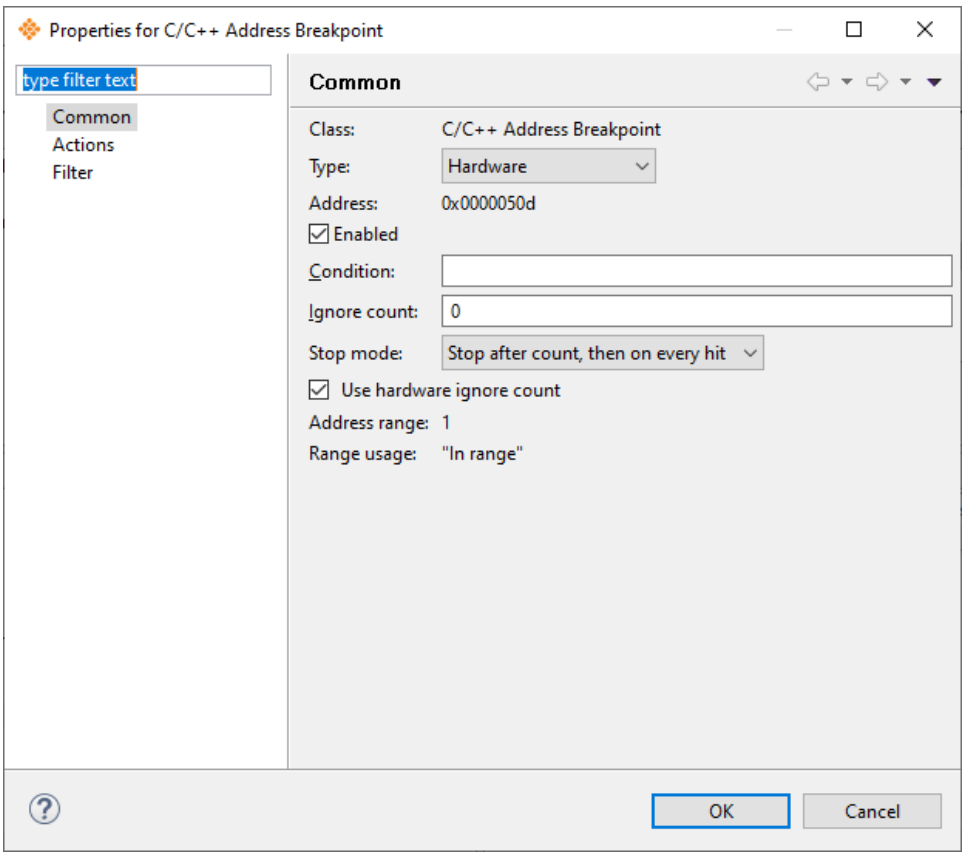


Рисунок 203. Окно «Properties for C/C++ Address Breakpoint». Вкладка «Common»

Вкладка «**Common**» содержит следующие поля:

Type

Тип точки останова.

Поддерживаются следующие:

- Regular (программная);
- Hardware (аппаратная);
- Temporary (временная программная);
- Hardware Temporary (временная апппаратная).

Процессоры серии 1967 имеют один стандартный модуль WP0, который используется для внутренней работы отладчика при выполнении команд пошаговой отладки. Начиная с версии 2.02.00, WP0 также применяется для установки пользовательской аппаратной точки останова.

Аппаратная точка останова, как и программная, может быть зарегистрирована по абсолютному адресу или по номеру строки в исходном файле.

Адресная аппаратная точка останова может быть использована для определения факта выхода программы за пределы ее стандартных адресов выполнения с целью перехвата исключений (прерываний) или в других специальных случаях.

Врменные точки останова (Temporary и Hardware Temporary), в отличие от регулярных (Regular и Hardware), срабатывают один раз.

Address

Только для адресных точек останова. Нередактируемое поле. Адрес устанавливается при установке адресной точки останова.

File

Файл, в котором установлена точка останова. Только для Line Breakpoint. Нередактируемое поле. Устанавливается при установке точки останова на строку.

Line

Номер строки, на которой установлена точка останова. Только для Line Breakpoint. Нередактируемое поле. Устанавливается при установке точки останова на строку.

Condition

Условие срабатывания, при выполнении которого точка останова должна сработать. Может отсутствовать.

Ignore count

Счетчик пропуска срабатываний. По умолчанию - 0 (счетчик не используется).

Пропуск установленного не нулевого числа срабатываний выполняется только при выполнении условия срабытывания, если оно также задано.

Stop mode

Режим остановки в случае срабатывания точки останова.

Возможные варианты:

- Stop after count, then on every hit (выполняется останов после заданного числа пропусков срабатываний "Ignore count" только 1 раз, после чего последующие остановы выполняются по каждому срабатыванию точки);
- Stop after every count (останов выполняется всегда после заданного числа пропусков срабатываний "Ignore count").

Только для регуляирных точек останова (с типами Regular и Hardware).

Use hardware Ignore count

Флаг "Use hardware ignore count" позволяет использовать аппаратный счетчик пропуска срабатываний. Только для аппаратных точек останова.

Если флаг установлен и отсутствует условие срабатывания по выражению "Condition", применяется аппаратный счетчик пропуска срабатываний и его автоматический программный перезапуск при превышении диапазона аппаратного счетчика (у WPi используются 16-ти битные счетчики).

Примечание. Опцию рекомендуется всегда оставлять включенной, так как это на порядки повышает производительность пропуска срабатываний и позволяет выполнять прогон фрагментов кода почти в реальном времени (с учетом времени на перезапуск). Максимальное значение "Ignore count" - UINT32_MAX.

Adress range

Количество 32-х разрядных слов интервала. Только для аппаратных точек останова. Нередактируемое. Настраивается при установке точки останова.

Range usage

Способ применения интервала.

Возможные значения:

- In range (точка останова срабатывает при выполнении команды процессора по любому адресу в интервале);

- Out of range (точка останова срабатывает по любому адресу за пределами заданного интервала).

Только для аппаратных точек останова. Нередактируемое. Настраивается при установке точки останова.

3. Перейти на вкладку «**Actions**» и настроить список действий при срабатывании точки останова, при необходимости.

   Для того чтобы добавить действие, необходимо выбрать нужное действие из списка «**Available actions**» (нижняя панель) и нажать кнопку «**Attach**».

   При наличии нескольких настроенных действий их последовательность выполнения можно менять, используя кнопки «**Up**» и «**Down**».

   Удалить действие из списка «**Actions for this breakpoint**» можно с помощью кнопки «**Remove**».

   При отсутствии нужного действия в списке «**Available actions**» можно его добавить. Для этого необходимо:

   1) Нажать на кнопку «**New...**».
   2) В окне «**New Breakpoint Action**» указать имя действия, тип действия, дополнительные параметры и нажать на кнопку «**OK**».

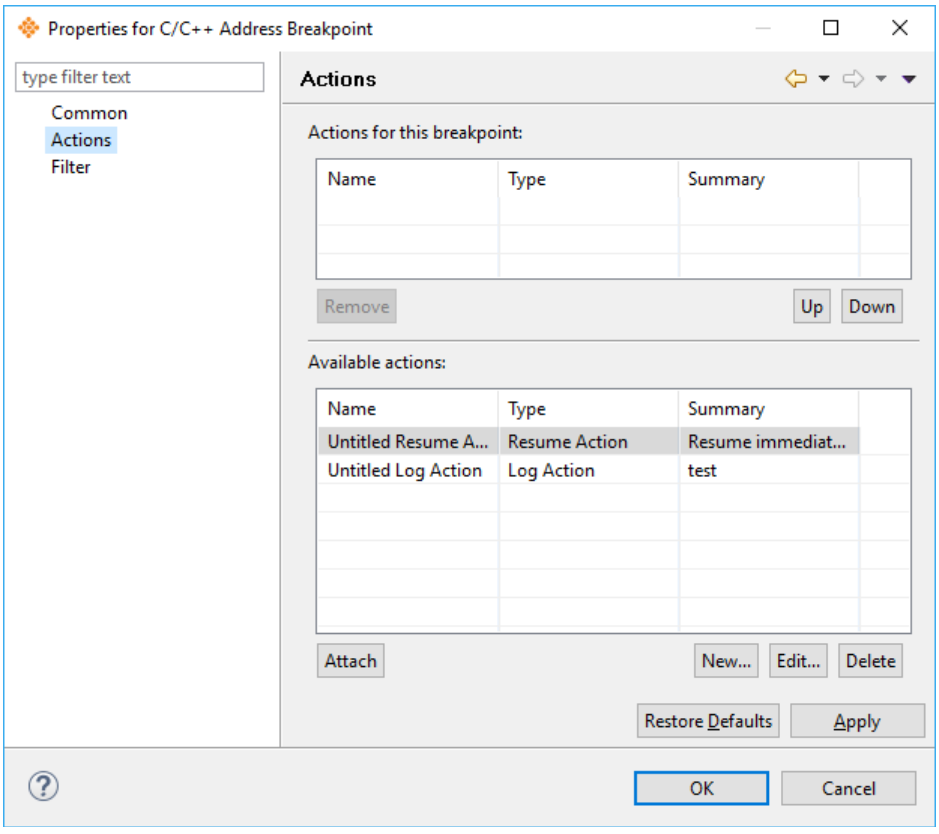


Рисунок 204. Окно «Properties for C/C++ Line Breakpoint». Вкладка «Actions»

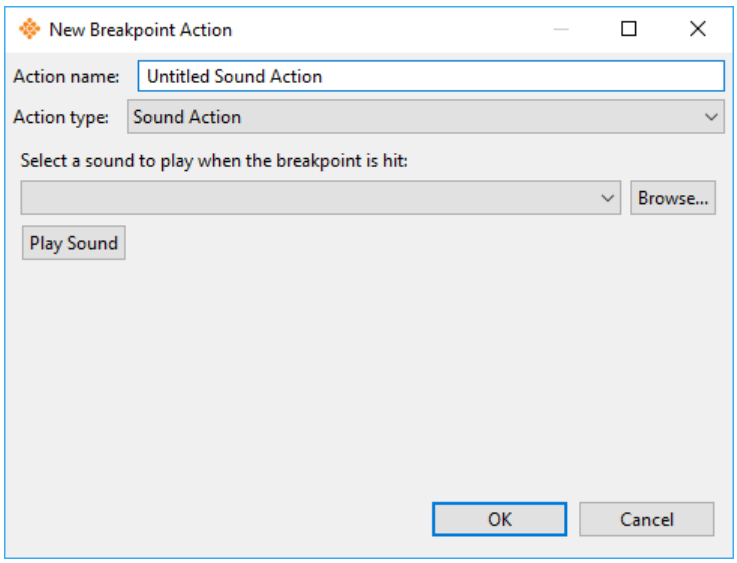


Рисунок 205. Окно «New Breakpoint Action»

4. Перейти на вкладку «**Filter**» и настроить ограничения при необходимости.

5. По завершению настройки следует нажать кнопку «**Apply**».

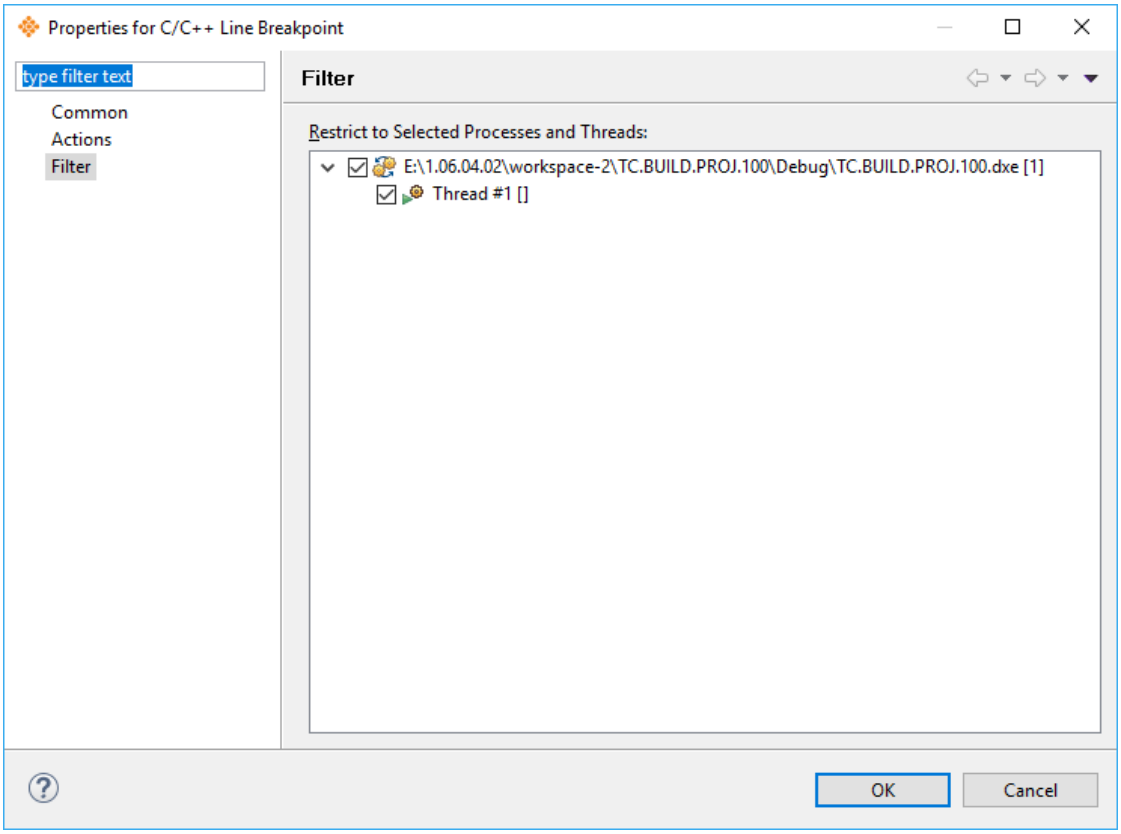


Рисунок 206. Окно «Properties for C/C++ Line Breakpoint». Вкладка «Filter»

6.11.7.2. Импорт точек останова

При необходимости, точки останова могут быть импортированы в среду из файла. Для этого необходимо:

1. Выбрать в контекстном меню на вкладке «**Breakpoints**» пункт «**Import Breakpoints...**».

2. Указать путь к файлу с информацией о точках останова, который следует импортировать.

3. Изменить при необходимости значения флагов:

Update existing breakpoints	Обновить существующие точки останова. Если флаг включен (по умолчанию), то при импорте будут добавлены только точки останова, отличные от существующих в проекте
Create breakpoint working set	Создать рабочий набор точек останова. Если флаг отключен (по умолчанию), рабочий набор не создается

4. Нажать на кнопку «**Next**».

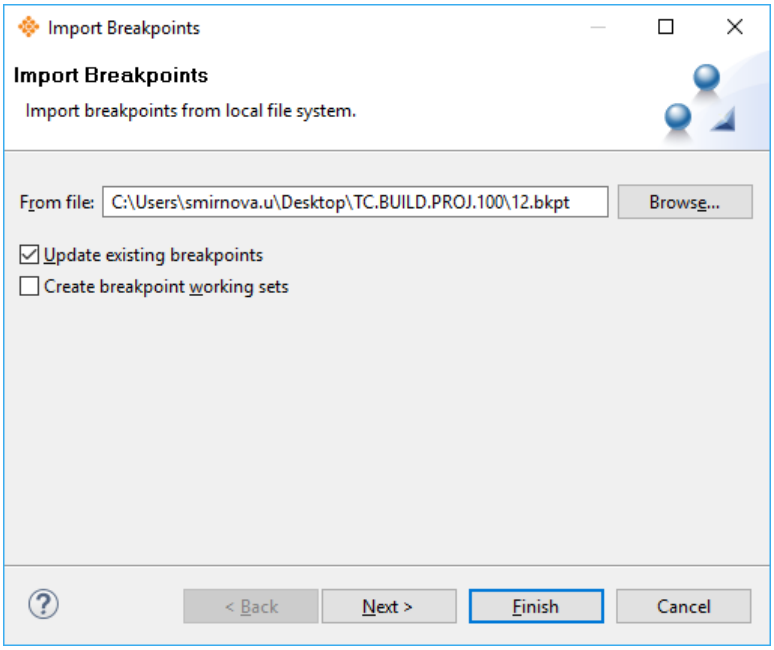


Рисунок 207. Окно «Imports Breakpoints». Выбор файла

5. По умолчанию, импортируются все точки останова из файла.

   Если необходимо какие-то из точек останова не импортировать, следует снять напротив них флаги.

6. Нажать на кнопку «**Finish**».

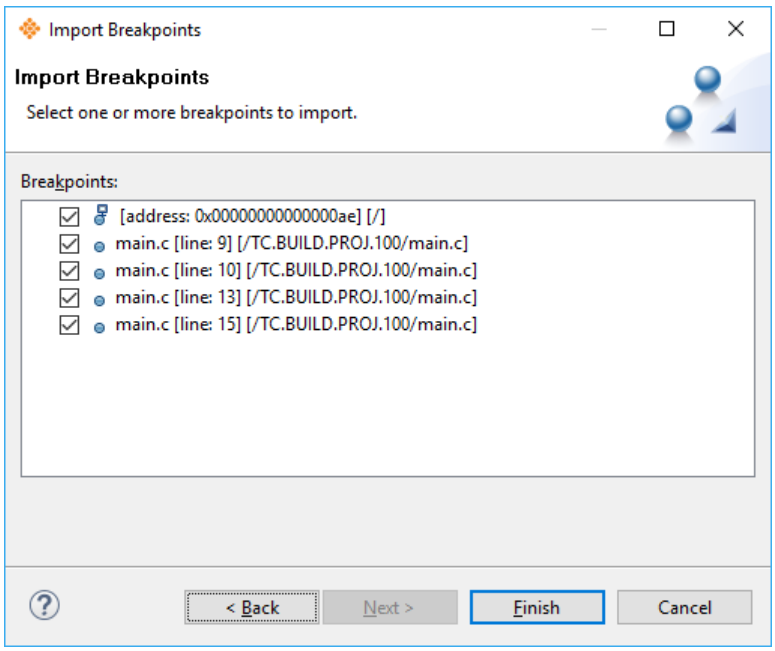


Рисунок 208. Окно «Imports Breakpoints». Выбор точек останова для импорта

6.11.7.3. Экспорт точек останова

При необходимости, точки останова могут быть экспортированы в файл. Для этого необходимо:

1. Выбрать в контекстном меню на вкладке «**Breakpoints**» пункт «**Export Breakpoints...**».
2. Указать точки останова для экспорта, проставив флаги напротив экспортируемых.
3. Указать файл, в который будут экспортированы точки останова.
4. Нажать кнопку «**Finish**».

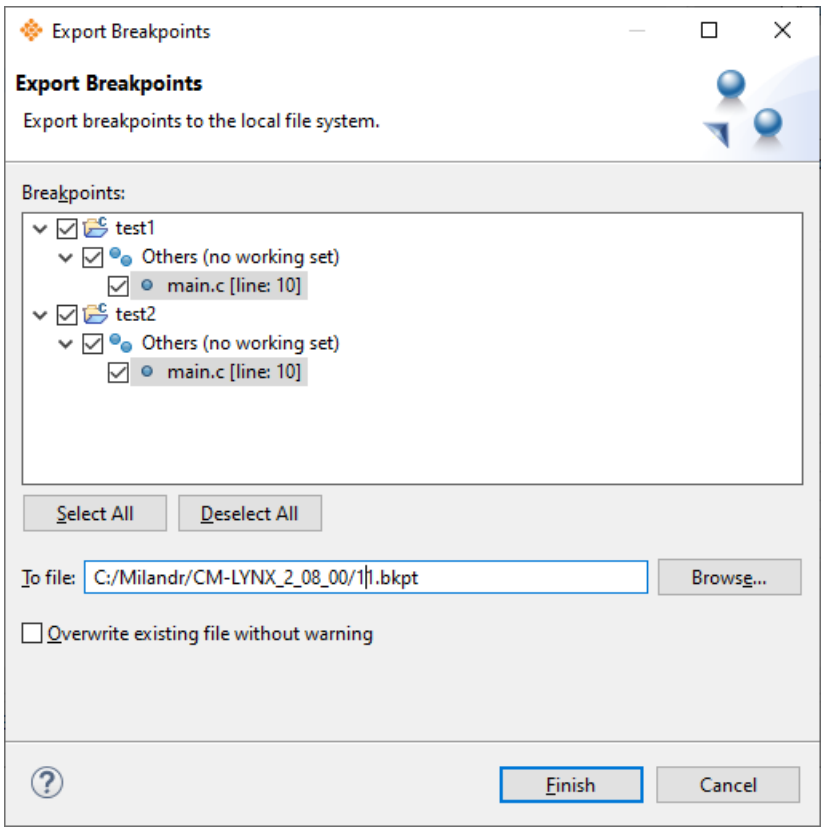


Рисунок 209. Окно «Export Breakpoints»

6.11.8. Установка точки наблюдения

Точки наблюдения «**Watchpoint**» останавливают выполнение при изменении значения какой-либо переменной или при обращении к переменной.

Для того чтобы установить точку наблюдения, необходимо:

1. На вкладке «**Variables**» (см. п. 5.4.3.3) выбрать переменную, на которую необходимо установить точку наблюдения.

2. Выбрать в контекстном меню пункт «**Add Watchpoint (C/C++)**».

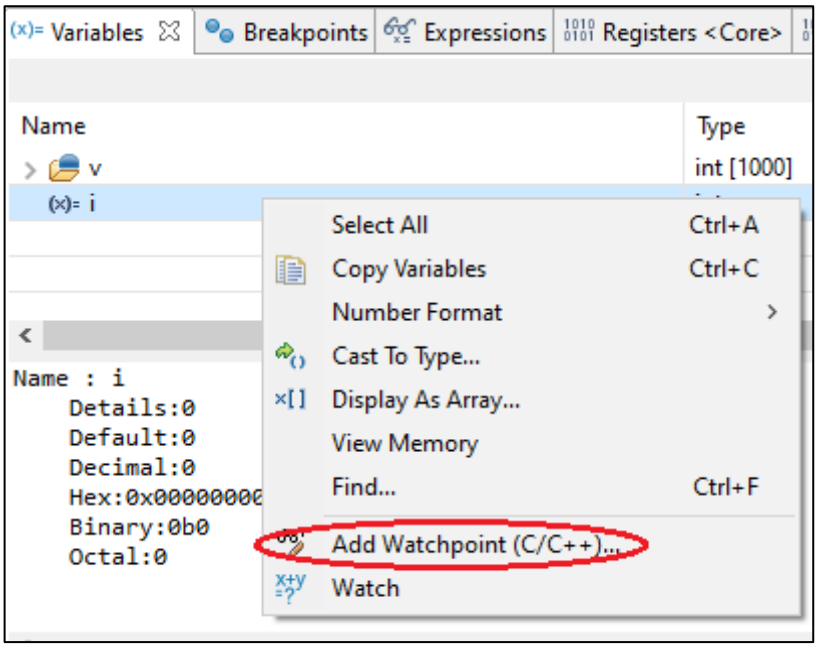


Рисунок 210. Установка точки наблюдения

3. Задать настройки для точки наблюдения:

Таблица 98. Параметры точки наблюдения

Параметр	Описание
Expression to watch	Выражение, на которое ставится точка наблюдения. По умолчанию подставляется та переменная, на которой было вызвано контекстное меню
Range	Наблюдаемый диапазон элементов относительно выбранной переменной. Размер элемента определяется размером выбранной переменной. Указывается в условных единицах. Для выражений, которые соответствуют массивам, необходимо указывать число элементов массива. Во всех остальных случаях это значение задается в 32-х битных словах (в текущей версии CM-LYNX).
Read	Если флаг установлен, то происходит приостановление выполнения при чтении переменной
Write	Если флаг установлен, то происходит приостановление выполнения при записи переменной
Enabled	Флаг активации точки наблюдения. При снятом флаге переменная не отслеживается
Watch target	Источник наблюдения. Возможные значения: ▪ All (режим наблюдения за всеми раздельными ресурсами, устанавливается по одной аппаратной точке наблюдения каждого типа); ▪ JK-bus; ▪ S-bus.
Condition	Условие, при котором происходит приостановка выполнения

Параметр	Описание
Ignore count	Количество событий, удовлетворяющих условию, или обращений к переменной (если проставлены соответствующие флаги) которые следует пропустить перед приостановкой выполнения
Stop mode	Режим остановки в случае срабатывания точки наблюдения. Возможные варианты: ▪ Stop after count, then on every hit (выполняется останов после заданного числа пропусков срабатываний "Ignore count" только 1 раз, после чего последующие остановы выполняются по каждому срабатыванию точки); ▪ Stop after every count (останов выполняется всегда после заданного числа пропусков срабатываний "Ignore count"); ▪ Stop after count, one hit (после срабатывания с учетом "Ignore count" WP отключается внутри отладчика без информирования IDE)
Use hardware ignore count	Флаг "Use hardware ignore count" позволяет использовать аппаратный счетчик пропуска срабатываний. Если флаг установлен и отсутствует условие срабатывания по выражению "Condition", применяется аппаратный счетчик пропуска срабатываний и его автоматический программный перезапуск при превышении диапазона аппаратного счетчика (у WPi используются 16-ти битные счетчики). Примечание. Опцию рекомендуется всегда оставлять включенной, так как это на порядки повышает производительность пропуска срабатываний и позволяет выполнять прогон фрагментов кода почти в реальном времени (с учетом времени на перезапуск).

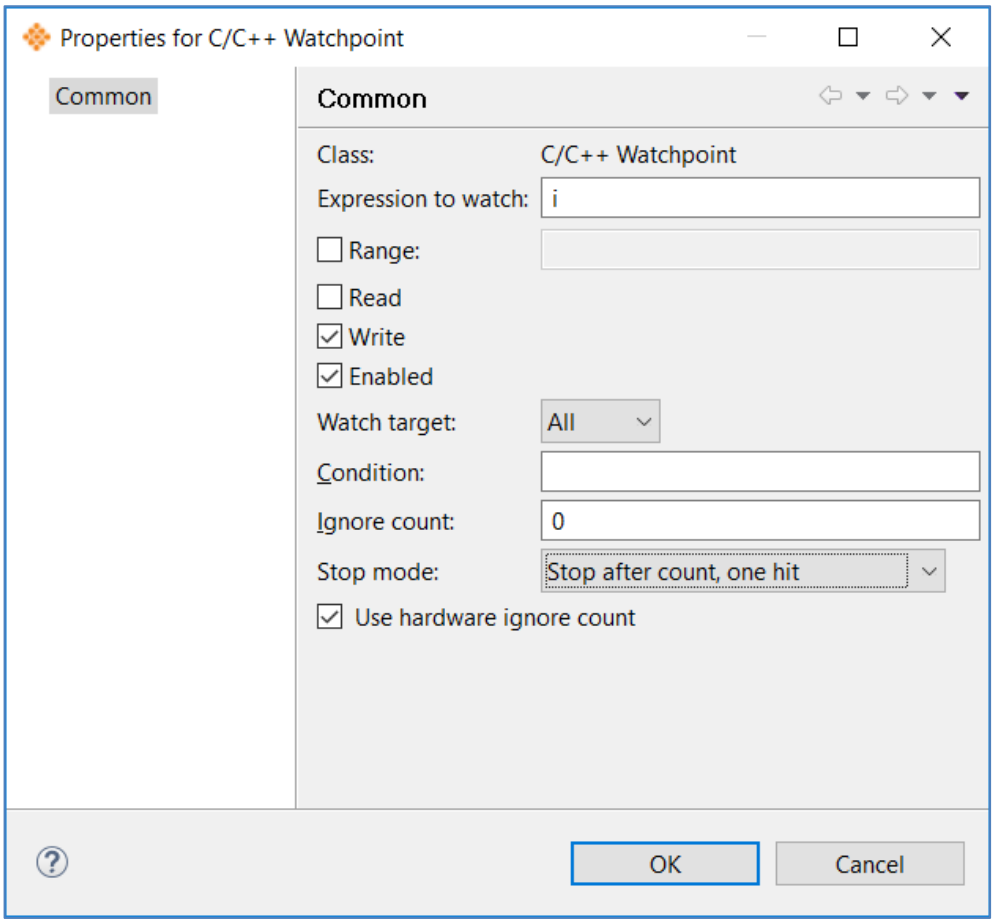


Рисунок 211. Настройка точки наблюдения

4. Нажать кнопку «**OK**».

   Новая точка наблюдения появится на вкладке «**Breakpoints**».

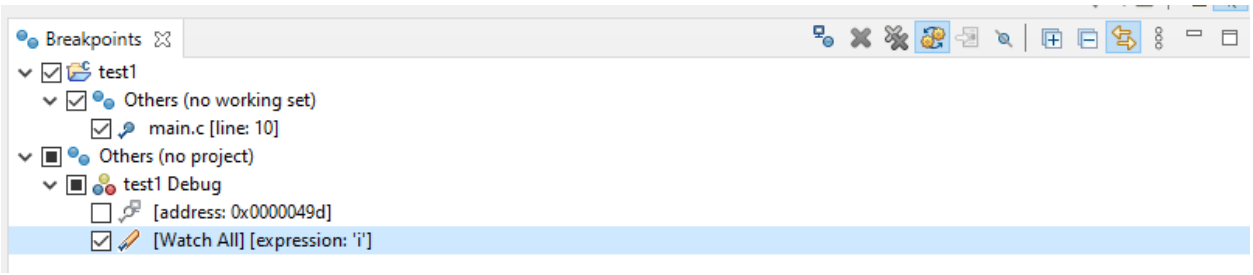


Рисунок 212. Список существующих точек наблюдения

Аналогично точку наблюдения можно установить на переменную в окне «**Expressions**» (см. п. 5.4.3.5. Вкладка «Expressions») или ячейку памяти в окне «**Memory**» (см. п. 5.4.3.7. Вкладка «Memory»).

При запуске отладки с активной точкой наблюдения на локальную переменную может появляться сообщение об ошибке:

"An enabled watchpoint is disabled. The reasons are as follows: the target has a limit on the number of enabled watchpoints or the watchpoint cannot be resolved in the current context."

Это связано с тем, что на момент выполнения команды «exec-run» отладчик ещё не перешёл в контекст main, и не может разрешить адрес локальной переменной. В результате точка наблюдения автоматически отключается. Эта проблема не возникает для глобальных переменных и адресных WP. Чтобы наблюдение за локальной переменной стало активным, её нужно повторно включить вручную после входа в нужный контекст.

Начиная с версии CM-LYNX 2.08.00, точки наблюдения привязаны к отладочным конфигурациям. Привязка означает, что работать такие точки останова будут только при отладке соответствующей конфигурации, а в других – не будут. До версии 2.08.00 все расставленные в разных проектах и сессиях точки наблюдения были общими в рамках рабочего пространства.

Пользователь может включить/отключить нужную точку наблюдения на вкладке «**Breakpoints**» одним из способов:

- установить/ снять флаг рядом с точкой наблюдения;
- выбрать в контекстном меню команду «**Enable**»/«**Disable**».

В режиме редактирования, если задан параметр "**Range**", показывается его значение. Иначе указывается "**default, all expression object**". Также, показываются адреса диапазона наблюдения **"Start address**", "**End address**". Всегда можно изменить цель наблюдения, а также условия срабатывания. После изменения любого параметра точка наблюдения переустанавливается, счетчик подсчета срабатываний сбрасывается (такие же действия выполняются при отключении/включении точки наблюдения) в окне Breakpoints.

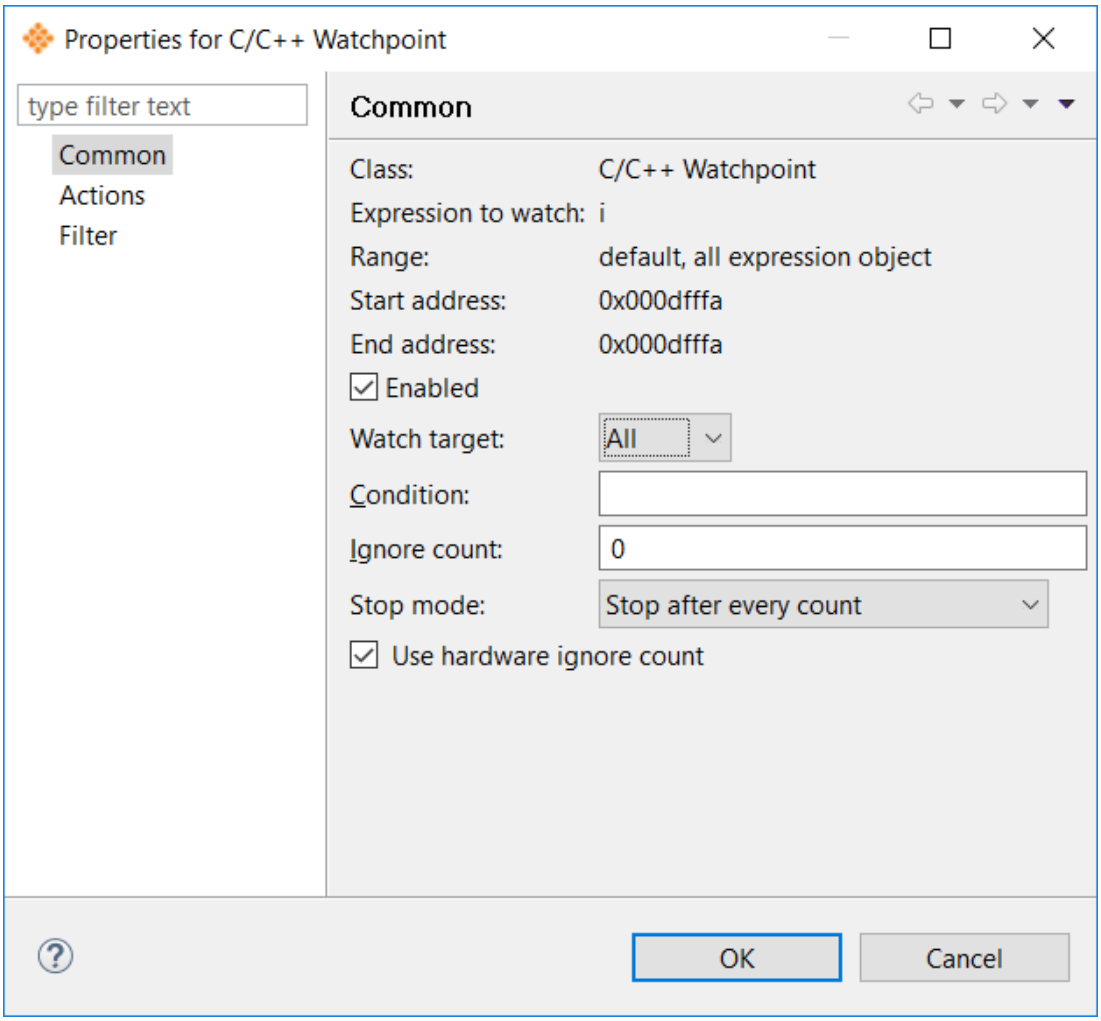


Рисунок 213. Редактирование точки наблюдения

Примечания:

1. Все адресные точки наблюдения, привязанные к отладочным конфигурациям проекта, по умолчанию отключаются при сборке данного проекта (если сборка сопровождается генерацией новых файлов), а не привязанные к проектам – при сборке любого проекта. За данное поведение отвечает настройка «**Disable address breakpoints and watchpoints on rebuild**», см. п. 6.12.3.

2. При запуске отладки с активной точкой наблюдения на локальную переменную может появляться сообщение об ошибке "An enabled watchpoint is disabled. The reasons are as follows: the target has a limit on the number of enabled watchpoints or the watchpoint cannot be resolved in the current context." Это связано с тем, что на момент выполнения команды «exec-run» отладчик ещё не перешёл в контекст main, и не может разрешить адрес локальной переменной. В результате точка наблюдения автоматически отключается. Эта проблема не возникает для глобальных переменных и адресных WP. Чтобы наблюдение за локальной переменной стало активным, её нужно повторно включить вручную после входа в нужный контекст.

6.11.9. Просмотр и редактирование значений регистров

Информацию о регистрах ядра и периферийных устройств, а также их значениях можно просматривать на вкладках «**Registers (Core)**» (см. Рисунок 214) и «**Registers (Peripherals)**» (см. Рисунок 215), расположенных на правой верхней панели окна. Регистры объединены в группы регистров. В регистрах большей разрядности могут находиться регистры меньшей разрядности и/или битовые поля.

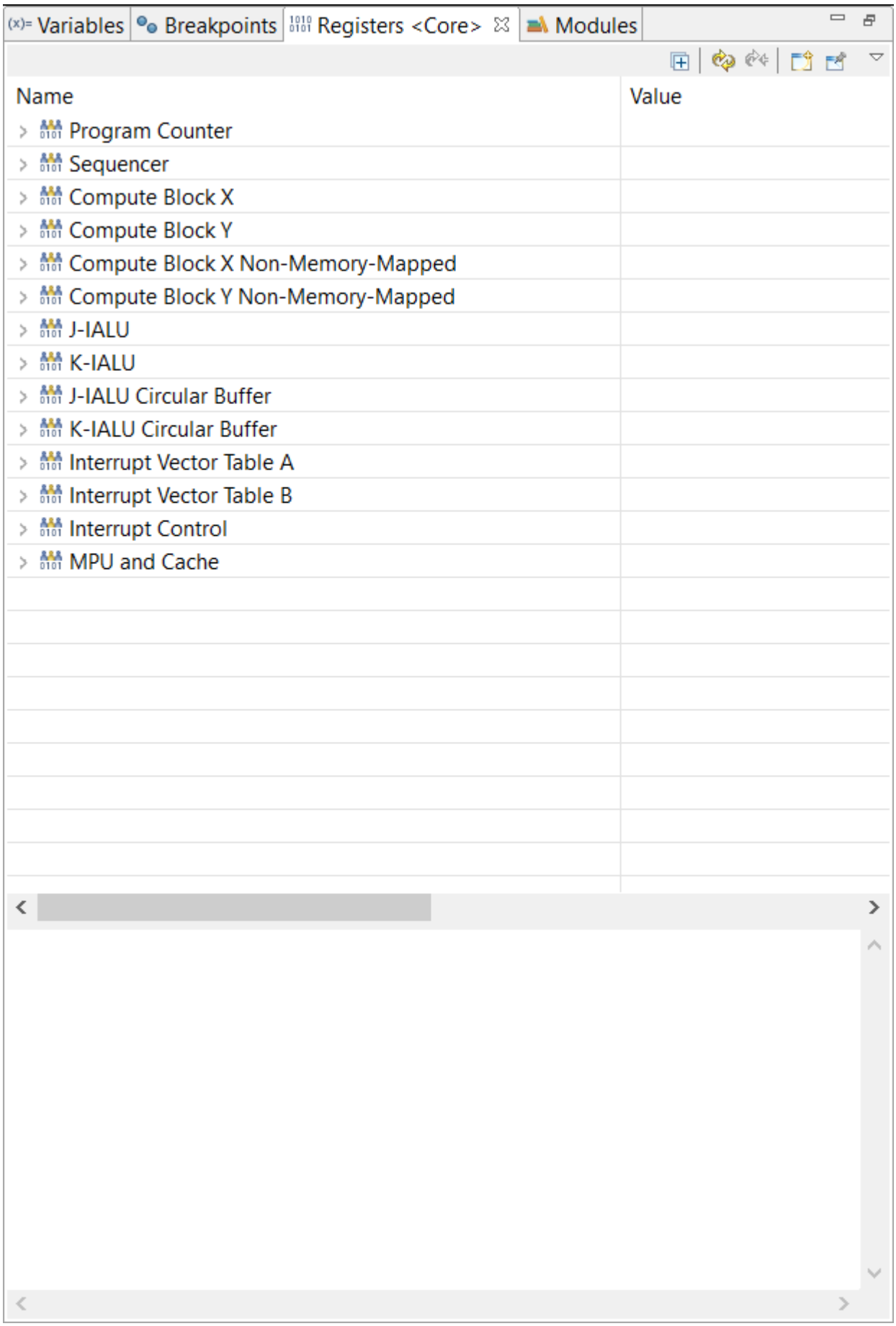


Рисунок 214. Вкладка «Registers (Core)»

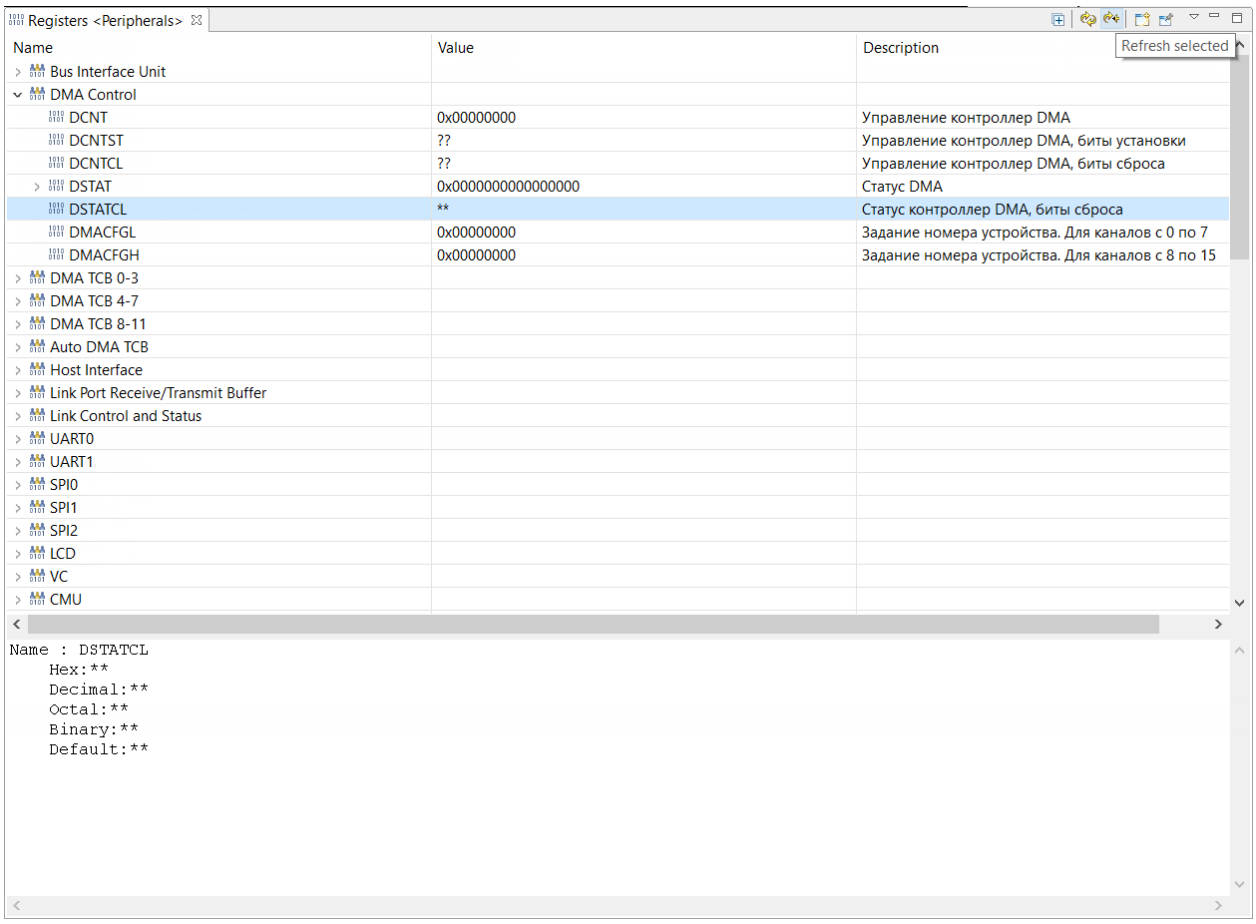


Рисунок 215. Вкладка «Registers (Peripherals)»

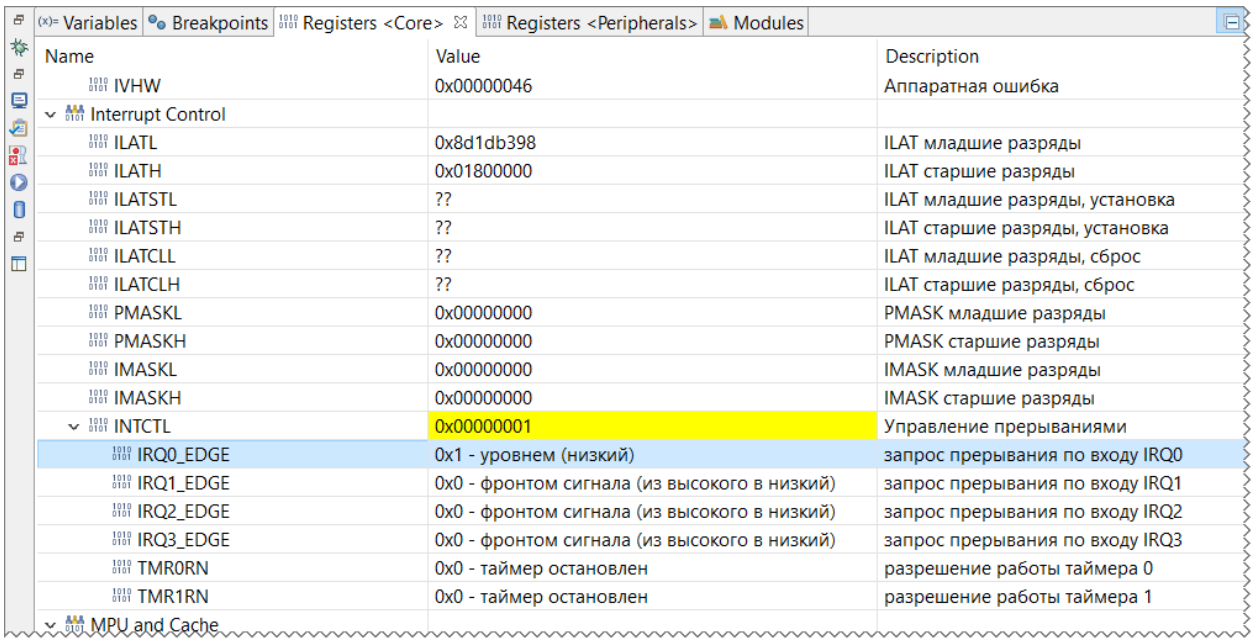
Для просмотра всех регистров на одной вкладке необходимо выбрать «**Window->Show View->Other...->Debug->Registers**».

Для регистров особого типа, т.е. влияющих на работу процессора, есть свои правила:

- ***Read-only*** – регистры данного типа недоступны для редактирования. У данных регистров в поле «*Value*» отображается значение «****»(см. Рисунок 215). Для просмотра их значений следует выбрать нужные регистры и нажать кнопку «**Refresh selected**» на панели инструментов. После чтения значение регистра будет отображаться до следующей команды пошаговой отладки (шаг, запуск с остановом);
- ***Write-only*** – регистры данного типа недоступны для чтения, но их можно отредактировать. У данных регистров в поле «*Value*» отображается значение «*??*» (см. Рисунок 215). При редактировании значений регистров WO введённые до этого значения стираются из памяти;
- ***Read-write*** – регистры данного типа доступны и для чтения, и для записи, но так как они влияют на работу процессора, то относятся к особому типу регистров со своими правилами. У данных регистров в поле «*Value*» отображается значение «****»(см. Рисунок 215). Для просмотра их значений следует выбрать нужные регистры и нажать кнопку «**Refresh selected**» на панели инструментов. После чтения значение регистра будет отображаться до следующей команды пошаговой отладки (шаг, запуск с остановом)) или ввода нового значения в редакторе. После записи регистр автоматически не читается, значение отображается символом «**».

Внимание: при нажатии кнопки «**Refresh**» чтение особых регистров не выполняется.

Значения, которые были изменены, выделяются цветом при остановке программы.



Name	Value	Description
IVHW	0x00000046	Аппаратная ошибка
Interrupt Control		
ILATL	0x8d1db398	ILAT младшие разряды
ILATH	0x01800000	ILAT старшие разряды
ILATSTL	??	ILAT младшие разряды, установка
ILATSTH	??	ILAT старшие разряды, установка
ILATCLL	??	ILAT младшие разряды, сброс
ILATCLH	??	ILAT старшие разряды, сброс
PMASKL	0x00000000	PMASK младшие разряды
PMASKH	0x00000000	PMASK старшие разряды
IMASKL	0x00000000	IMASK младшие разряды
IMASKH	0x00000000	IMASK старшие разряды
INTCTL	0x00000001	Управление прерываниями
IRQ0_EDGE	0x1 - уровнем (низкий)	запрос прерывания по входу IRQ0
IRQ1_EDGE	0x0 - фронтом сигнала (из высокого в низкий)	запрос прерывания по входу IRQ1
IRQ2_EDGE	0x0 - фронтом сигнала (из высокого в низкий)	запрос прерывания по входу IRQ2
IRQ3_EDGE	0x0 - фронтом сигнала (из высокого в низкий)	запрос прерывания по входу IRQ3
TMR0RN	0x0 - таймер остановлен	разрешение работы таймера 0
TMR1RN	0x0 - таймер остановлен	разрешение работы таймера 1
MPU and Cache		

Рисунок 216. Изменённое значение

Пользователь может изменить значения регистров для записи (Write). Для этого необходимо:

1. Выбрать регистр, которые необходимо изменить.
2. Щелкнуть внутри поле «**Value**».
3. Изменить значение.
4. Нажать клавишу «**Enter**».

Группу регистров или отдельный регистр из группы можно добавить в список выражений, который доступен для работы на вкладке «**Expressions**» (см. п. 6.11.11. Просмотр и редактирование значений выражений). Для этого необходимо выбрать регистр или группу регистров и в контекстном меню выбрать пункт «**Watch**». Кроме того, можно вручную добавить выражение, содержащее символ $ и название регистра, например, `$J0` для отображения регистра `J0`.

6.11.10. Просмотр и редактирование значений переменных

Просмотр и редактирование значений переменных доступно на вкладке «**Variables**», расположенной на правой верхней панели окна (см. 5.4.3.3. Вкладка «Variables»).

При отладке программы C/C++ данные, которые содержат переменные, можно развернуть, чтобы показать переменные и поля, содержащие переменные.

Для изменения значения переменной во время отладки необходимо:

1. Поставить курсор на значение в столбце **«Value»** изменяемой переменной.
2. Ввести новое значение.
3. Нажать клавишу **«Enter»**.

Примечание: значения для переменных типа __builtin_quad можно задавать только в hex-формате.

Переменную можно добавить в список выражений, который доступен для работы на вкладке «**Expressions**» (см. п. 6.11.11). Для этого необходимо выбрать переменную и в контекстном меню выбрать пункт «**Watch**».

Переменные отображаются в следующем порядке:

1. Локальные (всегда),
2. Статические (если включена кнопка «**Show Static Variables**»),
3. Глобальные файла (если включена кнопка «**Show File Global Variables**» или «**Show Module Global Variables**»),
4. Глобальные других файлов в составе исполняемого модуля (если включена кнопка «**Show Module Global Variables**»).

Примечание: Глобальные переменные могут быть перекрыты локальными при совпадении имен. В этих случаях перекрываемые переменные не отображаются.

Для преобразования переменной к другому типу можно воспользоваться пунктом контекстного меню «**Cast To Type**».

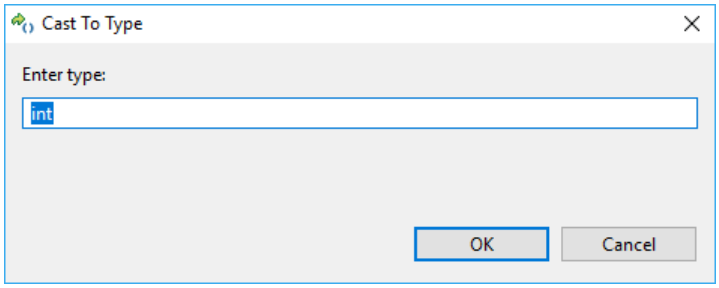


Рисунок 217. Окно «Cast To Type»

Для восстановления первоначального типа для переменной можно использовать пункт «**Restore Original Type**».

Для просмотра содержимого ячеек памяти, в которых хранится переменная, можно выбрать пункт «**View Memory**» контекстного меню. Откроется вкладка «**Memory**», подробнее см. п. 6.11.12. Просмотр и редактирование содержимого памяти целевого процессора.

Примечание. Поскольку текущая кодировка по умолчанию UTF-8, то для корректного отображения русскоязычных названий переменных на вкладках «**Variables**» и «**Expressions**» в среде CM-LYNX следует использовать тип **`wchar_t`**.

