
Разработка приложения для распределенных вычислений с использованием технологии RPC

Лабораторная работа

Ревизия: 0.1

История изменений

07.05.2010 – Версия 0.1. Первичный документ. Влад Ковтун

Содержание

История изменений	2
Содержание	3
Лабораторная работа 8. Разработка приложения для распределенных вычислений с использованием технологии Microsoft RPC	4
Вопросы	4
Постановка задачи	4
Цель	4
Задачи	4
Задание	5
Пример	5
Вывод	5
Требования	5
Методические указания для самостоятельной работы	6
Тестирование	6
Теоретические сведения	6
Алгоритм	6
Компиляция проекта	8
Установка	9
Безопасность	10
Устранение неполадок	10
Литература	11

Лабораторная работа 8. Разработка приложения для распределенных вычислений с использованием технологии Microsoft RPC

Вопросы

- Постановка задачи.
- Методические рекомендации.

Постановка задачи

Цель

1. Разработать консольное серверное приложение анализа логов web-сервера Apache с помощью технологии Microsoft RPC.
2. Разработать консольное клиентское приложение анализа логов web-сервера Apache с помощью технологии Microsoft RPC.

Задачи

1. Ознакомиться с технологией Microsoft RPC (Remote Procedure Calls) [2].
2. Ознакомиться с этапами разработки распределенного приложения с использованием Microsoft RPC [2].
3. Ознакомиться с языком IDL (Interface Definition Language) описания интерфейса вызовов удаленных процедур [17].
4. Ознакомиться с правилами оформления ACF (Application Configuration File) – файла описания конфигурации приложения [18].
5. Ознакомиться с последовательностью формирования stub (для сервера) и проху (для клиента) файлов исходного кода посредством MIDL компилятора [15, 16, 19].
6. Воспользоваться классическим консольным приложением, разработанным на лабораторной работе 2, для построения клиентского и серверного приложений для анализа логов web-сервера Apache.
 - 6.1. На основе файлов логов web-сервера, следует сформировать следующую статистику:
 - количество уникальных посетителей за каждый день.
 - наиболее популярные браузеры.
 - наиболее популярные ОС.
 - наиболее популярные страницы (ссылки).
 - наиболее активные пользователи (IP).
7. Разработать тестовое приложение, которое формирует все возможные тестовые комбинации ключей командной строки для консольного приложения.
8. Разработать тестовое приложение, которое формирует все возможные тестовые комбинации данных для формирования запросов к серверному приложению.
9. Произвести оценки различных характеристик клиентского приложения.
 - 9.1. Оценить производительность, при различном количестве заданий (файлов для анализа).
10. Произвести оценки различных характеристик серверного приложения.
 - 10.1. Оценить производительность, при различном количестве подключений клиентов от 1 до 10.
11. Оформить функциональную спецификацию на приложение. В функциональной спецификации обязательно указать, каким образом производится тестирование на корректность.
12. Оформить отчет к лабораторной работе.

Задание

Опишем основные ключи командной строки, которые следует реализовать для клиентского приложения.

Для управления клиентским приложением предлагается использовать ключи командной строки:

- /id:<filedir> – полный сетевой путь к директории, в которой хранятся файлы логов для анализа. Данный параметр является обязательным, если он не указан, то происходит вывод на экран соответствующего сообщения и подсказки по использованию данного приложения.
- /if:<filenames> - перечисление имен файлов логов через пробел, которые необходимо проанализировать. Файлы обязаны находиться в директории указанной с помощью ключа /id. Данный ключ позволяет осуществлять выборочный анализ файлов логов.
- /o:<filename> – полный путь к файлу, либо имя файла, который будет хранить отчет. Если данный параметр не указывается, то вывод производится на экран.
- /sa <servers IP addresses> - перечисление IP адресов серверов через пробел, которые будут производить чтение данных из файлов и их последующий анализ.
- /sp:<server port> - номер порта сервера, на который ожидается подключение клиентов. Согласно рекомендации Microsoft следует использовать диапазон портов [5001, 5021]. Следует отметить, что порты для серверного и клиентского приложения должны совпадать.
- /? - вывод информации о допустимых ключах командной строки.