Например,

```
#include <stdlib.h>

wchar_t rus[] = { L'а', L'б', L'в' };

int main()

{

	wchar_t elem_rus_1 = rus[0];

	return 0;

}
```

6.11.11. Просмотр и редактирование значений выражений

Просмотр и редактирование значений выражений доступно на вкладке «**Expressions**», расположенной на правой верхней панели окна (см. п. 5.4.3.5. Вкладка «Expressions»).

Выражения – фрагменты кода, которые могут быть оценены для получения результата. Контекст выражения зависит от конкретной модели отладки. Некоторые выражения могут потребоваться для оценки в определенном месте в программе, чтобы на переменные можно было ссылаться.

Для того чтобы добавить выражение, необходимо:

1. Нажать на кнопку  на панели инструментов или выбрать пункт «**Add Watch Expression**» в контекстном меню.

2. Ввести выражение в поле «**Expressions**».

3. Нажать кнопку «**OK**».

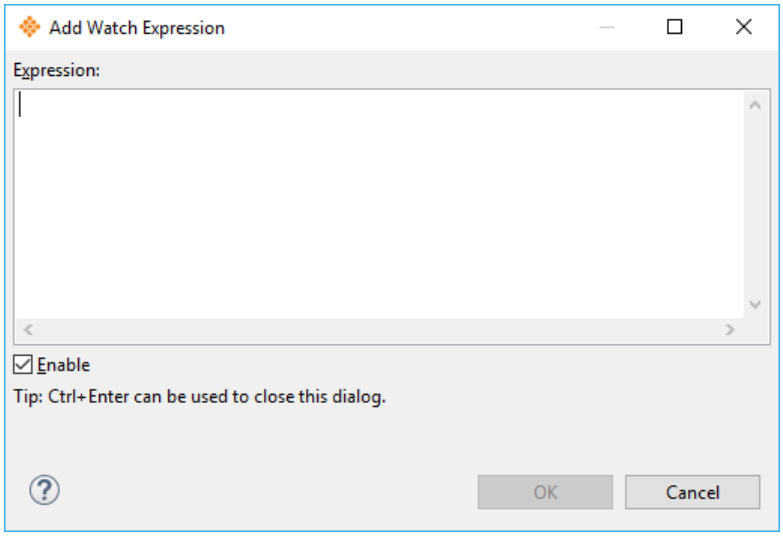


Рисунок 218. Окно «Add Watch Expressions»

Примечание: для добавления переменной на вкладку «**Expressions**», объявленной в ассемблерном файле с использованием макросов `__ASM_FUN` и `__ASM_VAR` из файла **decor.**h следует имя такой переменой указывать в формате **_var**, где **var** – имя переменной, вызываемой в C/C++ коде и объявленной в ассемблерном файле.

Добавление выражений доступно также на вкладках «**Registers**» и «**Variables**». В этом случае вкладка «**Expressions**» открывается автоматически при добавлении выражения.

При отладке программы C/C++ области, которые содержат переменные, могут быть увеличены для отображения переменных и полей с переменными.

Во время отладки на вкладке **«Expressions»** можно менять значение переменных.

Для изменения значения выражения во время отладки необходимо:

1. Поставить курсор на значение в столбце **«Value»** изменяемой переменной.

2. Ввести новое значение.

3. Нажать клавишу **«Enter»**.

Примечание: значения для переменных типа __builtin_quad можно задавать только в hex-формате.

Если включена кнопка «**Expressions for selected debug context**» отображаются глобальные выражения и привязанные к выбранному контексту отладки выражения. Новые вводимые выражения становятся привязанными к текущему контексту отладки. Если кнопка «**Expressions for selected debug context**» выключена, то отображаются только глобальные выражения, а новые вводимые становятся глобальными.

6.11.12. Просмотр и редактирование содержимого памяти целевого процессора

Просмотр и редактирование содержимого ячеек памяти целевого процессора доступны:

- На вкладке «**Memory**» (см. п. 5.4.3.7. Вкладка «Memory»);
- На вкладке «**Memory Browser**».

Примечание: вкладка «**Memory Browser**» полностью соответствует представлению «**Traditional**» вкладки «**Memory**» (при указании идентичного адреса или выражения).

6.11.12.1. Добавление объекта мониторинга

Список объектов мониторинга можно просмотреть и настроить в области «**Monitors**» вкладки «**Memory**». Каждый пункт списка «**Monitors**» представляет собой секцию памяти с определенным адресом, называемым *базовым адресом*.

Для того чтобы добавить объект мониторинга в список «**Monitors**», необходимо:

1. В списке «**Monitors**» нажать на кнопку .
2. Ввести в поле адрес или выражение (согласно стандартной адресации C/C++).

   Например, поддерживаются выражения вида:
   - 0x0847d3c;
   - (&y)+1024;
   - *ptr;
3. Нажать кнопку «**OK**».

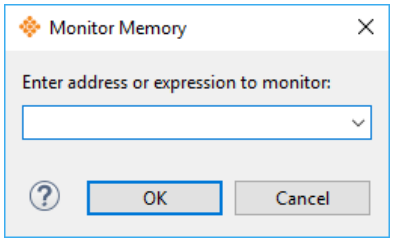


Рисунок 219. Окно «Monitor»

6.11.12.2. Настройка формата отображения ячеек памяти

Для каждого объекта мониторинга информация может отображаться в разных предустановленных форматах данных. Поддерживаются следующие форматы:

- Floating Point;
- Traditional;
- Hex (по умолчанию);
- ASCII;

- Signed Integer;
- Unsigned Integer;
- Hex Integer;
- Data Chart.

При создании объекта мониторинга автоматически отображается представление по умолчанию.

Представления отображаются в области «**Rendering**». Для одного объекта можно настроить несколько разных форматов отображения.

Для того чтобы добавить формат отображения ячеек памяти, необходимо:

1. Нажать на кнопку « **New Rendering**» или выбрать пункт контекстного меню « **Add Rendering»**.
2. Выбрать формат из предложенных вариантов.
3. Нажать кнопку «**Add Rendering(s)**»

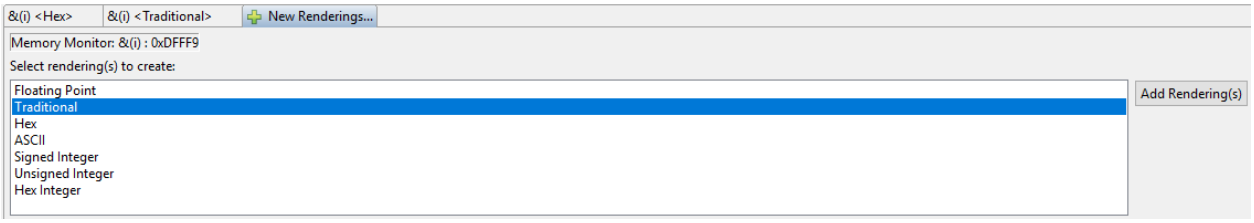


Рисунок 220. Панель выбора формата

6.11.12.3. Импорт содержимого ячеек памяти

Примечание: импорт доступен только в области памяти, размеченные как RAM.

Поддерживаются следующие форматы:

- Plain Text (данные в текстовом виде);
- RAW Binary (данные в виде кода);
- SRecord (данные в S-формате).

Для того чтобы импортировать содержимое ячейки памяти из файла, необходимо:

1. Нажать на кнопку .
2. Выбрать формат файла в поле «**Format**».
3. При необходимости, скорректировать диапазон для импорта в полях «**Restore from address**» и «**End address**», а также длину экспортируемого участка в поле «**Length**».
4. Указать путь к файлу в поле «**File name**», из которого следует импортировать.
5. Задать в поле «**Byte order**» порядок (Little endian или Big endian).
6. Нажать на кнопку «**OK**».

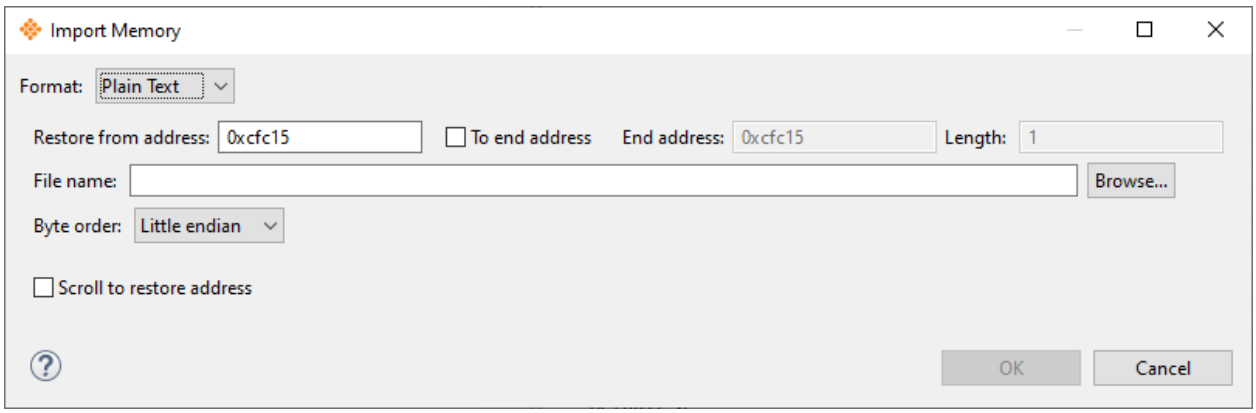


Рисунок 221. Окно «Import Memory»

6.11.12.4. Экспорт содержимого ячеек памяти

Поддерживаются следующие форматы:

- Plain Text (данные в текстовом виде);
- RAW Binary (данные в виде кода);
- SRecord (данные в S-формате).

Для того чтобы экспортировать содержимое ячейки памяти в файл, необходимо:

1. Выделить фрагмент на вкладке «**Memory**» и нажать на кнопку .
2. Выбрать формат файла в поле «**Format**».
3. При необходимости, скорректировать диапазон для экспорта в полях «**Start address**» и «**End address**», а также длину экспортируемого участка в поле «**Length**».
4. Указать путь к файлу в поле «**File name**».
5. Задать в поле «**Byte order**» порядок (Little endian или Big endian).
6. Задать в поле «**Number format**» формат, в котором сохраняются данные (Hex Integer, Floating Point 32-bit, Floating Point 64-bit).
7. Нажать на кнопку «**OK**».

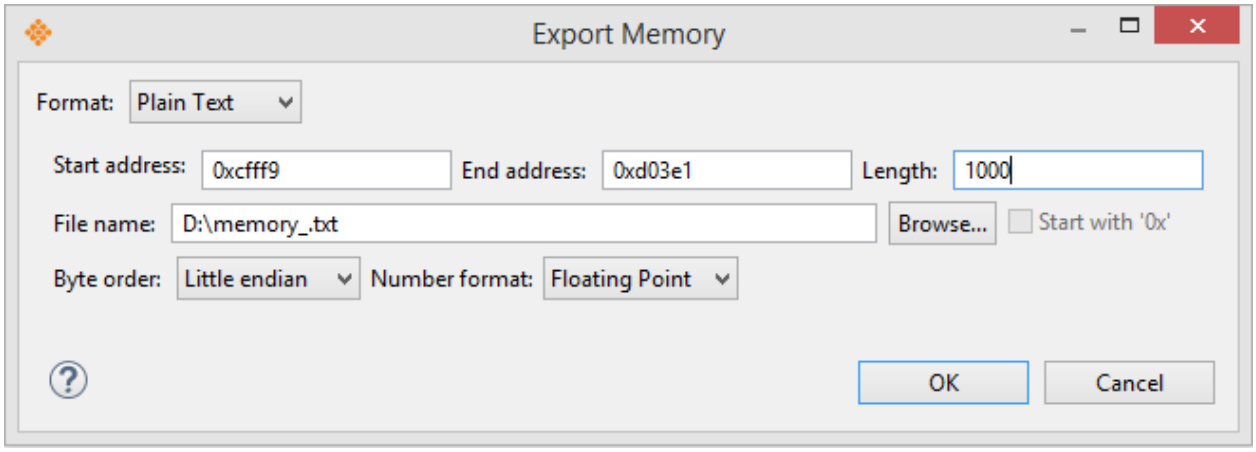


Рисунок 222.Окно «Export Memory»

6.11.12.5. Копирование выделенной области в буфер обмена

Во вкладках «**Hex**», «**ASCII**», «**Signed Integer**», «**Unsigned Integer**», «**Hex Integer**» область памяти, которая отображается в окне, можно скопировать командой «**Copy To Clipboard**» контекстного меню.

Во вкладке «**Traditional**» область памяти, которая выделена, можно скопировать командой контекстного «**Copy**» в форматах «Hex», «Decimal Signed», «Decimal Unsigned», «Octal», «Binary» в виде «**Binary**» (только данные), «**Text**» (только текст), «**Address**» (только адрес), «**All**» (всё).

Существует ограничение на копирование большого объёма данных в 0x40000 байт. При копировании более 0x40000 байт, будет появляться предупреждающий диалог о блокировке копирования.

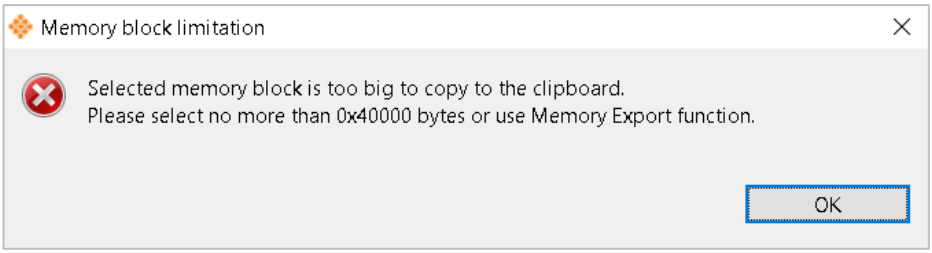


Рисунок 223. Предупреждающий диалог «Memory block limitation»

Для записи фрагмента памяти в файл в текстовом формате можно воспользоваться копированием нужного фрагмента памяти в буфер обмена, а затем буфер обмена скопировать в файл.

6.11.12.6. Особенности отображения ячеек памяти для переменной в режиме char size=8

В режиме `char size=8` вкладки **«Memory»** и **«Memory Browser»** по запросу «`&var`» отображают адрес слова, содержащего значение переменной `var`, который в общем случае отличается от адреса переменной `var`.

Например,

- если переменная `var` занимает один байт, то

```
(адрес слова) = (адрес переменной) / 4
(смещение внутри слова) = (адрес переменной) % 4;
```

- если переменная var занимает два байта, то

```
(адрес слова) = (адрес переменной) / 2
(смещение внутри слова) = (адрес переменной) % 2;
```

- если переменная `var` занимает четыре байта (слово), то адрес слова и адрес переменной совпадают.

6.11.12.7. Построение графиков

Формат представления «**Data Chart**» позволяет отображать содержимое памяти в виде графиков. Для построения графика необходимо:

1. Добавить массив или указатель в список «**Monitors**» - см. п. 6.11.12.1;
2. Изменить формат отображения на «**Data Chart**» - см. п. 6.11.12.2;
3. Настроить отображение графика, если необходимо.

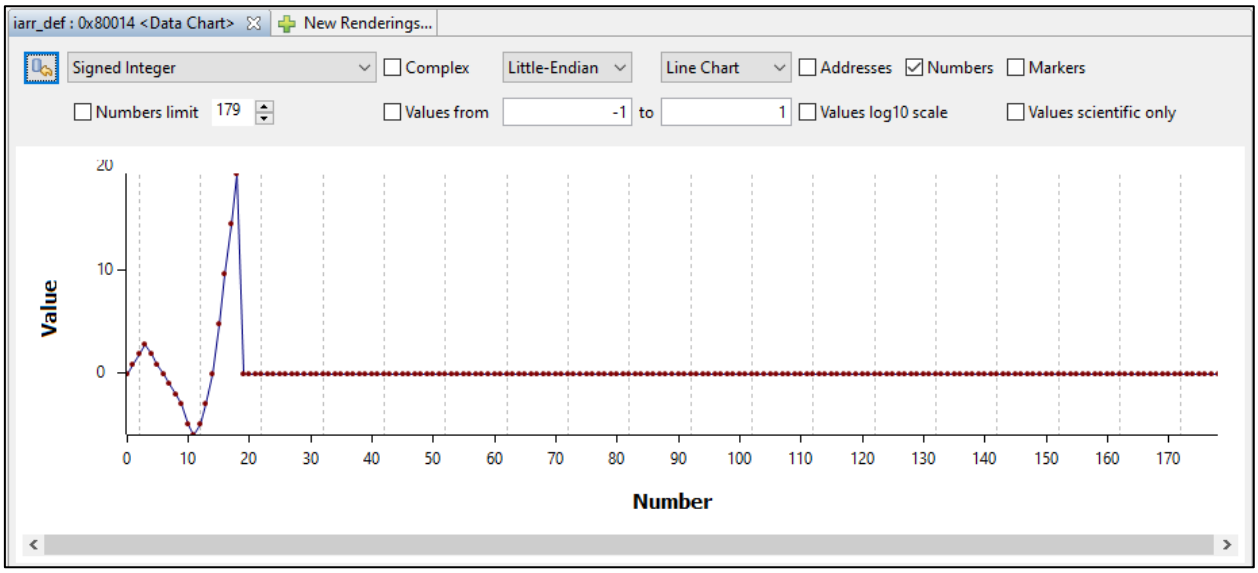


Рисунок 224. Вид панели «Renderings» при отображении графика

Таблица 99. Настройки и элементы управления формата представления «Data Chart»

Значок	Название	Описание функционирования
	Reset to Base Address	Обновить содержимое монитора и установить диапазон наблюдения по умолчанию.
	Numbers type	Выпадающий список для выбора типа входных данных: • «**Signed Integer**» - знаковые целые, 32 бита (по умолчанию); • «**Unsigned Integer**» - беззнаковые целые, 32 бита; • «**32-bit Single Precision Floating Point**» - 32-битные числа с плавающей запятой; • «**64-bit Double Precision Floating Point**» - 64-битные числа с плавающей запятой.
☐	Complex	Интерпретировать данные как комплексные. На графике будет отображаться модуль комплексного числа.
	Data Endiannes	Выпадающий список для выбора порядка байт в данных: • «**Little-Endian**» (по умолчанию), • «**Big-Endian**».
	Chart type	Выпадающий список для выбора тип графика: • «**Line Chart**» - Линейный график (по умолчанию), см. Рисунок 224, • «**Area Chart**» - Линейный график с заливкой области, см. Рисунок 225, • «**Bar Chart**» - Гистограмма, см. Рисунок 226, • «**Scatter Chart**» - Точечный график, см. Рисунок 227.
☐	Addresses	Включение отображения дополнительной оси адресов значений сверху (по умолчанию отключено).
☐	Numbers	Включение отображения основной оси порядковых номеров значений (с 0, по умолчанию включено).

Значок	Название	Описание функционирования
☐	Markers	Включение отображения маркера - информации о положении курсора мыши в окне (по умолчанию отключено).
☐	Numbers limit	Включение лимита окна отслеживаемых данных (количество значений, в конкретный момент времени может отображаться меньше элементов, по умолчанию отключено).
	-	Значение лимита окна данных (по умолчанию - 128 или размерность массива, вычисленная отладчиком).
☐	Values from, to	Включение ограничения диапозона значений. Неполные или пустые значения подсвечиваются красной рамкой, а корректные значения - зеленой. Корректными считаются положительные или отрицательные целые, а также double значения в нотации scientific. При нажатии «Enter» в одном из полей ввода, при наличии корректных начального и конечного значений и включенном режиме отображения в заданном диапазоне предельных значений, выполняется автоматическое обновление отображения с применением новых значений диапазона.
☐	Values log10 scale	Включение постороения оси значений «Value в логарифмическом масштабе. Это означает, что позиции отображения точек входного массива на графике будут соответствовать функции log10(V(N)), где N - номер числа по оси «Number», а V(N) - соответствующее значение входного массива данных. При этом на оси «Value» и в маркерах отображаются не логарифмические значения, а исходные. См. Рисунок 228.
☐	Values scientific only	Включение отображения всех дробных значений в формате scientific, то есть: ▪ 0.0#######E0## -для данных типа float/double32; ▪ 0.0##################E0## - для double64, где 0 минимальная всегда отображаемая цифра, а количество цифр после запятой "#" определяется разрядностью отображаемого значения. По умолчанию, при отключенном флаге «Values grid scientific only», применяется адаптивный режим отображения значений. Для дробных чисел используется формат «########0.0#####» как приоритетный. Если абсолютное значение меньше 1E9, и число цифр после запятой не более 6, то такое число представляется в указанном формате десятичной дроби. Остальные дробные числа представляются в формате scientific. Для типов Integer всегда отображаются полные десятичные значения. Введенные режимы отображения не влияют на формат значений во всплывающих подсказках и маркере позиции курсора. Там он всегда scientific.

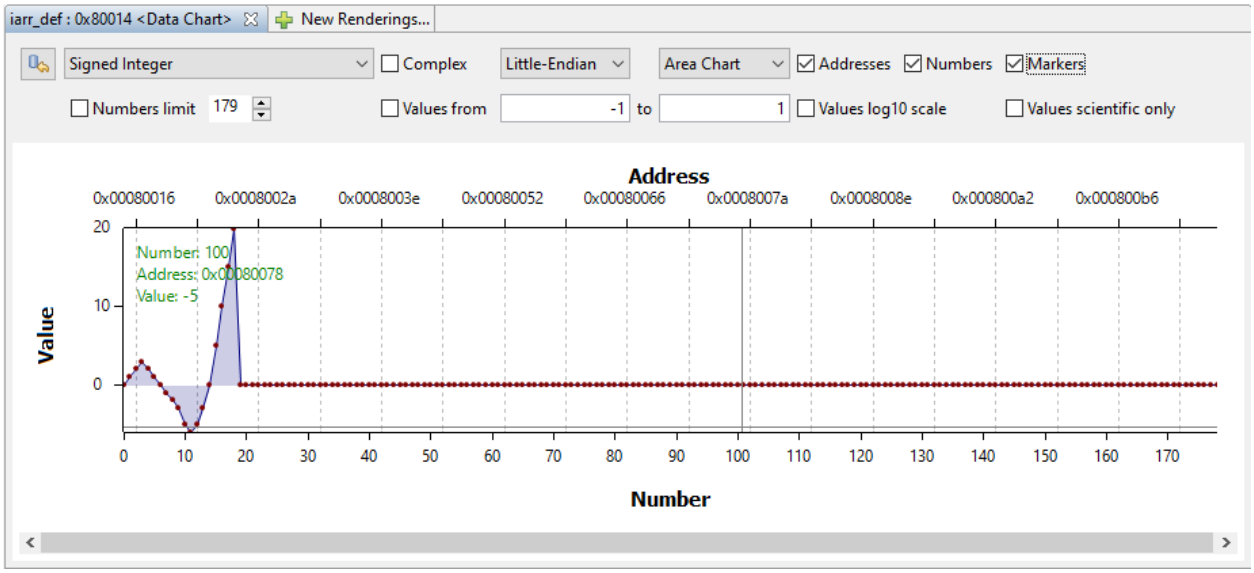


Рисунок 225. График типа «Area Chart» со включенной осью адресов, маркером и расширенной подсказкой

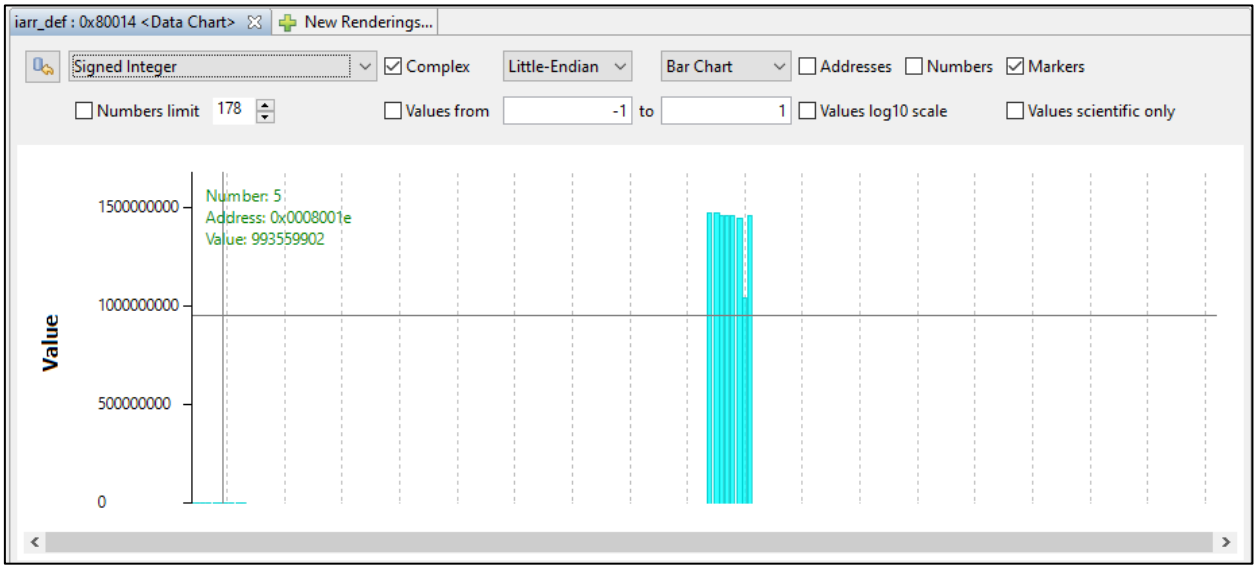


Рисунок 226. График типа «Bar Chart» для комплексных чисел с маркерами и расширенной подсказкой, при выключенных горизонтальных осях

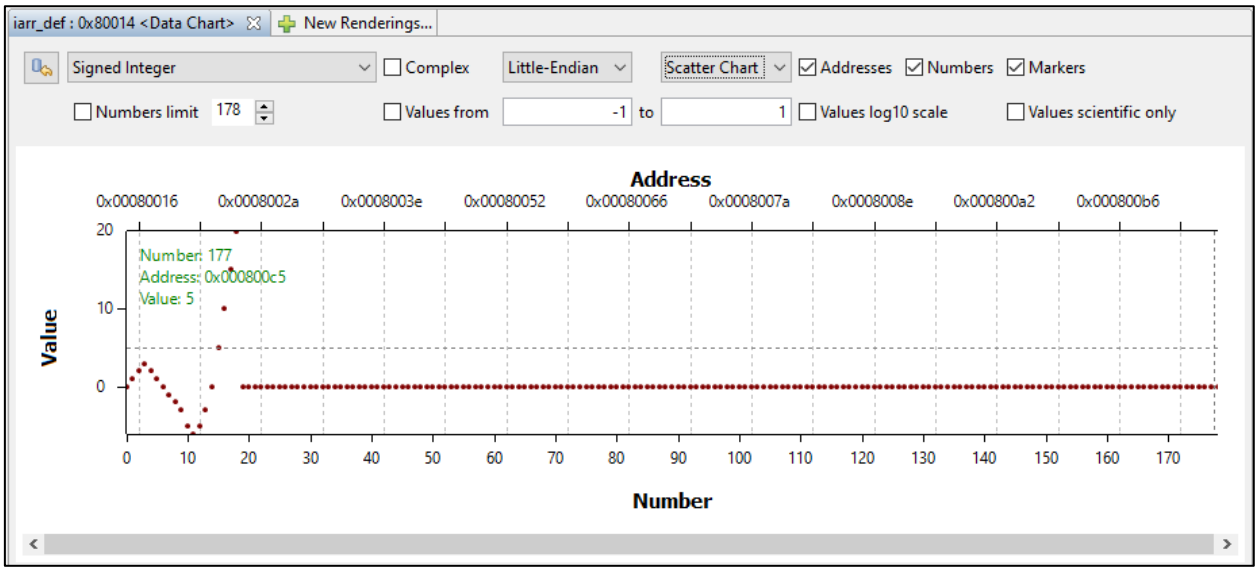


Рисунок 227. График типа «Scatter Chart» для целых чисел со включенной осью адресов, маркером и расширенной подсказкой

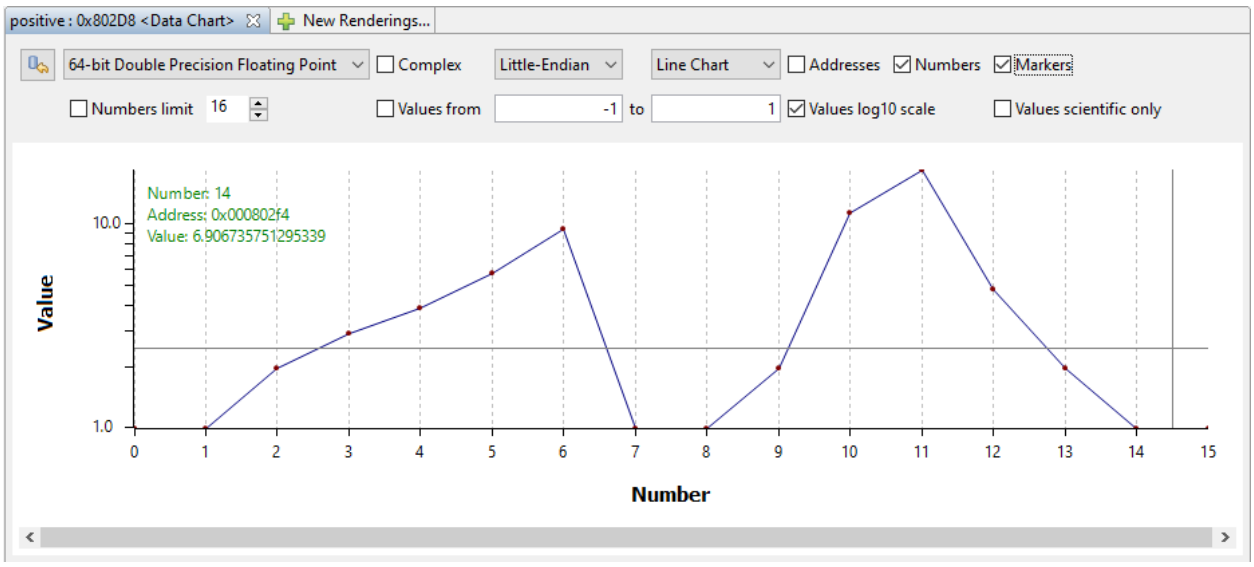


Рисунок 228. График с логарифимическим представлением оси «Value»

Формат представления «**Data Chart**» поддерживает следующие функции:

- Автоматический выбор диапазона значений по вертикальной оси. Формат отображения значений по вертикальной оси выбирается в соответствии с выбранным типом входных данных.
- Выделение области с помощью удержания левой кнопки мыши. После выделения области выполняется автоматическое определение диапазона по вертикальной оси и отображение выделенного фрагмента.
- Масштабирование колесом мыши с автоматическим расширением диапазона наблюдения значений по горизонтальной оси при «Zoom Out», если не превышены установленный и максимальный пределы (2048).
- Расширение по горизонтальной оси, по нажатию на правую стрелку горизонтальной полосы прокрутки, при условии нахождения в крайнем правом положении в текущей области отслеживания значений. Расширение возможно только если не превышены установленные ограничения.

- Отображение расширенной подсказки о значении по всем осям – появляется при наведении курсора мыши на точку графика или вершину столбца гистограммы.

Сразу после открытия отображение данных выполняется в режиме по умолчанию – происходит автоматическое определение типов входных данных для массивов, включая комплексные типы «std::complex<?>[]», с обновлением типов на локальной панели управления, чтение и интерпретация данных с отображением графика типа «**Line Chart**». В поле значения лимита окна наблюдения автоматически подставляется размерность массива, если выражение в текущем контексте отладки соответствует статическому массиву. Для массивов установленной размерности выполняется отображение сразу всего массива значений (максимально - 2048). Для других вариантов (например, указатели), отображаются первые 128 значений. При этом пользовательский лимит окна наблюдения остается отключенным, а поле значения лимита может использоваться как подсказка, которая показывает размерность текущего окна наблюдения (установленную размерность массива). В дальнейшем, пользователь может активировать режим ограничения окна наблюдения, чтобы не допустить выход за пределы массива или введенного другого ограничения во время выполнения действий по автоматическому расширению окна наблюдения (по «Zoom Out» с помощью мыши или по правой стрелке горизонтальной полосы прокрутки). Ограничение необходимо, чтобы исключить попытки чтения, интерпретации и отображения не валидных данных, которые могут повлиять на состав и качество отображаемой информации.

В процессе отладки монитор формата «**Data Chart**» функционирует в двух основных режимах чтения, интерпретации и отображения данных:

1. Автоматический режим отслеживания контекста и выражения наблюдения с автоматическим выбором режимов интерпретации и отображения данных.
    a. Режим активен полностью, когда включена кнопка «**Automatic refresh»** у вкладки «**Memory**» и отключен флаг «**Numbers limit**». На каждом останове (шаге) отладки, как и по нажатию кнопки «**Refresh**», выполняется автоматическое определение объекта выражения относительно активного (или выбранного по «**Pin to Debug Context**») контекста отладки. Если объект выражения не изменился (тип данных, стартовый адрес, размерность), то выполняется чтение данных и отображение графика в выбранном диапазоне, если он не выходит за пределы размерности. Если параметры отображения данных менялись пользователем и не соответствуют обновленной информации об объекте выражения, то они будут сброшены в значения по умолчанию (для массивов с установленной размерностью - наблюдение и отображение данных всего массива).
    b. Режим активен частично, когда не включена кнопка «**Automatic refresh»** у вкладки «**Memory**». Автоматическая установка параметров интерпретации и отображения данных в «**Data Chart**» выполняется в ручном режиме по нажатию кнопки «**Refresh**». В этом случае пользовательские установки параметров интерпретации и отображения не сбрасываются на каждом шаге (останове). Окно может быть использовано для сохранения и отображения определенного состояния объекта выражения в целевой системе на момент последнего чтения данных в момент открытия окна или выполнения обновления по нажатию кнопки «**Refresh**».

2. Пользовательский режим отображения, допускающий установку любых возможных параметров интерпретации и отображения данных. Активен, когда установлено ограничение окна наблюдения данных с помощью «**Numbers limit**». В этом режиме при чтении данных (по нажатию кнопки «**Refresh**» или при включенной кнопке «**Automatic refresh»**) не выполняется автоматическая установка параметров интерпретации и отображения данных в соответствии с полученной информацией об объекте наблюдения от отладчика. Установленное пользователем значение в поле ввода ограничения не меняется. После ввода нового значения ограничения для его применения необходимо дважды сменить состояние включения ограничения или выполнить обновление данных по нажатию кнопки «**Refresh**». Режим удобно применять тогда, когда точная информация об объекте наблюдения не определяется автоматически, а известна заранее разработчику.

6.11.13. Управление процессорами при многопроцессорной отладке

6.11.13.1. Режим «Multicore»

Режим «**Multicore**» позволяет выполнять отладочные команды (**`Resume`**, **`Suspend`**, **`Terminate`**, **`Step Into`**, **`Step Over`**, **`Step Return`**) одновременно для всех активных процессоров при симметричной многопроцессорной отладке (SMP).

Для переключения к режиму отладки «**Multicore**» следует нажать значок  «**Multicore Commands Mode**» на панели инструментов. Для отключения режима следует нажать на этот значок повторно.



Рисунок 229. Кнопка включения режима отладки «Multicore»

6.11.13.2. Режим «Multisession»

Режим «**Multisession**» позволяет выполнять отладочные команды (**`Resume`**, **`Suspend`**, **`Terminate`**) одновременно для всех активных процессоров при асимметричной многопроцессорной отладке (AMP).

Для переключения к режиму отладки «**Multisession**» следует нажать значок  («**Multisession Mode**») на панели инструментов. Для отключения режима следует нажать на этот значок повторно.



Рисунок 230. Кнопка включения режима отладки «Multisession»

6.11.14. Просмотр карты памяти процессора

Работа с картами виртуальной памяти процессора 1967ВН058 доступна на вкладке «**Memory Map**». Информация, отображаемая во вкладке «**Memory Map**», может существенно упрощать отладку и поиск ошибок ПО, работающего в защищенном режиме процессора.

Также на вкладке «**Memory Map**» поддерживается отображение карт физической памяти всех процессоров серии 1967 (ВН028, ВН044, ВН058). Эту информацию предоставляет отладчик из файлов SFR (если в них присутствует дополнительный раздел описания блоков физической памяти).

6.11.14.1. Переключение режима отображения данных

Для выбора режима отображения данных используется кнопка в крайней левой позиции панели инструментов вкладки «**Memory Map**».

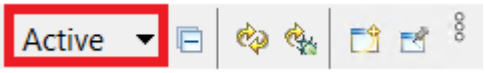


Рисунок 231. Кнопка смены режима

По умолчанию выбран режим отображения активной карты памяти «Active». В этом режиме отображается информация о карте доступной памяти, которая соответствует выбранному контексту процессора. Это может быть карта физической или виртуальной памяти, в зависимости от типа процессора и состояния контекста.

Все адреса и размерности блоков памяти отображаются в шестнадцатеричном формате.

Независимо от режима отображения данных в заголовке поля детализации отображается информация о типе процессора и режиме его работы, а также тип отображаемой карты памяти.

Для процессоров 1967ВН028 и 1967ВН044 определяется и показывается наличие флага EXT_MODE, а интерпретация и отображение дополнительной информации по карте памяти модуля MPU (в регистрах группы PU) не поддерживается.

Для процессора 1967ВН058 определяется режим активной работы модуля MMU ("V0", "VZ", "VT", "VX", а также размерность раблиц трансляции L1 по значениям полей TMASK и ZMASK регистра TTBCR), с пояснениями по таблицам трансляции L1 и предполагаемым методам управления памятью (программная загрузка TLB, аппаратная загрузка с использованием таблиц трансляций).

В поле детализации также отображаются более подробные сведения о выбранном блоке памяти, включая возможные сообщения об ошибках или предупреждения.

6.11.14.2. Просмотр карты физической памяти процессора

Карту физической памяти можно открыть, выбрав режим отображения данных «Physical», если активной является карта виртуальной памяти.

Карта физической памяти процессора показывается со строкой «Map type: 'Physical'» в поле детализации.

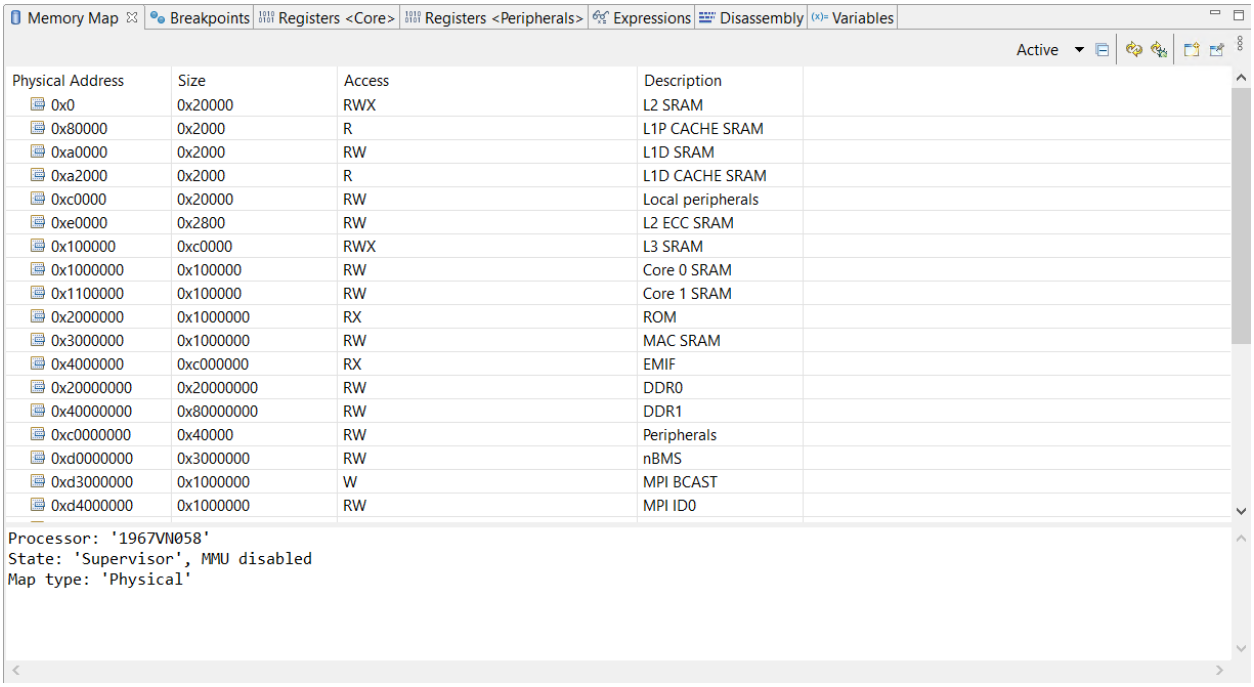


Рисунок 232. Карта физической памяти на вкладке «Memory Map»

В этом режиме для каждого блока физической памяти отображаются:

- **Physical Address** - физический адрес;
- **Size** - размер блока в адресуемых словах;
- **Access** - атрибуты доступа, которые указаны в спецификации на процессор (R – чтение, W – запись, X – выполнение кода);
- **Description** - краткое описание, назначение.

Все блоки физической памяти отсортированы по возрастанию физического адреса.

6.11.14.3. Просмотр карты виртуальной памяти процессора

Карту виртуальной памяти можно открыть, выбрав режим отображения данных «Virtual», если активной является карта физической памяти.

Карта виртуальной памяти процессора показывается при включенном модуле MMU со строкой «Map type: 'Virtual'» в поле детализации.

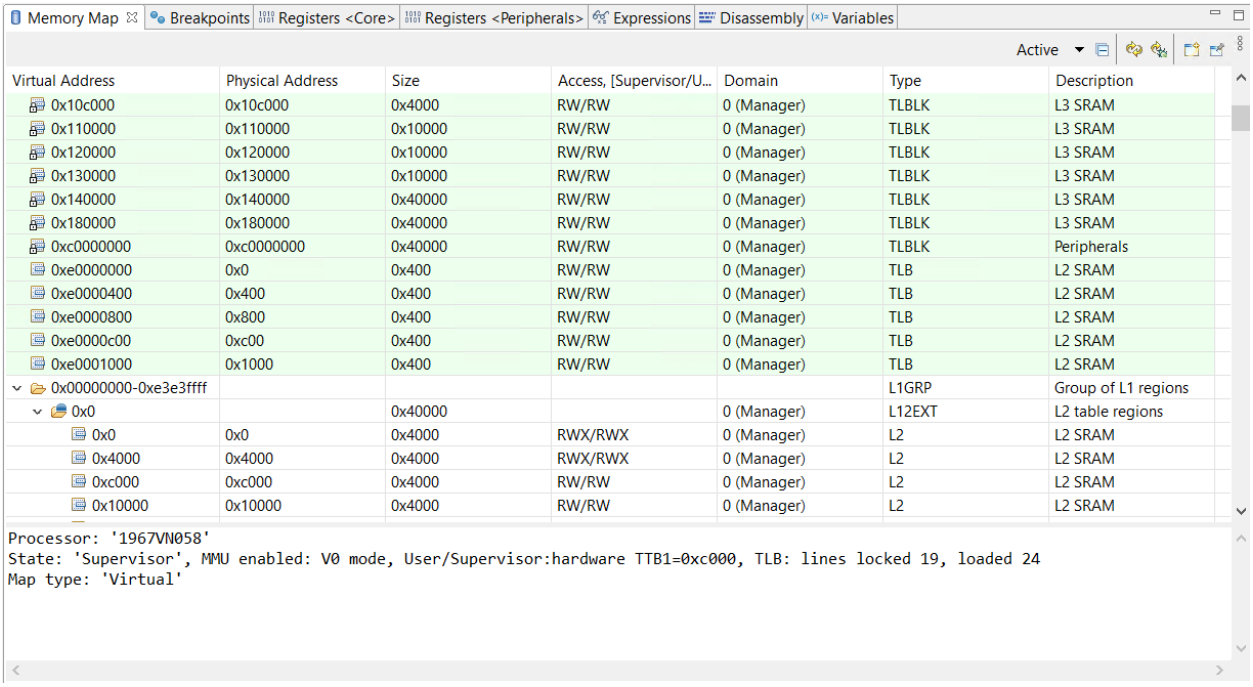


Рисунок 233. Карта вирутальной памяти на вкладке «Memory Map»

Карта виртуальной памяти состоит из следующих полей:

- **Virtual Address** - виртуальный адрес;
- **Physical Address** - физический адрес;
- **Size** - размер блока в адресуемых словах;
- **Access, [Supervisor/User]** - атрибуты доступа для разных режимов процессора (R – чтение, W – запись, X – выполнение кода);
- **Domain** - номер, ID домена и его тип в скобках: «Manager» или «Client»;
- **Type** - тип дескриптора блока памяти. Возможные значения:
  - ***TLBLK*** - заблокированный дескриптор в таблице TLB процессора. Соответствует фиксированной части доступной памяти;
  - ***TLB*** - загруженный неблокированный дескриптор TLB, который может быть автоматически аппаратно перезаписан модулем MMU при выполнении процессором команд. Соответствует динамической части доступной памяти;
  - ***L1GRP*** - виртуальная группа блоков таблицы трансляции L1. Максимально может включать 256 блоков. Предназначена для организации отображаемых данных в отдельной ветке дерева (с целью обеспечения приемлемых временных характеристик функций вкладки «Memory Map»);
  - ***L1*** - дескриптор таблицы L1;
  - ***L12EXT*** - дескриптор таблицы L1, который является расширением на таблицу L2. Группирует входящие блоки уровня L2 в отдельную ветку дерева;

- ***L2*** - дескриптор таблицы L2.

- **Description** - краткое описание, назначение.

Для блоков «L1GRP» показывается диапазон виртуальных адресов входящих блоков, а некоторые не имеющие смысла поля показываются пустыми. Для блоков «L12EXT» показывается начальный виртуальный адрес в поле «Virtual Addres», размер покрытого пространства виртуальных адресов, равный 1Mb - в поле «Size», не имеющие смысла поля «Physical Address» и «Access» - пустыми.

Всегда отображаются первыми блоки доступной памяти активной таблицы TLB в порядке возрастания виртуальных адресов (не в порядке следования дескрипторов в таблице).

Далее отображаются блоки доступной памяти в соответствии с текущим режимом процессора («Supervisor», «User») и атрибутами доступа по информации из действующей таблицы L1 и связанных с ней таблиц L2. Они всегда отображаются в естественном порядке возрастания виртуальных адресов, вместе с дополнительными типами группировки блоков в отдельных ветках «L1GRP» и «L12EXT».

Блоки памяти «TLB» и «TLBLK» в окне подсвечиваются светло-зеленым фоном, так как процессор 1967ВН058 во время останова в режиме эмулятора с включенным MMU в стандартном режиме отображения не может иметь доступ к памяти, которая не отображена по текущему состоянию таблицы TLB. Пользователь может работать в останове только с данными по активной части трансляции TLB. Прозрачная отладка во всем адресном пространстве с автоматической аппаратной подкачкой TLB в процессоре не поддерживается. Несмотря на отсутствие доступа к полной карте памяти со стороны процессора, в режиме эмулятора из отладчика допускаются обращения к областям недоступной памяти без влияния на состояние остановленного контекста процессора или соответствующие области памяти. По недоступным областям памяти могут читаться какие-либо константы, а запись в них не выполняется.

Блоки памяти «L1» 16Mb («Size» 0x40000) и «L2» 64Kb («Size» 0x4000) отображаются с сообщением об ошибке в поле детализации, подсветкой светлым красно-коричневым фоном и пиктограммой , когда в таблицах трансляции отсутствует необходимое число одинаковых дескрипторов.

Active

Virtual Address	Physical Address	Size	Access, [Supervisor...	Domain	Type	Description
0x0		0x40000		0 (Manager)	L12EXT	L2 table regions
0x80000		0x40000		0 (Manager)	L12EXT	L2 table regions
0xc0000		0x40000		0 (Manager)	L12EXT	L2 table regions
0x100000		0x40000		0 (Manager)	L12EXT	L2 table regions
0x100000	0x100000	0x4000	RWX/RWX	0 (Manager)	L2	L3 SRAM
0x104000	0x104000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x108000	0x108000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x10c000	0x10c000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x110000	0x110000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x114000	0x114000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x118000	0x118000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x11c000	0x11c000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x120000	0x120000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x124000	0x124000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x128000	0x128000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x12c000	0x12c000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM
0x130000	0x130000	0x4000	RW/RW	0 (Manager)	L2	L3 SRAM

```
Processor: '1967VN058'
State: 'Supervisor', MMU enabled: V0 mode, User/Supervisor:hardware TTB1=0xc000, TLB: lines locked 0, loaded 24
Map type: 'Virtual'
```

Рисунок 234. Отображение блока памяти при недостатке одинаковых дескрипторов в таблице трансляции (не выбран блок для детализации информации)

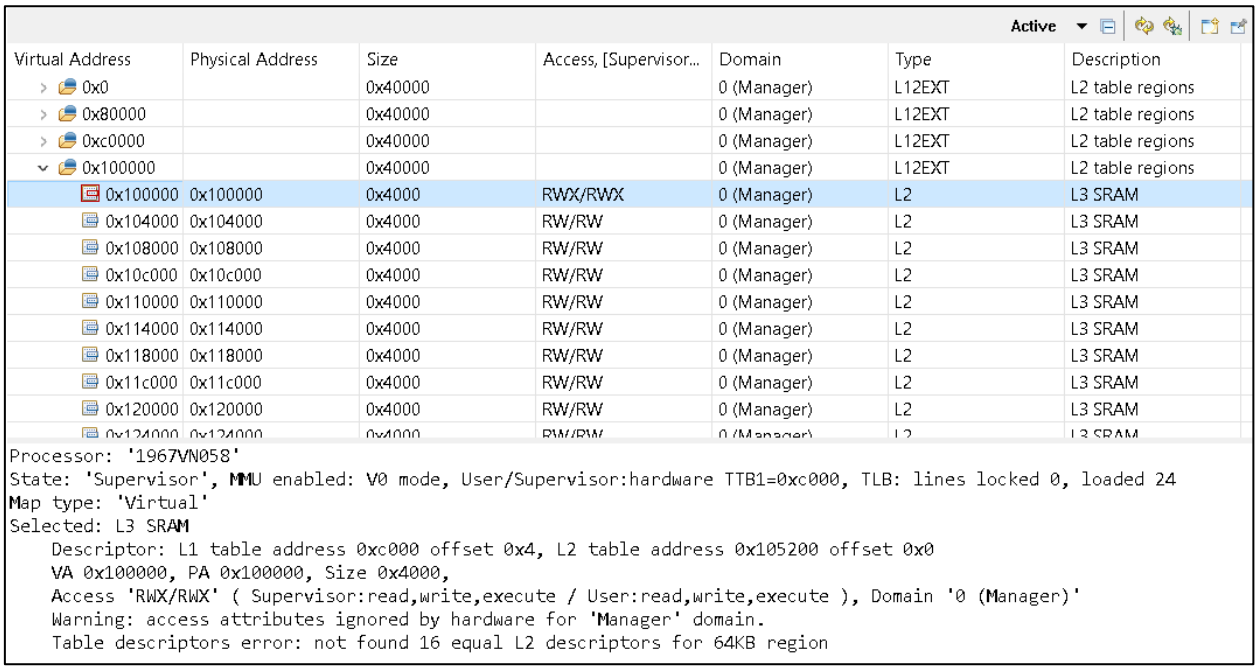


Рисунок 235. Отображение блока памяти при недостатке одинаковых дескрипторов в таблице трансляции (выбрана запись блока с ошибочным дескриптором)

Примечание. Настройки цветовой подсветки могут быть изменены в стандартном интерфейсе настроек «Preferences/General/Appearance/Colors and Fonts/Debug».

Для блоков памяти доменов типа «Manager» в поле детализации показывается предупреждающее сообщение о том, что атрибуты доступа «Access» (поле AP дескрипторов) не проверяются аппаратно, то есть такие блоки являются всегда доступными.

При работе процессора 1967ВН058 в режиме "TA_mode" показывается активная карта виртуальной памяти, состоящая из доступных блоков по таблице TLB и блоков отображения VA=PA карты физической памяти. При работе процессора в режиме совместимости с 1967ВН28 также показывается активная карта виртуальной памяти, состоящая из доступных блоков по TLB и блоков фиксированного отображения виртуального (для 1967ВН058) адресного пространства процессора 1967ВНВН028 на физическую память 1967ВН058.

6.11.14.4. Просмотр детализации для выбранного блока памяти

Для просмотра детальной информации по какому-либо блоку памяти, необходимо выделить интересующую строку (щелчком левой клавишей мыши).

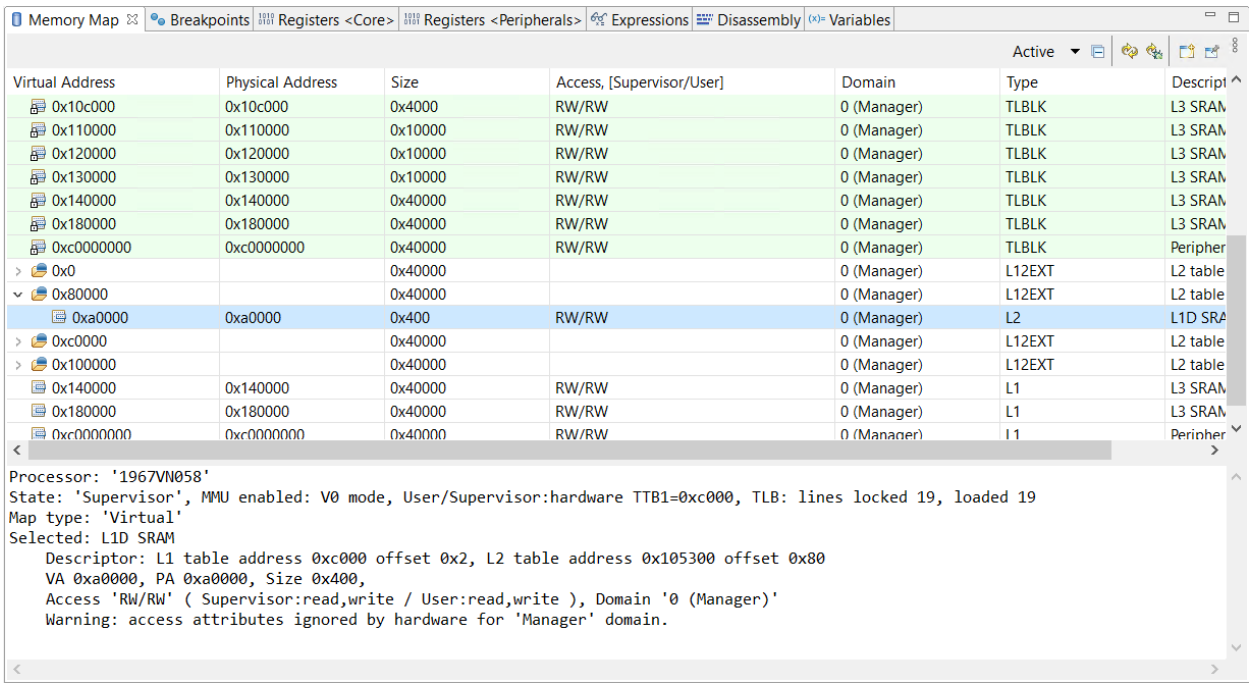


Рисунок 236. Просмотр детализации по выбранному блоку памяти

Отображаемая информация в поле детализации не только содержит общую информацию о режиме работы процессора, но и подробную расшифровку информации о выбранном блоке памяти. Эта информация может быть использована в том числе для определения адресов таблиц трансляций и отдельных дескрипторов в них.

Для просмотра содержимого блока памяти можно дважды щелкнуть левой клавишей мяши. Откроется вкладка «Memory Browser».

6.11.14.5. Обновление отображаемой информации

По умолчанию (без включенного режима «Automatic refresh») данные обновляются только по событиям останова процессора выбранного контекста, если процессор находился в автоматическом режиме выполнения программы после команды «Resume», смене выбранного контекста отладки в окне «Debug», переключении активного окна «Memory Map» на другую вкладку и обратно или по внутренним событиям, связанным с аналогичными действиями. При пошаговой отладке по командам «Step Into», «Step Over», «Step Return» автоматическое обновление данных выполняется только при включенном режиме «Automatic refresh». Это может быть удобно только в особых случаях, поэтому по умолчанию такой режим отключен. Для включения режима нажмите кнопку .

Поддерживается принудительное обновление данных по команде «Refresh» (по кнопке). Команду можно применять в выборочных точках при пошаговом выполнении программы, когда необходимо отобразить предполагаемые изменения в отображаемой информации. Также она может быть востребована при прямом редактировании памяти с таблицами трансляций или системных регистров процессора.

6.11.15. Просмотр результатов профилирования МАКС

Если в отлаживаемом проект разработан с применением ОСРВ МАКС-Lynx, включена и определенных образом настроена трассировка событий ОСРВ МАКС-Lynx, то во время отладки можно просматривать результатов профилирования МАКС в отдельном приложении «МАКС Профилирование. Многопроцессорные системы» (MACSProf). Подробнее о работе с приложением MACSProf см. документ «МАКС Профилирование. Многопроцессорные системы. Руководство пользователя».

Сразу после запуска программа MACSProf начинает работу в режиме приема и накопления данных я перехода в режим отображения результатов обработки и диаграмм необходимо нажать значок («**MACSProf - Stop and View**») на панели инструментов.

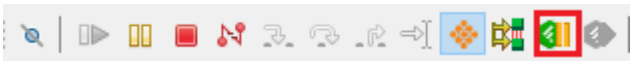


Рисунок 237. Кнопка перехода к просмотру результатов профилирования МАКС

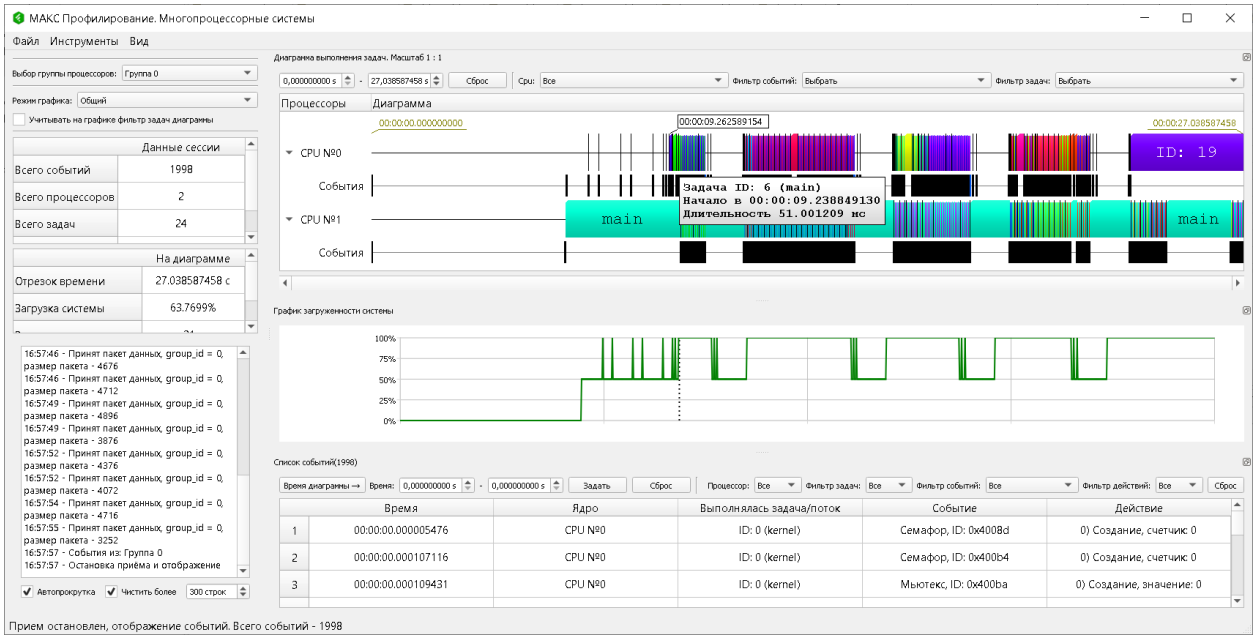


Рисунок 238. Приложение «MACSProf»

Программа MACSProf после перехода в режим отображения результатов обработки и диаграмм прекращает прием данных, а обратный переход в режим приема данных возможен только по команде «**MACSProf – Continue**».

Для возобновления приема и накопления данных профилирования необходимо нажать значок («**MACSProf - Continue**») на панели инструментов .

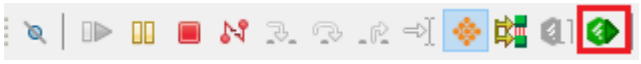


Рисунок 239. Кнопка перехода к режиму накопления данных профилирования

Важно! Команды «**MACSProf - Stop and View**» и «**MACSProf – Continue**» поддерживают режим «**Multisession Commands Mode**». Когда он включен, команды выполняются для всех сессий и связанных с ними контекстов подключения и процессов MACSProf.

6.11.16. Отладка с помощью нескольких сред разработки

В некоторых случаях может потребоваться запуск двух или более сеансов отладки одновременно на одном ПК:

- Пошаговое сравнение работы похожих программ или одной программы в разных условиях;
- Работа с платой, имеющей два или более интерфейсов отладки JTAG;

Для одновременно запуска двух сеансов отладки рекомендуется:

1. Установить вторую копию CM-LYNX в отдельный каталог;
2. Создать во второй CM-LYNX своё рабочее пространство (workspace);
3. Импортировать в рабочее пространство второй CM-LYNX нужный проект, создать там отладочную конфигурацию и т.п., убедиться, что отладка во второй CM-LYNX работает;
4. В отладочной конфигурации второй CM-LYNX задать порт, отличный от порта отладочной конфигурации первой CM-LYNX (чтобы сеансы отладки не мешали друг другу):
    a. Во вкладке «Debugger» окна «Debug Configuration» выбрать тип отладки **JEM** и тип «Connection»: **Remote**;
    b. В появившемся окне **Port** задать любое число (в диапазоне от 1024 до 65535), отличное от 11000 (например, 12000), см. Рисунок 240;
    c. Переключить обратно тип «Connection» на **Local**;
    d. При необходимости – вернуть тип отладки на **SIM**;
    e. Сохранить настройки отладочной конфигурации.
5. Запустить сеансы отладки в обеих CM-LYNX.

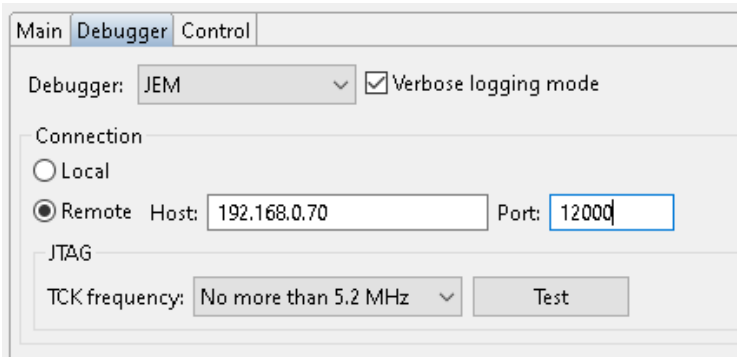


Рисунок 240. Вкладка «Debugger» окна «Debug Configuration» при настройке порта отладочной конфигурации

При работе с несколькими аппаратными отладчиками (например, при работе с платой, имеющей два или более интерфейсов отладки JTAG) необходимо учитывать текущие особенности работы:

- При запуске сеанса отладки CM-LYNX он «захватывает» первый аппаратный отладчик, доступный в операционной системе, после чего этот аппаратный отладчик становится недоступен для другого сеанса отладки (пока первый сеанс не завершится).
- Порядок аппаратных отладчиков в списке определяет операционная система, как правило он всегда одинаков для одного набора отладчиков.

Таким образом, первый запускаемый сеанс отладки из CM-LYNX, будет всегда захватывать «первый» аппаратный отладчик, а второй запускаемый сеанс отладки из другой CM-LYNX (одновременно с первым), будет захватывать «второй» аппаратный отладчик.

В результате порядок запуска сеансов отладки будет определять, с каким аппаратным отладчиком (и, соответственно, – каким портом JTAG) будет работать каждая из CM-LYNX.

6.11.17. Отладка с помощью буфера трассировки

Помимо стека вызовов можно также использовать буфер трассировки, встроенный в процессор. Буфер трассировки позволяет хранить от 16 до 32 записей о последних переходах, включая переходы внутри функций, возвраты и т.д. Более подробная информация о буфере есть в спецификации на процессор.

Буфер трассировки позволяет выполнять отладку в ситуациях, когда разрушается стек вызов и т.п.

Работа с буфером трассировки осуществляется с помощью вкладки «**Trace Core**» (см. также п. 5.4.3.9).

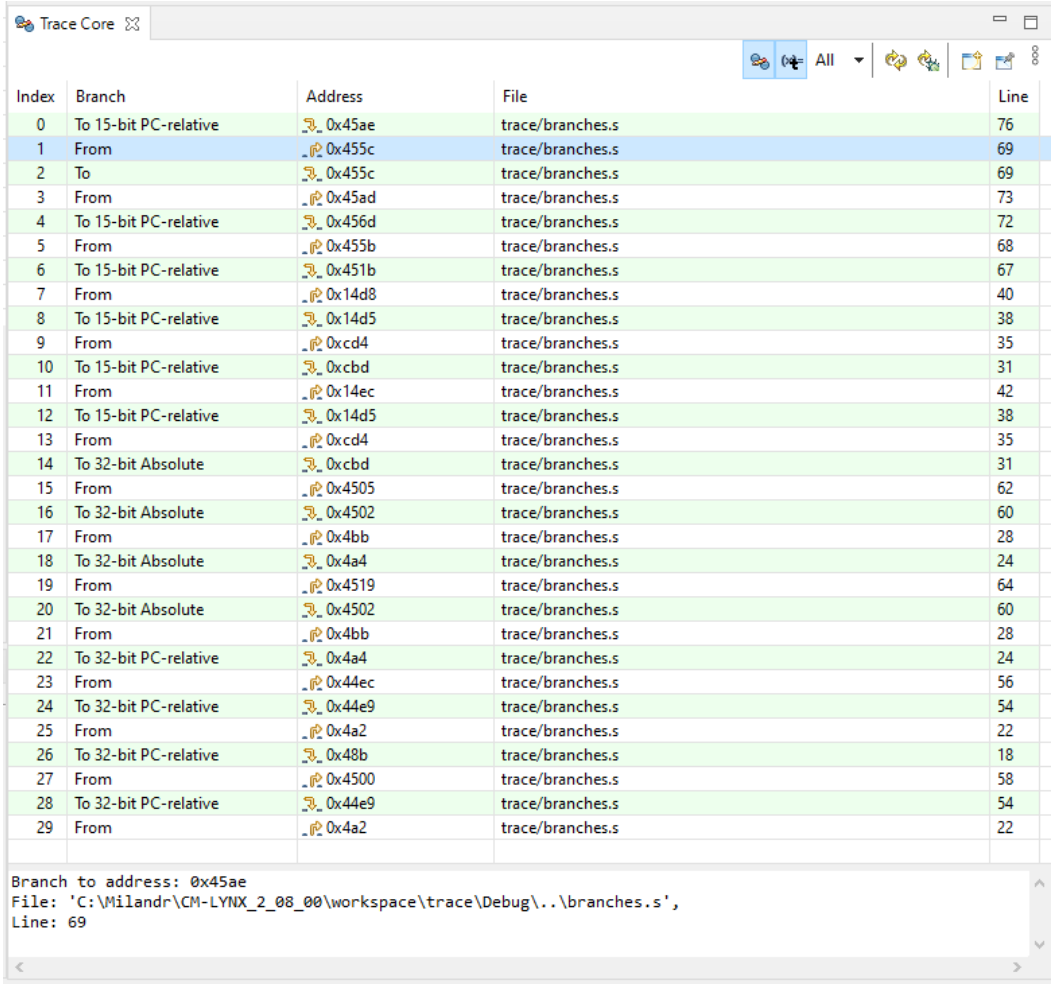


Рисунок 241. Вкладка «Trace Core»

На вкладке есть кнопка состояния и управления «», позволяющая включать и выключать буфер трассировки; а также настройки отображения записей буфера.

В основной части вкладки отображается таблица с записями буфера, а внизу – подробная информация о выбранной записи. Информация об исходном файле и строке определяется по отладочной информации на основе адреса.

Кнопка состояния и управления реагирует также на включение и выключение буфера трассировки программно, через запись в регистр и т.п.

При двойном нажатии левой кнопкой мыши на записи открывается исходный файл, связанный с соответствующим адресом, и выполняется позиционирование курсора на строке в редакторе. При отсутствии необходимой отладочной информации, а также при невозможности открыть исходного файла, открывается окно дизассемблера с позиционированием курсора на адресе.

Поддерживается копирование выделенных записей в буфер обмена в текстовом виде.

Настройки автоматического открытия вкладки «**Trace Core**», обновления кнопки состояния и управления при переключении контекстов, а также формата колонки «**File**» находятся в настройках рабочего пространства – см. п. 6.12.2.4.7.

6.12. Персональные настройки

Настройка основных параметров рабочего пространства осуществляется в окне «**Preferences**» (см. Рисунок 242). Для открытия окна следует выбрать в главном меню **Window →Preferences**.

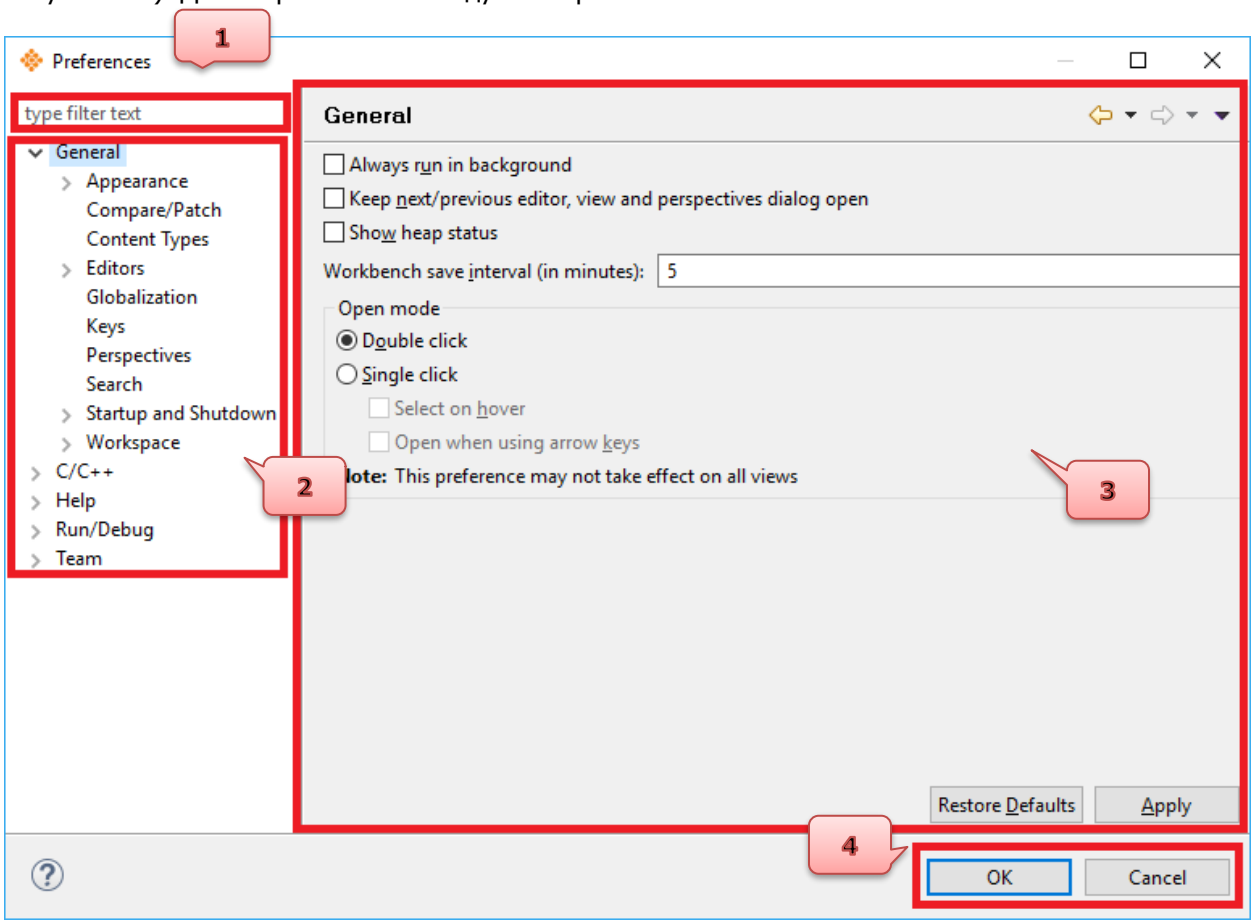


Рисунок 242. Окно настроек «Preferences»

В окне настроек расположены:

1. Поле для поиска параметров с помощью фильтра. Поиск осуществляется как по названиям заголовков, так и, по ключевым словам, например, «appearance» и «C/C++»;

2. Перечень объектов, доступных для настройки;

3. Вкладка с настройками выбранного в перечне объекта;

4. Функциональные кнопки принятия/отмены изменений.

Стрелки и , расположенные в правой верхней части окна, позволяют перемещаться по ранее просмотренным позициям. Также можно выбрать нужный объект из списка ранее просмотренных, который раскрывается при нажатии на кнопку .

6.12.1. Общие настройки

Общие настройки задаются на панели «**General**».

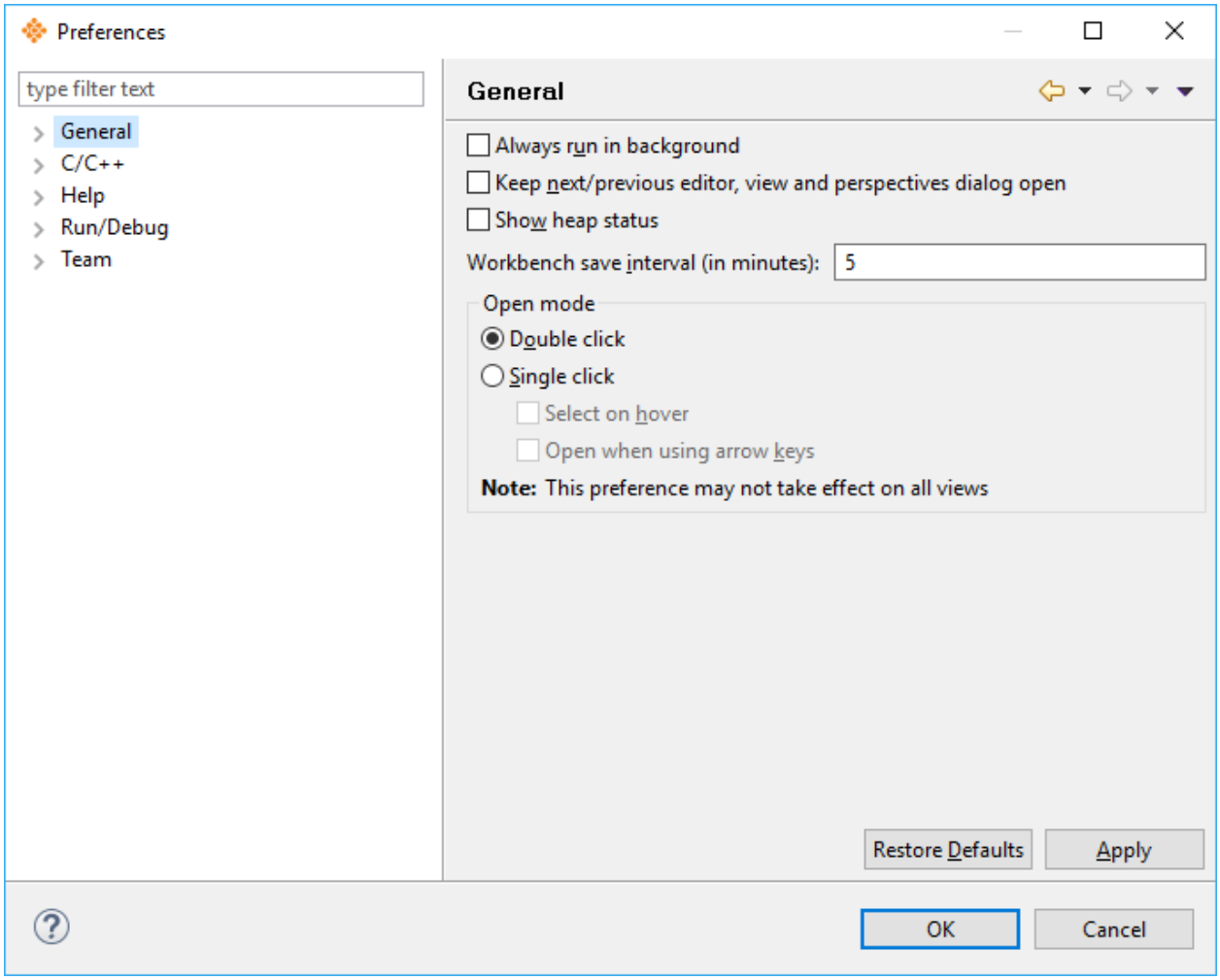


Рисунок 243. Панель «General»

Таблица 100. Элементы управления панели «General»

Название элемента	Описание функционирования	По умолчанию
Always run in background	Установить режим выполнения долгих операций в фоновом режиме без отрыва от другой работы.	Выкл
Keep next/ previous part dialog open	Установить режим, при котором всплывающие диалоговые окна остаются открытыми, после того как клавиша, вызывающая их, отпущена. Обычно диалоговые окна закрываются, как только отпускается комбинация клавиш.	Выкл
Show Heap Status	Установить режим отображения индикатора, отображающего информацию о текущем использовании динамической области.	Выкл
Workbench save interval (in minutes)	Установить интервал автоматического сохранения в минутах.	5

Название элемента	Описание функционирования	По умолчанию
Open mode...	Выбрать способ открытия ресурса: ▪ **Double click** – при установке флажка выбран режим, при котором один щелчок мыши выбирает ресурс, двойной – открывает его в редакторе; ▪ **Single click** – при установке флажка можно выбрать один из способов выбора ресурса: – **Select on hover** – выбор режима, при котором ресурс выбирается наведением курсора и щелчок мыши открывает его в редакторе; – **Open when using arrow keys** – выбор режима, при котором ресурс выбирается с помощью клавиш стрелки. Примечание: в зависимости от того, на какой вкладке расположен фокус, выбор и открытие ресурсов может быть разным.	Двойной щелчок

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.1. Настройки внешнего вида

Настройки внешнего вида находятся на панели «**Appearance**». Для переключения на панель «**Appearance**» для настройки внешнего вида следует перейти в перечне к пункту **General → Appearance**.

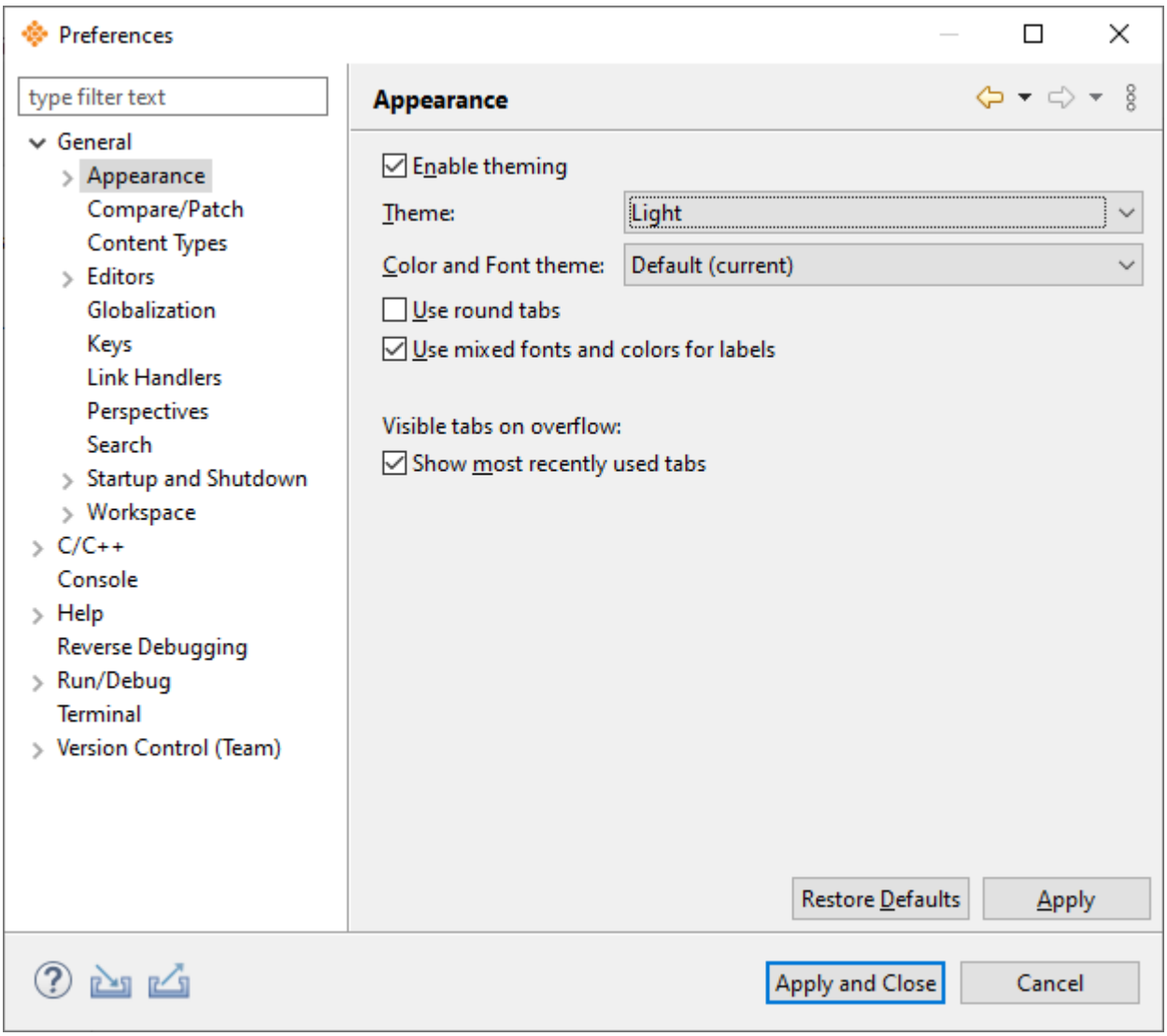


Рисунок 244. Панель «Appearance»

Таблица 101. Параметры, доступные для настройки на панели «Appearance»

Название элемента	Описание функционирования	По умолчанию
Theme	Задать внешний вид активной темы.	Зависит от операционной системы
Color and Font theme	Задать цвет и тип шрифта. Подробнее см п. 6.12.1.1.1 Настройки шрифтов.	Зависит от операционной системы
Enable animations	Разрешить показывать анимированные элементы интерфейса.	Выкл
Use mixed fonts and colors for labels	Использовать смешанный набор шрифтов и цветовых схем для ярлыков.	Вкл
Show most recently used tabs	Показывать недавно открытые вкладки.	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

Примечания:

1. После смены темы рекомендуется перезапустить CM-LYNX;
2. Некоторые элементы могут отображаться в цветах темы по-умолчанию;

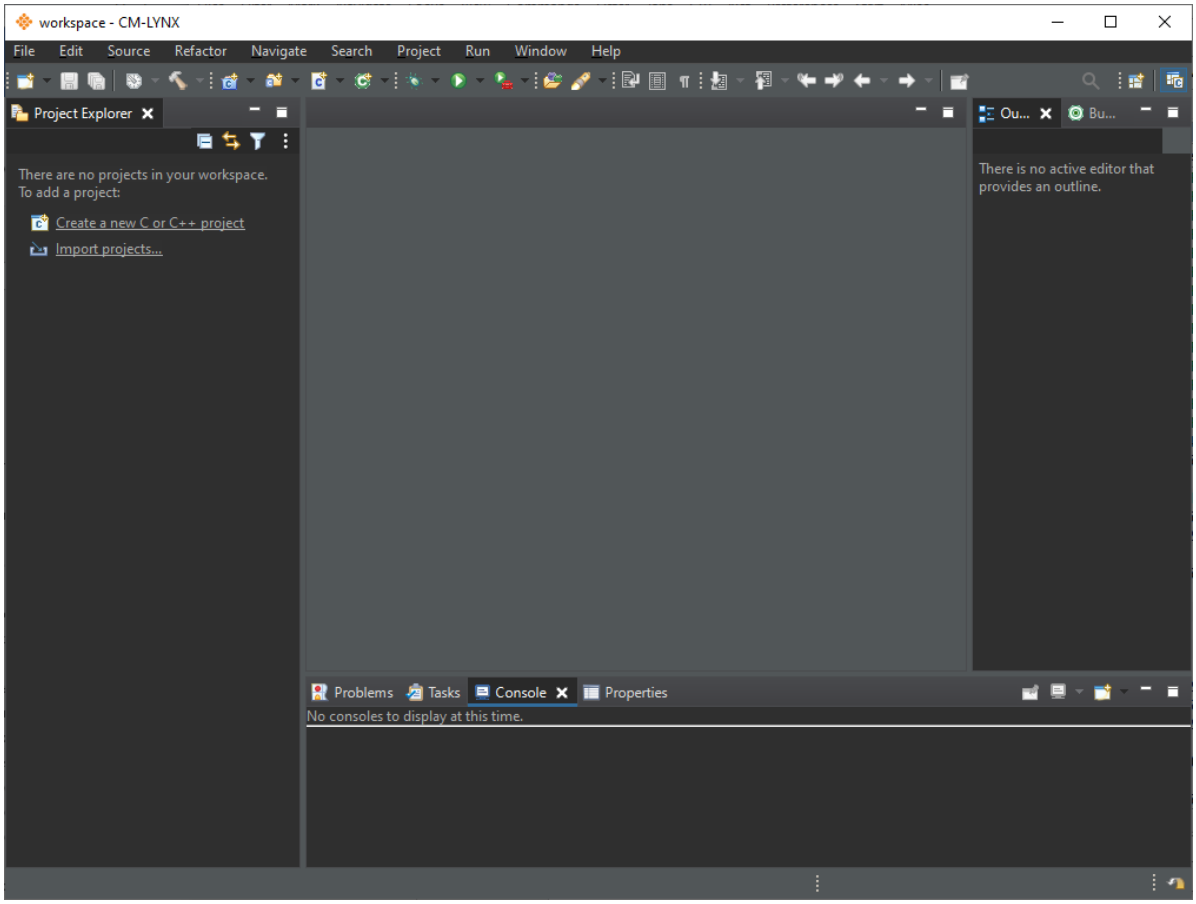


Рисунок 245. Главное окно CM-LYNX при использовании темы "Dark"

6.12.1.1.1. Настройки шрифтов

Большинство шрифтов и цветов, используемых компонентами среды, могут быть установлены на панели настроек **General → Appearance → Colors and Fonts**.

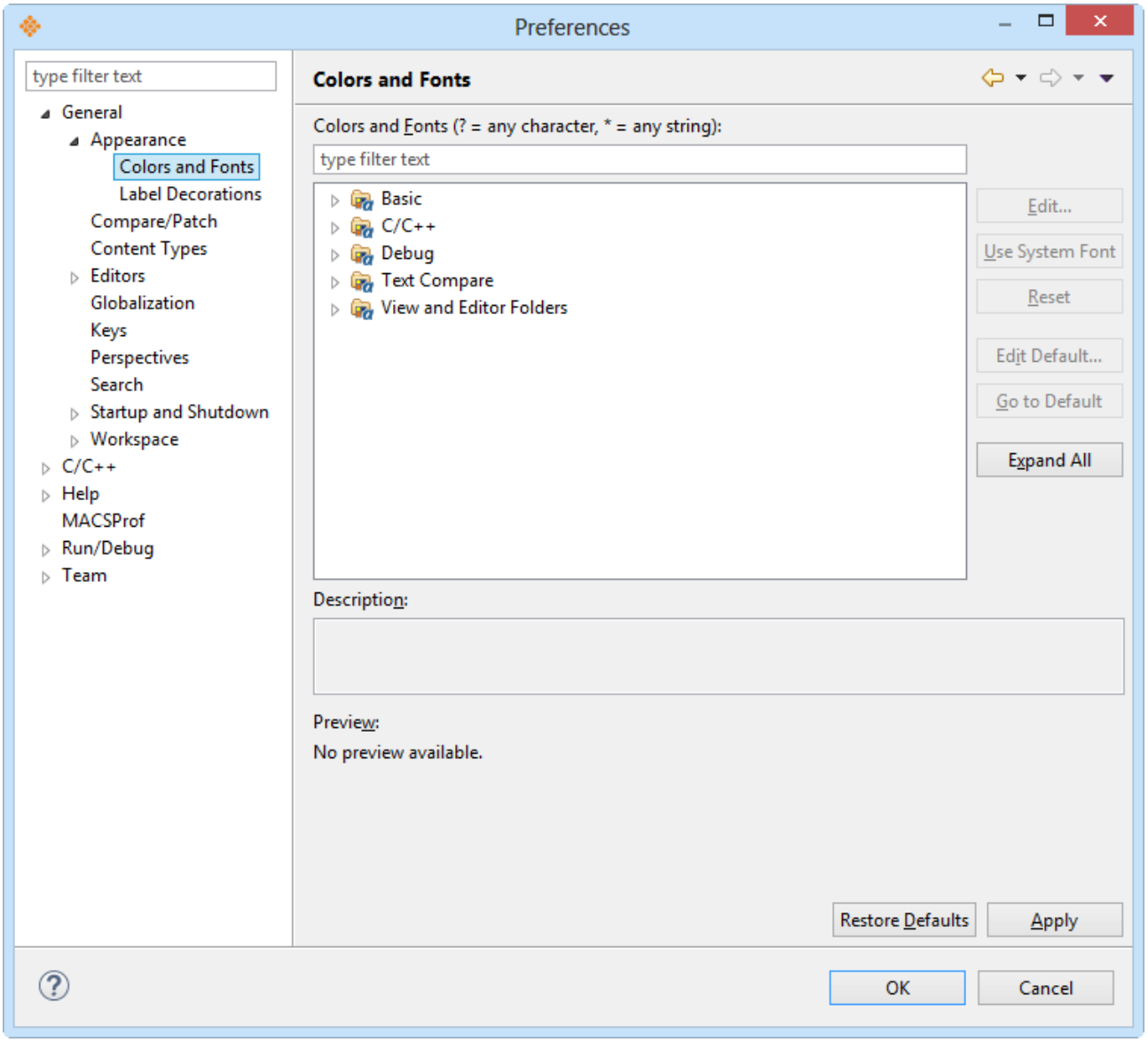


Рисунок 246. Панель «Colors and Fonts»

Для работы и предварительного просмотра шрифтов и цветов используется древовидное представление. Внешний вид шрифта и цвет показаны на соответствующем ему значке.

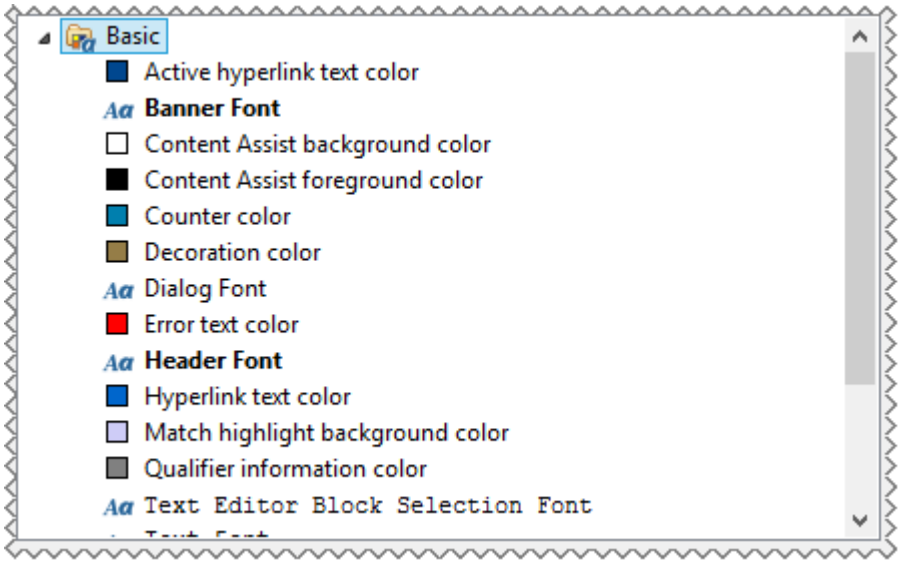


Рисунок 247. Перечень шрифтов

Для смены шрифта следует выбрать нужный шрифт в списке и нажать кнопку «**Use System Font**» или нажать кнопку «**Edit**» и в открывшемся окне выбрать новый шрифт.

Настройки цвета можно изменить, дважды щелкнув на цветной значок в древовидном списке.

Для возврата к шрифту по умолчанию следует нажать кнопку «**Reset**».

Для просмотра шрифта, который является шрифтом по умолчанию в данной группе, следует нажать кнопку «**Go to Default**». Курсор перейдёт в списке на шрифт по умолчанию.

Для изменения шрифтов по умолчанию следует выбрать в группе шрифт и нажать кнопку «**Edit Default...**». В открывшемся окне задать шрифт, стиль написания, размер и нажать кнопку «**OK**».

Для удобного поиска нужного шрифта можно воспользоваться строкой фильтра. Ключевое слово вводится в строке фильтра в верхней части панели.

Если необходимо просмотреть все имеющиеся шрифты в древовидном списке, следует нажать кнопку «**Expand All**».

При выборе цвета или шрифта из списка его описание отображается в поле предварительного просмотра «**Preview**» (см. Рисунок 248), расположенного в нижней части панели.

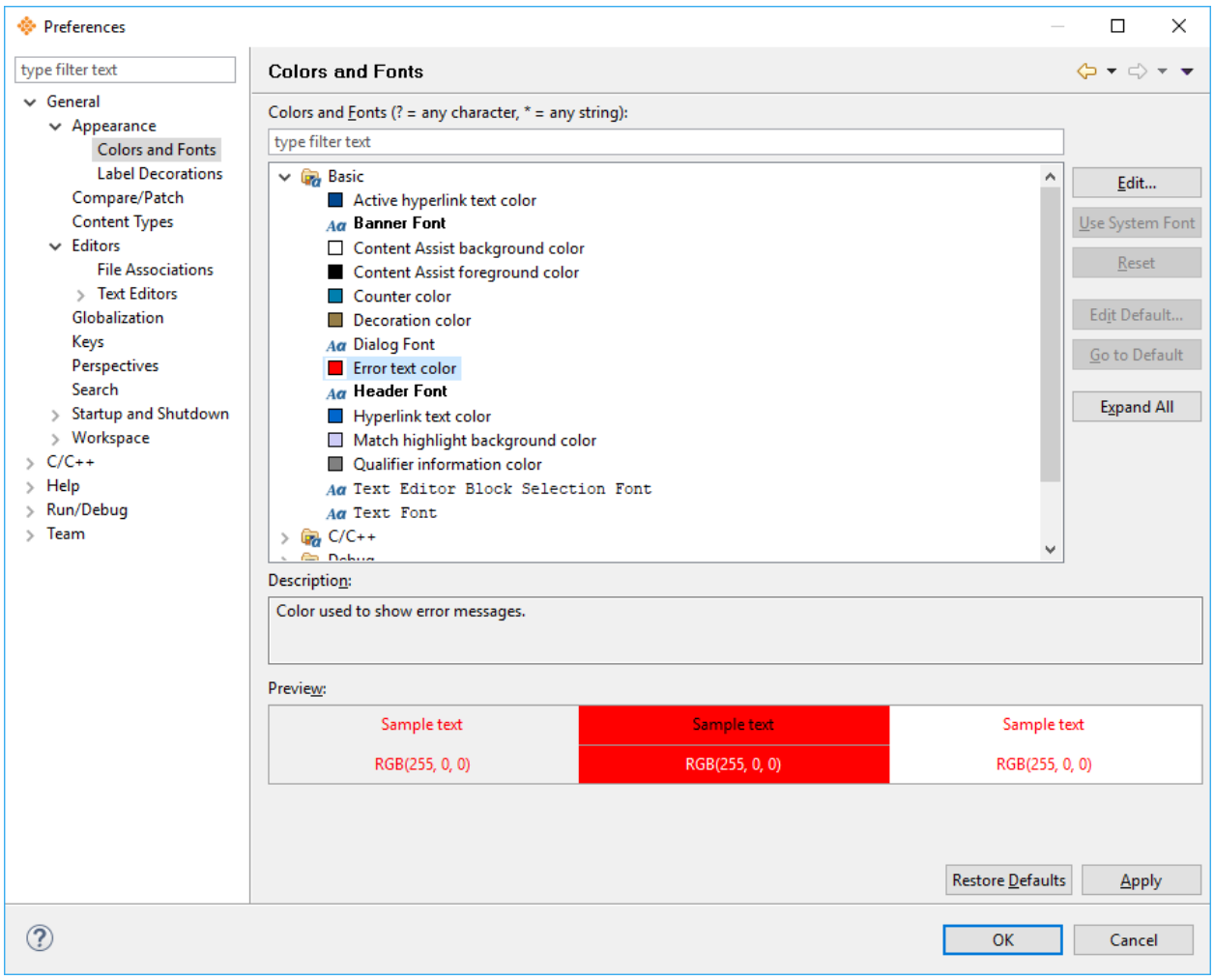


Рисунок 248. Предварительный просмотр шрифтов

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.1.2. Настройки отображения дополнительной информации на значке объекта

Панель «**Label Decorations**» позволяет настроить отображение дополнительной информации на значке объекта.

Для перехода на панель необходимо выбрать **General → Appearance → Label Decorations**. На панели содержится список вариантов оформления, доступных для выбора.

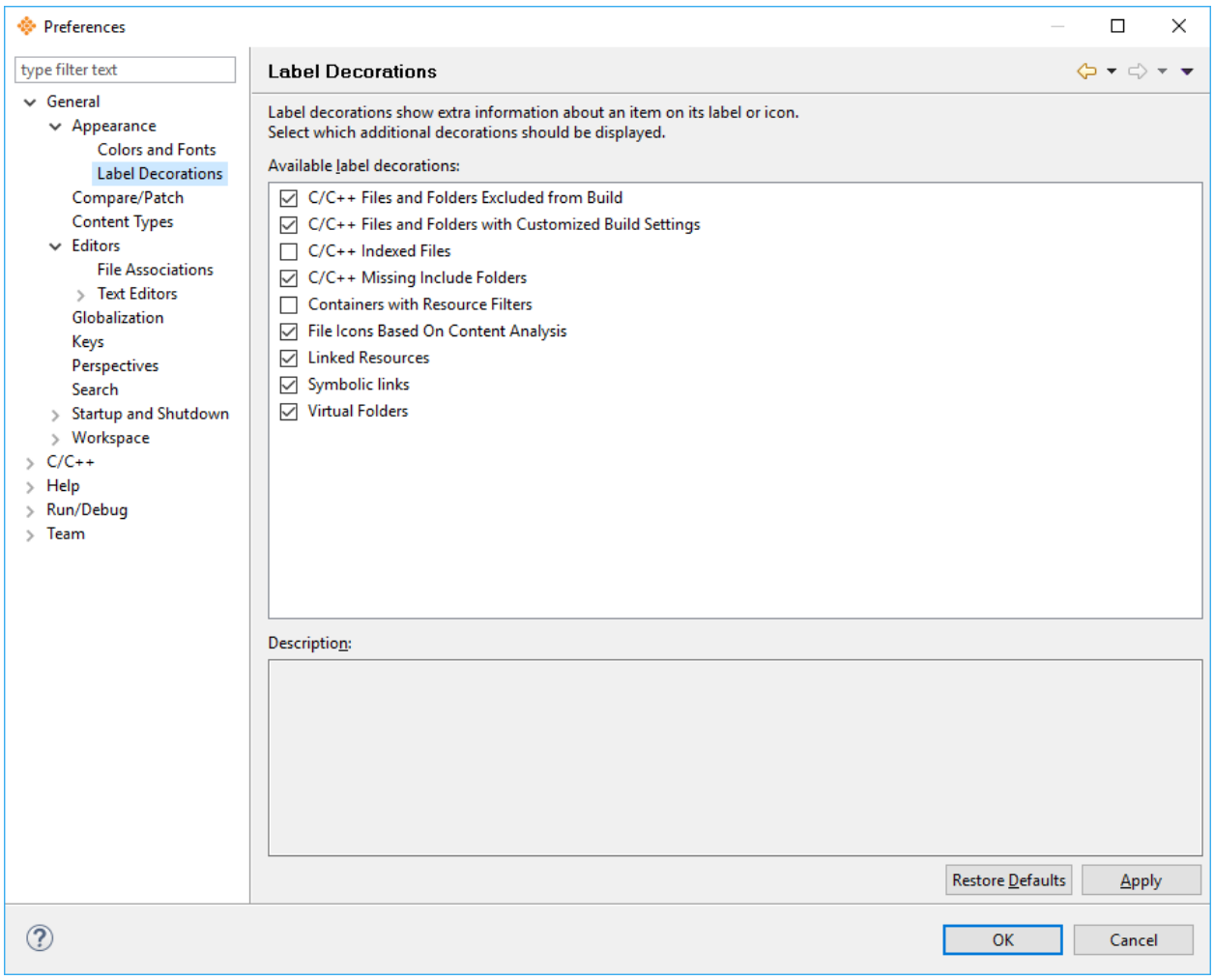


Рисунок 249. Панель «Label Decorations»

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.2. Настройки сравнения и патча элементов кода

Настройки сравнения и патча элементов кода находятся на панели «**Compare/Patch**». Для перехода на панель «**Compare/Patch**» следует выбрать **General → Appearance → Compare/Patch**.

Панель «**Compare/Patch**» содержит две вкладки «**General**» и «**Text Compare**».