Для управления серверным приложением предлагается использовать ключи командной строки:

- /sp:<server port> - номер порта сервера, на который ожидается подключение клиентов. Согласно рекомендации Microsoft следует использовать диапазон портов [5001, 5021].
- /cc:<max clients count> - максимально допустимое число клиентских подключений.
- /? - вывод информации о допустимых ключах командной строки.

Пример

Для анализа 3 файлов логов web-сервера Apache на клиентской стороне, результат анализа выводится в файл report.dat. анализ производится на удаленном сервере с IP адресом 192.168.1.10, порт подключения 5051. Отметим, что файловое хранилище расположено на публичном хранилище \\192.168.1.12\logs.

```
C:/>analyserclient.exe /id:\\192.168.1.12\logs /if access-www.1.log access-www.2.log access-www.3.log /sa 192.168.1.10 /sp:5051 /o:report.dat
```

На серверной стороне (удаленный сервер) следует запустить приложение с указанием порта, на который следует производить подключения клиентам, а также максимальное количество клиентских подключений.

```
C:/>analyserserver.exe /sp:5051 /cc:5
```

Вывод

Во время работы приложения клиентского и серверного приложений, рекомендуется выводить информацию о статусе приложения, а также о корректности его работы на консоль.

Другими словами, на серверной стороне следует анализировать, сколько процентов из каждого файла (запроса) – проанализировано.

Допускается перенаправление вывода консоли в текстовый файл, например:

```
C:/>analyserclient.exe /id:\\192.168.1.12\logs /if access-www.1.log access-www.2.log access-www.3.log /sa 192.168.1.10 /sp:5051 /o:report.dat >log.txt
```

Требования

- Архитектура приложения строится по модульному принципу.
- За основу принимается стандартная библиотека C++ (в случае разработки на C++).

- Для реализации распределенных вычислений следует использовать технологию Microsoft RPC либо аналогичную.
- Рекомендуется использовать защищенные ресурсы и указатели.
- Обязательным является обработка исключений.
- Исходный код обязан быть комментирован. Для C++ следует использовать нотацию `doxygen` [3].

Методические указания для самостоятельной работы

При подготовке к лабораторной работе необходимо:

- Ознакомиться с понятиями поддержки нескольких процессоров в ОС. Известно, что ОС Windows поддерживает Symmetric Multiprocessing (SMP) Model [1].
- Подключение библиотеки OpenMP к компилятору Microsoft Visual C++ [1].
- Ознакомиться с функциями ОС Windows, используемыми для поддержки нескольких процессоров [1].
- Ознакомиться с директивами библиотеки OpenMP [1].
- Ознакомиться с операторами OpenMP [1].
- Ознакомиться с типами данных OpenMP [1].
- Ознакомиться с переменными окружения OpenMP [1].
- Ознакомиться с функциями OpenMP [1].

Тестирование

Тестирование приложения осуществляется в несколько этапов:

Разработка Unit Test в рамках проекта. Данный подход позволит проверить корректность реализации алгоритма.

Воспользоваться тестовыми пакетными файлами для проверки корректности функционирования.

Разработать простейшее приложение, которое формирует большие тестовые пакетные файлы, которые затем использовать для тестирования основного приложения.

Теоретические сведения

Алгоритм

Коротко остановимся на алгоритме функционирования распределенного приложения.