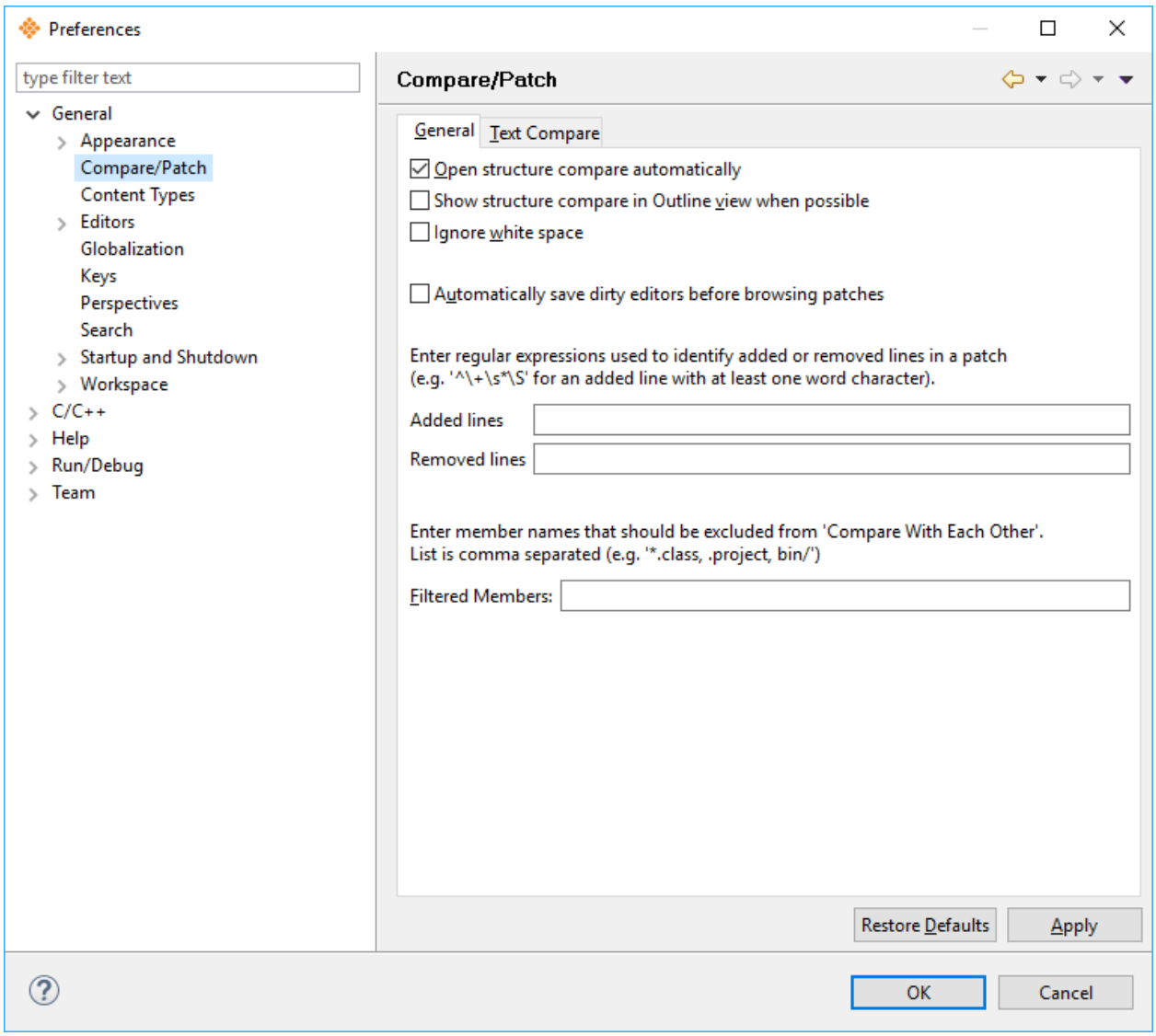


Рисунок 250. Панель «Compare/Patch», вкладка «General»

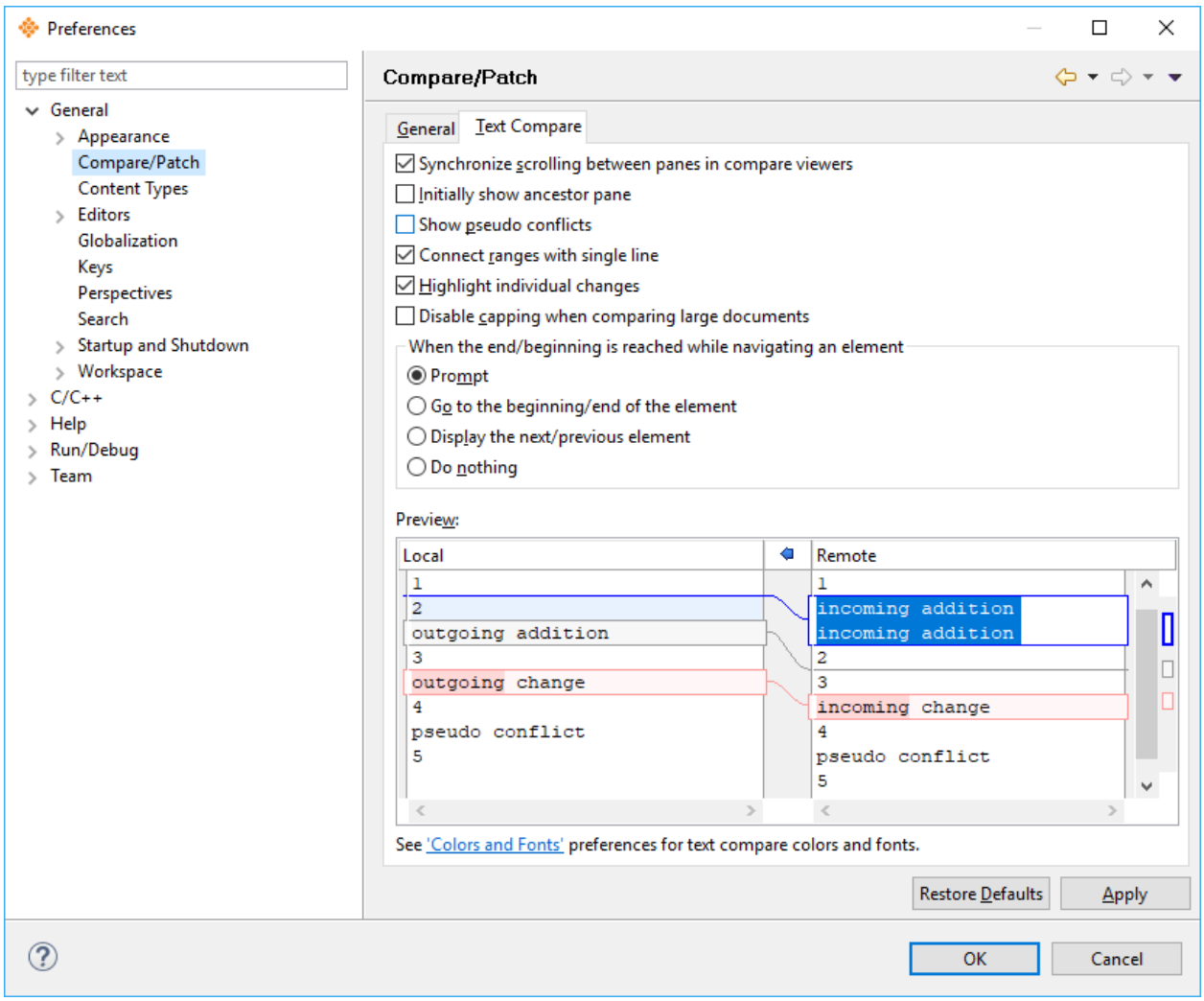


Рисунок 251. Панель «Compare/Patch», вкладка «Text Compare»

Таблица 102. Элементы управления панели «Compare/Patch», вкладка «General»

Название элемента	Описание функционирования	По умолчанию
Open structure compare automatically	Отключить/Включить автоматическое сравнение структуры каждый раз, когда заканчивается сравнение содержимого.	Вкл
Show structure compare in Outline view when possible	Отображать/Скрыть сравнение структуры на вкладке «**Outline**», когда это возможно.	Выкл
Show additional compare information in the status line	Отображать/Скрыть дополнительную информацию об изменениях в строке состояния.	Выкл
Ignore white space	Отображать/Скрыть неотображаемые символы на панели сравнения.	Выкл
Automatically save dirty editors before browsing patches	Включить/Отключить автоматическое сохранение несохраненных изменений до применения обновления.	Выкл
Added/Removed lines	Указать строки, которые должны считаться добавленными или удаленными при применении обновления. Обе опции основаны на регулярных выражениях.	

Название элемента	Описание функционирования	По умолчанию
Filtered Members	Указать участников, которых надо исключить при сравнении «**Compare With Each Other**». Примечание: названия в списке должны отделяться запятыми.	

Таблица 103. Элементы управления панели «Compare/Patch», вкладка «Text Compare»

Название элемента	Описание функционирования	По умолчанию
Synchronize scrolling between panes in compare viewers	Включить/Отключить одновременную прокрутку в двух областях просмотра на соседних панелях для отображения идентичных фрагментов кода при сравнении.	Вкл
Initially show ancestor pane	Показать/Скрыть панель с общим предком, которая всегда отображается при сравнении двух версий ресурса с предыдущей версией, от которой они получены (общий предок). Исходная версия отображается на отдельной панели при сравнении трех версий.	Выкл
Show pseudo conflicts	Показать/Скрыть псевдоконфликты, возникающие, когда два разработчика делают одно и то же изменение (например, оба добавляют или удаляют одну и ту же строку кода или комментарий).	Выкл
Connect ranges with single line	Соединить разные диапазоны визуально одной линией или ограничить диапазон двумя линиями.	Вкл
Highlight individual changes	Выделить (подчеркнуть) отдельные изменения внутри конфликтов.	Вкл
Disable capping when comparing large documents	Включить/Отключить механизм наложения для быстрого нахождения различий в больших файлах. При наложении не будет пропущено ни одно изменение, даже в случае большого количества различий.	Выкл
When the end/beginning is reached while navigating an element	Задать действие по достижению начала/конца во время навигации по элементу. ▪ **Prompt** – если опция включена, и для сравнения выбран один элемент, при достижении начала/конца элемента отобразится предложение перейти в начало/конец элемента. Если для сравнения выбраны два и более элементов, будет предложено переместиться в начало/конец текущего элемента или перейти к предыдущему/следующему элементу. Если выбрать опцию «**Запомнить выбор**», то будет применён один из двух вариантов из описанных ниже: ▪ **Go to the beginning/end of the element** – если опция включена, по достижении конца/начала элемента произойдет перемещение обратно в начало/конец; ▪ **Display the next/previous element** – если опция включена, и сравниваются два или более элементов, по достижении конца/начала элемента откроется предыдущий/следующий элемент; ▪ **Do nothing** – если опция включена, никаких действий предпринято не будет.	Prompt

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.3. Настройки типов содержимого

Панель «**Content Types**» позволяет редактировать типы содержимого, связанные с ними названия файлов и наборы символов. Также можно установить связь между произвольными именами файлов или их расширением с типами содержимого (например, XML-файлы). IDE использует это описание в различных ситуациях, таких как просмотр редактора и сравнение файла.

Для перехода на панель «**Content Types**» следует выбрать **Window → Preferences → General → Content Types**.

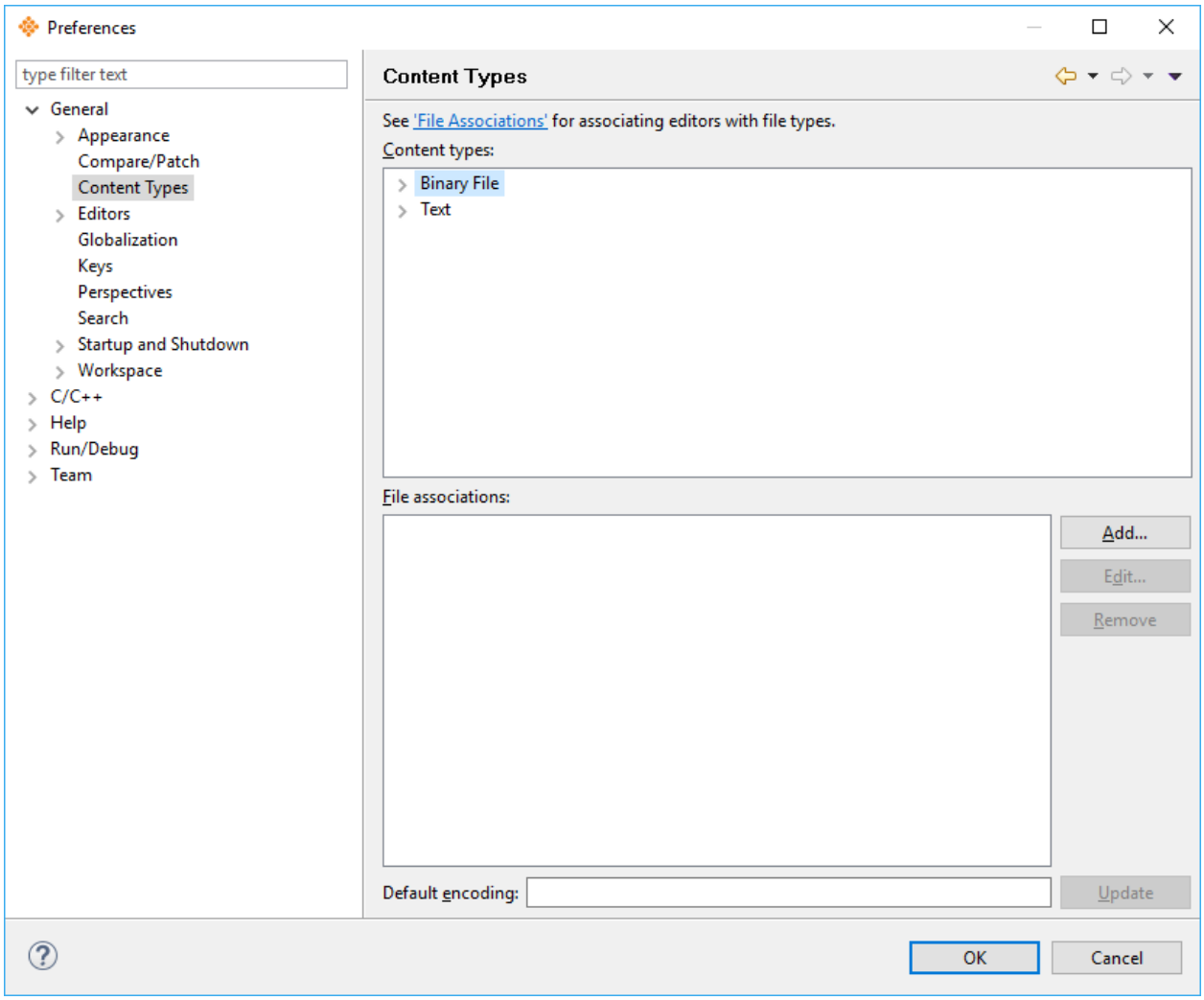


Рисунок 252. Панель «Content Types»

Примечание: некоторые элементы могут быть отмечены как «locked» (заблокированные). Элемент блокируется, если одна из связей, обеспеченных дополнительным программным модулем, указывает на тип содержимого. Другими словами, можно изменять только те связи, которые были добавлены пользователем.

Для того чтобы добавить связь, следует:

1. Нажать «**Add**...».
2. В открывшемся диалоговом окне (см. Рисунок 253) ввести имя файла или расширение.
3. Нажать кнопку **«OK»**.

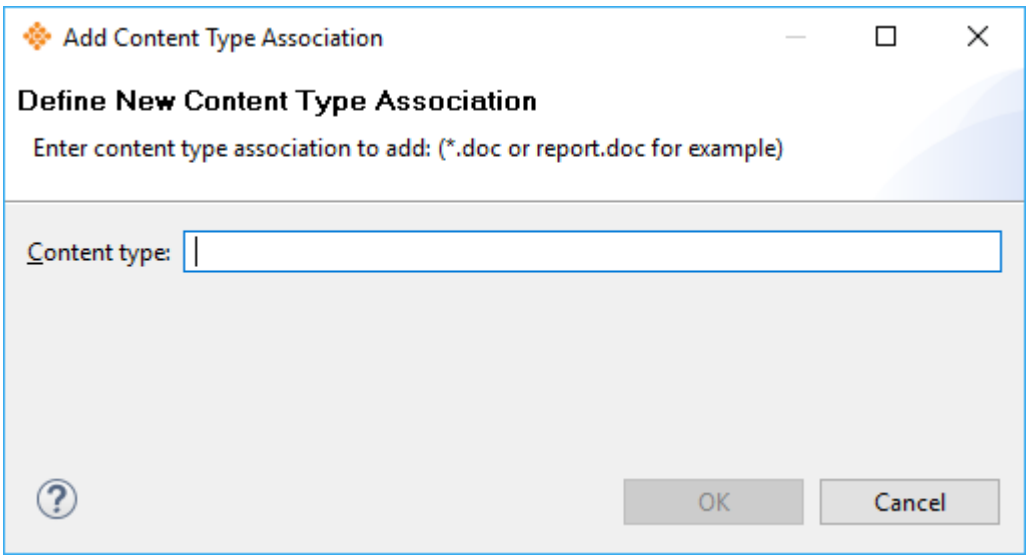


Рисунок 253. Окно «Add Content Type Association»

Для редактирования и удаления названий и расширений файлов используются кнопки «**Edit**» и «**Remove**» соответственно.

Также можно установить кодировку по умолчанию для данного типа содержимого. Для этого следует ввести название кодировки в текстовом поле «**Default encoding»** в нижней части панели «**Content Types**» и нажать кнопку «**Update»**.

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.4. Настройки редакторов

Настройки работы редакторов находятся на панели «**Editors**». Для перехода на панель «**Editors**» следует выбрать **Window → Preferences → General → Editors**.

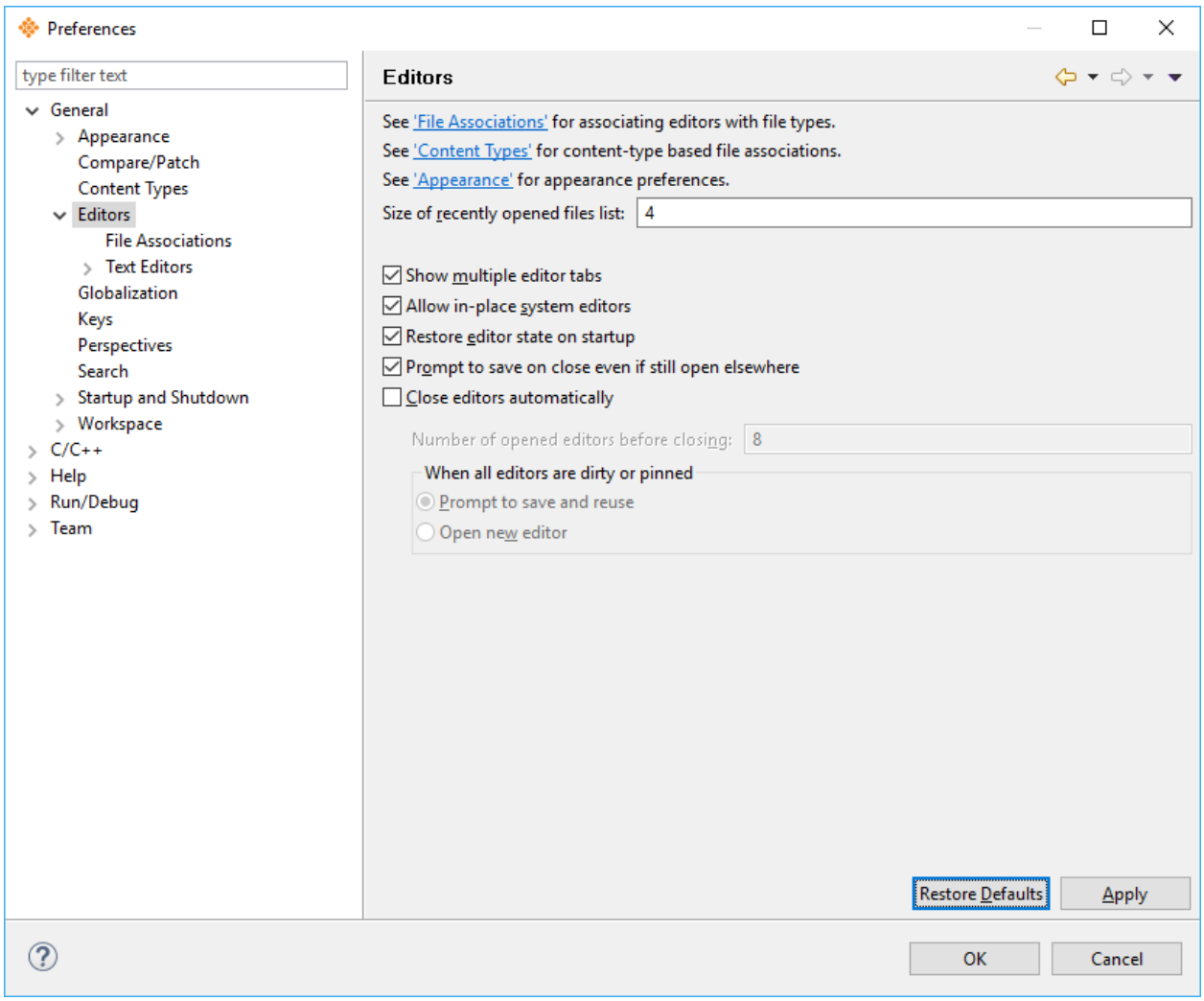


Рисунок 254. Панель «Editors»

Таблица 104. Элементы управления панели «Editors»

Название элемента	Описание функционирования	По умолчанию
Size of recently opened files list	Задать количество файлов, отображаемых в меню «**File**», в списке недавно использованных файлов.	4
Show multiple editor tabs	Показать/Скрыть несколько вкладок редактора. Если опция выключена (флажок снят), в редакторе отображается одна вкладка, и скрытые вкладки доступны только из раскрывающегося списка.	Вкл
Restore editor state on start-up	Включить/Отключить опцию восстановления вида редактора при запуске программе в том виде, в каком он был в последнюю рабочую сессию.	Вкл
Prompt to save on close even if still open elsewhere	Сохранить изменения при закрытии, даже если какие-то окна открыты.	Вкл

Название элемента	Описание функционирования	По умолчанию
Close editors automatically	Повторно использовать редакторы. Если опция включена, можно указать количество редакторов для использования перед повторным использованием (по умолчанию – 8). Также с помощью переключателя можно выбрать одно из действий в случае, когда все редакторы «грязные» (есть несохраненные изменения) или закреплены: ▪ **Prompt to save and reuse** – открыть диалоговое окно приглашения сохранить или использовать повторно; ▪ **Open new editor** – открыть новый редактор; Когда опция включена, на панель инструментов добавляется элемент «**Pin Editor**» для закрепления редактора. Закрепленные редакторы не используются повторно.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.4.1. Настройки ассоциации файлов с редактором

Панель «**File Associations**» предназначена для добавления или удаления типов файлов, распознаваемых в среде разработки, а также установки связи редактора с типом файлов в списке.

Для перехода на панель «**File Associations**» следует выбрать **General → Editors → File Associations**.

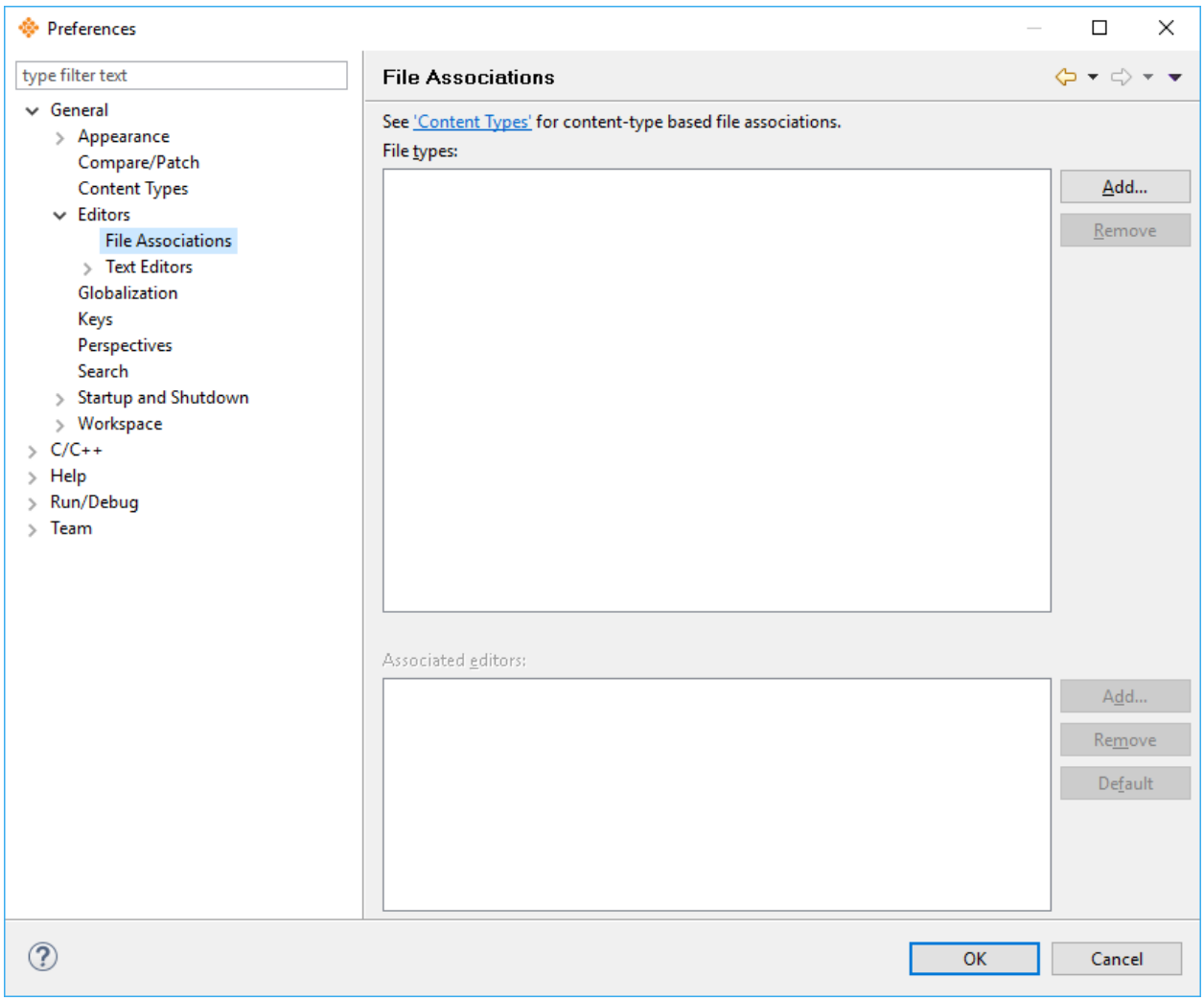


Рисунок 255 Панель «File Associations»

Элементы управления при работе со списком типов файлов (область «**File types**»):

- кнопка «**Add...**» – добавляет новый файл или тип файла (расширение) в готовый список. В отобразившемся диалоговом окне «**Add File Type**» в текстовом поле укажите имя файла или расширение. При добавлении расширения обязательно ставится «.» или комбинация «.*» перед типом (например, «.xml» или «*.xml»).

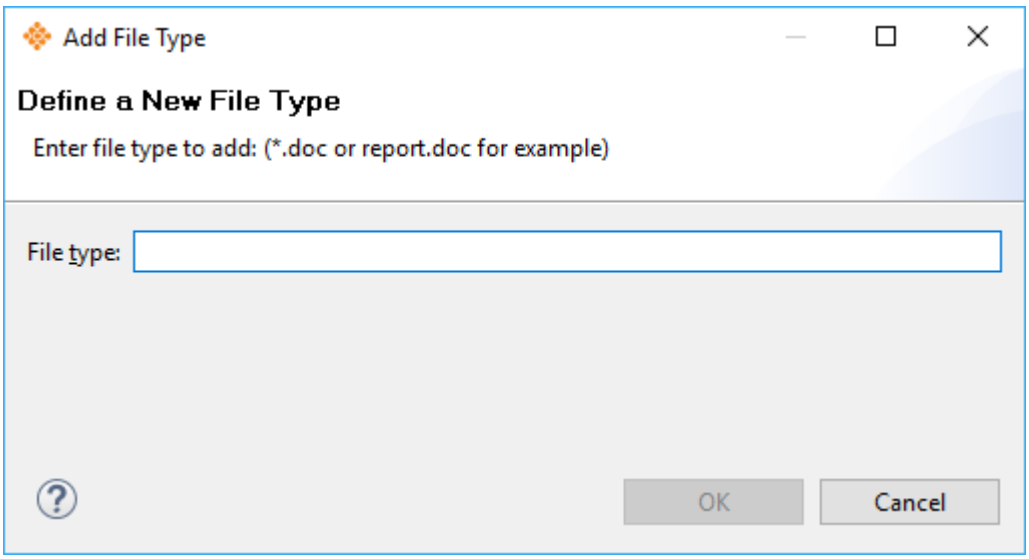


Рисунок 256. Диалоговое окно «Add File Type»

- кнопка «**Remove**» – удаляет выбранный тип файла из списка.

Элементы управления при работе со списком связанных редакторов (область «**Associated editors**»):

- кнопка «**Add...**» – добавляет новый редактор к списку редакторов, связанных с файлами, выбранными выше. В отобразившемся диалоговом окне «**Editor Selection**» можно выбрать редактор для запуска в среде разработки (внутренний) или вне среды (внешний). Если редактор не отображается в списке, следует нажать кнопку «**Browse**» для самостоятельного поиска редактора.

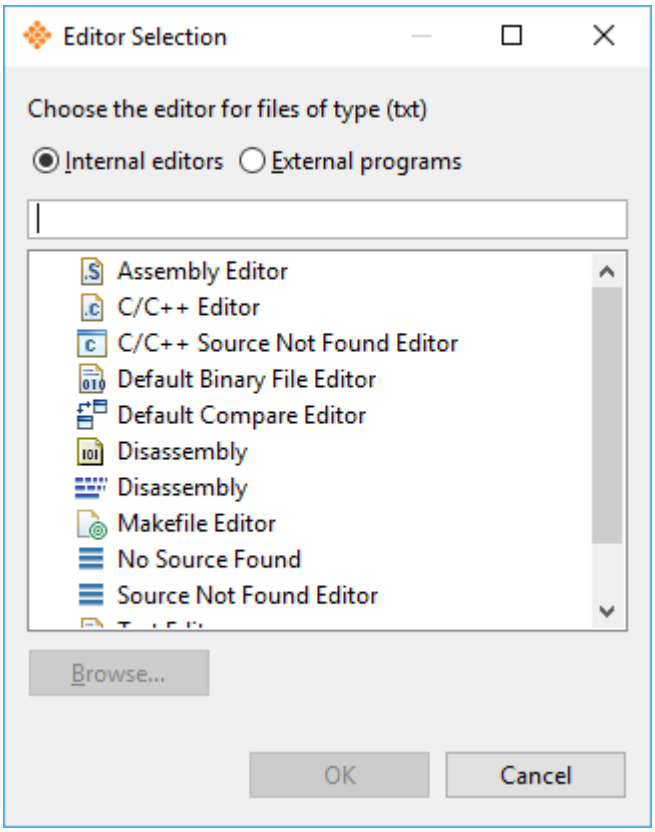


Рисунок 257. Диалоговое окно «Editor Selection»

- кнопка «**Remove**» – удаляет связь между редактором и типом файлов, выбранным выше.

Примечание: любой редактор, связанный типом содержимого, не может быть удален из списка. На данный момент нет возможности удалять такие редакторы.

- кнопка «**Default»** – устанавливает выбранный редактор в качестве редактора по умолчанию для типа файла, выбранного выше. Редактор перемещается в верх списка «**Associated editors**», чтобы указать, что этот редактор является редактором по умолчанию для данного типа файлов.

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.4.2. Настройки текстовых редакторов

Для перехода на панель «**Text Editors**» следует выбрать **General → Editors → Text Editors.**

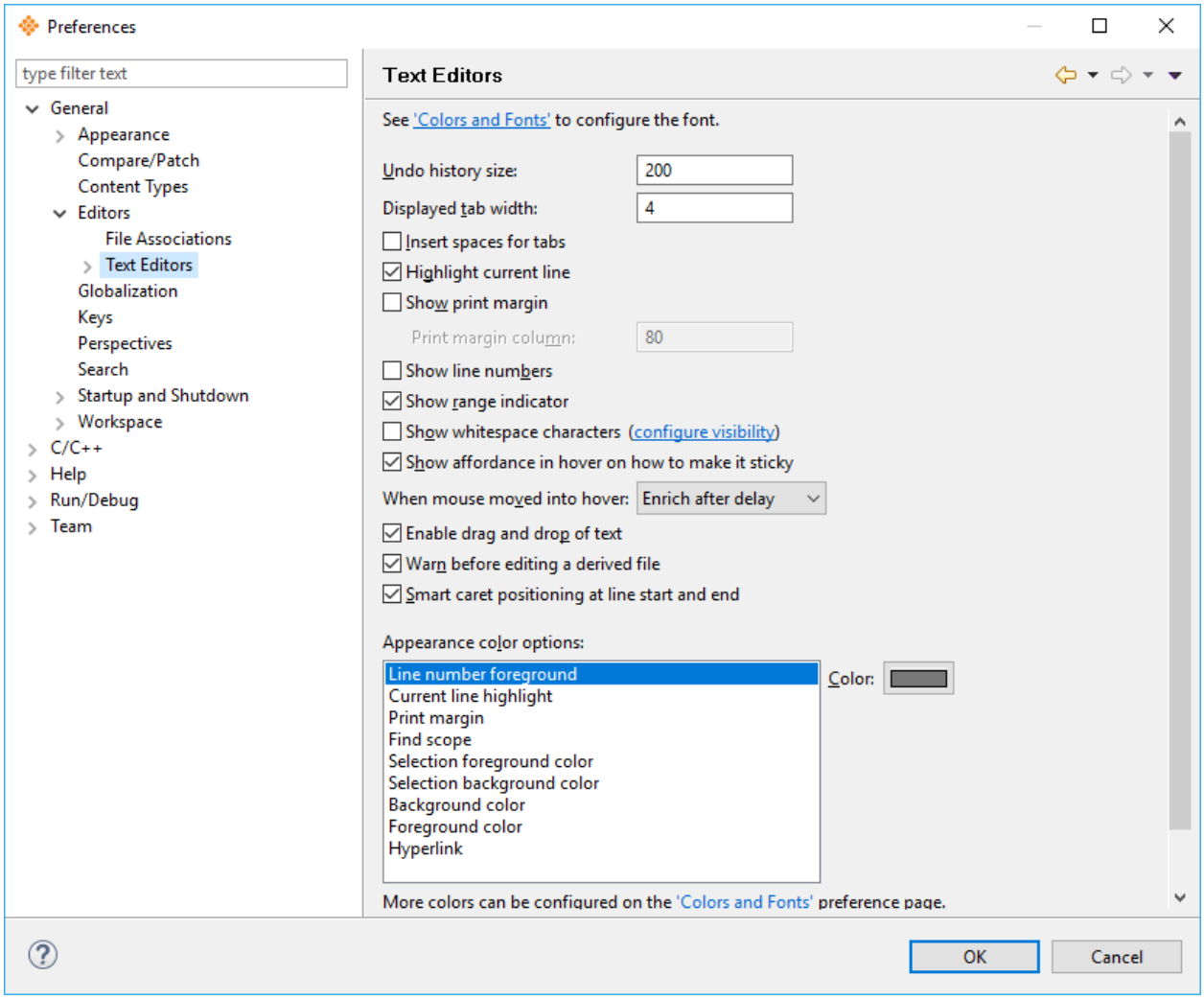


Рисунок 258. Панель «Text Editors»

Таблица 105. Элементы управления панели «Text Editors»

Название элемента	Описание функционирования	По умолчанию
Undo history size	Задать размер отменяемой истории для текстовых редакторов.	200
Displayed tab width	Задать ширину отображаемых закладок для текстовых редакторов.	4

Название элемента	Описание функционирования	По умолчанию
Insert spaces for tabs	Вставить символ пробела вместо символа табуляции.	Выкл
Highlight current line	Выделить цветом текущую строку.	Вкл
Show print margin	Установить видимость границ печатного поля.	Выкл
Print margin column	Задать границы колонок при печати.	80
Show line numbers	Показать/Скрыть номера строк в левой части редактора.	Вкл
Show range indicator	Показать/Скрыть индикатор промежутков в текстовом редакторе.	Вкл
Show whitespace characters	Показать/Скрыть скрытые знаки в текстовом редакторе.	Выкл
Show affordance in hover on how to make it sticky	Показать/Скрыть всплывающую подсказку о том, как закрепить действие.	Выкл
When mouse moved into hover	Выбрать один из сценариев поведения курсора мыши, когда он заходит на область всплывающей подсказки: ▪ **Close hover** – закрыть всплывающее окно, как только курсор на него наводится; ▪ **Enrich immediately** – открыть всплывающую подсказку сразу; ▪ **Enrich after delay** – открыть всплывающую подсказку с небольшой задержкой; ▪ **Enrich on click** – открыть всплывающую подсказку по щелчку мыши.	Enrich after delay
Enable drag and drop of text	Включить/Отключить возможность перетаскивания текста.	Вкл
Warn before editing a derived file	Выводить предупреждение при попытке редактировать исходный файл.	Вкл
Smart caret positioning at line start and end	Автоматически установить курсор в начале или в конце строки.	Вкл
Show affordance in hover on how to make it sticky	Показать/Скрыть возможность действия при наведении курсора и как закрепить ее.	Вкл
Appearance color options	Выбрать цвет элементов из списка.	–
Color	Выбрать цвет внешнего вида выбранной опции.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.4.2.1 Настройки редактора ассемблерных файлов

Данные настройки управляют отображением подсветки синтаксиса для ассемблерных файлов для 1967ВН0ХХ. В настоящее время поддерживается подсветка однострочных и многострочных линий инструкций. См. также п. 6.6.14.

Для перехода на панель «**Assembly** » следует выбрать **General → Editors → Text Editors → Assembly.**

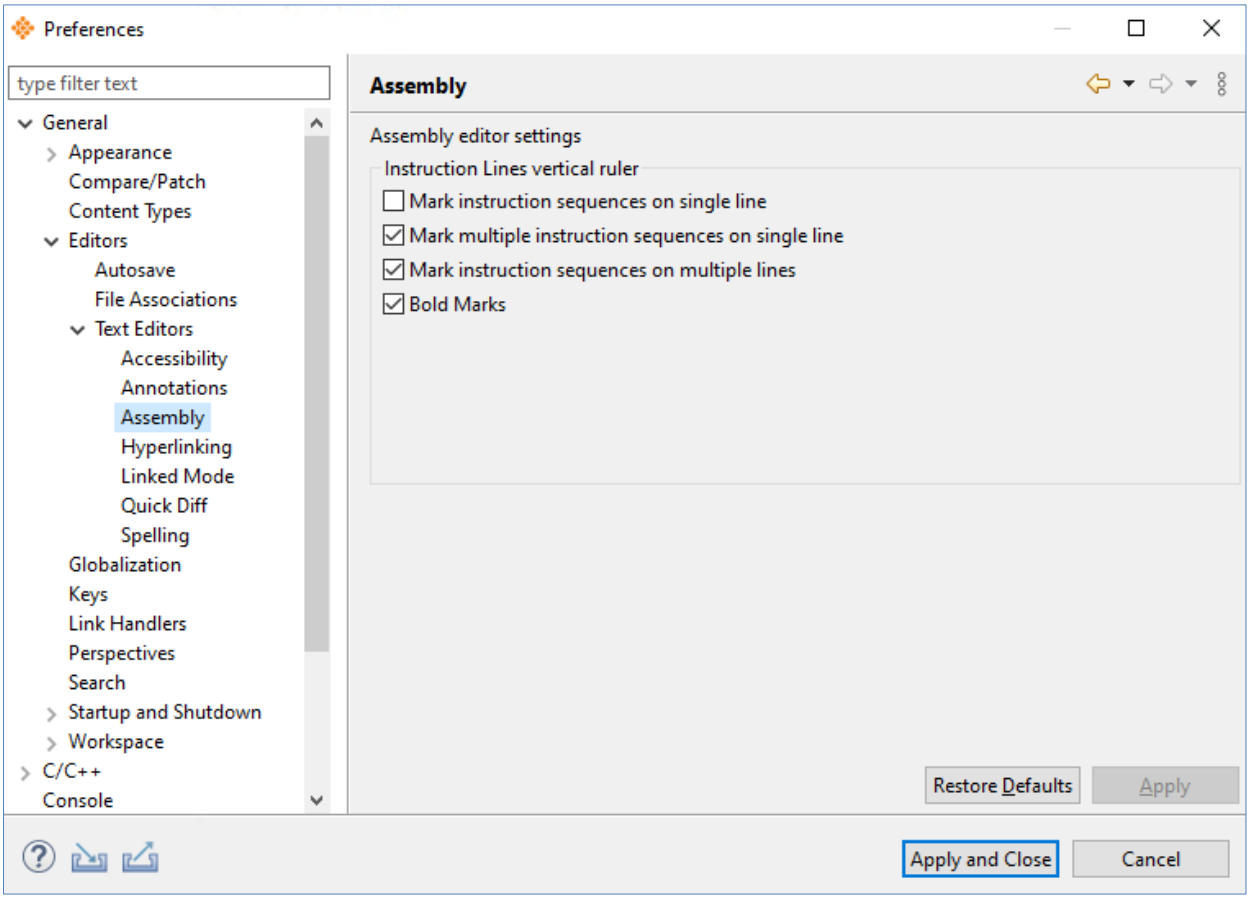


Рисунок 259. Панель «Assembly»

Таблица 106. Элементы управления панели «Assembly»

Название элемента	Описание функционирования	По умолчанию
Mark instruction sequences on single line	Подсветка однострочных линий инструкций.	Выкл
Mark multiple instruction sequences on single line	Подсветка нескольких последовательностей инструкций в одной строке.	Вкл
Mark instruction sequences on multiple lines	Подсветка многострочных линий инструкций.	Вкл
Bold Marks	Подсветка жирными метками	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.5. Настройка параметров ключей

Панель «**Keys**» предназначена для настройки сочетаний клавиш для различных команд. Для перехода на панель «**Keys**» следует выбрать **General → Keys**.

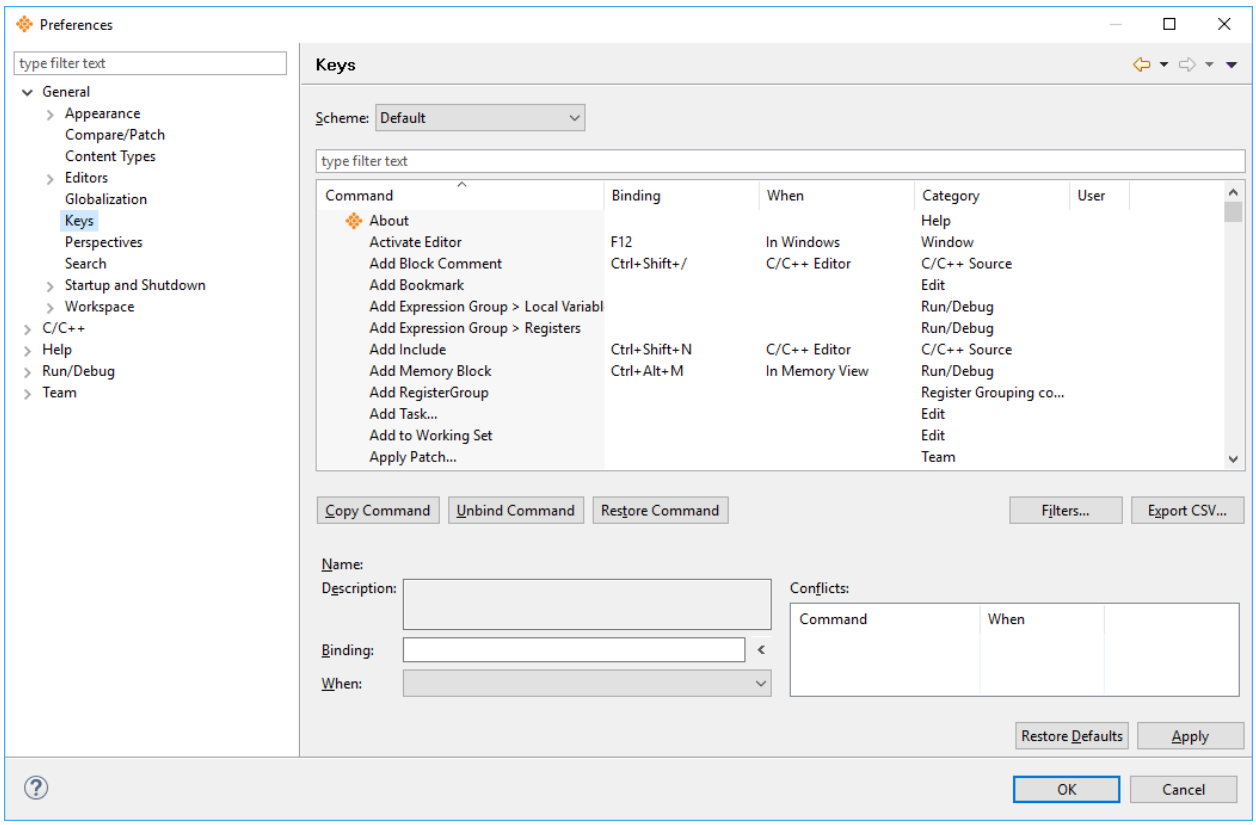


Рисунок 260. Панель «Keys»

6.12.1.6. Настройка режимов работы (перспектив)

Панель «**Perspectives**» предназначена для управления режимами работы, возможными в среде разработки.

Для перехода на панель «**Perspectives**» следует выбрать **General → Perspectives**.

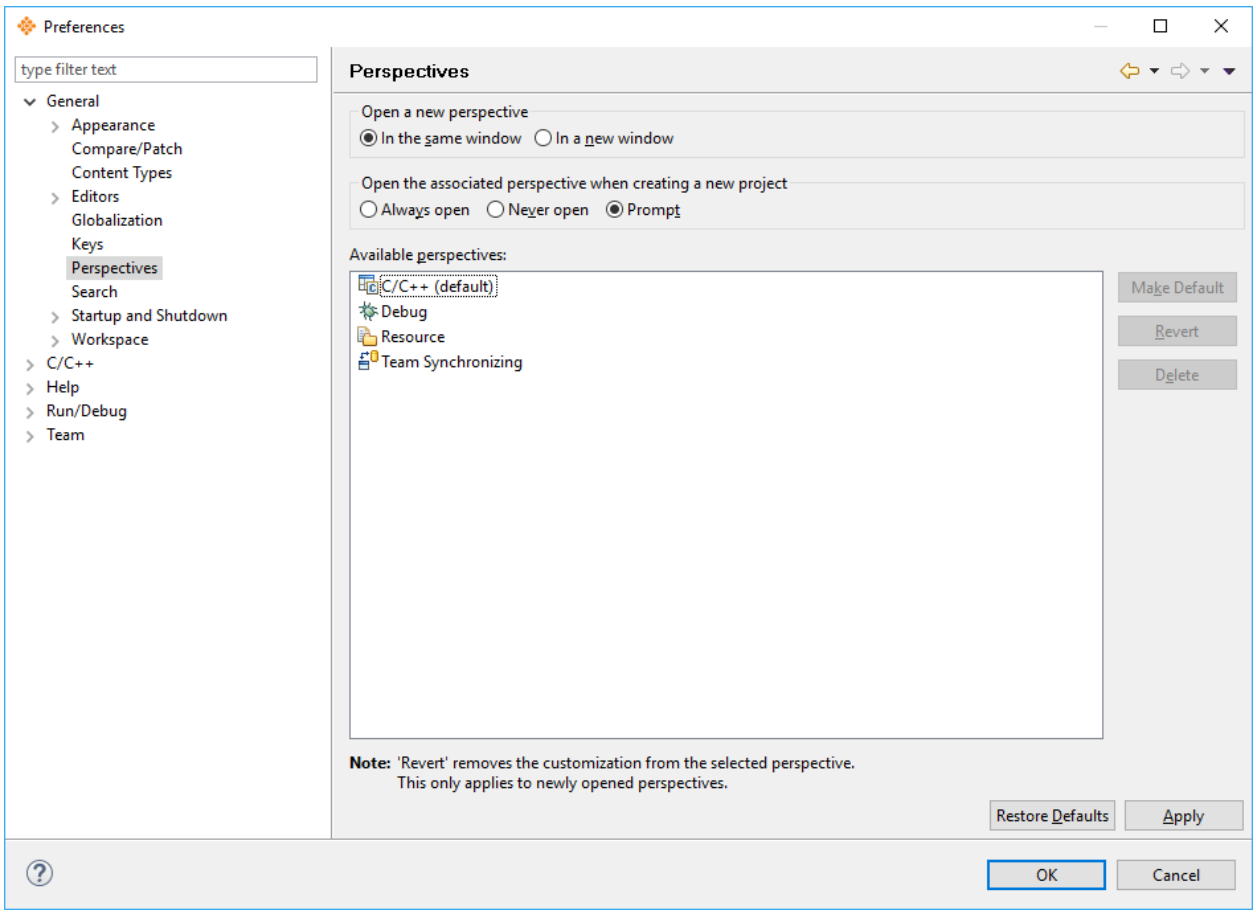


Рисунок 261. Панель «Perspectives»

Таблица 107. Элементы управления панели «Perspectives»

Название элемента	Описание функционирования	По умолчанию
Open a new perspective	Выбрать способ перехода к новому режиму работы: ▪ **In the same window** – в текущем окне; ▪ **In a new window** – открыть в новом окне.	В том же окне
Open the associated perspective when creating a new project	Открыть связанную перспективу при создании нового проекта: ▪ **Always open** – всегда открывать; ▪ **Never open** – никогда не открывать; ▪ **Prompt** – незамедлительно.	Незамедлительно
Available Perspectives	Список доступных режимов работы.	–
Make Default	Установить выбранные режимы работы в качестве режимов по умолчанию.	–
Revert	Перезагрузить выбранный режим работы до конфигурации по умолчанию. Применимо только ко встроенным режимам работы, которые были перезаписаны с использованием **Window → Save Perspective As**…	–
Delete	Удалить выбранные режимы работы. Применимо только к пользовательским режимам работы (встроенные режимы удалить нельзя).	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.7. Настройки параметров поиска

Настройки параметров поиска находятся на панели «Search». Для перехода на панель «**Search**» следует выбрать **General → Search**.

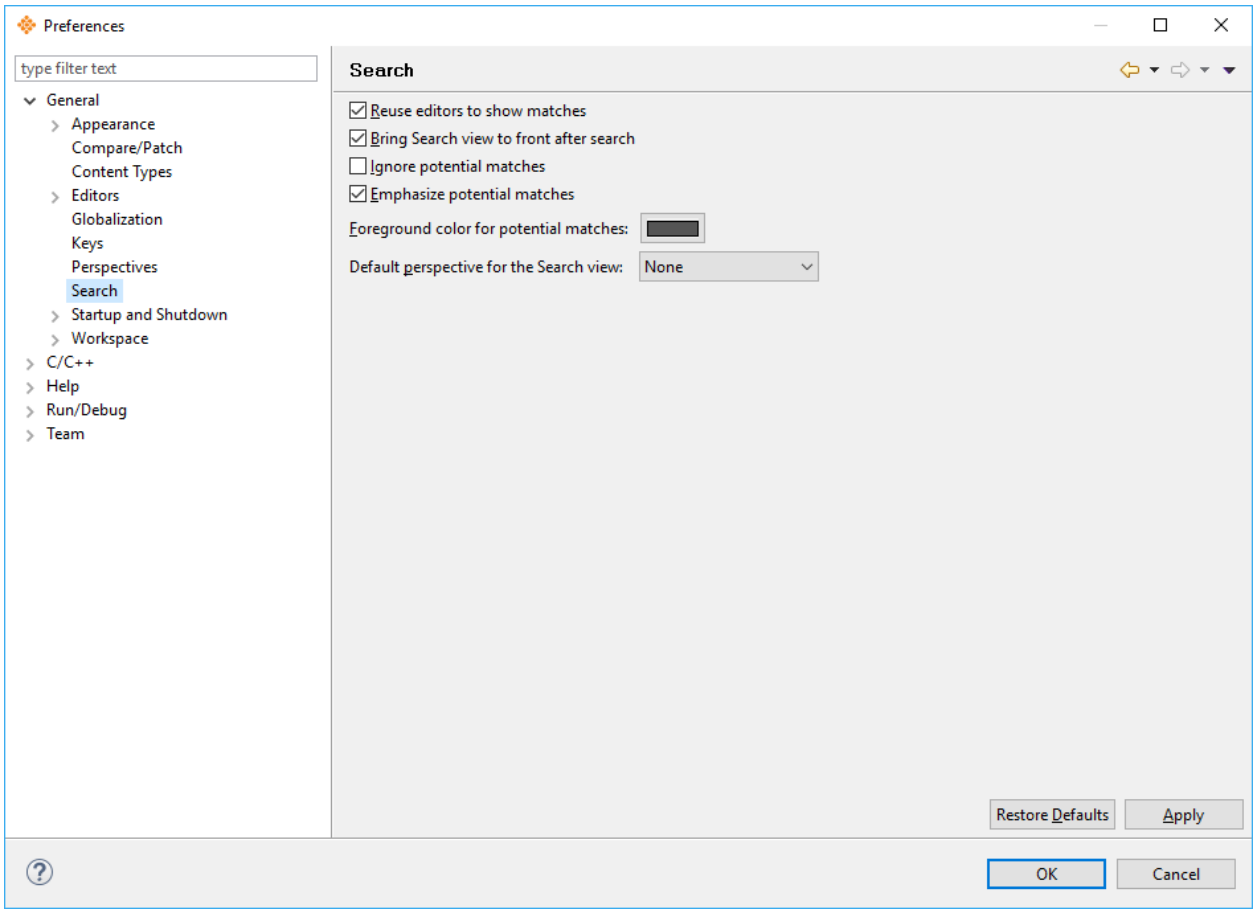


Рисунок 262. Панель «Search»

Таблица 108. Элементы управления панели «Search»

Название элемента	Описание функционирования	По умолчанию
Reuse editors to show matches	Использовать тот же редактор для поиска, чтобы сократить количество открытых редакторов.	Вкл
Bring Search view to front after search	Отобразить вид поиска поверх других окон после выполнения поиска.	Вкл
Ignore potential matches	Показывать точные совпадения.	Выкл
Emphasize potential matches	Показывать потенциальные совпадения во вкладке «**Search**». Если поисковая системы не на 100% уверена в совпадении, оно будет расценено как потенциальное.	Вкл
Foreground color for potential matches	Выбрать цвет заливки для потенциальных совпадений.	Чёрный

Название элемента	Описание функционирования	По умолчанию
Default perspective for the Search view	Выбрать режим работы, который будет выводиться на передний план при появлении новых результатов поиска.	Нет

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.8. Настройки запуска и завершения работы приложения

Панель «**Startup and Shutdown**» предназначена для выбора дополнительных программных модулей, которые будут запускаться автоматически при запуске среды разработки. Для перехода на панель «**Startup and Shutdown**» следует выбрать **General → Startup and Shutdown**.

Обычно дополнительные модули не запускаются, пока они не нужны, хотя некоторые могут запускаться во время запуска.

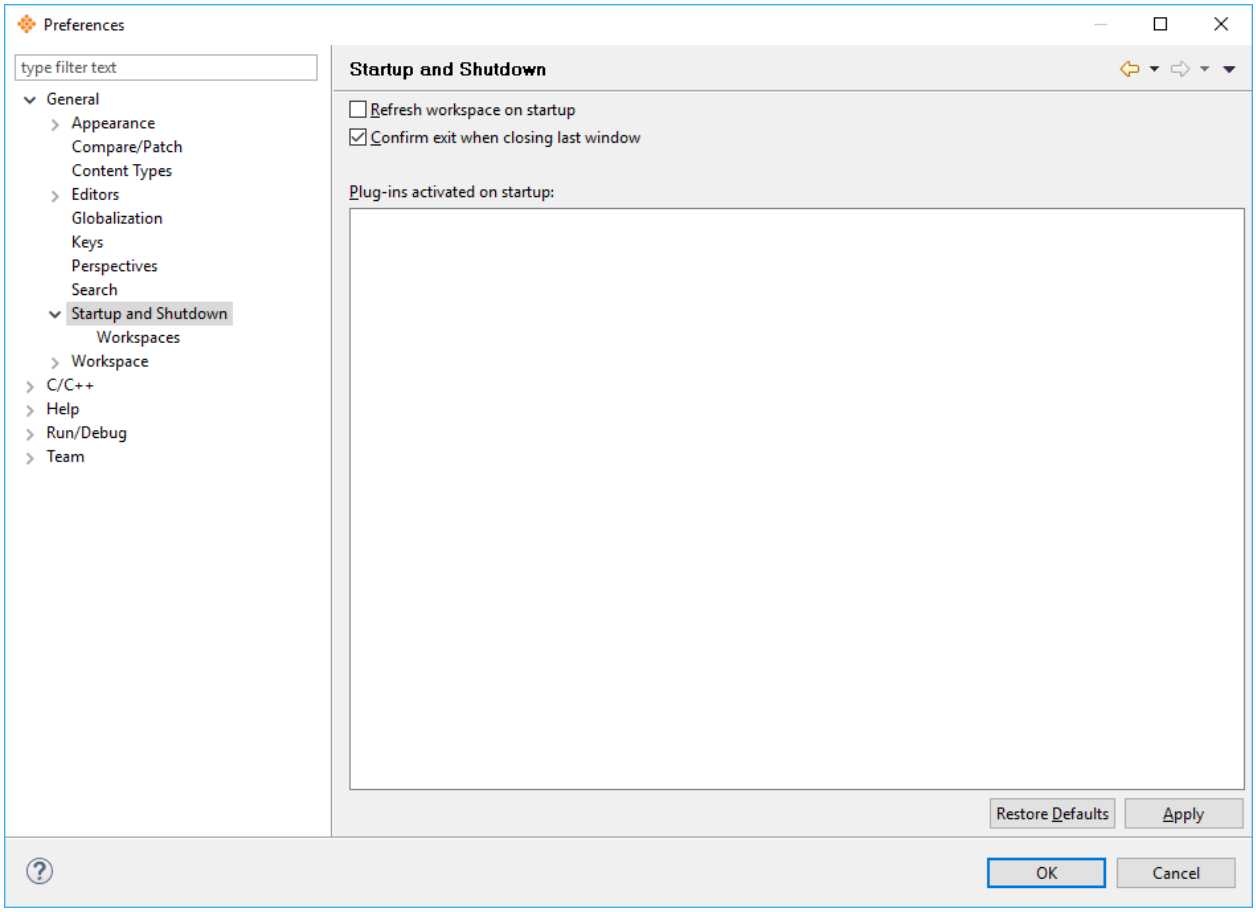


Рисунок 263. Панель «Startup and Shutdown»

Таблица 109. Элементы управления панели «Startup and Shutdown»

Название элемента	Описание функционирования	По умолчанию
Refresh workspace on startup	Синхронизировать при запуске содержимое среды с файловой системой.	Выкл

Название элемента	Описание функционирования	По умолчанию
Confirm exit when closing last window	Включить/Отключить запрос подтверждения выхода при закрытии последнего окна программы.	Вкл
Plug-ins activated on startup	Выбрать дополнительные программные модули из списка доступных для активации при запуске.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.8.1. Настройки подсказок в окне выбора рабочего пространства при запуске CM-LYNX

Панель «**Workspaces**» предназначена для настройки подсказок в окне выбора рабочего пространства при запуске CM-LYNX (см. Рисунок 36). Для перехода на панель «**Workspaces**» следует выбрать **General → Startup and Shutdown → Workspaces**.

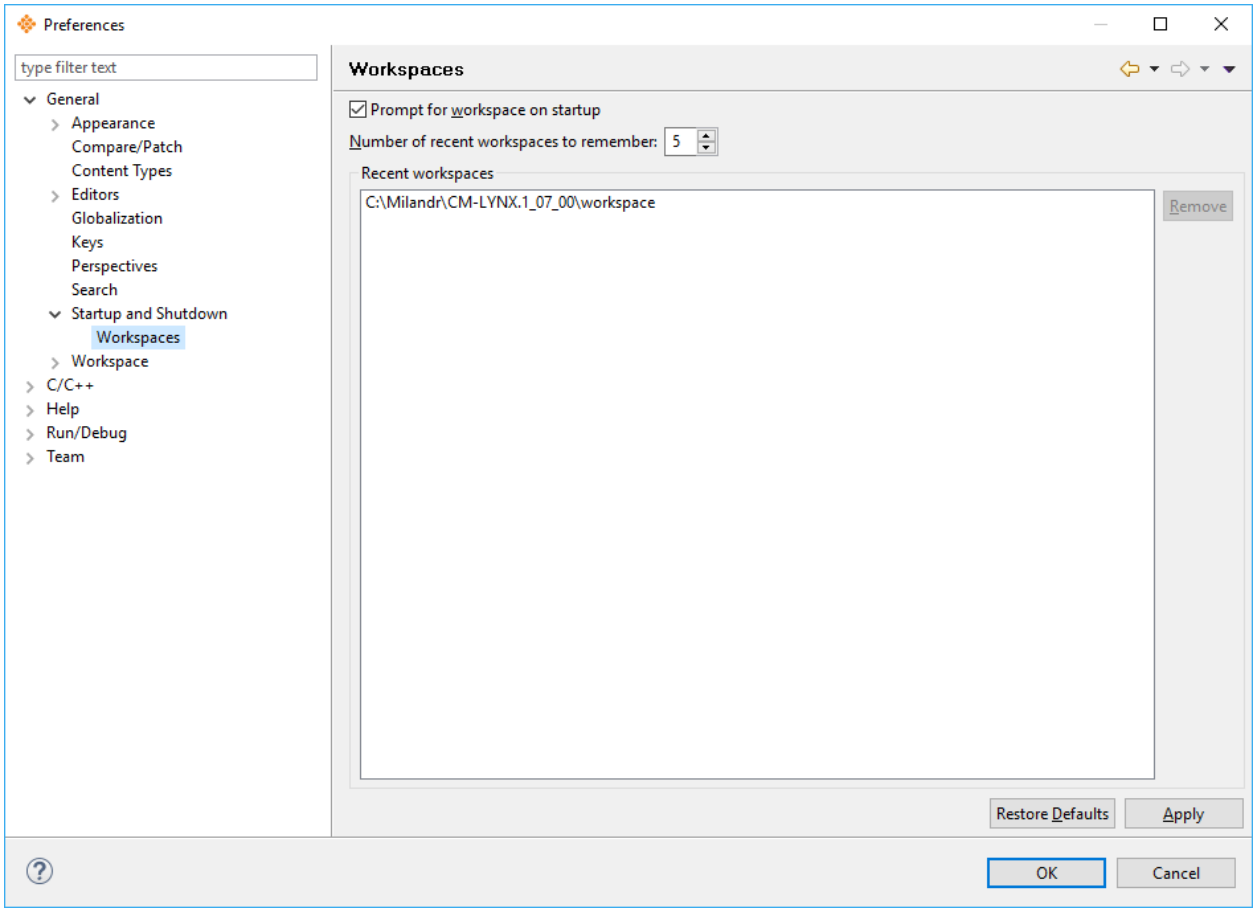


Рисунок 264. Панель «Workspaces»

Таблица 110. Элементы управления панели «Workspaces»

Название элемента	Описание функционирования	По умолчанию
Prompt for workspace on startup	Включить/отключить запрос рабочего пространства при каждом запуске.	Вкл
Number of recent workspaces to remember	Задать максимальное количество недавно использованных рабочих пространств, которые будут запоминаться и которые будут представлены в диалоговом окне «**Workspace Launcher**».	5

Название элемента	Описание функционирования	По умолчанию
Recent workspaces	Список недавно использованных рабочих пространств, представленных в диалоговом окне «**Workspace Launcher**».	–
Remove	Удалить выбранное рабочее пространство из списка недавно использованных.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.9. Настройки рабочего пространства

Панель «**Workspace**» (Рисунок 265) предназначена для указания настроек рабочего пространства. Для перехода на панель «**Workspace**» следует выбрать **General → Workspace**.

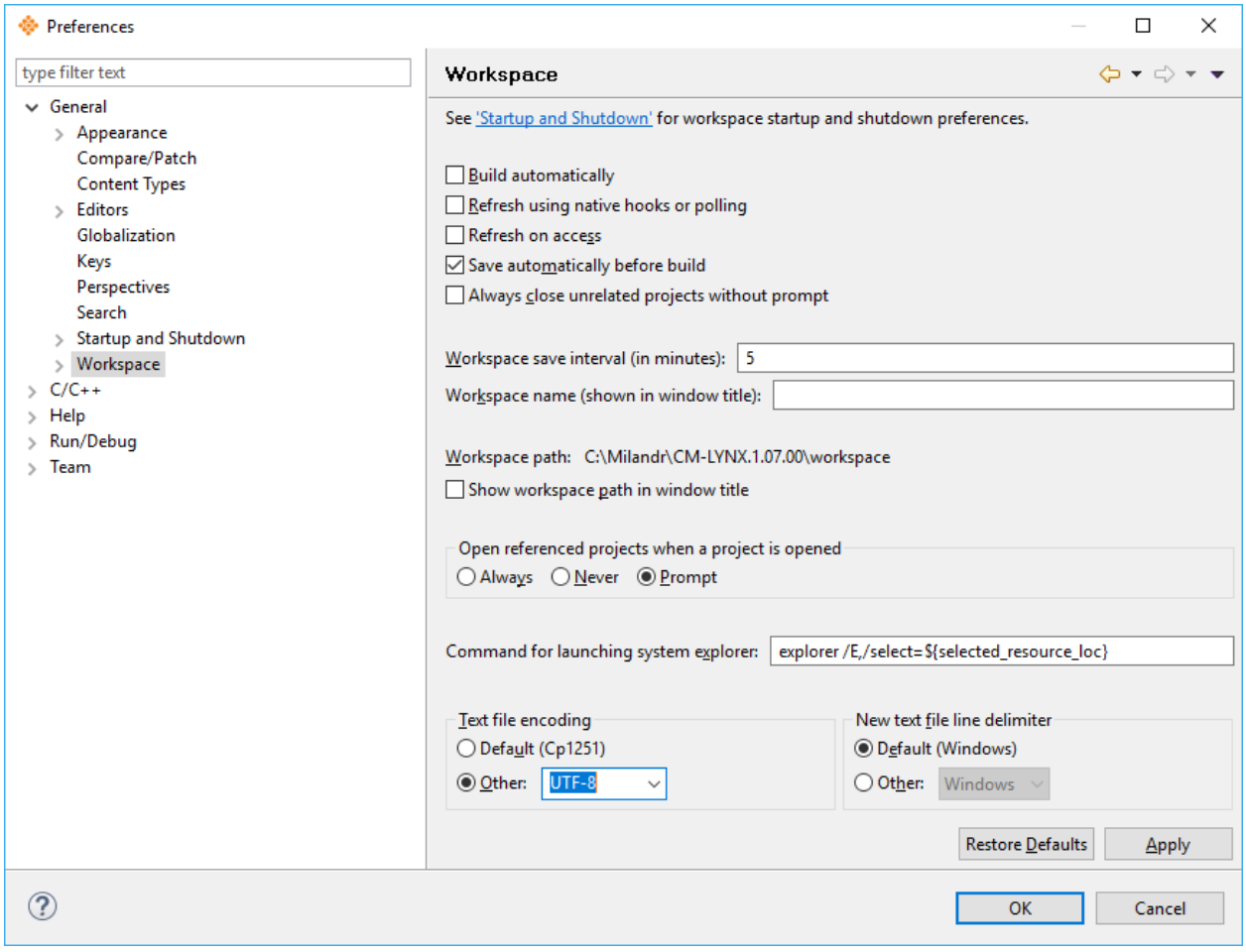


Рисунок 265. Панель «Workspace»

Таблица 111. Элементы управления панели «Workspace»

Название элемента	Описание функционирования	По умолчанию
Build automatically	Выполнить автоматическую сборку вне зависимости от того, был ли сохранен измененный ресурс.	Вкл

Название элемента	Описание функционирования	По умолчанию
Refresh using native hooks or polling	Выполнить автоматическую синхронизацию ресурсов рабочего пространства с соответствующими ресурсами в файловой системе с использованием собственного инструмента обновления (в Windows) или механизма опроса. Примечание: процесс возможно может занимать длительное время в зависимости от количества ресурсов рабочего пространства.	Выкл
Refresh on access	Выполнить автоматическое асинхронное обновление файлов, обнаруженных вне синхронизации рабочего пространства, например, при доступе к содержимому файла редактором.	Выкл
Save automatically before build	Выполнить автоматическое сохранение всех ресурсов, которые были изменены со времени последней сборки, при выполнении ручной сборки рабочего пространства.	Выкл
Always close unrelated projects without prompt	Закрыть несвязанные проекты без запроса.	Выкл
Workspace save interval (in minutes)	Указать периодичность автоматического сохранения состояния рабочего пространства на диск (в минутах).	5
Workspace name (shown in window title)	Включить/Отключить добавление в текстовом поле информации к названию каждого окна среды вместе со знаком «–».	–
Show workspace path in window title	Включить/Отключить добавление пути к рабочему пространству в названии окна.	Выкл
Open referenced projects when a project is opened	Выбрать действие при открытии проекта: ▪ **Always** – при открытии проекта всегда открываются и закрытые проекты, с которыми он связан; ▪ **Never** – при открытии проекта никогда не открываются закрытые проекты, с которыми он связан; ▪ **Prompt** – незамедлительно.	Незамедлительно
Command for launching system explorer	Задать команду для запуска проводника системы.	–
Text file encoding	Выбрать кодировку при сохранении текстовых файлов в редакторе. Примечание: начиная с версии CM-LYNX 1.07.00, кодировка по умолчанию изменена на UTF-8.	По умолчанию (UTF-8)
New text file line delimiter	Выбрать ограничитель строки для новых текстовых файлов. Примечание: не влияет на ограничение строк в текущих файлах.	По умолчанию (Windows)

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.9.1. Настройки очереди сборки

Панель «**Build Order**» предназначена для определения порядка сборки в проектах. Для перехода на панель «**Build Order**» следует выбрать **General → Workspace → Build Order.**

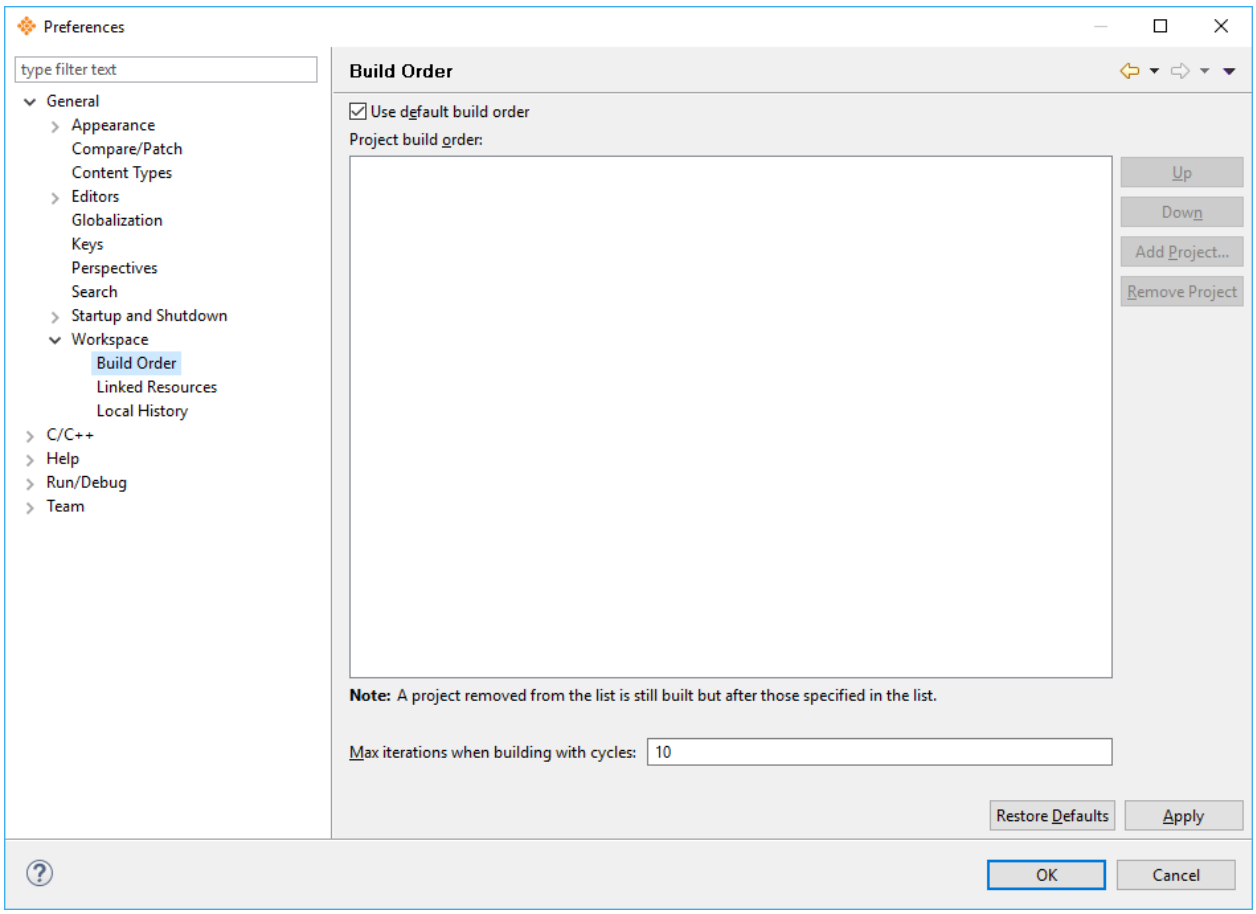


Рисунок 266. Панель «Build Order»

Таблица 112. Элементы управления панели «Build Order»

Название элемента	Описание функционирования	По умолчанию
Use default builder order	Определить порядок сборки по умолчанию. Выключение опции (флажок снят) позволяет получить доступ к списку проектов, порядком сборки которых можно управлять.	Вкл
Project build order	Список проектов, порядком которых можно управлять, используя кнопки справа. ▪ **Up, Down** – перемещают выделенный проект вверх/вниз по списку; ▪ **Add Project..., Remove Project** – добавляет/ удаляет проекты. **Примечание:** проекты, удаленные из списка порядка сборки, будут собраны после всех проектов в списке.	–
Max iterations when building with cycles	Указать максимальное количество повторений при сборке, содержащей циклы. Желательно избегать циклов между проектами. Проекты с циклами действительно логично принадлежат к одному проекту, и, таким образом, они должны быть сведены в один проект, если это возможно. Если циклы необходимы, возможно для правильного выполнения понадобится несколько повторений сборки.	10

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.9.2. Настройки параметров связанных ресурсов

Панель «**Linked Resources**» предназначена для работы со связанными ресурсами. Для перехода на панель «**Linked Resources**» следует выбрать **General → Workspace → Linked Resources**.

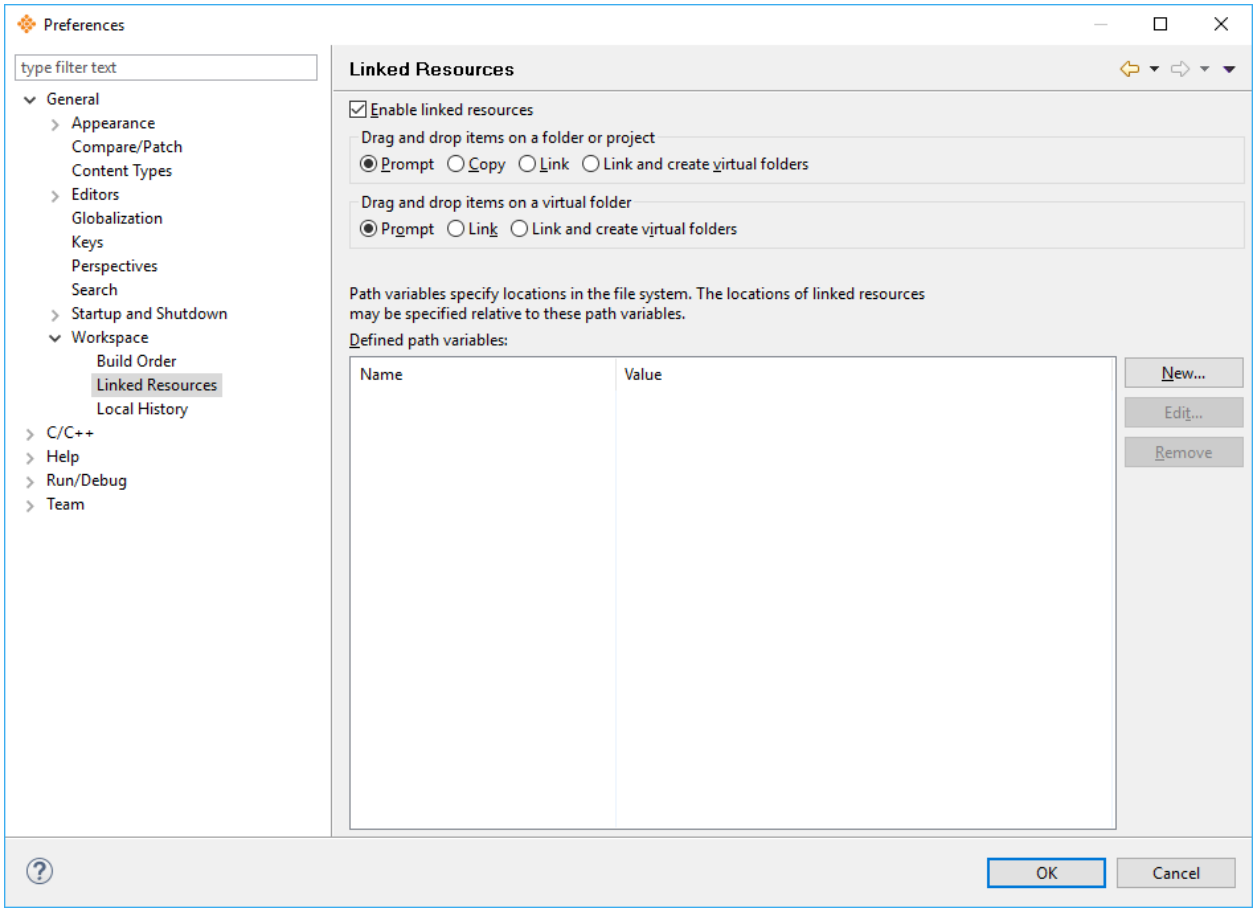


Рисунок 267. Панель «Linked Resources»

Таблица 113. Элементы управления панели «Linked Resources»

Название элемента	Описание функционирования	По умолчанию
Enable linked resources	Включить/Выключить глобальное использование связанных ресурсов для всего рабочего пространства. Если опцию отключить, то пользователь не сможет создать новые связанные ресурсы или импортировать существующие проекты, в которых содержатся связанные ресурсы.	Вкл
Drag and drop items on a folder or project	Выбрать порядок перетаскивания объектов связанных папок или проектов. ▪ **Prompt** – сразу; ▪ **Copy** – перемещение объектов путём копирования; ▪ **Link** – перемещение объектов по ссылке; ▪ **Link and create virtual folders** – перемещение объектов по ссылке и создание виртуальных папок.	Сразу

Название элемента	Описание функционирования	По умолчанию
Drag and drop items on a virtual folder	Выбрать порядок перетаскивания объектов в виртуальные папки. ▪ **Prompt** – сразу; ▪ **Copy** – перемещение объектов путём копирования; ▪ **Link** – перемещение объектов по ссылке; ▪ **Link and create virtual folders** – перемещение объектов по ссылке и создание виртуальных папок.	Сразу

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.1.9.3. Настройки локальной истории

Панель «**Local History**» предназначена для сохранения локальной история файла при его создании или изменении. Для перехода на панель «**Local History**» следует выбрать **General → Workspace → Local History**.

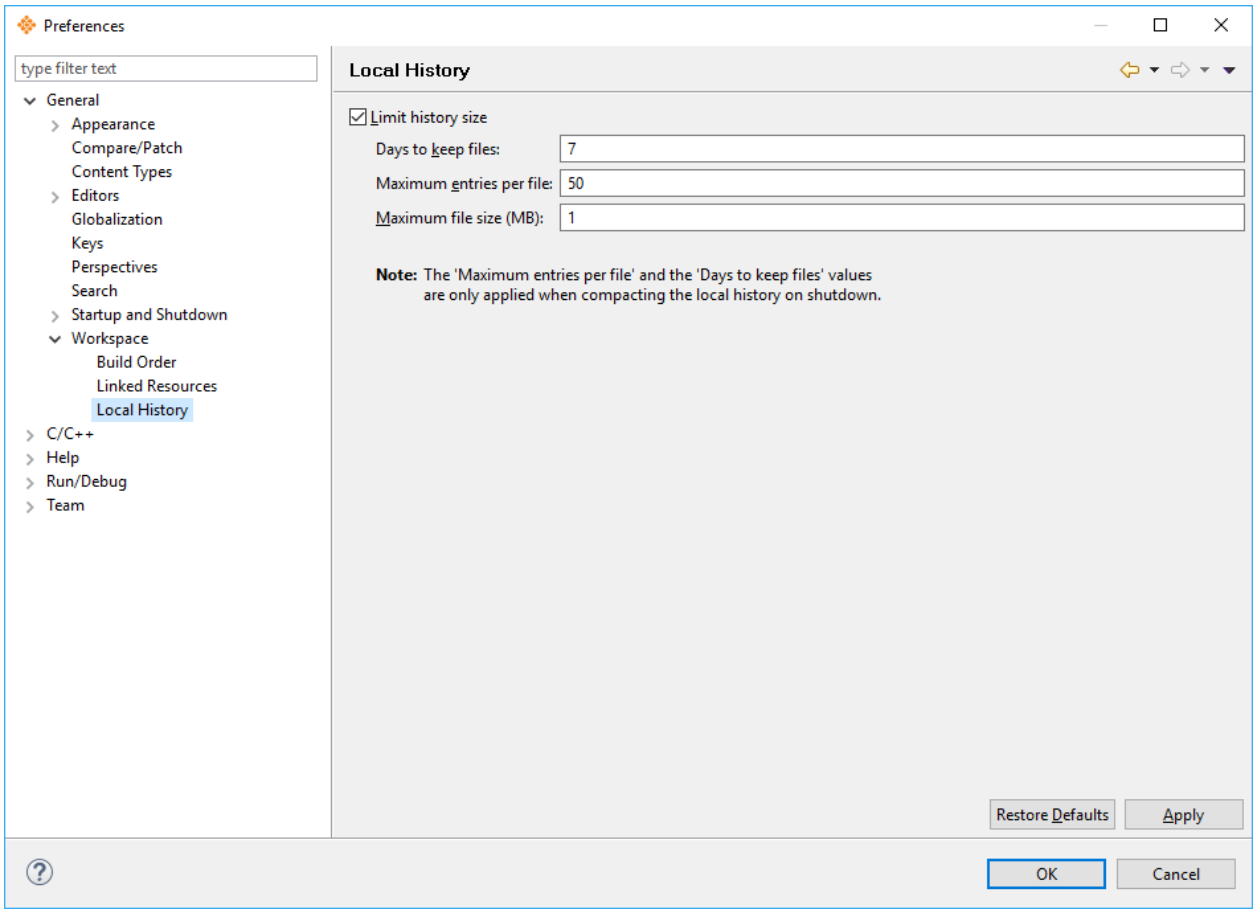


Рисунок 268. Панель «Local History»

Таблица 114. Элементы управления панели «Local History»

Название элемента	Описание функционирования	По умолчанию
Limit history size	Ограничить размер локальной истории на диске.	Вкл

Название элемента	Описание функционирования	По умолчанию
Days to keep files	Указать количество дней для хранения изменений в локальной истории. История более поздних изменений будет утеряна.	7 дней
Maximum entries per file	Указать количество исторических состояний для файла, которые необходимо хранить в локальной истории. При превышении этого значения, старая история будет удалена, чтобы освободить место для новой.	50 записей
Maximum file size (MB)	Указать максимальный размер отдельного состояния, занимаемый в области хранения истории. Файлы, превышающие указанный размер, храниться не будут.	1 МБ

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2. Настройки режима работы C/C++

Панель «**C/C++**» предназначена для настроек режима работы (перспективы) C/C++. Для перехода на панель следует выбрать **C/C++**.

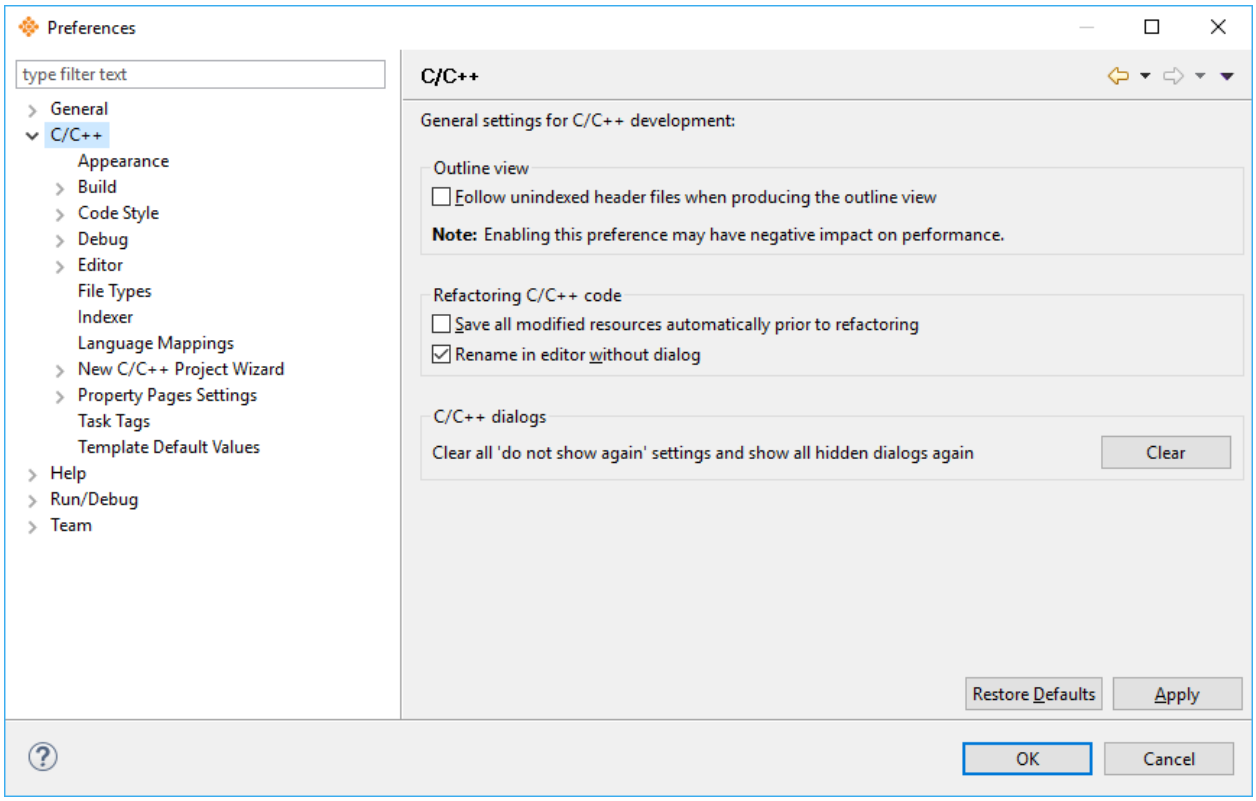


Рисунок 269. Панель «C/C++»

Таблица 115. Элементы управления панели «C/C++»

Название элемента	Описание функционирования	По умолчанию
Follow unindexed header files when producing the outline view	Активировать более полный поиск путей заголовков для создания более полного представления на вкладке «**Outline**». **Примечание:** включение флага может негативно сказаться на скорости работы.	Выкл
Save all modified resources automatically prior to refactoring	Задать автоматическое сохранение всех изменённых ресурсов перед рефакторингом кода.	Выкл
Rename in editor without dialog	Переименовать элементы в редакторе без отдельного диалогового окна.	Вкл
Clear all 'do not show again' settings and show all hidden dialogs again	Сбросить все настройки уведомлений и отображение всех скрытых диалоговых окон.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.1. Настройки внешнего вида элементов C/C++

Панель «**Appearance**» предназначена для настройки внешнего вида элементов C/C++. Для перехода на панель «**Appearance**» следует выбрать **C/C++ → Appearance**.

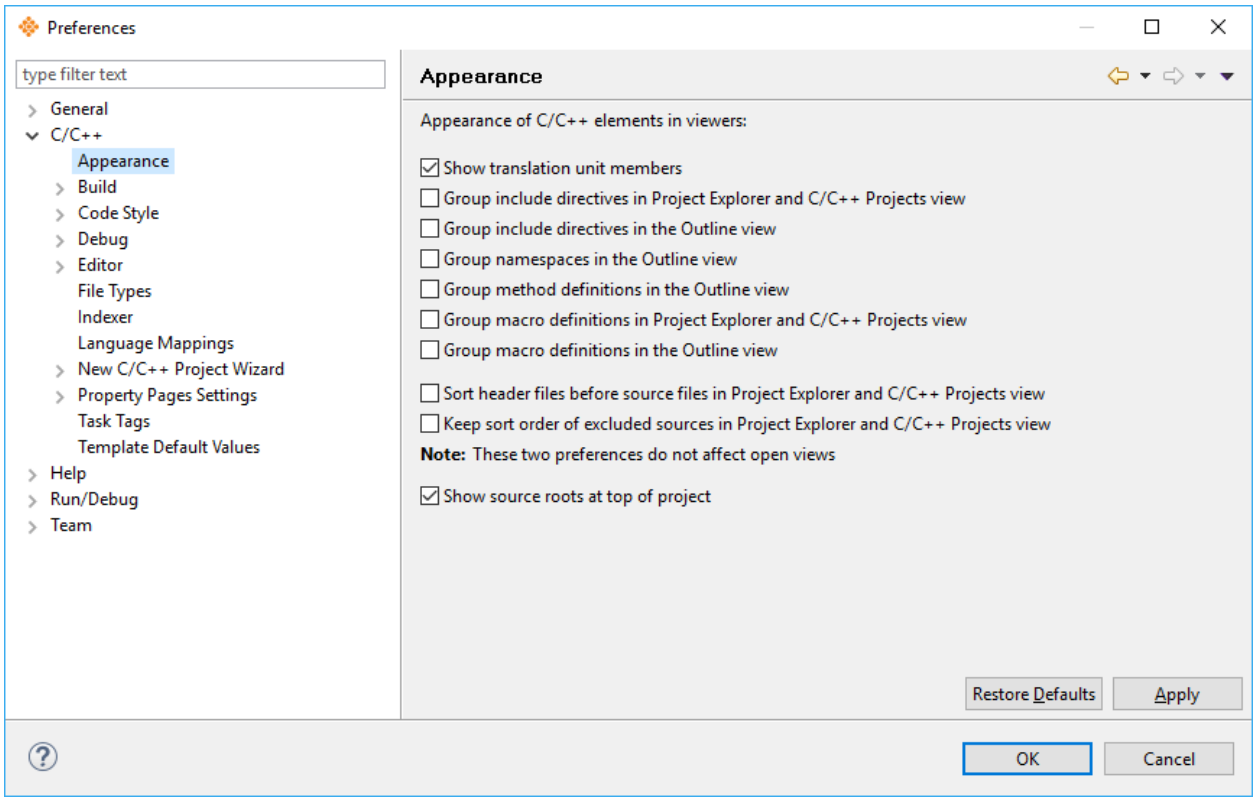


Рисунок 270. Панель «Appearance»

Таблица 116. Элементы управления панели «Appearance»

Название элемента	Описание функционирования	По умолчанию
Show translation unit members	Показать объекты приложения во вкладках «**Project Explorer**» и «**C/C++ Projects**».	Вкл
Group include directives in Project Explorer and C/C++ Projects view	Группировка директив `include` во вкладках «**Project Explorer**» и «**C/C++ Projects**».	Выкл
Group include directives in the Outline view	Группировка директив `include` на вкладке «**Outline**».	Выкл
Group namespaces in the Outline view	Группировка объявлений в пространстве имен на вкладке «**Outline**».	Выкл
Group method definitions in the Outline view	Группировка определений методов одного типа на вкладке «**Outline**».	Выкл
Group macro definitions in the Project Explorer and C/C++ Project views.	Группировка определений методов одного типа во вкладках «**Project Explorer**» и «**C/C++ Projects**».	Выкл
Group macro definitions in the Outline view	Группировка определений макрокоманды на вкладке «**Outline**».	Выкл
Sort header files before source files in Project Explorer and C/C++ Projects view	Разделение заголовочных и исходных файлов во вкладках «**Project Explorer**» и «**C/C++ Projects**».	Выкл
Keep sort order of excluded sources in Project Explorer and C/C++ Projects view	Сохранить порядок сортировки исключенных исходных файлов во вкладках «**Project Explorer**» и «**C/C++ Projects**».	Выкл
Show source roots at top of project	Группировка исходных корневых объектов сверху на вкладках «**Project Explorer**» и «**C/C++ Projects**».	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.2. Настройки сборки

Панель «**Build**» предназначена для адаптации настроек сборки для разработки C/C++. Для перехода на панель «**Build**» следует выбрать **C/C++ → Build**.

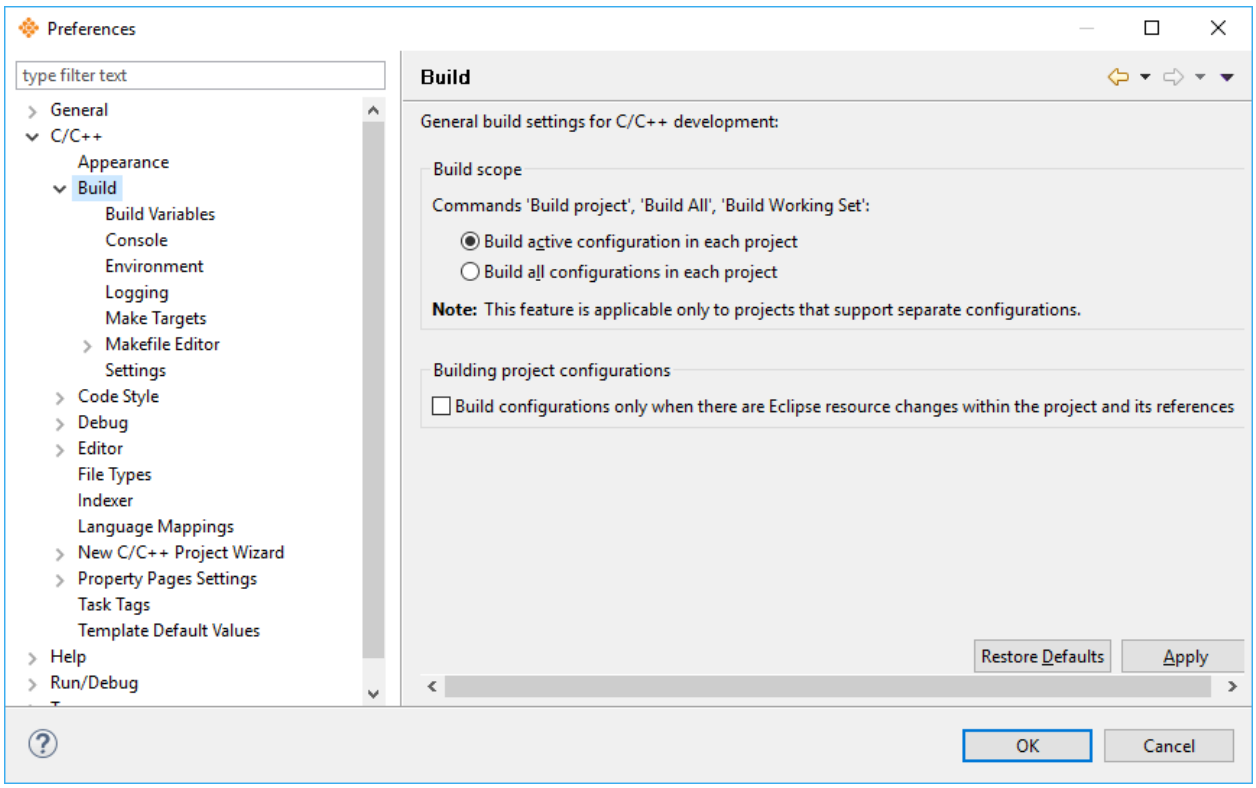


Рисунок 271. Панель «Build»

Таблица 117. Элементы управления панели «Build»

Название элемента	Описание функционирования	По умолчанию
Commands "Build project", "Build All", "Build Working Set"	Задать параметры команд сборки: ▪ **Build active configuration in each project** – сделать сборку только активной конфигурации в проекте; ▪ **Build all configurations in each project** – сделать сборку всех конфигураций в проекте.	Сборка только активной конфигурации
Build configurations only when there are Eclipse resource changes within the project and its resources	Выполнить сборку только в случае, когда в проекте и ресурсах произошли изменения.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.2.1. Настройки переменных сборки

Панель «**Build Variables**» предназначена для добавления, редактирования или удаления переменных сборки. Для перехода на панель **«Build Variables»** следует выбрать **C/C++ → Build → Variables**.

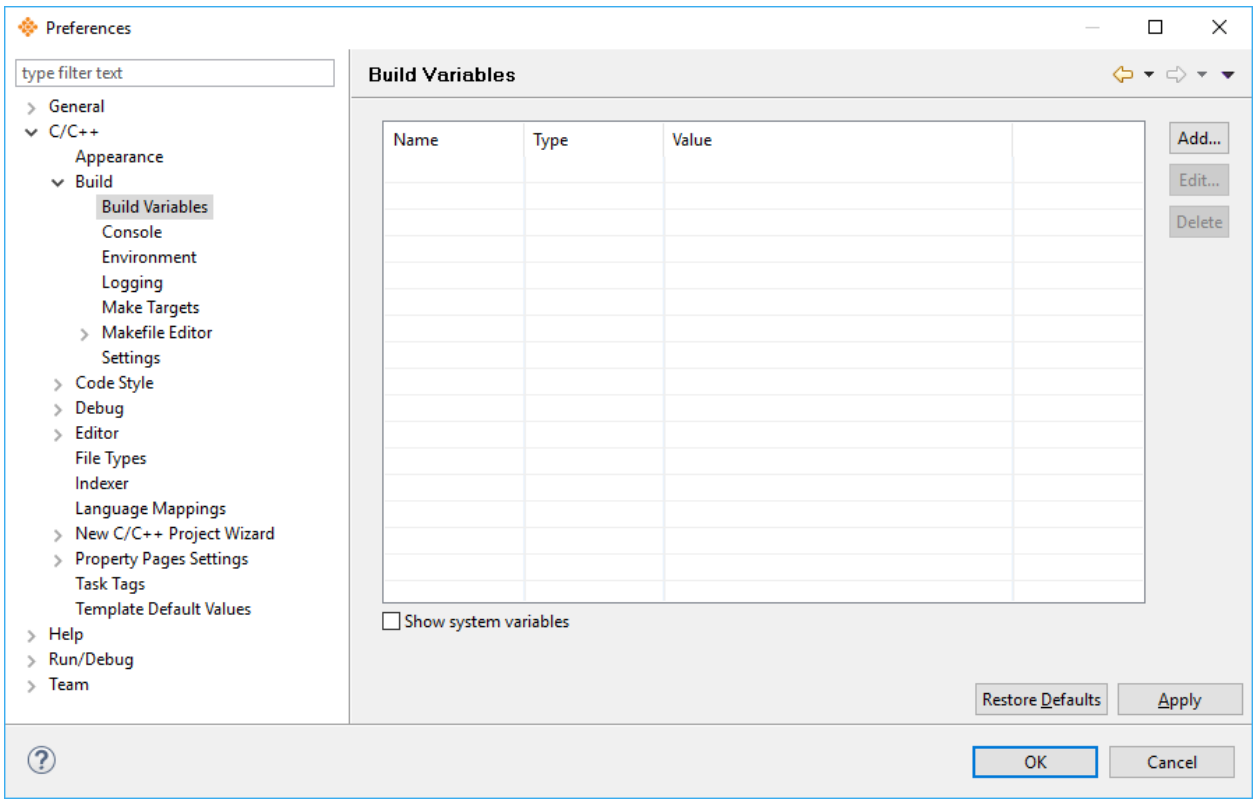


Рисунок 272. Панель «Build Variables»

Таблица 118. Элементы управления панели «Build Variables»

Название элемента	Описание функционирования	По умолчанию
Add...	Добавить новую переменную сборки в диалоговом окне «**Define a New Build Variable**».	–
Edit...	Перейти к редактированию выбранной переменной сборки.	–
Delete	Удалить выбранную переменную. Примечание: нельзя удалить системные переменные.	–
Show system variables	Показать/скрыть системные переменные в списке переменных.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

*6.12.2.2.2. Настройки внешнего вида сообщений на вкладке «**Console**»*

Панель «**Console**» предназначена для настройки внешнего вида сообщений на вкладке «**Console**». Для перехода на панель «**Console**» следует выбрать **C/C++ → Build → Console**.

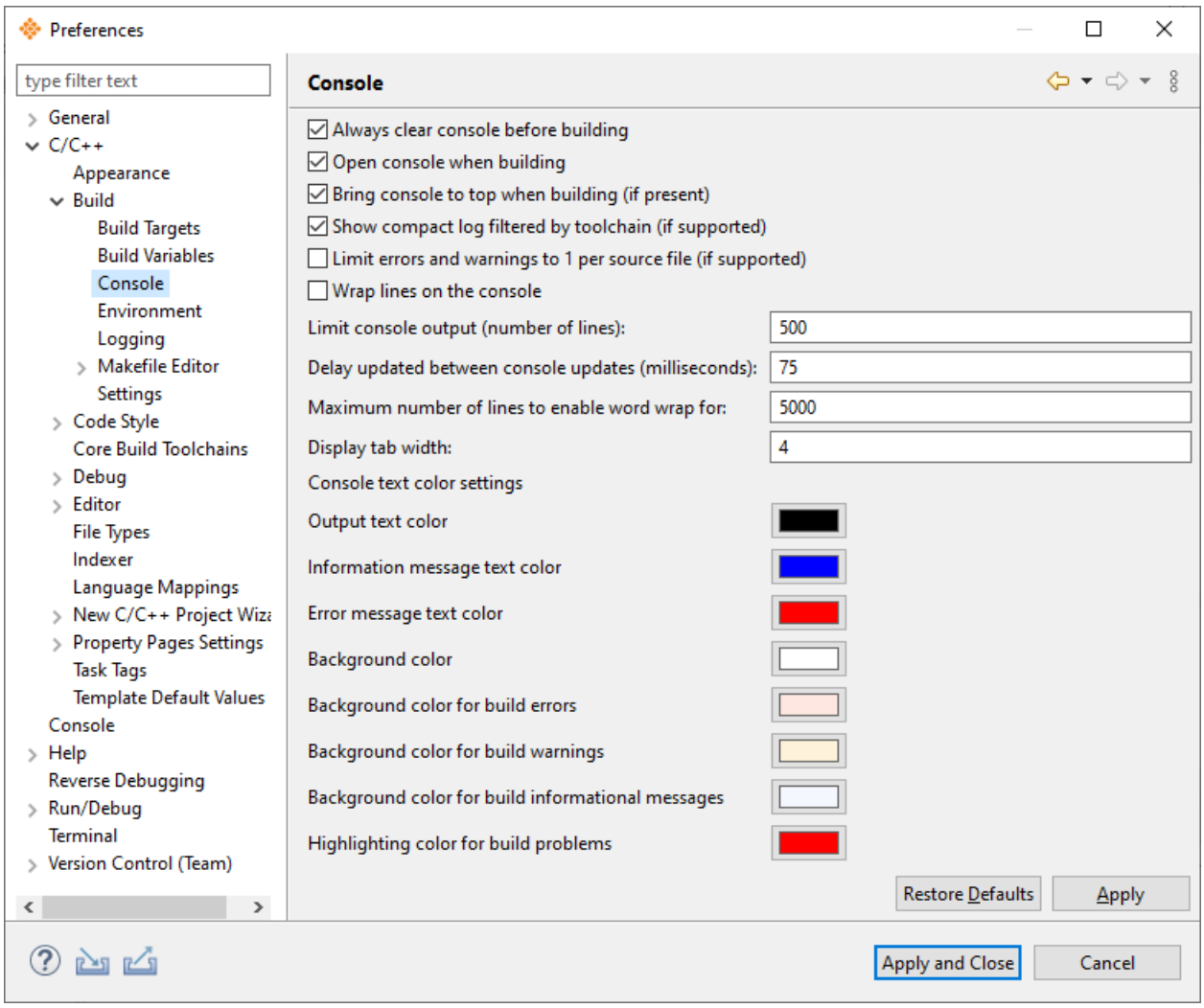


Рисунок 273. Панель «Console»

Таблица 119. Элементы управления панели «Console»

Название элемента	Описание функционирования	По умолчанию
Always clear console before building	Очищать содержимое вкладки «**Console**» каждый раз при выполнении сборки.	Вкл
Open console when building	Открыть вкладку «**Console**» при выполнении сборки.	Вкл
Bring console to top when building (if present)	Сделать вкладку «**Console**» активным окном при выполнении сборки.	Вкл
Show compact log filtered by toolchain (if supported)	Показывать компактный журнал сборки.	Выкл
Limit errors and warnings to 1 per source file (if supported)	Ограничивать в компактном журнале количество ошибок и предупреждений до 1 на файл.	Выкл
Wrap lines on the console	Переносить строки на вкладке «**Console**».	Выкл

Название элемента	Описание функционирования	По умолчанию
Limit console output (number of lines)	Указать максимальное количество строк, которые могут отображаться на вкладке «**Console**».	500
Display tab width	Указать количество пробелов для табуляции.	4
Console text color settings	Настройки цвета текста на вкладке «**Console**».	–
Output text color	Выбрать цвет текста на вкладке «**Console**».	–
Information message text color	Выбрать цвет информационных текстовых сообщений на вкладке «**Console**».	–
Error message text color	Выбрать цвет сообщений об ошибке на вкладке «**Console**».	–
Background color	Выбрать цвет фона на вкладке «**Console**».	–
Background color for build errors	Выбрать цвет фона ошибок при сборке на вкладке «**Console**».	–
Background color for build warnings	Выбрать цвет фона предупреждений при сборке на вкладке «**Console**».	–
Background color for build informational messages	Выбрать цвет фона информационных сообщений при сборке на вкладке «**Console**».	–
Highlighting color for build problems	Выбрать цвет выделения проблем при сборке на вкладке «**Console**».	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.2.3. Настройки среды для проектов рабочего пространства

Панель «**Environment**» предназначена для настройки среды для всех проектов рабочего пространства, а также позволяет управлять переменными среды, используемыми при сборке. Для перехода на панель «**Environment**» следует выбрать **C/C++ → Build → Environment**.