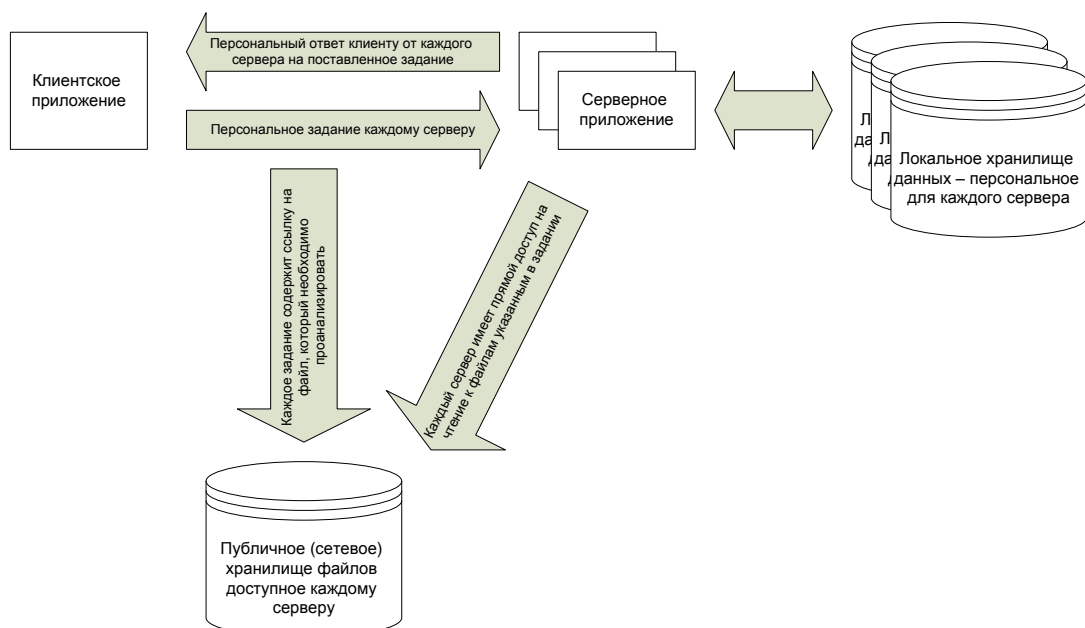


Рис. 1. Общая архитектура распределенного приложения

С целью упрощения, приведем учебное приложение, которое не подразумевает клиентской частью использования многих удаленных серверов.

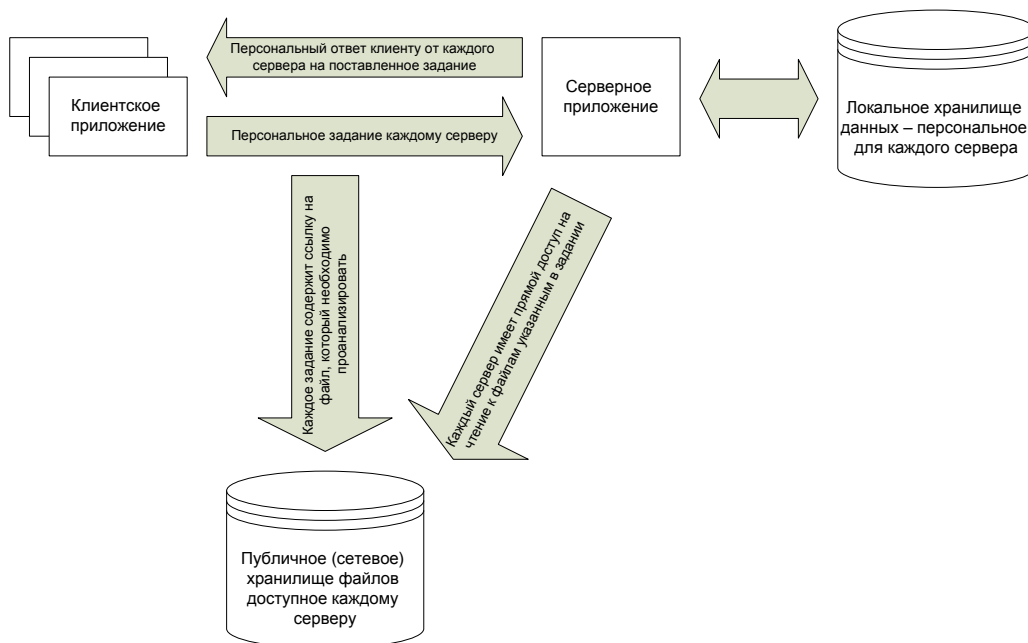


Рис. 2. Упрощенная архитектура распределенного приложения

Коротко остановимся на описании упрощенного (демонстрационного) распределенного приложения.