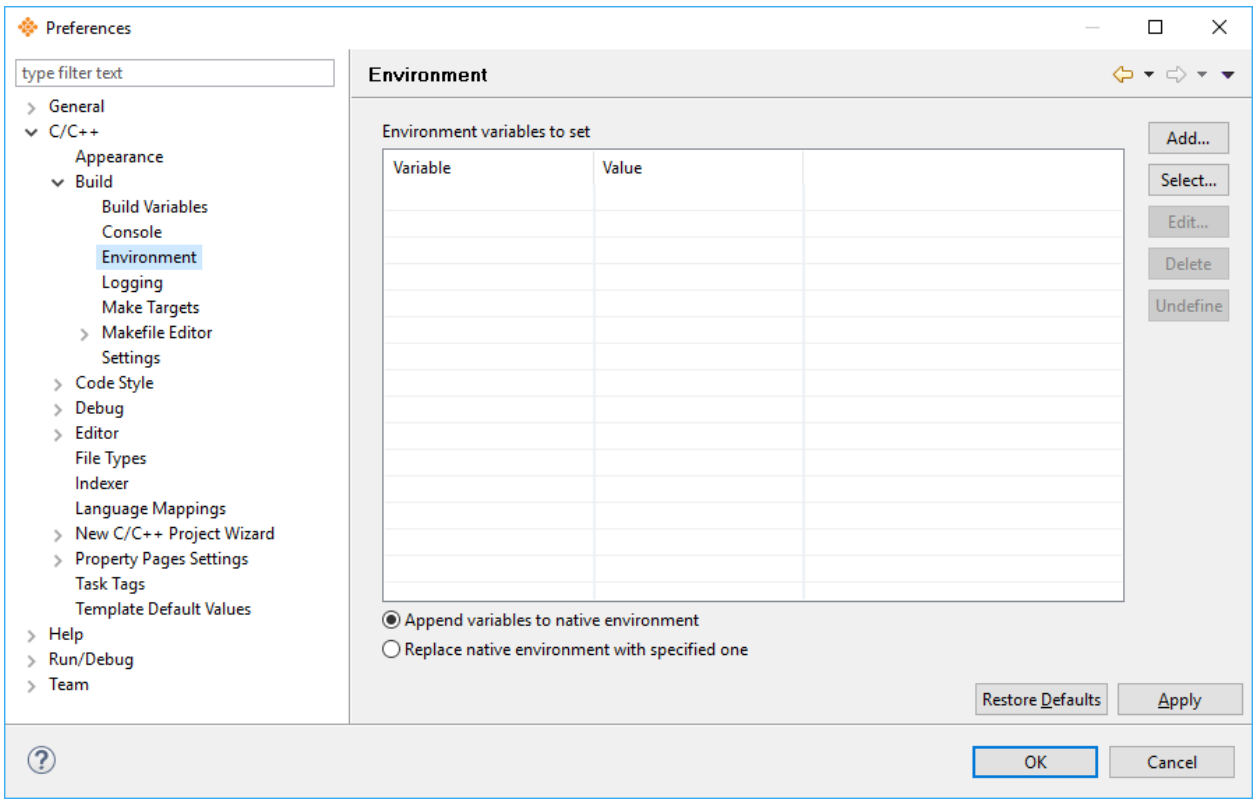


Рисунок 274. Панель «Environment»

Таблица 120. Элементы управления панели «Environment»

Название элемента	Описание функционирования	По умолчанию
Variable	Список имен переменных среды.	–
Value	Список значений соответствующих переменных среды.	–
Append variables to native environment	Добавить переменные к собственной среде при сборке.	Вкл
Replace native environment with specified one	Заменить собственную среду указанными переменными, а затем восстановить собственную среду после ее завершения.	Выкл
Add...	Открыть диалоговое окно «**New variable**» для создания переменной среды и ввода ее значения. Созданные переменные среды появляются в списке, выделенные жирным шрифтом.	–
Select...	Открыть диалоговое окно «**Select variables**» для выбора переменной из списка системных.	–
Edit...	Изменить имя и значение выбранной переменной среды.	–
Delete	Удалить выбранную переменную из списка.	–
Undefine	Удалить определение выбранной переменной. У некоторых переменных таких, как переменная **PATH**, не может быть удалено определение.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.2.4. Настройка логирования процесса сборки

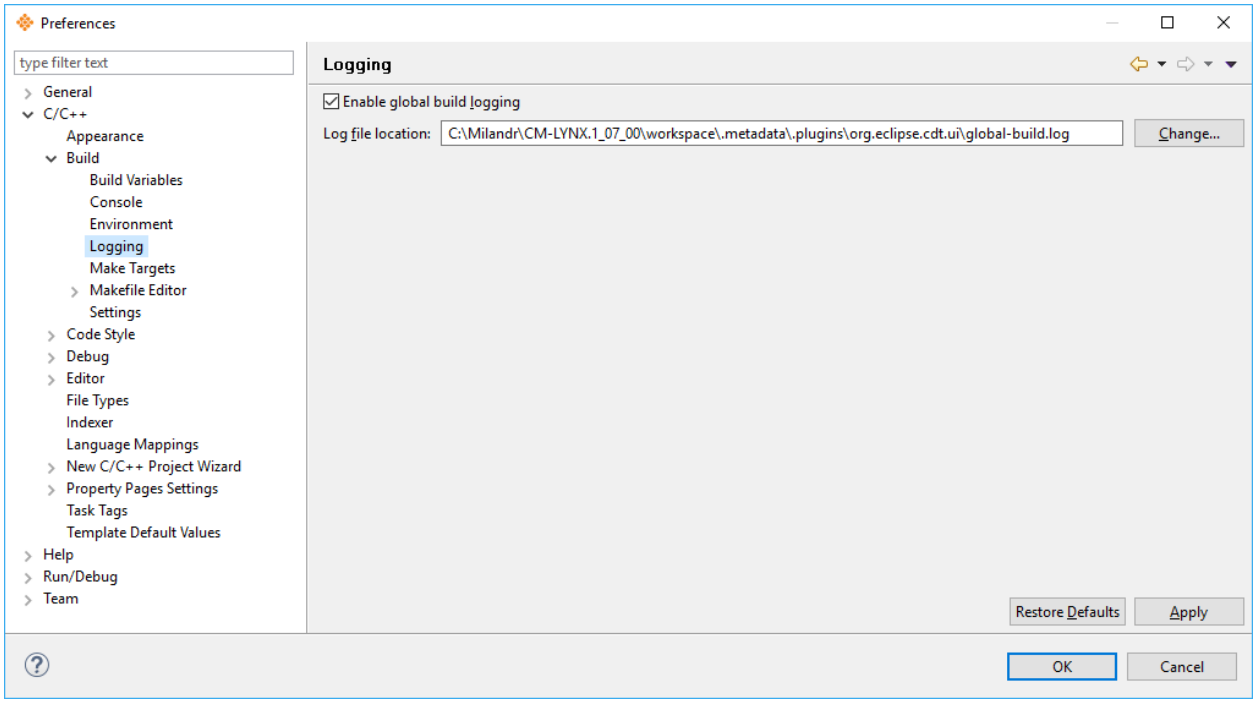


Рисунок 275. Панель «Logging»

Таблица 121. Элементы управления панели «Logging»

Название элемента	Описание функционирования	По умолчанию
Enable global build logging	Включить/отключить ведение журнала сборки.	–
Log file location	Указать путь к журналу сборки.	
Change...	Открыть диалоговое окно для выбора пути к файлу журнала сборки.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.2.5. Настройки опций компиляции

Панель «**Make Targets**» предназначена для настройки опций компиляции. Для перехода на панель «**Make Targets**» следует выбрать **C/C++ → Build → Make Targets**.

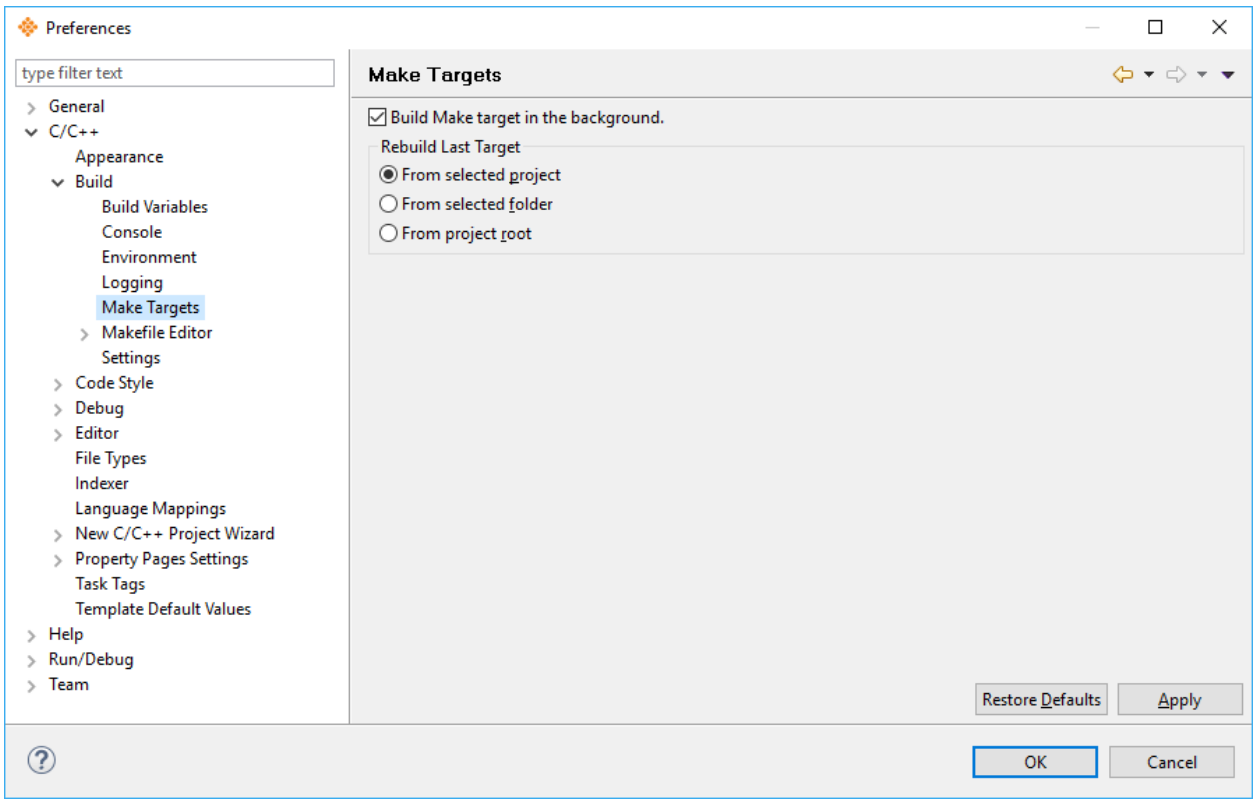


Рисунок 276. Панель «Make Targets»

Таблица 122. Элементы управления панели «Make Targets»

Название элемента	Описание функционирования	По умолчанию
Build Make target in the background	Выполнить сборку в фоновом режиме.	Вкл
Rebuild Last Target	Выбрать область действия команды «**Rebuild Last Target**» на вкладке «**Make Targets**»: ▪ **From selected project** – заново собирает ту же компиляцию, что и в прошлый раз в выбранном проекте; ▪ **From selected folder** – заново собирает ту же компиляцию, как при прошлом запуске в выбранной папке; ▪ **From project root** – заново собирает ту же компиляцию, как и при прошлом запуске, из корневой папки.	Сборка компиляции в выбранном проекте

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.2.6. Настройки синтаксиса и сворачивания в редакторе Makefile Editor

Панель «**Makefile Editor**» предназначена для установки параметров синтаксиса и сворачивания в редакторе Makefile Editor. Для перехода на панель «**Makefile Editor**» следует выбрать **C/C++ → Build → Makefile Editor**.

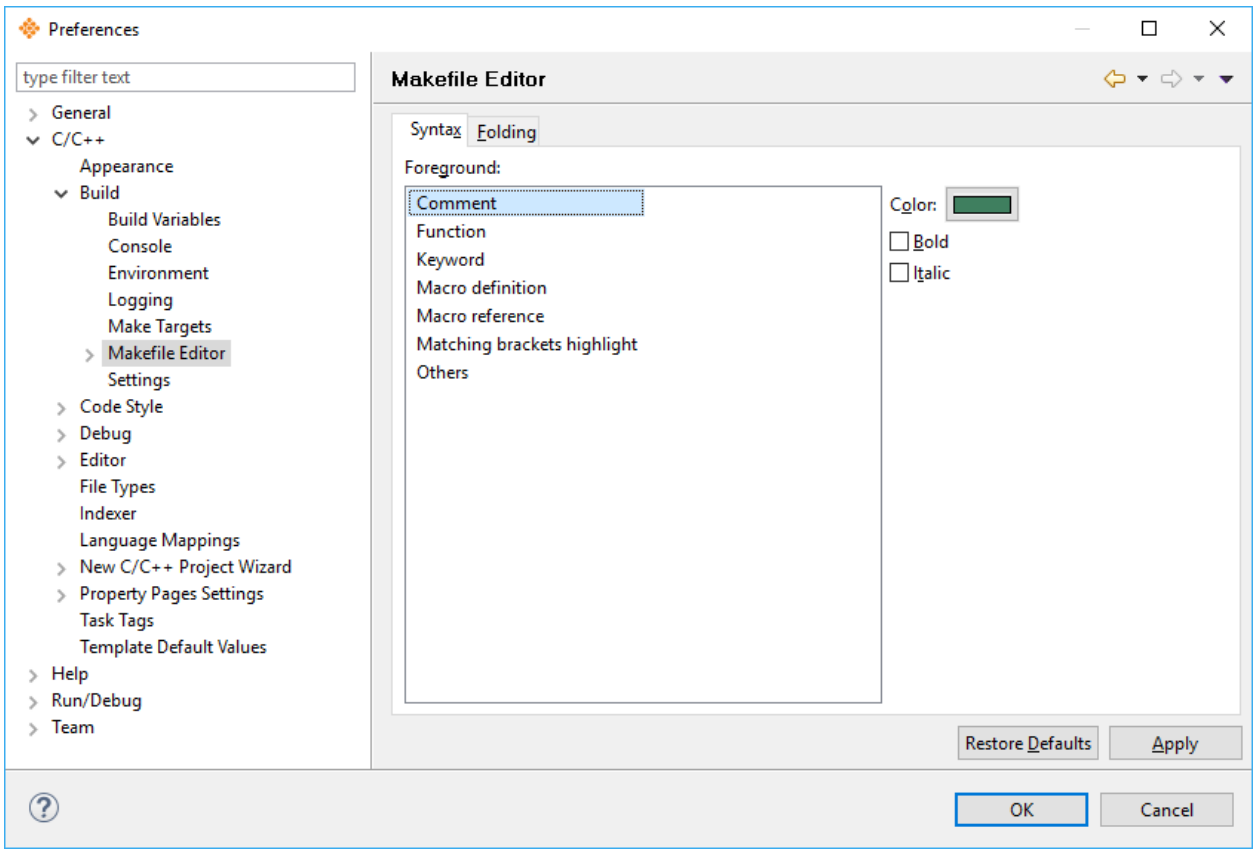


Рисунок 277. Панель «Makefile Editor», вкладка «Syntax»

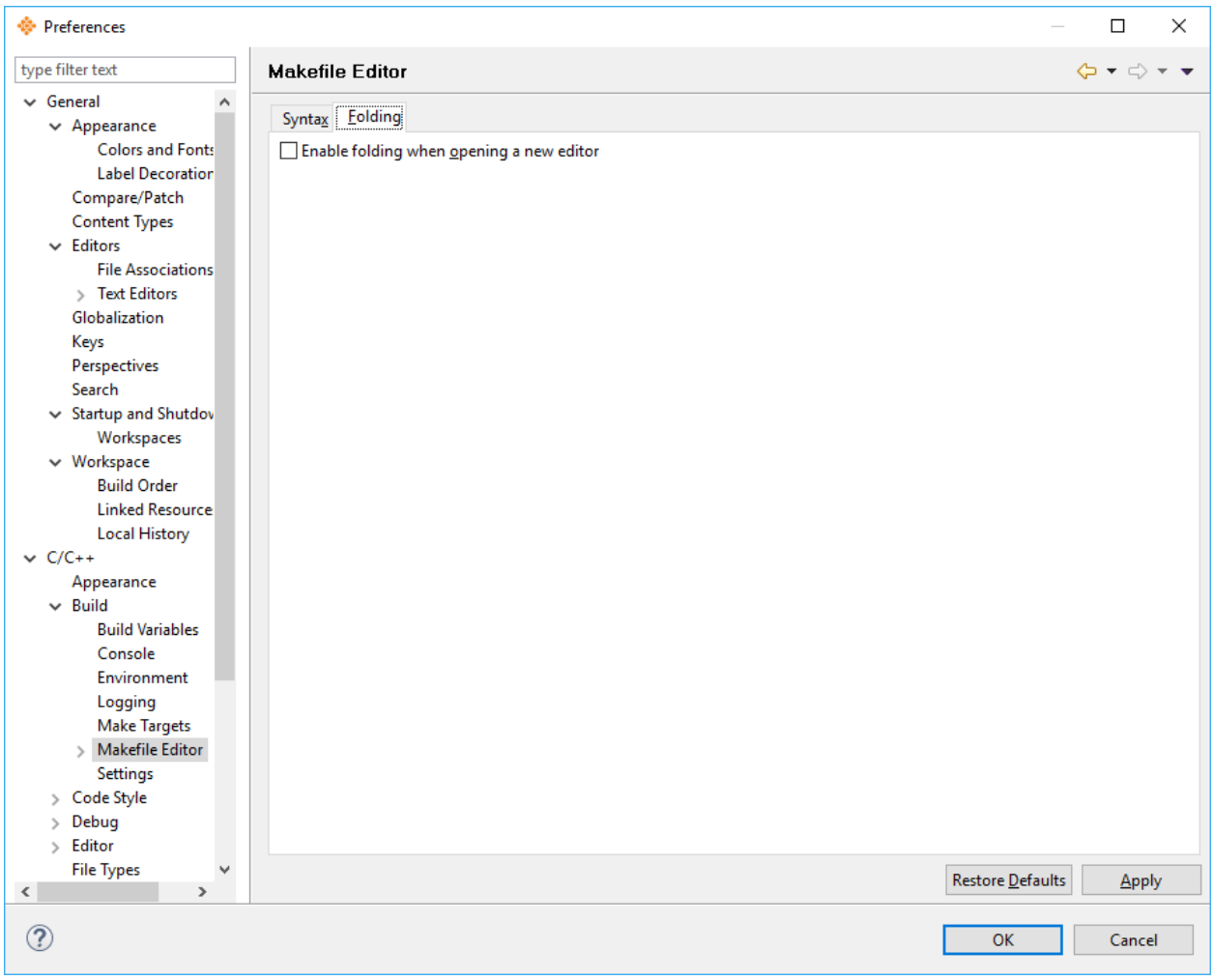


Рисунок 278. Панель «Makefile Editor», вкладка «Folding»

Таблица 123. Элементы управления панели «Makefile Editor»

Название элемента	Описание функционирования	По умолчанию
Вкладка Syntax		
Color	Выбрать цвет для указанного элемента синтаксиса.	Зелёный
Bold	Выделить жирным шрифтом выбранный элемент синтаксиса.	Выкл
Italic	Выделить курсивом выбранный элемент синтаксиса.	Выкл
Вкладка Folding		
Enable folding when opening a new editor	Включить/Отключить сворачивание при открытии нового редактора создания файлов.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.3. Настройки стиля оформления кода

Панель «**Code Style**» предназначена для адаптации настроек стиля оформления кода. Для перехода на панель «**Code Style**» следует выбрать **C/C++ → Code Style**.

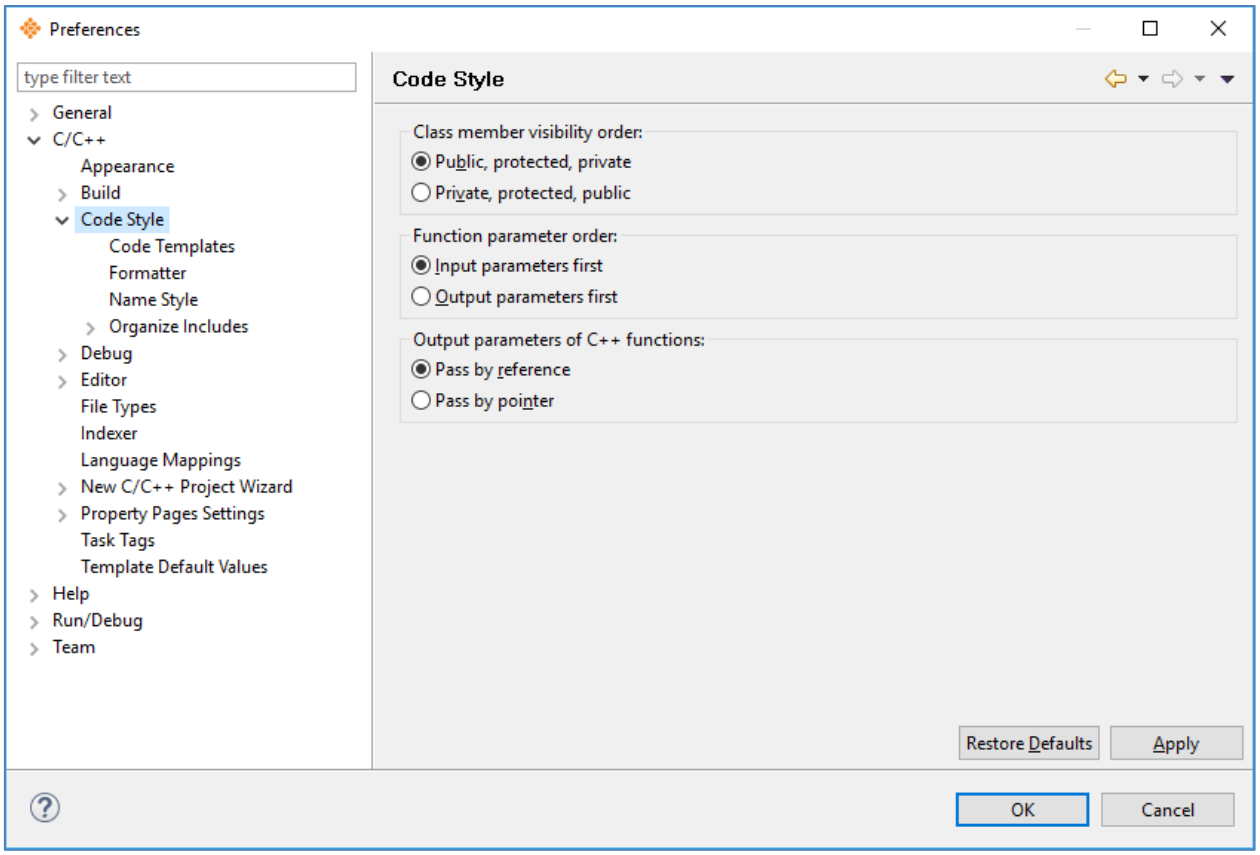


Рисунок 279. Панель «Code Style»

Таблица 124. Элементы управления панели «Code Style»

Название элемента	Описание функционирования	По умолчанию
Class member visibility order	Выбор порядка отображения членов класса: ▪ Public, protected, private; ▪ Private, protected, public	Public, protected, private
Function parameter order	Выбор порядка следования параметров функций: ▪ Input parameters first (сначала входные параметры); ▪ Output parameters first (сначала выходные параметры).	Input parameters first
Output parameters of C++ functions	Выбор способа передачи выходных параметров C++ .функций: ▪ Pass by reference (передача по ссылке); ▪ Pass by pointer (передача по указателю).	Pass by reference

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.3.1. Настройка шаблонов кода

Панель «**Code Templates**» предназначена для настройки шаблонов кода, комментариев и файлов. Для перехода на панель «**Code Templates**» следует выбрать **C/C++ → Code Style->Code Templates**.

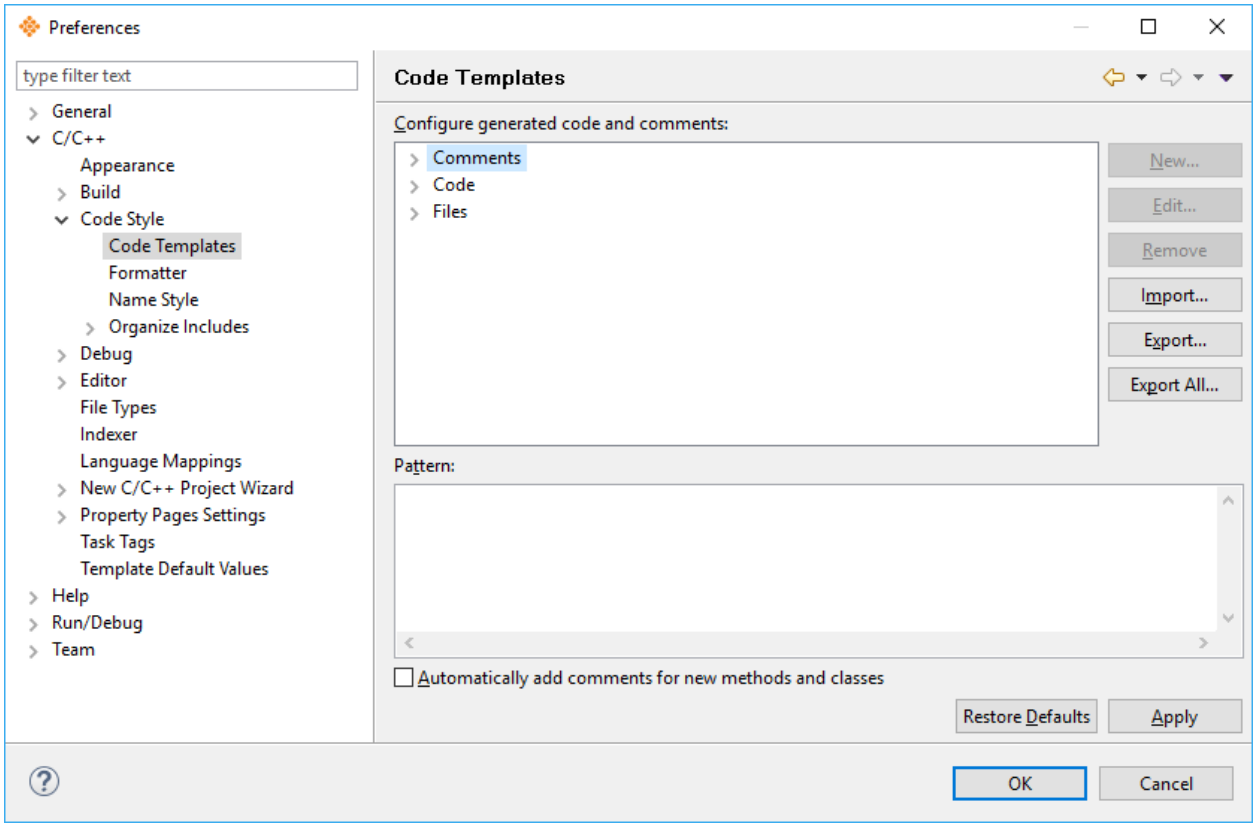


Рисунок 280. Панель «Code Templates»

Страница «**Code Templates»** содержит шаблоны, используемые действиями, создающими код и / или исходные файлы. Например, мастер создания классов использует шаблоны исходного и заголовочного файлов для создания новых файлов и шаблонов кода для вставки сгенерированных заготовок методов. Шаблоны содержат переменные, которые заменяются при применении шаблона. Некоторые переменные доступны во всех шаблонах, некоторые специфичны для отдельных шаблонов.

Таблица 125. Элементы управления панели «Code Templates»

Название элемента	Описание функционирования
New…	Добавить шаблон.
Edit..	Отредактировать шаблон.
Remove	Удалить шаблон.
Import…	Импортировать шаблон в среду из файловой системы.
Export…	Экспортировать шаблон из среды.
Export All	Экспортировать все шаблоны из среды.

Таблица 126. Встроенные шаблоны

Наименование	Описание
Comment Templates	
Files	Комментарий заголовка для новых исходных файлов или заголовочных файлов. Обратите внимание, что на этот шаблон можно ссылаться в шаблонах исходного и заголовочного файлов с помощью конструкции `${filecomment}`.
Types	Комментарий для новых типов. Обратите внимание, что на этот шаблон можно ссылаться в шаблонах исходного и заголовочного файлов с помощью конструкции `${typecomment}`.
Fields	Комментарий для новых полей.
Methods	Комментарий для новых методов.
Constructors	Комментарий для новых конструкторов.
Destructors	Комментарий для новых деструкторов.
Code Templates	
Beginnings of namespace declaration	Шаблон используется при объявлении начала пространства имен.
End of namespace declaration	Шаблон используется при объявлении конца пространства имен.
Class body	Шаблон используется при создании класса.
Method body	Шаблон используется при создании нового метода.
Constructor body	Шаблон используется при создании нового конструктора.
Destructor body	Шаблон используется при создании нового деструктора.
File Templates Шаблоны файлов могут быть определены и изменены не только для файлов C/C++, но и для других типов контента, таких как файлы Ассемблера или текстовые файлы. На основе одного из шаблонов, определенных на этой странице, можно создавать файлы с использованием шаблона.	
Default C++ source template	Используется при создании нового исходного файла C++. В шаблоне можно указать, куда будут добавлены комментарии. Обратите внимание, что шаблон может содержать переменные `${typecomment}` и `${filecomment}`.
Default C++ header template	Используется при создании нового заголовочного файла C++. В шаблоне можно указать, куда будут добавлены комментарии. Обратите внимание, что шаблон может содержать переменные `${typecomment}` и `${filecomment}`.
Default C source template	Используется при создании нового исходного файла C. В шаблоне можно указать, куда будут добавлены комментарии. Обратите внимание, что шаблон может содержать переменные `${typecomment}` и `${filecomment}`.

Наименование	Описание
Default C header template	Используется при создании нового заголовочного файла С. В шаблоне можно указать, куда будут добавлены комментарии. Обратите внимание, что шаблон может содержать переменные `${typecomment}` и `${filecomment}`.
Default assembly template	Используется при создании нового ассемблерного файла.
Default text file template	Используется при создании нового текстового файла.

6.12.2.3.2. Настройки форматирования

Панель «**Formatter**» предназначена для настройки стилей кода проекта для функций «умного» ввода, таких как автовыравнивание и форматирование.

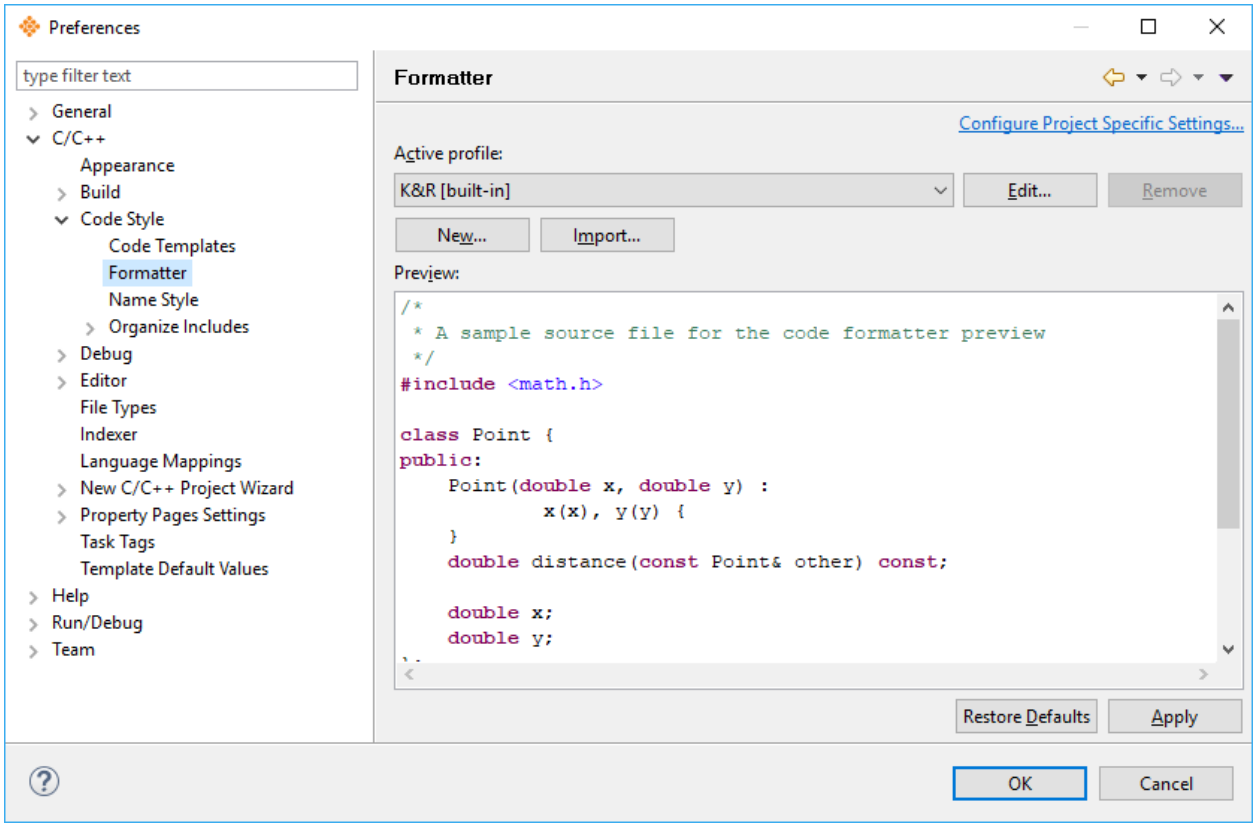


Рисунок 281. Панель «Formatter»

Таблица 127. Элементы управления панели «Formatter»

Название элемента	Описание функционирования
Configure Project Specific Settings...	Открыть диалоговое окно «**Project Specific Configuration**» для выбора проекта, к которому будут применены настройки форматирования.
Edit...	Открыть диалоговое окно для просмотра и редактирования настроек активного профиля. Редактирование доступно только для профилей, созданных пользователями.
Remove	Удалить выбранный профиль. Операция доступна только для профилей, созданных пользователями.

Название элемента	Описание функционирования
New...	Открыть диалоговое окно для создания нового профиля. В этом диалоговом окне необходимо ввести имя для нового профиля. Кроме того, можно выбрать встроенный или определенный пользователем существующий профиль для создания нового профиля.
Import...	Импортировать профили из файловой системы в среду.

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.3.3. Настройки организации подключаемых заголовков

Панель «**Organize includes**» предназначена для настройки организации подключаемых заголовок.

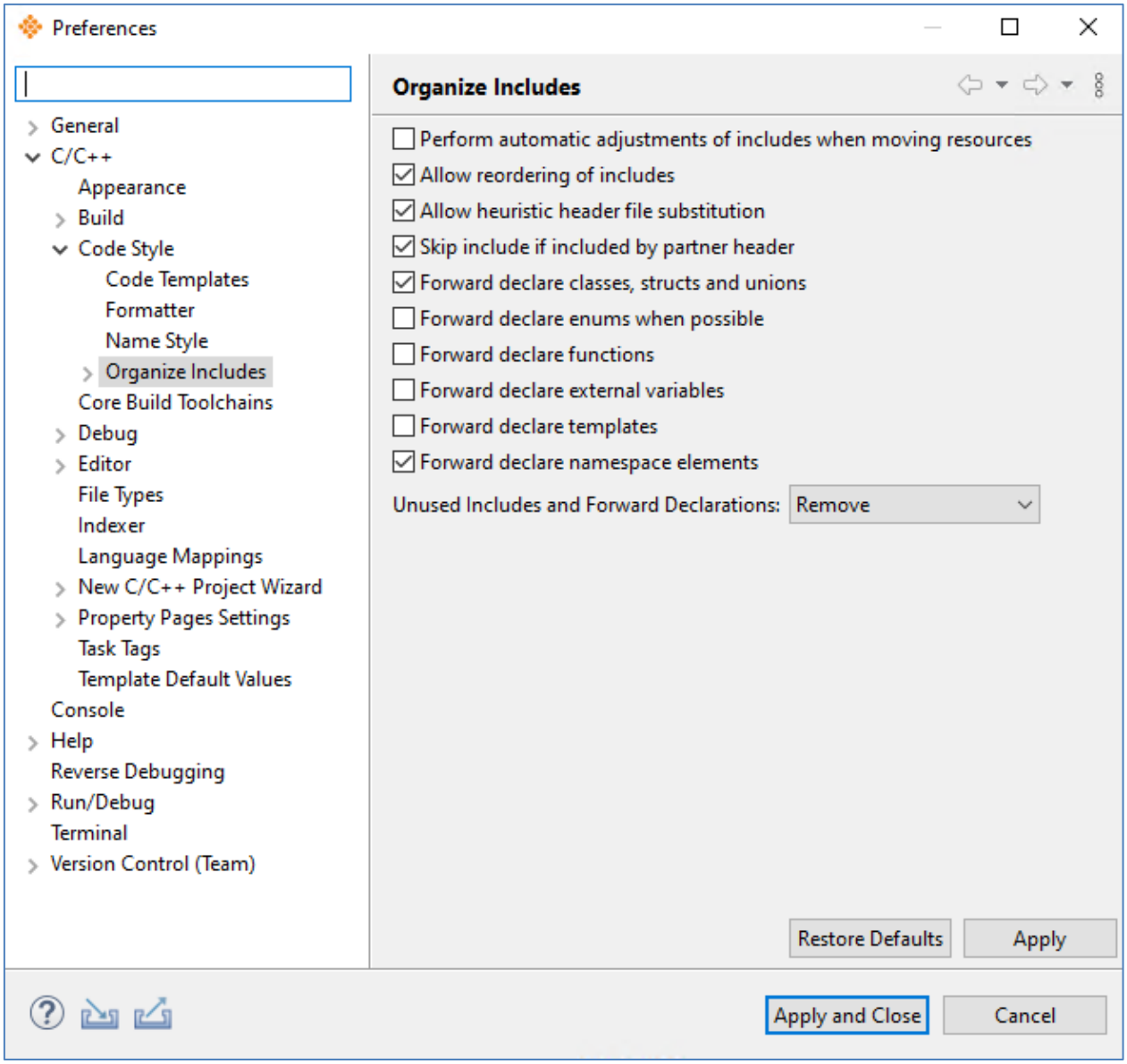


Рисунок 282. Панель «Organize includes»

Таблица 128. Элементы управления панели «Organize includes»

Название элемента	Описание функционирования	По умолчанию
Perform automatic adjustments of includes when moving resources	Включить/отключить автоматическую корректировку путей в директивах include при перемещении ресурсов.	Выкл
Allow reordering of includes	Включить/отключить переупорядочивание подключений.	Вкл
Allow heuristic header file substitution	Включить/отключить эвристическую замену заголовочных файлов.	Вкл
Skip include if included by partner header	Включить/отключить пропуск подключения, если оно уже выполнено в связанном заголовочном файле.	Вкл
Forward declare classes, structs and unions	Включить/отключить предварительное объявление классов, структур и объединений.	Вкл
Forward declare enums when possible	Включить/отключить предварительное объявление перечислений, когда это возможно.	Выкл
Forward declare functions	Включить/отключить предварительное объявление функций.	Выкл
Forward declare external variables	Включить/отключить предварительное объявление внешних переменных.	Выкл
Forward declare templates	Включить/отключить предварительное объявление шаблонов.	Выкл
Forward declare namespace elements	Включить/отключить предварительное объявление элементов пространства имён.	Вкл
Unused Includes and Forward Declarations	Выбрать, что делать с неиспользуемыми подключениями и предварительными объявлениями: Remove – полностью убрать из кода, Comment out – оставить в коде, но как комментарий, Keep – не изменять, оставить как есть.	Remove

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4. Настройки отладки

Панель «**Debug**» предназначена для управления основными настройками для отладки C/C++. Для перехода на панель «**Debug**» следует выбрать **C/C++ → Debug**.

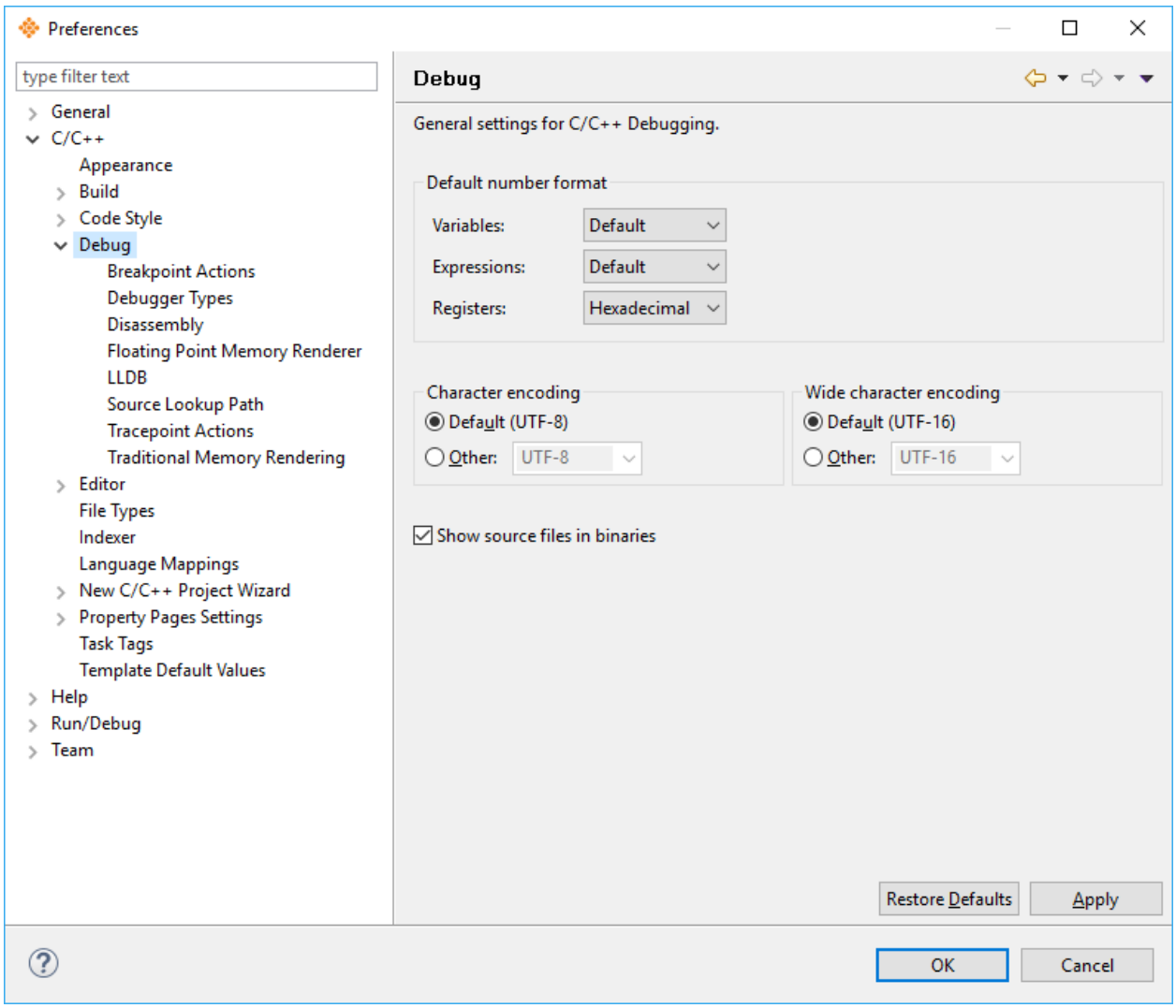


Рисунок 283. Панель «Debug»

Таблица 129. Элементы управления панели «Debug»

Название элемента	Описание функционирования	По умолчанию
Variables	Выбрать из списка формат переменной по умолчанию: ▪ **Default** – по умолчанию; ▪ **Hexadecimal** – шестнадцатеричный; ▪ **Decimal** – десятеричный; ▪ **Octal** – восьмеричный; ▪ **Binary** – бинарный.	Default
Expressions	Выбрать из списка формат выражения по умолчанию: ▪ **Default** – по умолчанию; ▪ **Hexadecimal** – шестнадцатеричный; ▪ **Decimal** – десятеричный; ▪ **Octal** – восьмеричный; ▪ **Binary** – бинарный.	Default

Название элемента	Описание функционирования	По умолчанию
Registers	Выбрать из списка формат регистра по умолчанию: ▪ **Default** – по умолчанию; ▪ **Hexadecimal** – шестнадцатеричный; ▪ **Decimal** – десятеричный; ▪ **Octal** – восьмеричный; ▪ **Binary** – бинарный.	Default
Character encoding	Выбрать кодировку символов отлаживаемой программы. Это применимо к строкам типа «`char`».	Default (UTF-8)
Wide character encoding	Выбрать кодировку «широких» символов отлаживаемой программы. Применяется к строкам типа «`wchar_t`».	Default (UTF-16)
Show source files in binaries	Показать/скрыть исходные файлы, связанные с двоичными файлами проекта.	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.1. Настройки для точек останова

Панель «**Breakpoint Actions**» предназначена для создания, добавления или удаления действий точек останова. Действия точек останова, описанных в подразделе, доступны во всех проектах текущего рабочего пространства. Для перехода на панель **«Breakpoint Actions»** следует выбрать **C/C++ → Debug → Breakpoint Actions**.

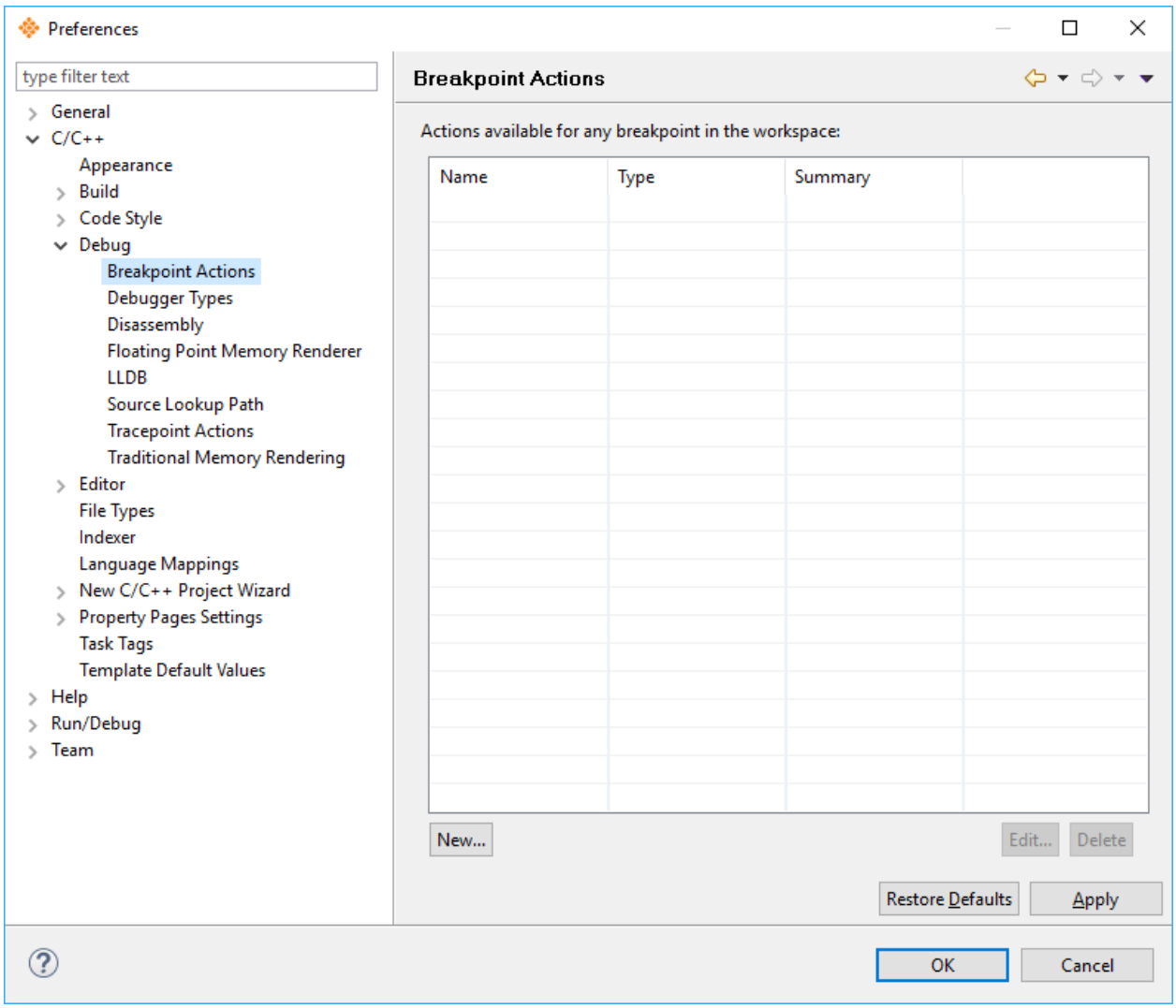


Рисунок 284. Панель «Breakpoint Actions»

Таблица 130. Элементы управления панели «Breakpoint Actions»

Название элемента	Описание функционирования	По умолчанию
New...	Создать новое действие для точки останова.	–
Edit...	Открыть диалоговое окно «**Action**» для изменения настроек выбранного действия.	–
Delete	Удалить выбранное действие.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.2. Настройки отладочных конфигураций

Панель «**Debug Configurations**» предназначена для настройки автоматического создания и удаления отладочных конфигураций при работе с проектами. Для перехода на панель «**Debug Configurations**» следует выбрать **C/C++ → Debug → Debug Configurations**.

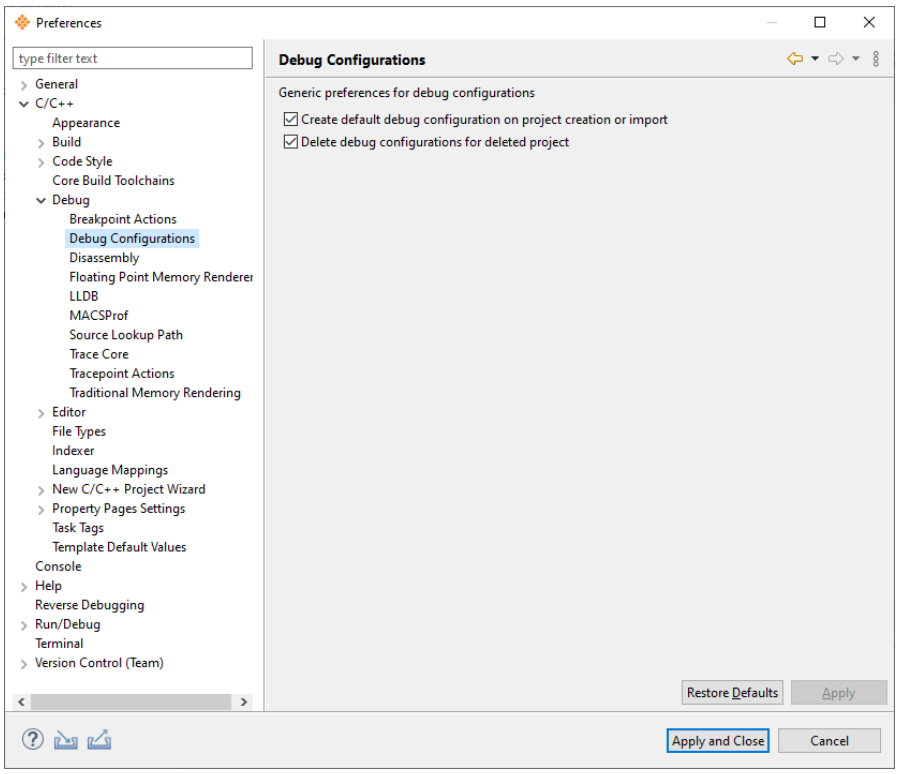


Рисунок 285. Панель «Debug Configurations»

Таблица 131. Элементы управления панели «Debug Configurations»

Название элемента	Описание функционирования	По умолчанию
Create default debug configuration on project creation or import	Создавать ли автоматически отладочную конфигурацию при создании или импорте проекта	Вкл
Delete debug configurations for deleted project	Удалять ли автоматически отладочную конфигурацию при удалении проекта	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.3. Настройки по умолчанию для вкладки «Disassembly»

Панель «**Disassembly**» предназначена для установки параметров по умолчанию для вкладки «**Disassembly**» при работе в режиме отладки. Для перехода на панель «**Disassembly**» следует выбрать **C/C++ → Debug → Disassembly**.

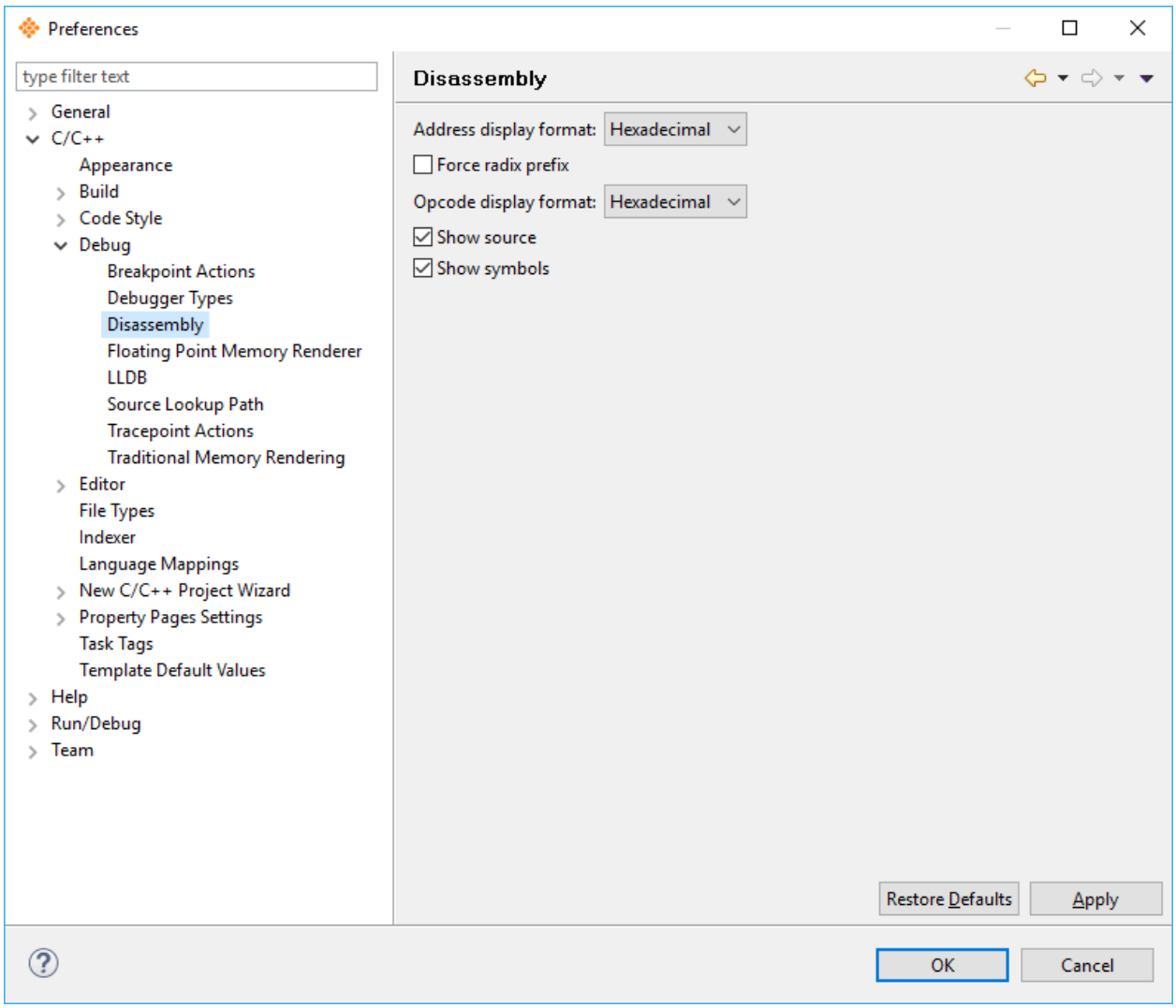


Рисунок 286. Панель «Disassembly»

Таблица 132. Элементы управления панели «Disassembly»

Название элемента	Описание функционирования	По умолчанию
Address display format	Выбрать формат по умолчанию для адреса: ▪ **Hexadecimal** – шестнадцатеричный; ▪ **Decimal** – десятеричный; ▪ **Octal** – восьмеричный.	Hexadecimal
Opcode display format	Выбрать формат по умолчанию для опкодов: ▪ **Hexadecimal** – шестнадцатеричный; ▪ **Decimal** – десятеричный; ▪ **Octal** – восьмеричный.	Hexadecimal
Show source	Показать/скрыть исходный текст.	Вкл
Show symbols	Показать/скрыть метки.	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.4. Настройка таймаутов выполнения операций отладки

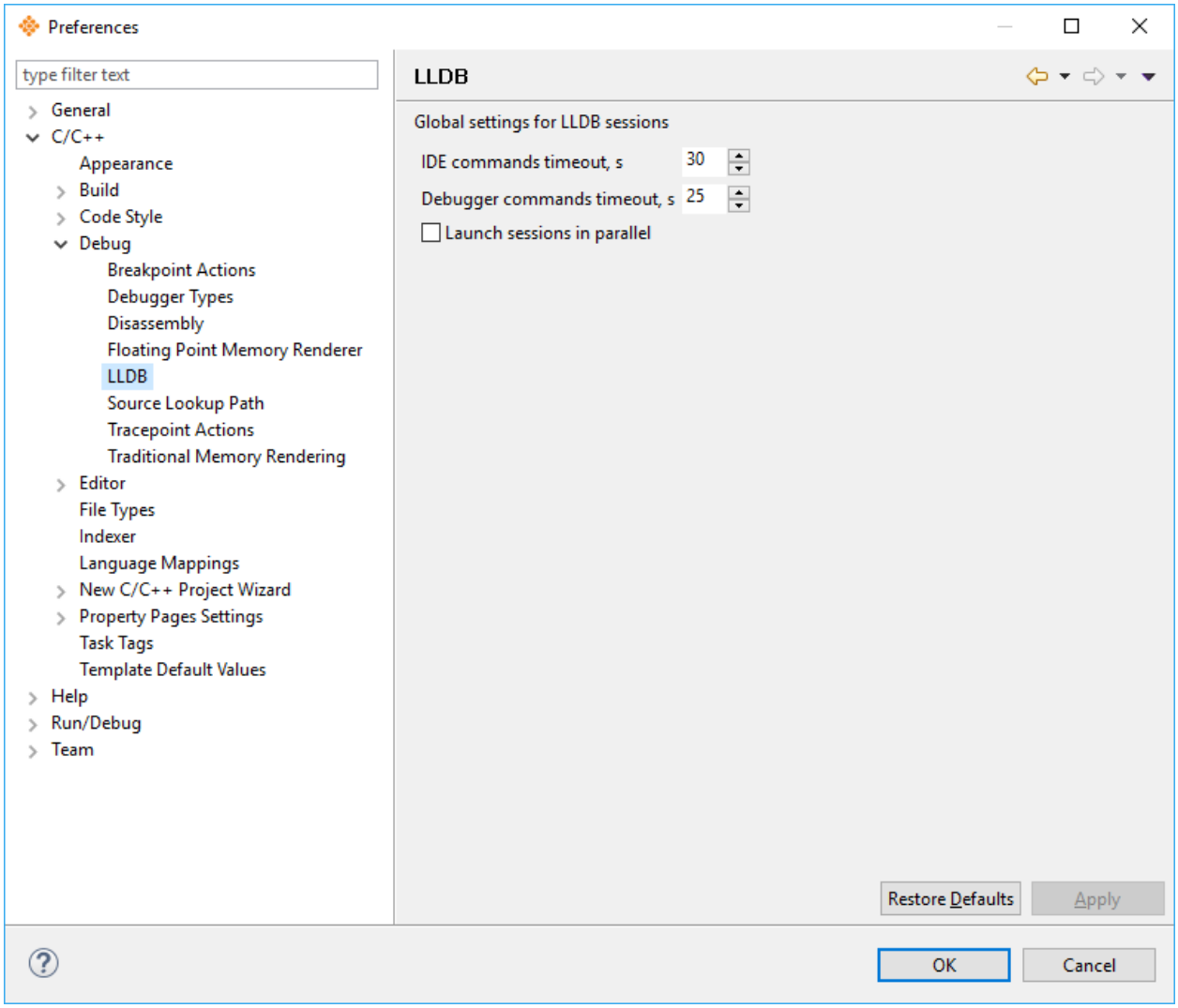


Рисунок 287. Панель «LLDB»

Таблица 133. Элементы управления панели «LLDB»

Название элемента	Описание функционирования	По умолчанию
IDE commands timeout, s	Временная задержка (таймаут) на операции отладки в IDE.	30сек
Debugger commands timeout, s	Временная задержка (таймаут) на операции отладки в отладчике LLDB. Рекомендуется устанавливать таймаут команд отладчика меньше на 5 секунд по сравнению с таймаутом команд отладки в IDE.	25 сек

Название элемента	Описание функционирования	По умолчанию
Launch sessions in parallel	Включить режим параллельного запуска сессий отладки. При отладке многопроцессорной системы формируется несколько сессий отладки (их количество равно числу процессоров в многопроцессорной системе). По умолчанию, эти сессии запускаются последовательно. Для ускорения отладки мультисессии можно включить режим параллельного запуска сессий отладки. При включении режима параллельного запуска сессий отладки может потребоваться корректировка таймаутов выполнения команд отладки (особенно, при увеличении объема исполняемых файлов, числа процессоров, или снижении частоты JTAG).	Отключен

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.5. Настройки приложения MACSProf

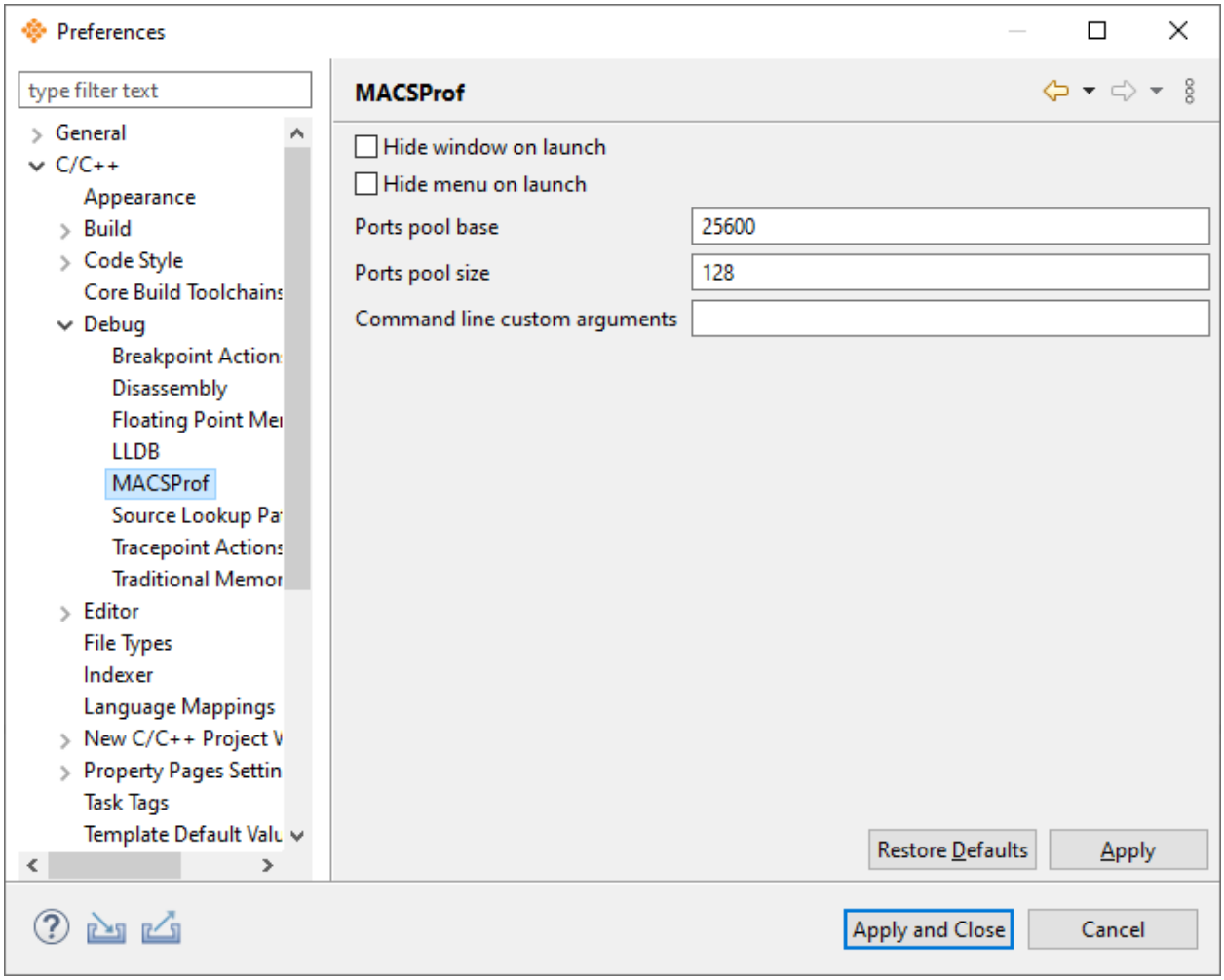


Рисунок 288. Панель «MACSProf»

Таблица 134. Элементы управления панели «MACSProf»

Название элемента	Описание функционирования	По умолчанию
Hide window on launch	При установленном флаге окно MACSProf делается невидимым сразу после запуска.	Выкл.
Hide menu on launch	При установленном флаге основное меню MACSProf делается невидимым	Выкл.
Ports pool base	Номер первого порта из пула портов, выделенного для запуска параллельных процессов MACSProf.	25600
Ports pool size	Количество портов в пуле портов, выделенном для запуска параллельных процессов MACSProf.	128
Command line custom arguments	Коммандная строка для задания дополнительных фиксированных параметров запуска. Перечень параметров запуска см. в руководстве пользователя на программу «МАКС Профилирование. Многопроцессорные системы».	Пустая

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.6. Настройки хранилищ источников

Панель «**Source Lookup Path**» предназначена для добавления или удаления хранилищ источников. Для перехода на панель «**Source Lookup Path**» следует выбрать **C/C++ → Debug → Source Lookup Path**.

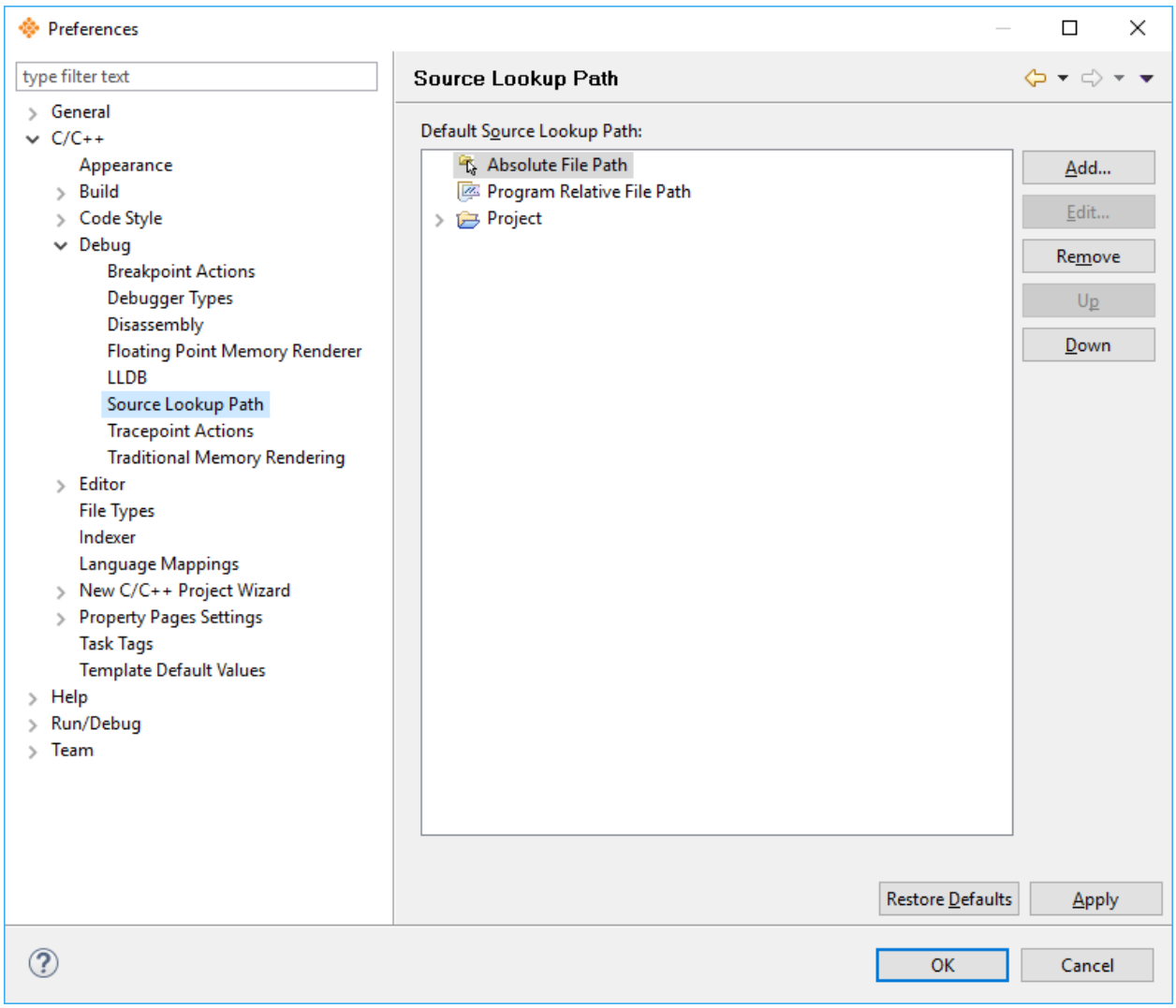


Рисунок 289. Панель «Source Lookup Path»

Таблица 135. Элементы управления панели «Source Lookup Path»

Название элемента	Описание функционирования	По умолчанию
Add...	Открыть диалоговое окно «**Add Source**» для добавления нового хранилища источника одного из следующих типов: ▪ **Absolute File Path** – абсолютный путь файла; ▪ **Compilation Directory** – папка компиляции; ▪ **File System Directory** – папка файловой системы; ▪ **Path Mapping** – метки пути; ▪ **Program Relative File Path** – путь файла, связанного с приложением; ▪ **Project** – проект; ▪ **Project-Path Relative to Source Folders** – путь проекта, связанного с исходными папками; ▪ **Workspace** – рабочее пространство; ▪ **Workspace Folder** – папка рабочего пространства.	–

Название элемента	Описание функционирования	По умолчанию
Edit...	Открыть диалоговое окно «**Edit File System Directory**» для изменения выбранное хранилища источника.	–
Remove	Удалить выбранное хранилище источника.	–
Up	Переместить выбранное хранилище источника вверх в списке.	–
Down	Переместить выбранное хранилище источника вниз в списке.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.4.7. Настройки буфера трассировки

Панель «**Trace Core**» предназначена для настройки работы с буфером трассировки (см. также п. 6.11.17). Для перехода на панель «**Trace Core**» следует выбрать **C/C++ → Debug → Trace Core**.

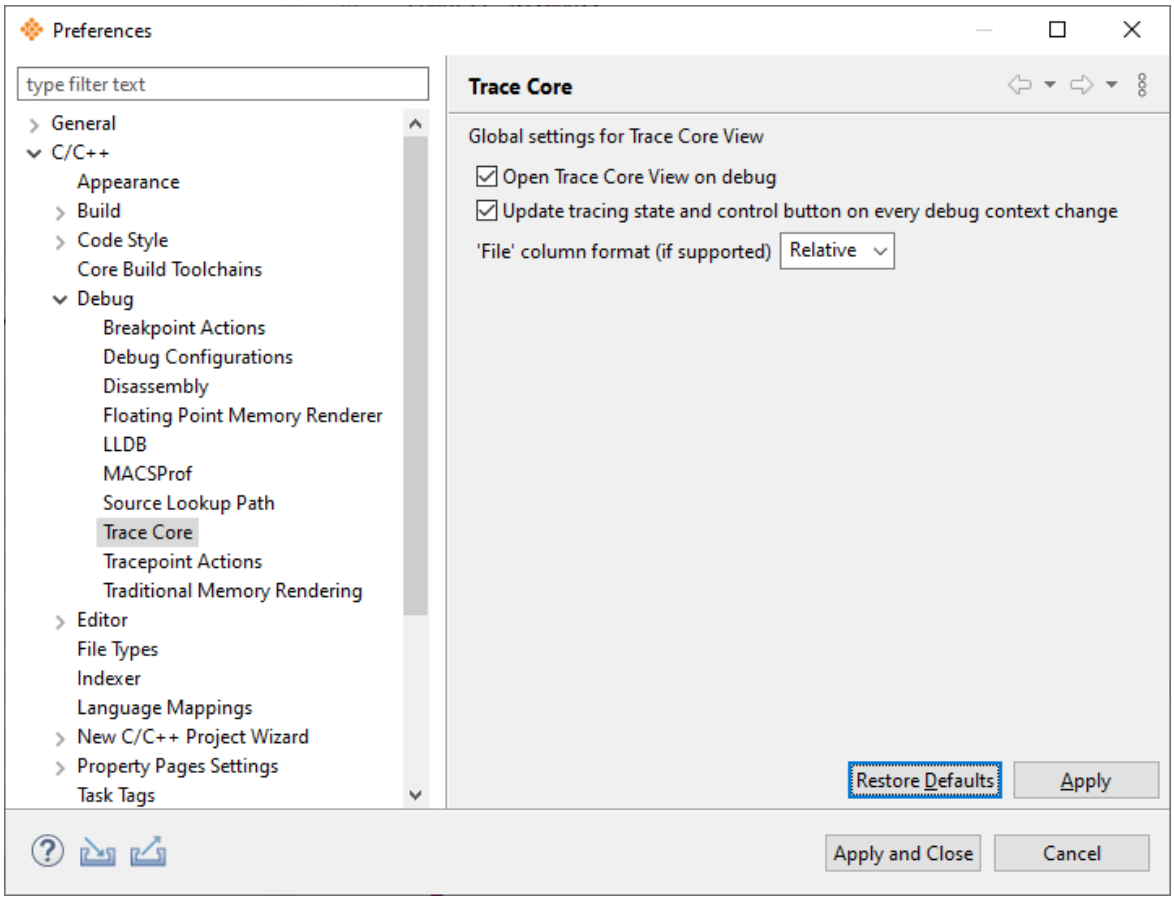


Рисунок 290. Панель «Trace Core»

Таблица 136. Элементы управления панели «Trace Core»

Название элемента	Описание функционирования	По умолчанию
Open Trace Core View on debug	Автоматически открывать вкладку «**Trace Core**» при запуске отладки.	Вкл.

Название элемента	Описание функционирования	По умолчанию
Update tracing and control button on every debug context change	Обновление кнопки состояния и управления при каждом изменении выбранного контекста отладки пользователем (например, при переключении между сессиями с разными процессорами) или в процессе работы среды.	Вкл.
'File' column format (if supported)	Формат колонки «**File**»: ▪ **Short** – только имя файла; ▪ **Relative** – имя файла и путь относительно рабочего пространста или среды; ▪ **Full** – полной путь и имя файла.	Relative

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5. Настройки редактора C/C++

Панель «**Editor**» предназначена для настройки внешнего вида подчеркивания текста в редакторах C/C++. Для перехода на панель «**Editor**» следует выбрать **C/C++ → Editor**.

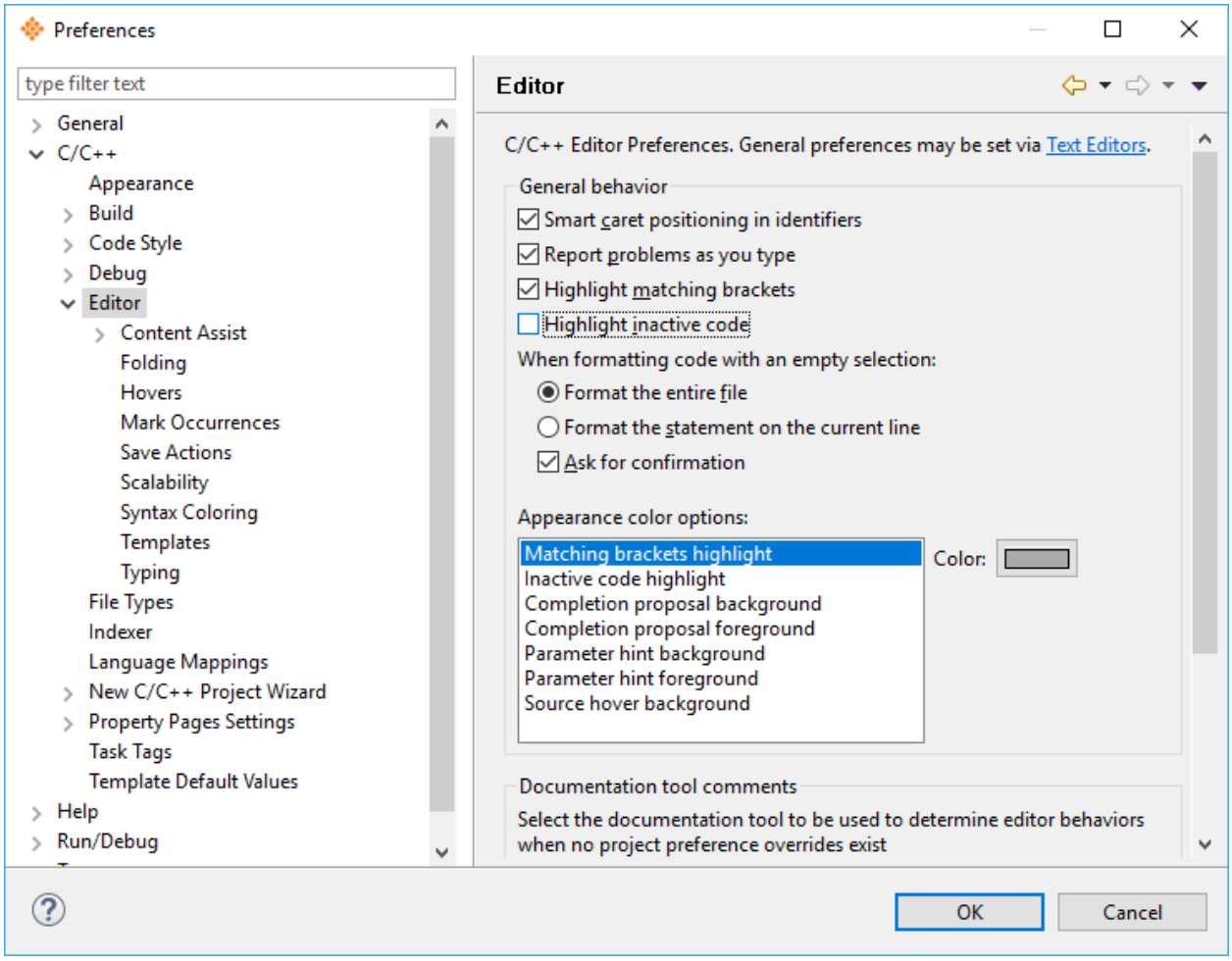


Рисунок 291. Панель «Editor»

Таблица 137. Элементы управления панели «Editor»

Название элемента	Описание функционирования	По умолчанию
Smart caret positioning in identifiers	Отобразить дополнительные границы слов внутри идентификаторов.	Вкл
Report problems as you type	Сразу выделить в редакторе найденные проблемы.	Вкл
Highlight matching brackets	Подсветить соответствующую скобку, когда курсор находится рядом со скобками.	Вкл
Highlight inactive code	Выделить неактивный код, когда курсор располагается внутри него.	Вкл
When formatting code with an empty selection	Задать форматировании кода с пустым выделением. Доступные варианты: ▪ **Format the entire file**–форматировать весь файл; ▪ **Format the statement on the current line** – форматировать выражение в текущей строке. Также можно задать флажок, запрашивающий подтверждение у пользователя.	Весь файл
Appearance color options	Показать список символов, для которых можно выбрать цвет.	–
Color	Задать цвет. Индикатор справа окрашивается в тот цвет, в котором будет отображаться выделенный элемент.	–
Documentation tool comments	Выбрать инструмент работы с документацией, который будет определять представление и поведение редактора. Такие функции редактора, как автозавершение, создание комментариев в документации и выделение синтаксиса. Выбранный инструмент будет использован для файлов, не связанных с проектом, или для файлов, в которых не были установлены настройки уровня проекта.	Нет

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.1. Настройки автозавершения

Панель «**Content Assist**» предназначена для настройки функции автозавершения в редакторе C/C++. Для перехода на панель «**Content Assist**» следует выбрать **C/C++ → Editor → Content Assist**.

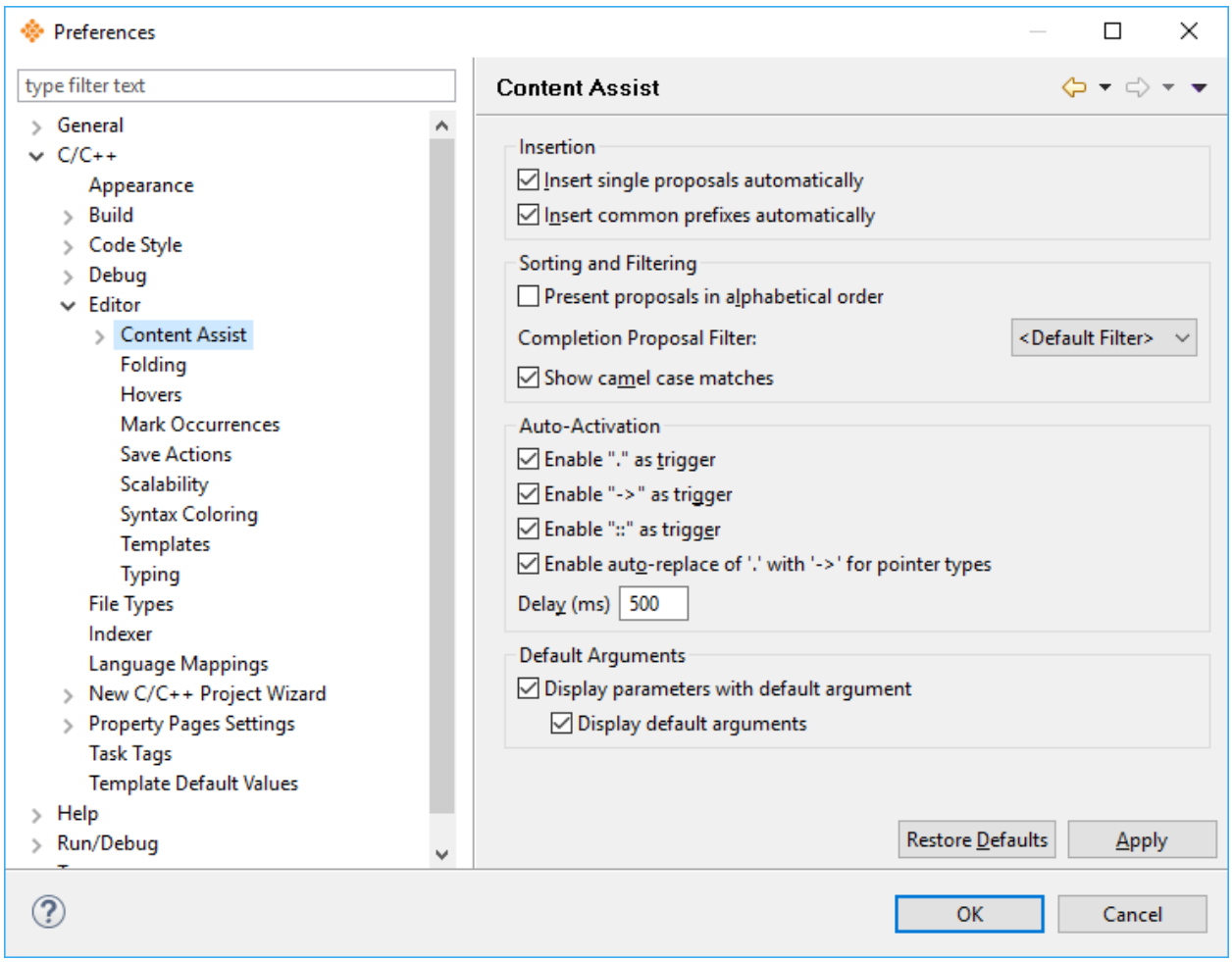


Рисунок 292. Панель «Content Assist»

Таблица 138. Элементы управления панели «Content Assist»

Название элемента	Описание функционирования	По умолчанию
Insert single proposals automatically	Вставить элемент в код, когда функция автозавершения находит только одно предложение.	Вкл
Insert common prefixes automatically	Вставить общие префиксы.	Вкл
Present proposals in alphabetical order	Представить предложения в алфавитном порядке.	Выкл
Completion Proposal Filter	Поставить фильтры предложения завершения. Фильтры предложения завершения могут сортировать предложения функции автозавершения для удобства работы. Для этого следует выбрать нужный фильтр автозавершения из списка доступных.	По умолчанию
Show camel case matches	Показать совпадения с «Горбатым Регистром».	Вкл
Enable "." as trigger	Включить/отключить использование "." как триггера для автозавершения.	Вкл
Enable "->" as trigger	Включить/отключить использование "->" как триггера для автозавершения.	Вкл

Название элемента	Описание функционирования	По умолчанию
Enable "::" as trigger	Включить/отключить использование "::" как триггера для автозавершения.	Вкл
Enable auto-replace of '.' with '->' for pointer types	Включить автоматическую замену '.' на '->' для типов указатель (`pointer`).	Вкл
Delay (ms)	Указать время до запуска автозавершения (в мс).	500
Display parameters with default argument	Отображать параметры с аргументом по умолчанию.	Вкл
Display default arguments	Отображать аргументы по умолчанию.	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.2. Настройки сворачивания

Панель «**Folding**» предназначена для настройки сворачивания в редакторе C/C++. Для перехода на панель «**Folding**» следует выбрать **C/C++ → Editor → Folding**.

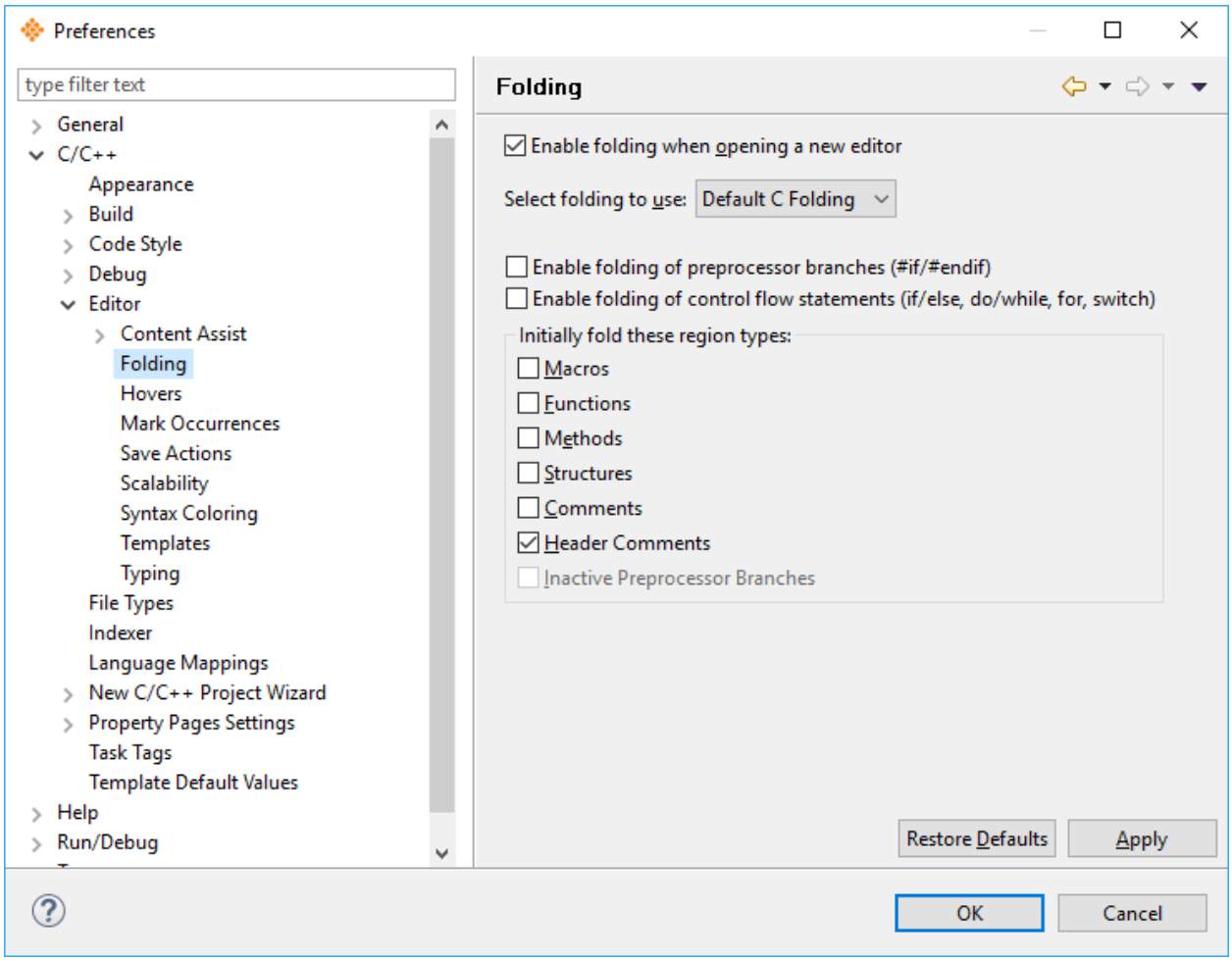


Рисунок 293. Панель «Folding»

Таблица 139. Элементы управления панели «Folding»

Название элемента	Описание функционирования	По умолчанию
Enable folding when opening a new editor	Применить правила сворачивания к новому редактору.	Вкл
Select folding to use	Выбрать вариант сворачивания из списка доступных.	Сворачивание С по умолчанию
Enable folding of preprocessor branches (#if/#endif)	Свернуть предпроцессорные ветки.	Выкл
Enable folding of control flow statements (if/else, do/while, for, switch)	Свернуть операторы управления.	Выкл
Initially folds these region types	Выбрать типов объектов для сворачивания: ▪ **Macros** – сворачивает макроопределения; ▪ **Functions** – сворачивает функции; ▪ **Methods** – сворачивает методы; ▪ **Structures** – сворачивает структуры; ▪ **Comments** – сворачивает комментарии; ▪ **Header Comments** – сворачивает комментарии к заголовку; ▪ **Inactive Preprocessor Branches** – сворачивает ветки процессора.	Выкл Выкл Выкл Выкл Выкл Вкл Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.3. Настройки поведения курсора

Панель «**Hovers**» предназначена для настройки поведения курсора в редакторе C/C++. Для перехода на панель «**Hovers**» следует выбрать **C/C++ → Editor → Hovers**.

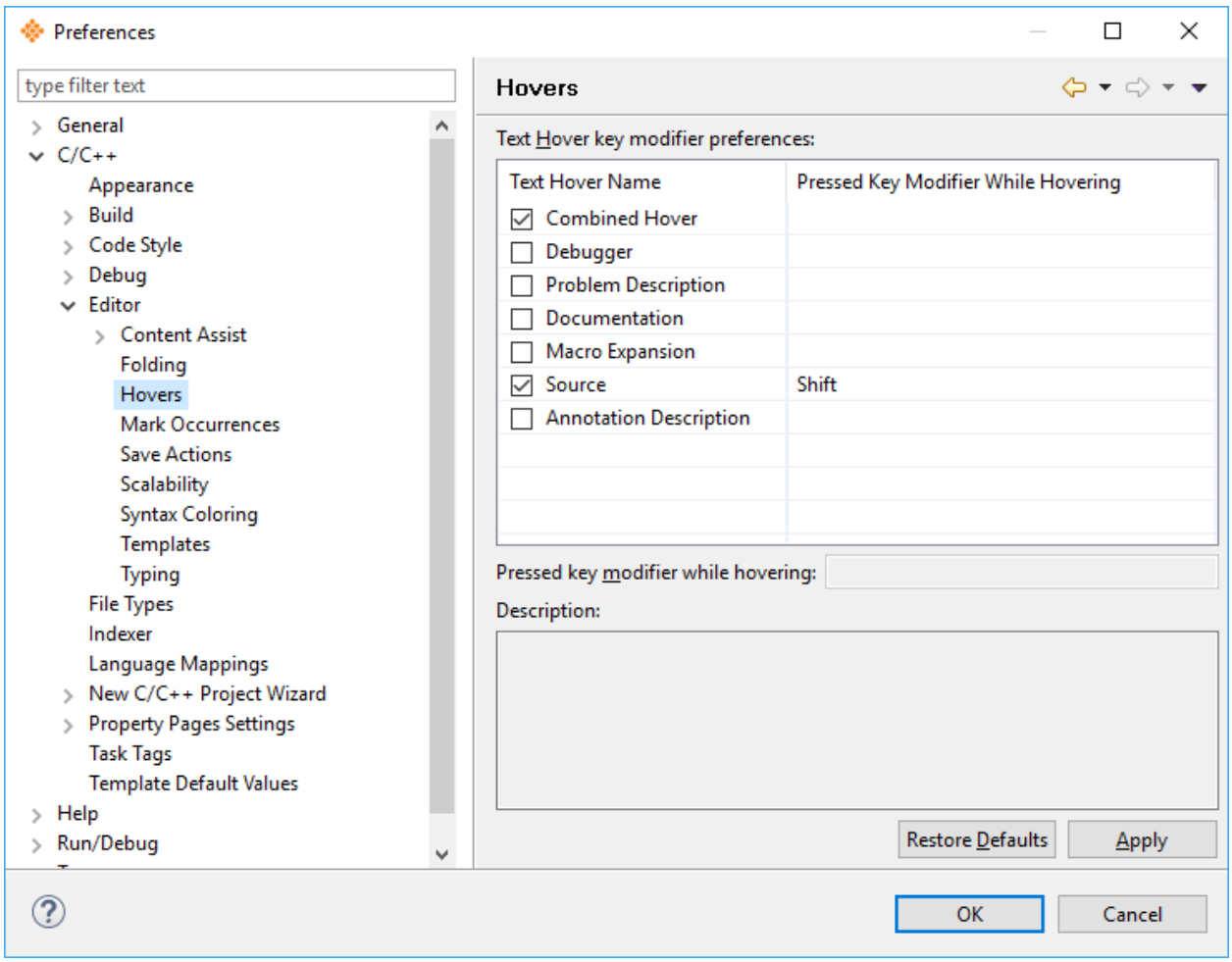


Рисунок 294. Панель «Hovers»

Таблица 140. Элементы управления панели» Hovers»

Название элемента	Описание функционирования	По умолчанию
Text Hover key modifier preferences	Задать горячие клавиши, чтобы включить альтернативное поведение курсора. Например, при наведении курсора и нажатии клавиши **Ctrl** отобразится ссылка на источник объявления элемента.	–
Pressed key modifier while hovering	Задать клавишу-модификатор для выбранного названия наведения курсора.	–
Description	Вывод описания выбранного наведения курсора.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.4. Настройки аннотаций

Панель «**Mark Occurrences**» предназначена для настройки аннотаций редактора C/C++. Для перехода на панель «**Mark Occurrences**» следует выбрать **C/C++ → Editor → Mark Occurrences**.

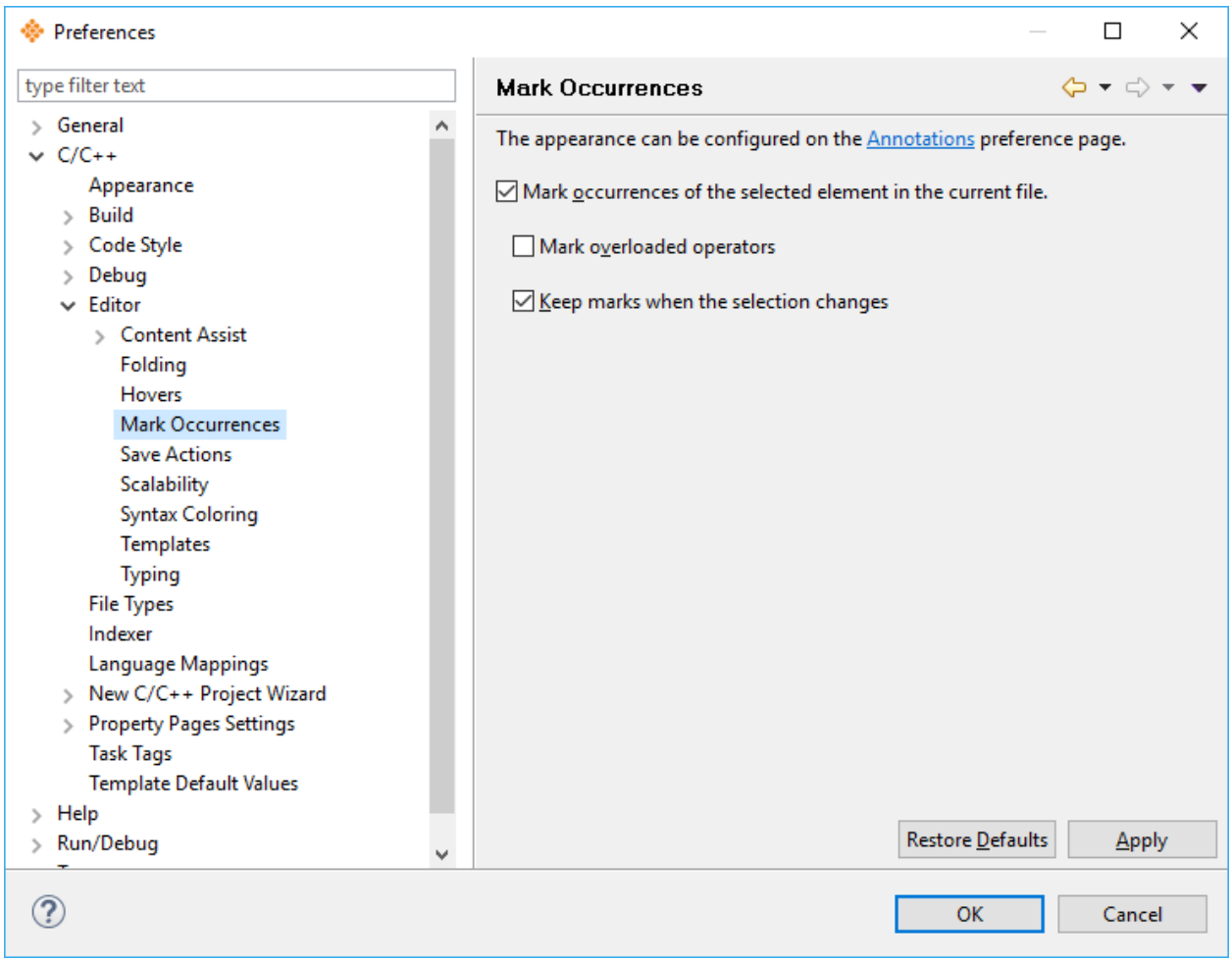


Рисунок 295. Панель «Mark Occurrences»

Таблица 141. Элементы управления панели «Mark Occurrences»

Название элемента	Описание функционирования	По умолчанию
Mark occurrences of the selected element in the current file	Показать, где в файле есть ссылки на переменную, функцию, метод, тип, макрос или другой элемент.	Вкл
Mark overloaded operators	Пометить перегруженные операторы.	Выкл
Keep marks when the selection changes	Закрепить расположение отметки о наличии при новом выборе.	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.5. Настройки действий при сохранении

Панель «**Save Actions**» предназначена для настройки действий редактора C/C++ при сохранении файла. Для перехода на панель «**Save Actions**» следует выбрать **C/C++ → Editor → Save Actions**.

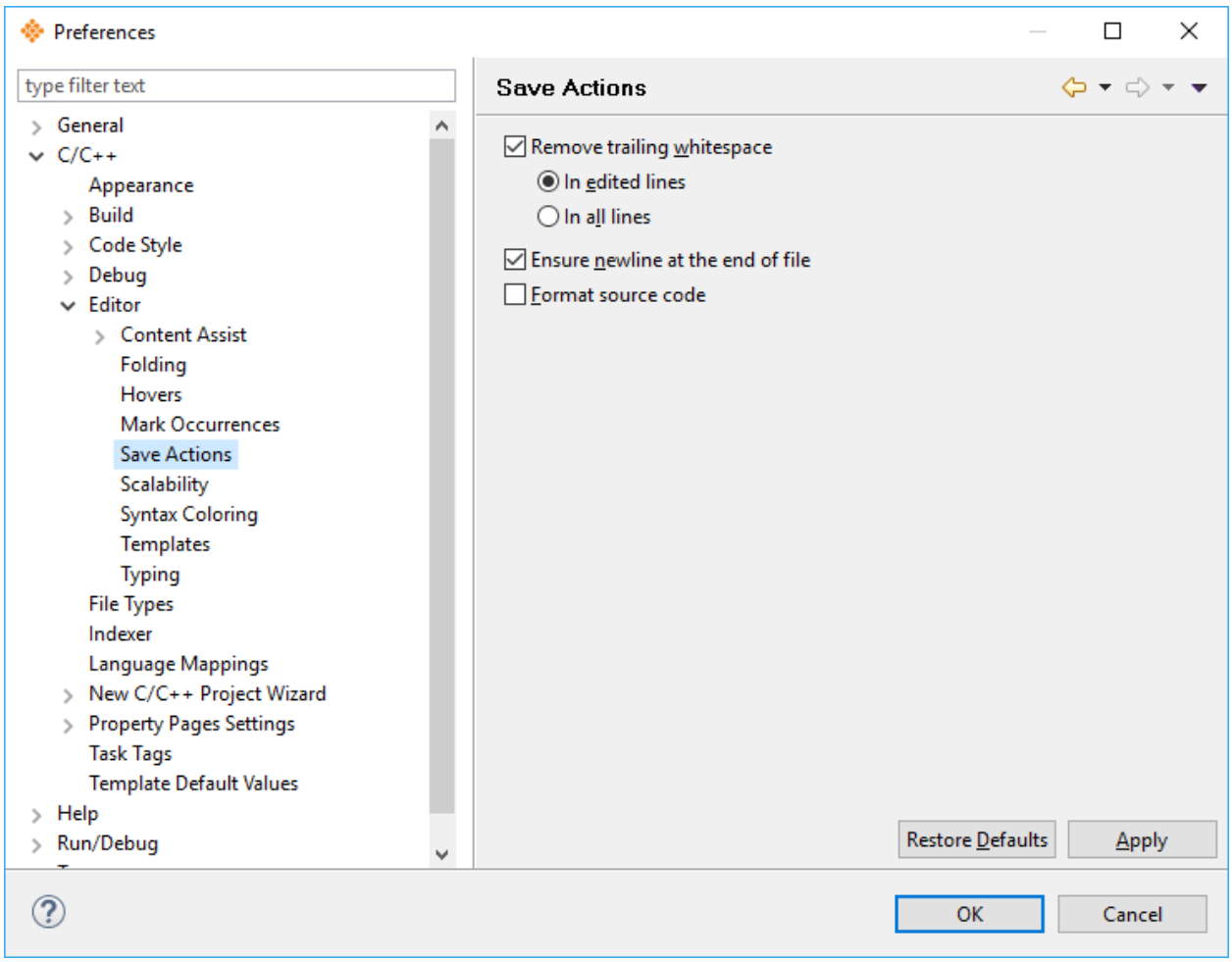


Рисунок 296. Панель «Save Actions»

Таблица 142. Элементы управления панели «Save Actions»

Название элемента	Описание функционирования	По умолчанию
Remove trailing whitespace	Удалить пробелы и знаки табуляции: ▪ в конце редактируемых строк; ▪ во всех строках.	Вкл В конце редактируемых строк
Ensure newline at the end of file	Вставить новую строку в конце последней строки, если у последней строки ее еще нет, при сохранении.	Вкл
Format source code	Отформатировать исходный код.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.6. Настройки опций для работы с большими исходными файлами

Панель «**Scalability**» предназначена для настройки опций для работы с большими исходными файлами. Для перехода на панель «**Scalability**» следует выбрать **C/C++ → Editor → Scalability**.

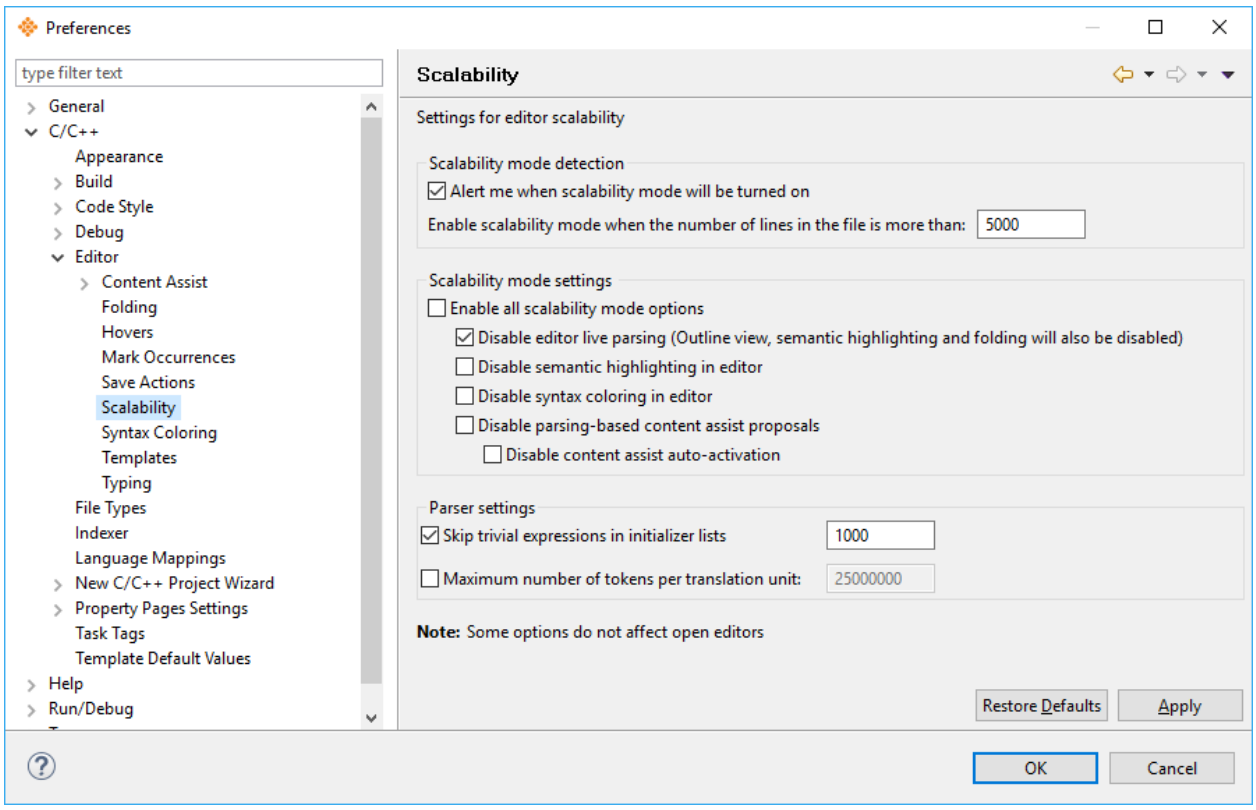


Рисунок 297. Панель «Scalability»

Таблица 143. Элементы управления панели «Scalability»

Название элемента	Описание функционирования	По умолчанию
Alert me when scalability mode will be turned on	Показать информационные диалоговые окна, когда в редакторе C/C++ открывается большой файл, размеры которого превышают установленный лимит строк.	Вкл
Enable scalability mode options when the number of lines in the file is more than	Указать количество строк в файле, при превышении которого редактор C/C++ запустит режим масштабируемости, и все нижеприведенные параметры режима масштабируемости вступят в силу.	5000
Enable all scalability mode options	Включить все опции ниже.	Выкл
Disable editor live parsing	Отключить синтаксический анализ при вводе текста. Также отключаются другие функции такие, как отображение контура, выделение синтаксиса, сворачивание и т.д.	Вкл
Disable semantic highlighting	Отключить выделение цветом идентификаторов C/C++.	Выкл
Disable syntax coloring	Отключить выделение цветом ключевых слов, комментариев и опечаток.	Выкл
Disable parsing-based content assist proposals	Отключить предложения автозавершения, которое требует синтаксической проверки файла.	Выкл
Disable content assist auto-activation	Отключить автоматическую активацию автозавершения по последовательности вызывающих символов таких, как '.', '::' или '->'.	Выкл

Название элемента	Описание функционирования	По умолчанию
Skip trivial expressions in initializer lists	Пропускать тривиальные инициализаторы в списках инициализации при синтаксическом анализе. Это повышает производительность разбора и индексирования при наличии очень больших списков инициализации.	Вкл 1000
Maximum number of tokens per translation unit	Максимальное количество токенов на единицу трансляции.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.7. Настройки отображения исходного кода

Панель «**Syntax Coloring**» предназначена для того, чтобы настроить отображение исходного кода C/C++. Каждый элемент категории (код, комментарии и предпроцессор) содержит список элементов языка, которые могут отображаться в своем цвете и стиле. Для перехода на панель «**Syntax Coloring**» следует выбрать **C/C++ → Editor → Syntax Coloring**.

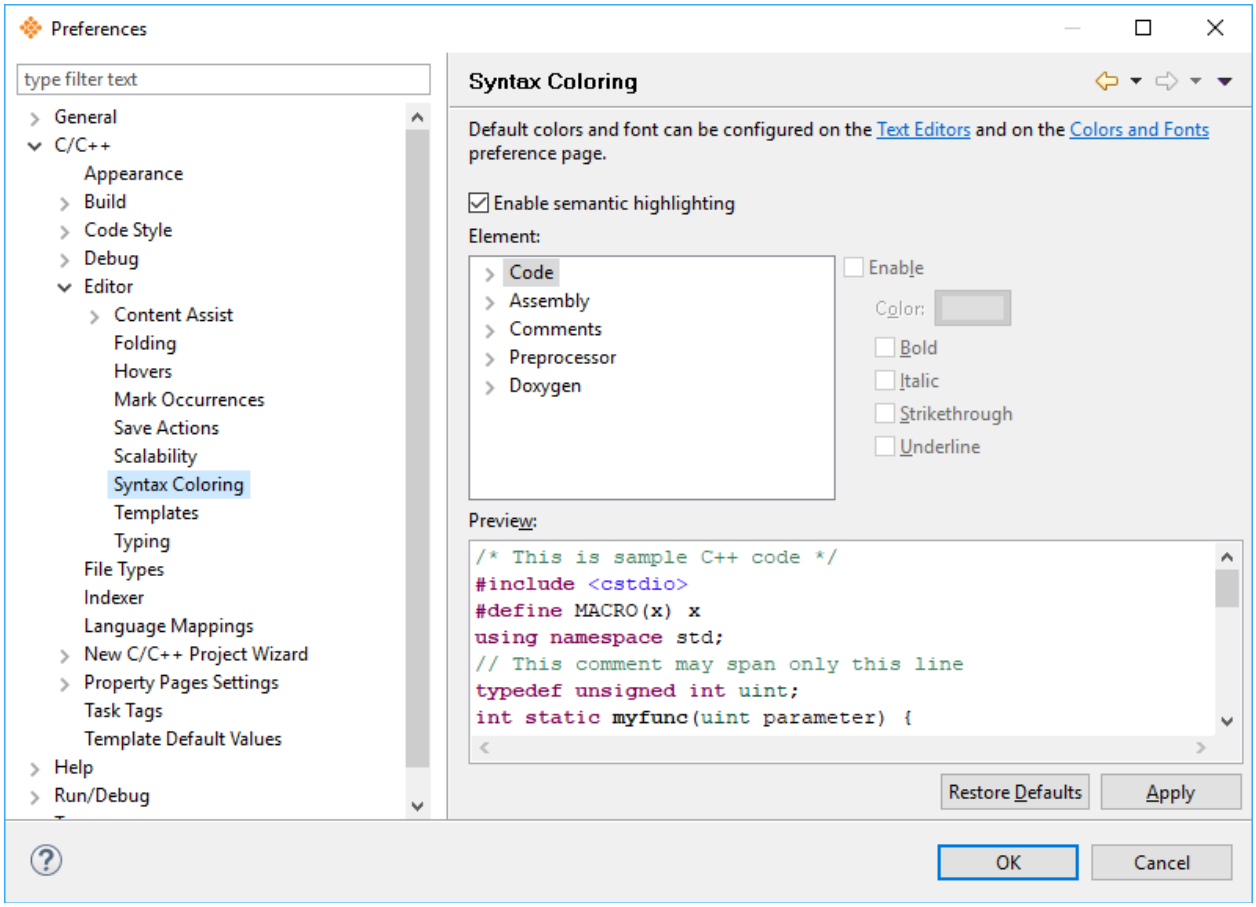


Рисунок 298. Панель «Syntax Coloring»

Таблица 144. Элементы управления панели «Syntax Coloring»

Название элемента	Описание функционирования	По умолчанию
Enable semantic highlighting	Включить/Отключить семантическое выделение некоторых структур в списке элементов.	Вкл
Enable	Включить/Отключить цветовую подсветку синтаксиса выбранного элемента.	Выкл
Color	Выбрать цвет, которым будет выделяться выбранный элемент.	Выкл
Bold	Выделить выбранный элемент жирным.	Выкл
Italic	Выделить выбранный элемент курсивом.	Выкл
Strikethrough	Выделить выбранный элемент перечеркиванием.	Выкл
Underline	Выделить выбранный элемент подчеркиванием.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.8. Настройки шаблонов

Панель «**Templates**» предназначена для управления всеми шаблонами: редактирования предустановленных или создания собственных. Для перехода на панель «**Templates**» следует выбрать **C/C++ → Editor → Templates**.

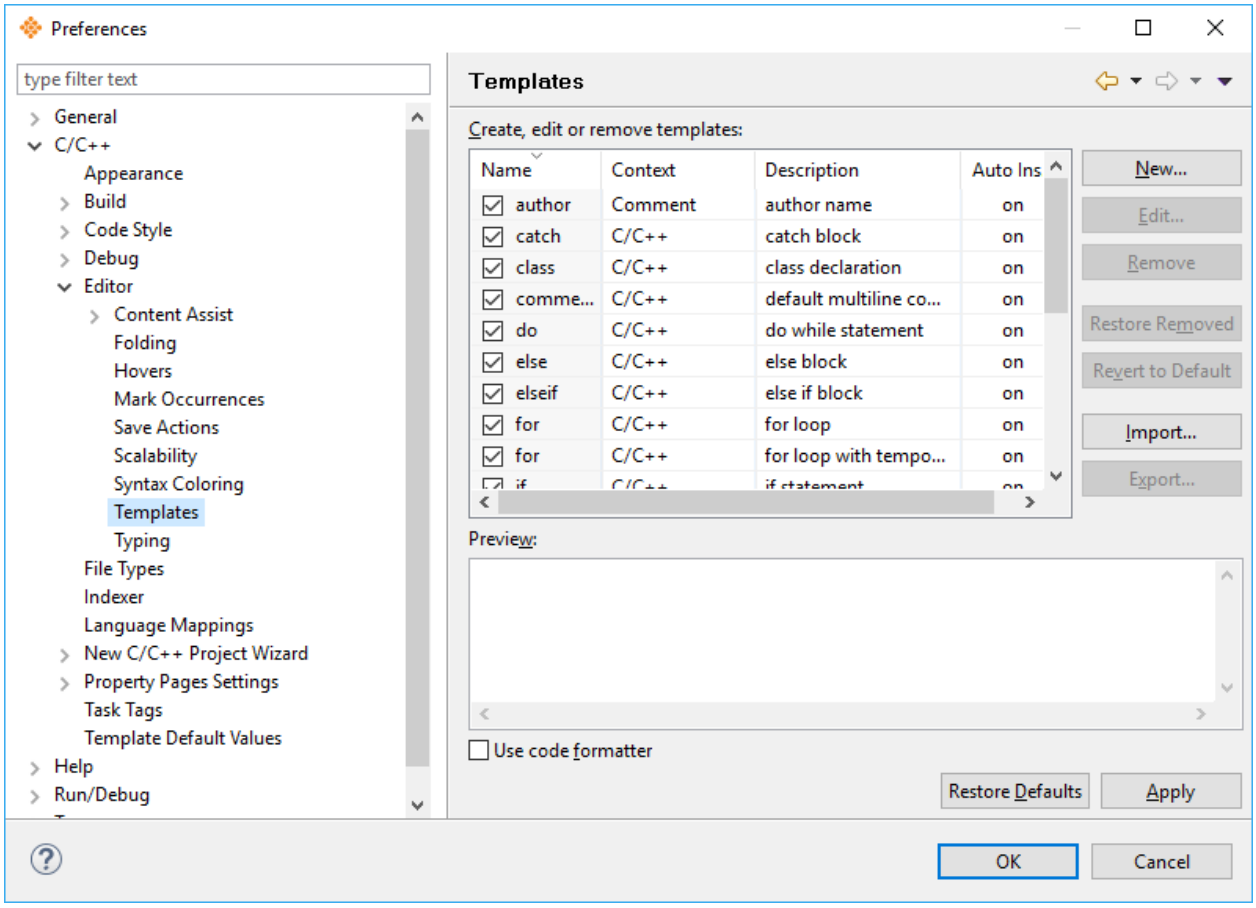


Рисунок 299. Панель «Templates»

Таблица 145. Элементы управления панели «Templates»

Название элемента	Описание функционирования	По умолчанию
New...	Открыть диалоговое окно «**New Template**» для создания нового шаблона.	–
Edit...	Открыть диалоговое окно «**Edit Template**» для редактирования выбранного шаблона.	–
Remove	Удалить выбранный шаблон из списка.	–
Restore Removed	Восстановить последний удалённый шаблон.	–
Revert to Default	Вернуть шаблон в изначальное состояние.	–
Import...	Импортировать шаблон.	–
Export...	Экспортировать выбранные шаблоны.	–
Use code formatter	Применить выбранный стиль кода к шаблону.	Выкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.5.9. Настройки форматирования

Панель «**Typing**» предназначена для настройки вспомогательных опций при форматировании вводимой информации при наборе текста. Для перехода на панель «**Typing**» следует выбрать **C/C++ → Editor → Typing**.

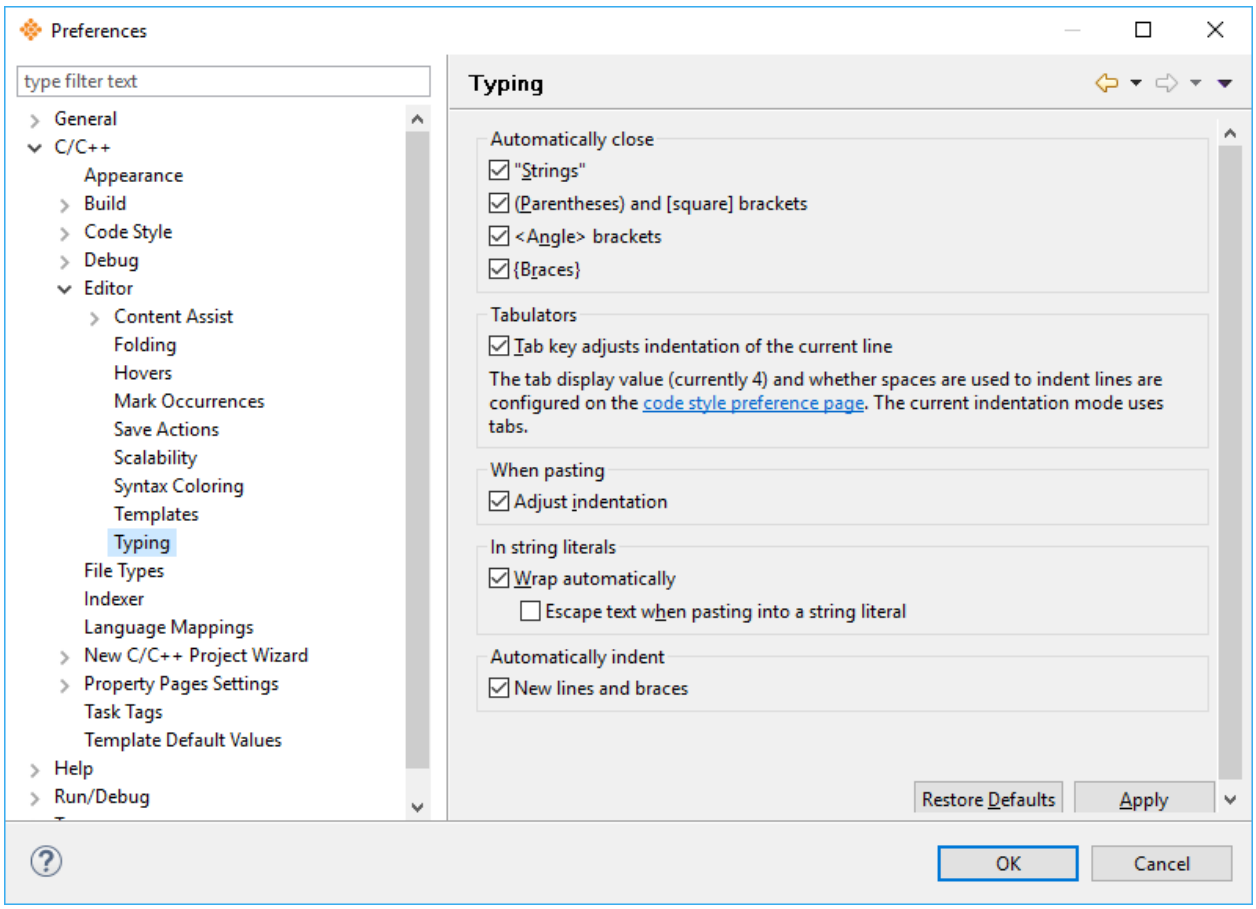


Рисунок 300. Панель «Typing»

Таблица 146. Элементы управления панели «Typing»

Название элемента	Описание функционирования	По умолчанию
"Strings"	Автоматически закрывать кавычки строки.	Вкл
(Parenthesis) and [square] brackets	Автоматически закрывать скобки и квадратные скобки.	Вкл
<Angle> brackets	Автоматически закрывать угловые скобки.	Вкл
{Braces}	Автоматически закрывать фигурные скобки.	Вкл
Tab key adjusts indentation of the current line	Отступить строку при нажатии клавиши **Tab**.	Вкл
Adjust indentation	Выровнять отступы вставленного текста в соответствии с текущим уровнем выравнивания.	Вкл
Wrap automatically	Включить автоматическое сворачивание строковых литералов, когда они превышают максимальную длину строки.	Вкл
Escape text when pasting into a literal string	Отключить специальные символы во вставленных строках при вставке в существующую строковую константу.	Выкл

Название элемента	Описание функционирования	По умолчанию
New lines and braces	Сместить новую линию или скобки вправо согласно настройкам Code Style. При снятом флажке применяются отступы по умолчанию (копируются с предыдущей строки).	Вкл

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.6. Настройки индексатора

Панель «**Indexer**» предназначена для выбора индексатора по умолчанию для новых проектов и настройки его поведения. Для перехода на панель «**Indexer**» следует выбрать **C/C++ → Indexer**.

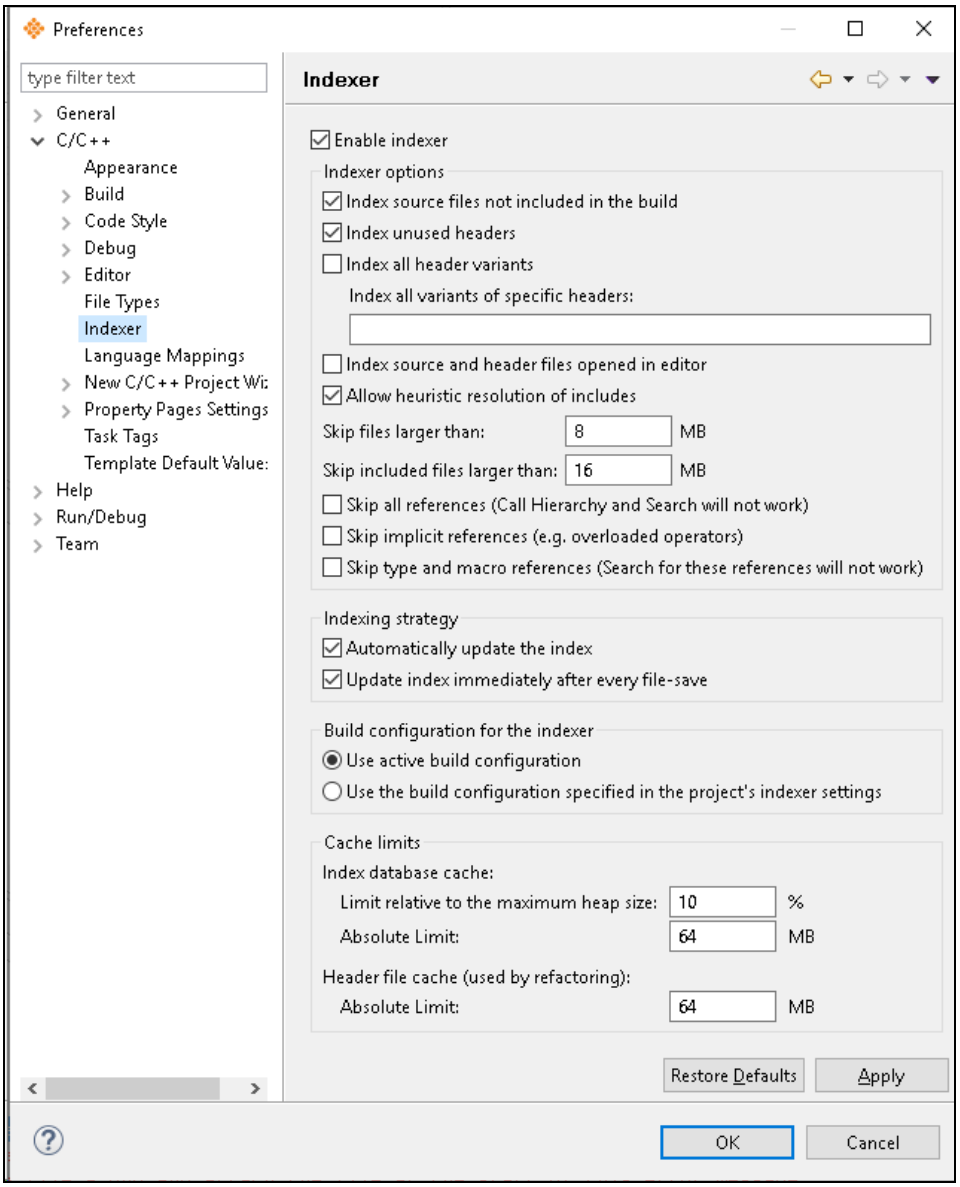


Рисунок 301. Панель «Indexer»

Таблица 147. Элементы управления панели «Indexer»

Название элемента	Описание функционирования	По умолчанию
Enable Indexer	Выбрать индексатор по умолчанию для всех новых проектов.	Вкл
Index source files not included in the build	Индексировать все исходные файлы, используемые в проекте.	Вкл
Index unused headers	Индексировать неиспользуемые заголовочные файлы.	Вкл
Index all header variants	Индексировать все варианты заголовков.	Выкл
Index all variants of specific headers	Индексировать все варианты конкретных заголовков.	–
Index source and header files opened in editor	Индексировать исходные и заголовочные файлы, открытые в редакторе.	Выкл
Allow heuristic resolution of includes	Разрешить индексатору пропускать при индексации повторяющиеся включаемые файлы.	Вкл
Skip files larger than	Указать максимальный размер файла для пропуска при индексации.	8 Мб
Skip included files lager than	Указать максимальный размер включенного файла для пропуска при индексации.	16 Мб
Skip all references (Call Hierarchy and Search will not work)	Разрешить не индексировать ссылки для экономии места и повышения производительности. Это отвлекает некоторые ресурсы от работы, такие как «**Call Hierarchy**» и поиск ссылок.	Выкл
Skip implicit references (e.g. overloaded operators)	Разрешить не индексировать косвенные ссылки.	Выкл
Skip type and macro references (Search for these references will not work)	Разрешить не индексировать макро или типовые ссылки. Это отвлекает некоторые ресурсы от работы, такие как «поиск».	Выкл
Automatically update the index	Автоматически обновить индекс по мере необходимости.	Вкл
Update index immediately after every file change	Принудительно обновить индекс при сохранении файла.	Вкл
Use active build configuration	Всегда использовать настройки активной конфигурации индексатора сборки, чтобы выполнить сборку индекса.	Вкл
Use the build configuration specified in the project's indexer settings	Всегда использовать настройки конфигурации, заданной в настройках проекта для сборки индекса.	Выкл
Index database cache: Limit relative to the maximum heap size	Указать относительный размер лимита, который индекс может достичь, основываясь на максимальном размере динамической памяти.	10

Название элемента	Описание функционирования	По умолчанию
Index database cache: Absolute limit	Указать максимальный размер кэша базы данных индекса.	64
Header file cache: Absolute limit	Указать максимальный размер кэша заголовочных файлов.	64

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.7. Настройки использования ассоциации

Панель «**Language Mappings**» предназначена для того, чтобы настроить использование ассоциации языков C/C++ для рабочего пространства. Для перехода на панель «**Language Mappings**» следует выбрать **C/C++ → Language Mappings**.

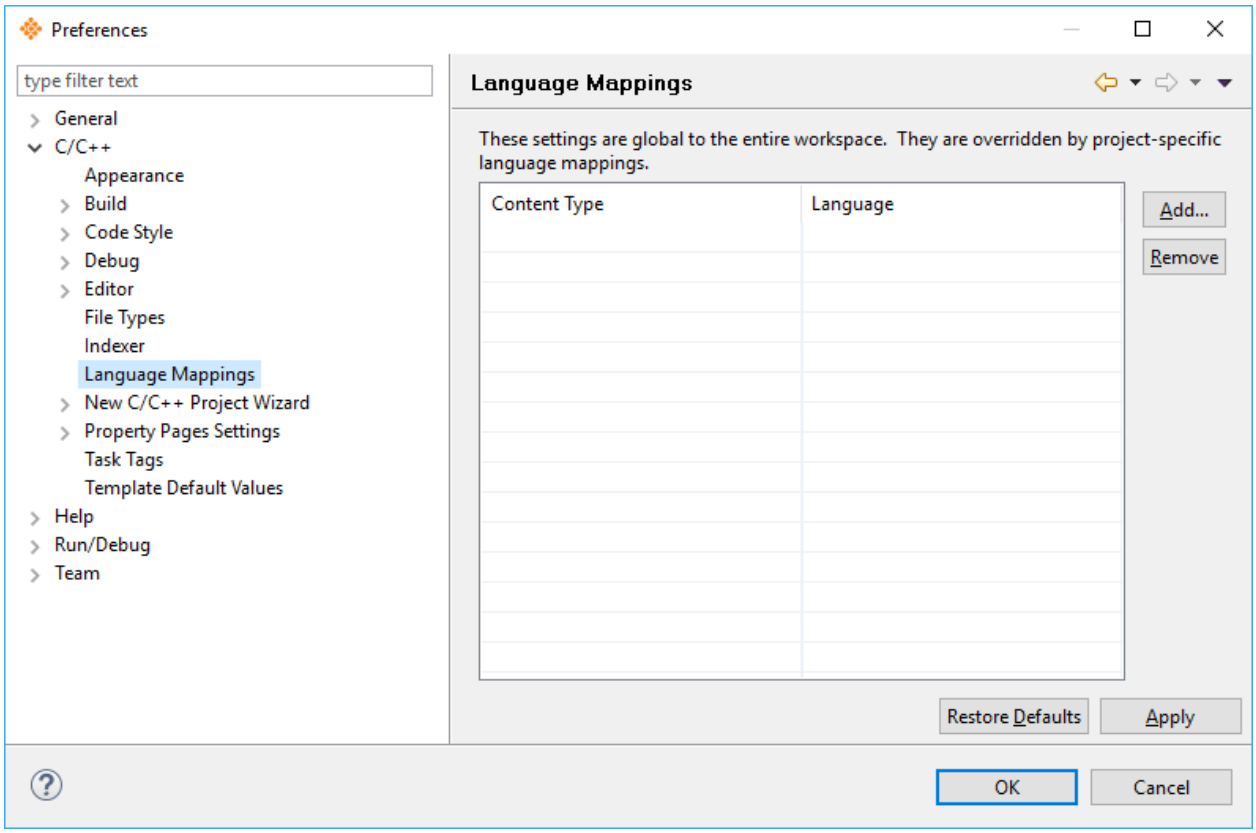


Рисунок 302. Панель «Language Mappings»

Таблица 148. Элементы управления панели «Language Mappings»

Название элемента	Описание функционирования	По умолчанию
Content Type	Задать тип содержимого, связанный с языком.	–
Language	Задать язык, связанный с типом содержимого.	–
Add...	Открыть диалоговое окно «**Add Mapping**», где можно определить тип содержимого и связанный с ним язык.	–

Название элемента	Описание функционирования	По умолчанию
Remove	Удалить выбранное отображение языка.	–

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.8. Настройки мастера создания C/C++ проектов

Панель «**New C/C++ Project Wizard**» предназначена для того, чтобы настроить внешний вид мастера создания C/C++ проектов. Для перехода на панель «**New C/C++ Project Wizard**» следует выбрать **C/C++ → New C/C++ Project Wizard**.

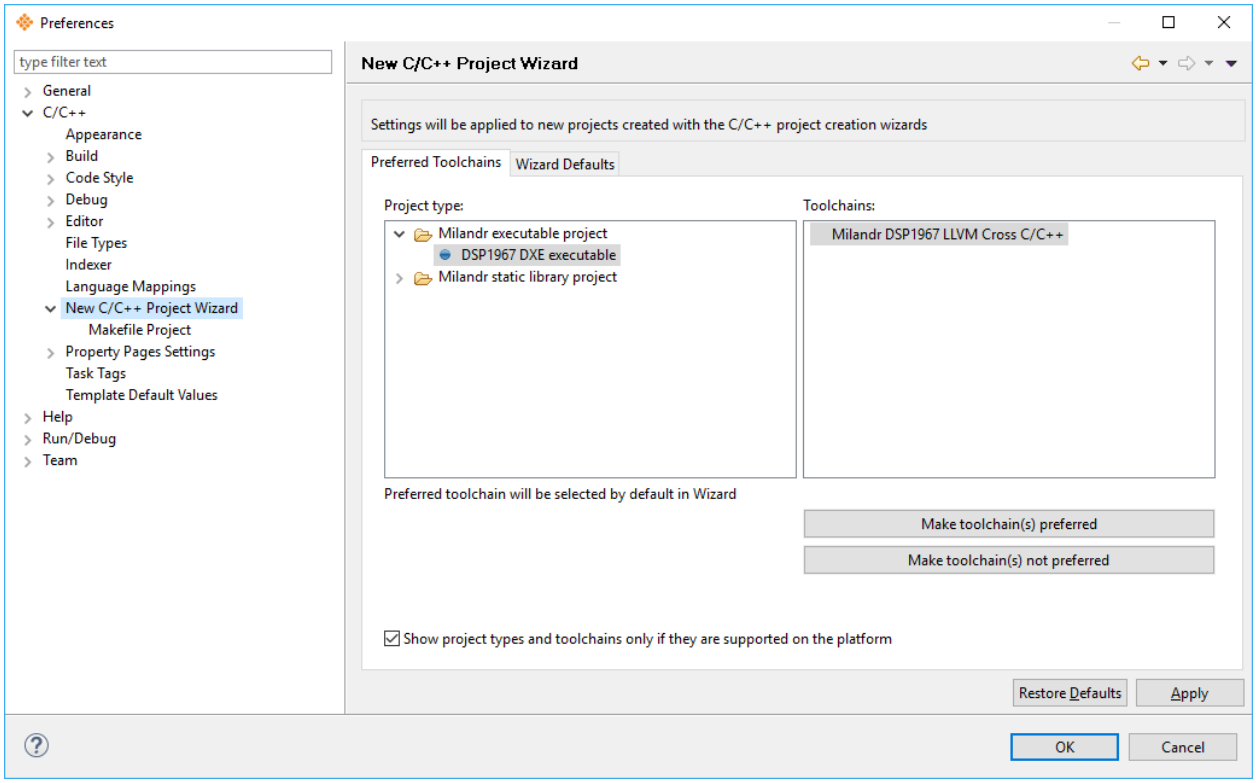


Рисунок 303. Панель «New C/C++ Project Wizard» (вкладка «Preferred Toolchains»)

Таблица 149. Элементы управления панели «New C/C++ Project Wizard» (вкладка «Preferred Toolchains»)

Название элемента	Описание функционирования	По умолчанию
Project types	Отображаются типы проектов, доступные в мастере создания C/C++ проектов.	–
Toolchain	Отображается набор инструментов (*toolchain*), доступный в мастере создания C/C++ проектов.	–

Название элемента	Описание функционирования	По умолчанию
Make toolchain(s) preferred	Делает предпочтительным выбранный набор инструментов. Предпочтительные наборы инструментов отмечены стрелкой (">"). В диалоговом окне мастера создания C/C++ проектов по умолчанию выбран предпочтительный набор инструментов. Если предпочтительными являются несколько наборов инструментов, то выделен первый из них. Если нет предпочтительных наборов инструментов, то выбран весь список.	–
Make toolchain(s) not preferred	Удалить из предпочитаемых выбранный набор инструментов.	–
Show project types and toolchains only if they are supported on the platform	Совпадает с элементом управления на странице мастера создания C/C++ проектов. Определяет, скрыты или показаны неподдерживаемые типы проектов и наборы инструментов.	Вкл.

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.9. Настройки окна «Properties» для ресурсов

Панель «**Property Pages Settings**» предназначена для того, чтобы настроить внешний окна «**Properties**» для ресурсов. Для перехода на панель «**Property Pages Settings**» следует выбрать **C/C++ → Property Pages Settings**.

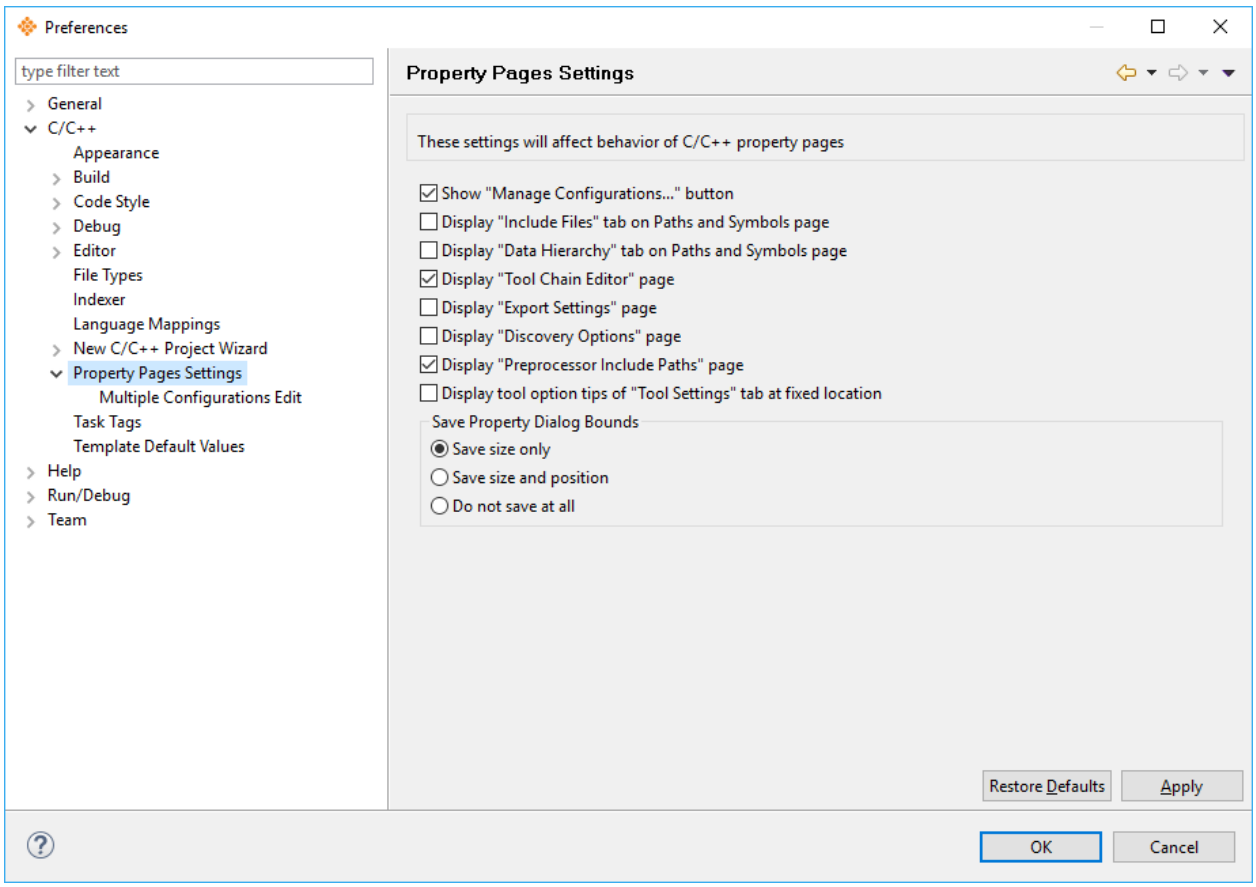


Рисунок 304. Панель «Property Pages Settings»

Таблица 150. Элементы управления панели «Property Pages Settings»

Название элемента	Описание функционирования	По умолчанию
Show 'Manage Configurations...' button	Показать/скрыть кнопку «**Manage Configurations...**», обычно размещенную на каждой странице свойств. Если эта кнопка скрыта, управление операцией можно получить из главного меню или контекстного меню.	Вкл.
Display 'Include Files' tab on Paths and Symbols page	Показать/скрыть вкладку «**Include Files**» на странице «**Paths and Symbols**», отображающую включенные по умолчанию файлы. Эти include файлы будут проанализированы перед исходным кодом, даже если они явно не включены в исходный файл.	Выкл.
Display 'Data hierarchy' tab on Paths and Symbols page	Показать/скрыть вкладку «**Data hierarchy**» на странице «**Paths and Symbols**», отображающую конфигурацию проекта. Эта опция полезна в основном для отладки, поэтому она отключена по умолчанию.	Выкл.
Display 'Tool Chain Editor' page	Показать/скрыть страницу «**Tool Chain Editor**». Обычно пользователям не нужно вручную редактировать наборы инструментов.	Вкл.
Display 'Export Settings' page	Показать/скрыть страницу «**Export Settings**». Обычно пользователям не нужно вручную редактировать параметры экспорта.	Выкл.

Название элемента	Описание функционирования	По умолчанию
Display 'Discovery Options' page	Показать/скрыть страницу «**Discovery Options**».	Выкл.
Display 'Preprocessor Include Paths' page	Показать/скрыть страницу «**Preprocessor Include Paths**».	Вкл.
Display tool option tips of 'Tool Settings' tab at fixed location	Показать/скрыть подсказки по выбору инструмента на вкладке «**Tool Settings**» в фиксированном месте.	Выкл.
Save Property Dialog Bounds	Определяет, как сохраняются параметры окна «**Properties**»: ▪ **Save size only** – положение определяется системой; ▪ **Save size and position** – в следующий раз диалоговое окно отображается в том же месте с тем же размером; ▪ **Do not save at all** – размер и положение устанавливается системой каждый раз, когда отображается диалоговое окно.	Save size only

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.2.9.1. Настройки объединения

Панель «**Multiple Configurations Edit**» предназначена для настройки объединения изменений при одновременном редактирование свойств для нескольких конфигураций проекта. Для перехода на панель «**Multiple Configurations Edit**» следует выбрать **C/C++ → Property Pages Settings → Multiple Configurations Edit**.

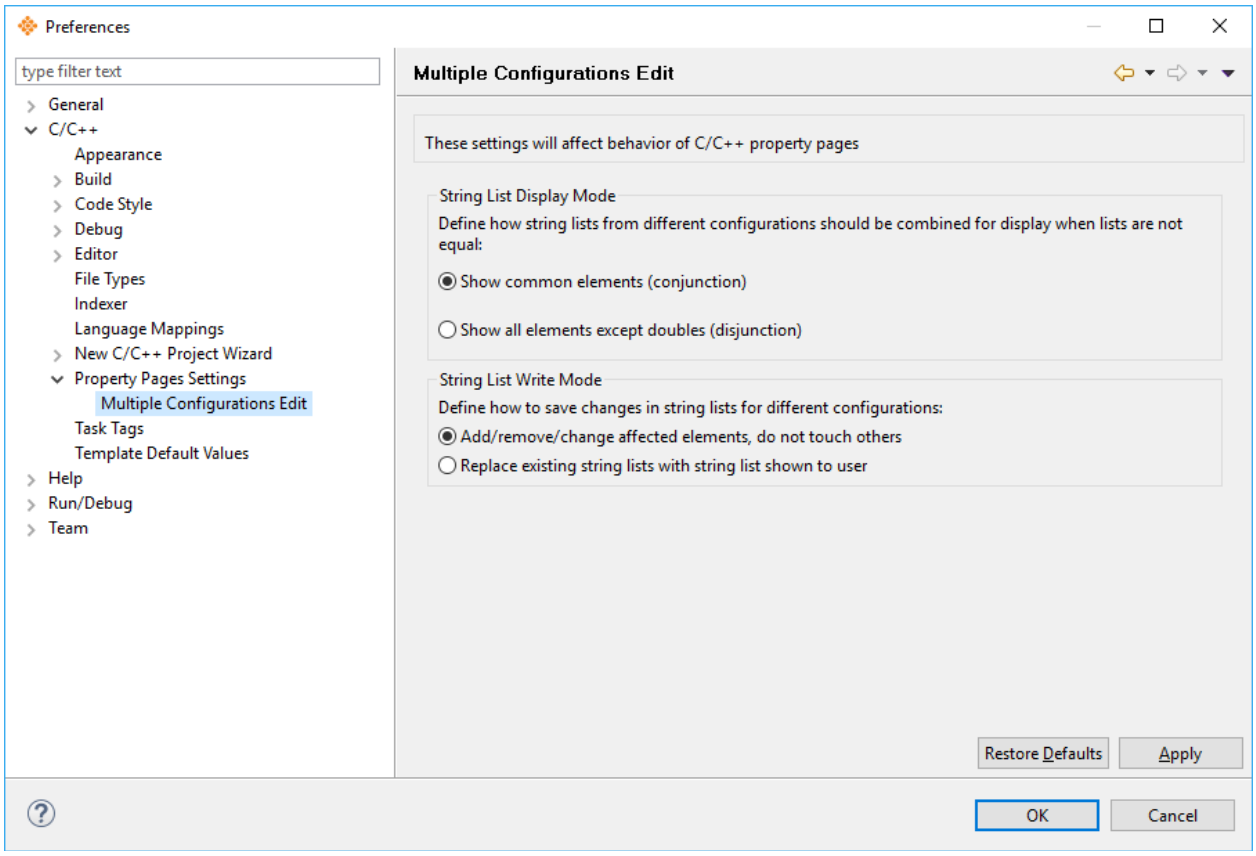


Рисунок 305. Панель «Multiple Configurations Edit»

Таблица 151. Элементы управления панели «Multiple Configurations Edit»

Название элемента	Описание функционирования	По умолчанию
String list Display Mode	Определяет, как строковые списки различных конфигураций должны быть объединены для отображения, когда списки различны. Возможны варианты: ▪ **Show common element (conjunction)** – показать все элементы (конъюнкция); ▪ **Show all elements except doubles** (**disjunction**) – показать все элементы, исключая дубли (дизъюнкция).	**Show common element (conjunction)**
String list Write Mode	Определяет вариант сохранения изменений в строковых списках для различающихся конфигураций. Возможны варианты: ▪ **Add/ removes/ change elements affected elements, do not touch other** – режим изменения; добавить/удалить/ изменить только выбранные пользователем элементы; ▪ **Replace existings string lists with string list shown user** – режим перемещения; отобразить видимый контент обеих конфигураций.	**Add/ removes/ change elements affected elements, do not touch other**

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.12.3. Настройки запуска и отладки

Панель «**Run/Debug**» предназначена для настроек запуска и отладки (см. Рисунок 306).

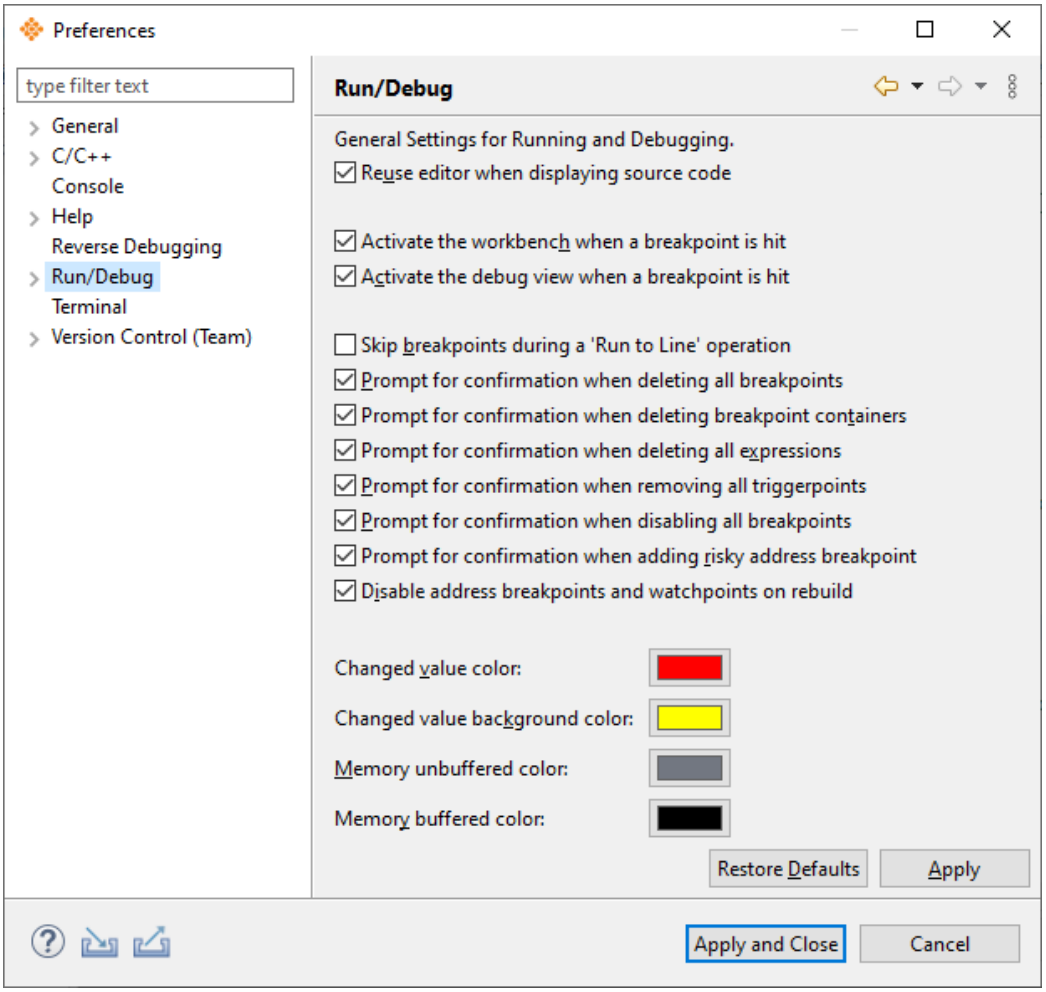


Рисунок 306. Панель «Run/Debug»

Таблица 152. Некоторые элементы управления панели «Run/Debug»

Название элемента	Описание функционирования	По умолчанию
Prompt for confirmation when adding risky address breakpoint	Определяет, запрашивать ли подтверждение при попытке установки адресной точки останова на не первый опкод в линии. См. также п. 6.11.7.	Вкл.
Disable address breakpoints and watchpoints on rebuild	Определяет, отключать ли все адресные точки останова и точки наблюдения в случае сборки проекта. См. также п. 6.11.7.	Вкл.

6.12.3.1. Настройки запуска

Панель «**Launching**» предназначена для настроек запуска. Для перехода на панель « **Launching** » следует выбрать **Run/Debug → Launching**.

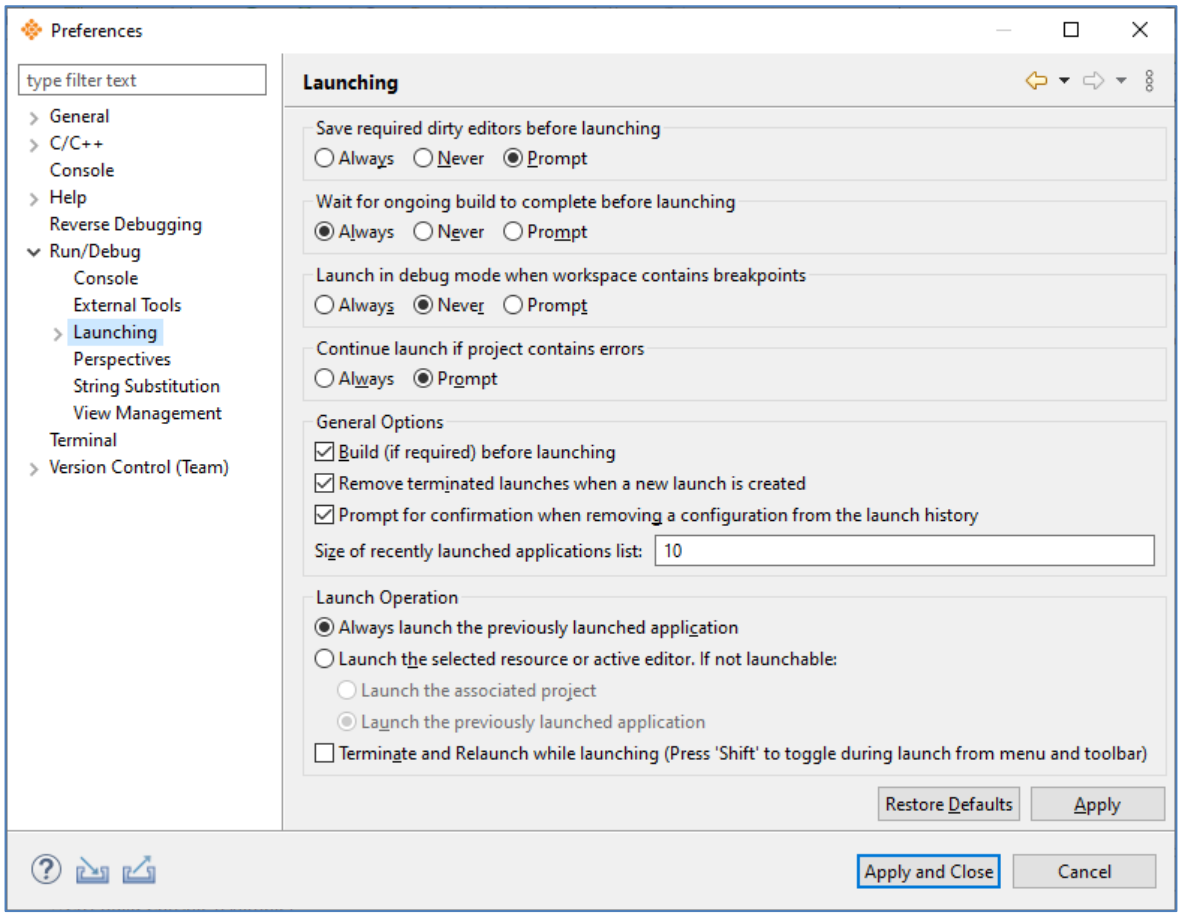


Рисунок 307. Панель «Launching»

Таблица 153. Некоторые элементы управления панели «Launching»

Название элемента	Описание функционирования	По умолчанию
Build (if required) before launching	Определяет, будет ли автоматически производиться сборка проекта перед запуском отладки. Если флаг выключен, то сборка не будет производиться совсем, даже при изменении файлов проекта.	Вкл.
Launch Operation	Определяет, какая конфигурация отладки будет запущена при нажатии на кнопку (Debug), см также п. 6.11.1.4. Возможные варианты: ▪ **Always launch the previously launched application** – всегда запускать предыдущую конфигурацию отладки; ▪ **Launch the selected resource of active editor. If not launchable:** – **Launch the associated project** – запускать только конфигурацию отладки, ассоциированную с текущим проектом; – **Launch the previously launched application** – запускать конфигурацию отладки, ассоциированную с текущим проектом, или предыдущую запущенную (если ассоциированной нет).	**Always launch the previously launched application**

Для того чтобы принять сделанные изменения, следует нажать кнопку «**Apply**».

Для того чтобы вернуться к настройкам по умолчанию, следует нажать кнопку «**Restore Defaults**».

6.13. Завершение работы

Для завершения работы с CM-LYNX следует нажать крестик в правом верхнем углу окна и подтвердить выход из приложения (см. Рисунок 308).



Рисунок 308. Подтверждение выхода из программы

Если при завершении одной из прошлых сессий работы в пользователь установил флаг «**Always exit without prompt**», то окно подтверждения более выводиться не будет.

7. Диагностические сообщения

Состояние работы ПО передаётся различными цветовыми сигналами индикатора на плате:

- Красный – красный индикатор загорается при включении питания и горит до установки связи с драйвером USB на ПК;
- Кратковременный красный – индикатор загорается красным на короткое время во время подачи **`Reset`** (в том числе во время инициализации обмена с целевым ПК);
- Мигающий жёлтый – индикатор начинает мигать жёлтым после установки связи с драйвером и до инициализации обмена с целевым ПК;
- Жёлтый – индикатор загорается жёлтым в останове, в частности сразу после инициализации обмена с целевым ПК;
- Зелёный – индикатор горит зелёным в режиме выполнения пользовательской программы.