Клиентское приложение

1. Разбор ключей командной строки.
2. Подготовка структур данных для подключения к удаленному серверу.
3. Подготовка защищенной структуры данных для хранения результатов анализа.
4. Вызов удаленной процедуры – Инициализация подключения к серверу клиентом.
5. Для каждого файла-лога, который следует проанализировать.
 - 5.1. Вызов удаленной процедуры – Инициализация подключения.
 - 5.2. Вызов удаленной процедуры – Запуск задания, на сервере для анализа задания - файла-лога. Блокирующий вызов.
 - 5.3. Вызов удаленной процедуры – Получение размера результатов анализа. Блокирующий вызов.
 - 5.4. Подготовка временной структуры данных для получения результатов анализа с удаленного сервера.
 - 5.5. Вызов удаленной процедуры – Получение результатов анализа. Блокирующий вызов.
 - 5.6. Добавление полученных результатов из временной структуры данных в общий отчет.
 - 5.7. Уничтожение промежуточных (временных) структур данных для результатов анализа.
6. Вызов удаленной процедуры – Закрытие подключения.
7. Вывод результатов анализа, которые находятся в структуре результатов анализа.

Серверное приложение

Следует отметить, что каждый удаленный вызов выполняется сервером в отдельном потоке. Это, в свою очередь, требует использования механизмов синхронизации при доступе к глобальным переменным. В данном случае массив с управляющими объектами-клиентами.

1. Разбор параметров командной строки.
2. Подготовка структур данных для подключения удаленных клиентов к серверу.

3. Запуск вечного цикла обработки удаленных вызовов от удаленных клиентов.
4. Обработка удаленного вызова - Инициализации подключения.
 - 4.1. Создание управляющего объекта-клиента для удаленного вызова.
 - 4.2. Добавление вновь созданного управляющего объекта-клиента в глобальный массив.
5. Обработка удаленного вызова – Закрытие подключения.
 - 5.1. Уничтожение управляющего объекта-клиента.
 - 5.2. Удаление управляющего объекта-клиента из глобального массива.
6. Обработка удаленного вызова – Запуск задания.
 - 6.1. Открытие файла-лога для чтения. Файл-лога может находиться, как локально на компьютере сервера, так и на публичном хранилище данных. Чтение осуществляется в НЕ ЭКСКЛЮЗИВНОМ режиме. Проверка корректности открытия.
 - 6.2. Построчное считывание записей.
 - 6.2.1. Анализирует запись и заносит во временную структуру результатов.
 - 6.3. Закрытие файла-лога.
7. Обработка удаленного вызова – Получение размера результатов анализа.
 - 7.1. Получение размера результатов анализ и возврат этого числа клиенту.
8. Обработка удаленного вызова – Получение результатов анализа.
 - 8.1. Проверка размера структуры данных для приема результатов анализа.
 - 8.2. Заполнение структуры данных результатами анализа.
 - 8.3. Очистка временной структуры результатов.

Описание примера

В качестве примера, следует рассмотреть распределенное демонстрационное приложение WebAnalyserRPC. В приложение входит:

- WebAnalyser – проект со standalone приложением, которое анализирует указанные лог файлы и результаты заносит в report.txt.
- WebAnalyserRPC проект с интерфейсом распределенного приложения. Компилирует IDL и ACF файлы.
- WebAnalyserClient – проект клиентской части распределенного приложения.
- WebAnalyserServer – проект серверной части распределенного приложения.

Компиляция проекта

Для клиентского и серверного проектов следует обязательно установить дополнительную зависимость от библиотеки Rpcrt4.lib и Rpcns4.lib, см. рис. 3.

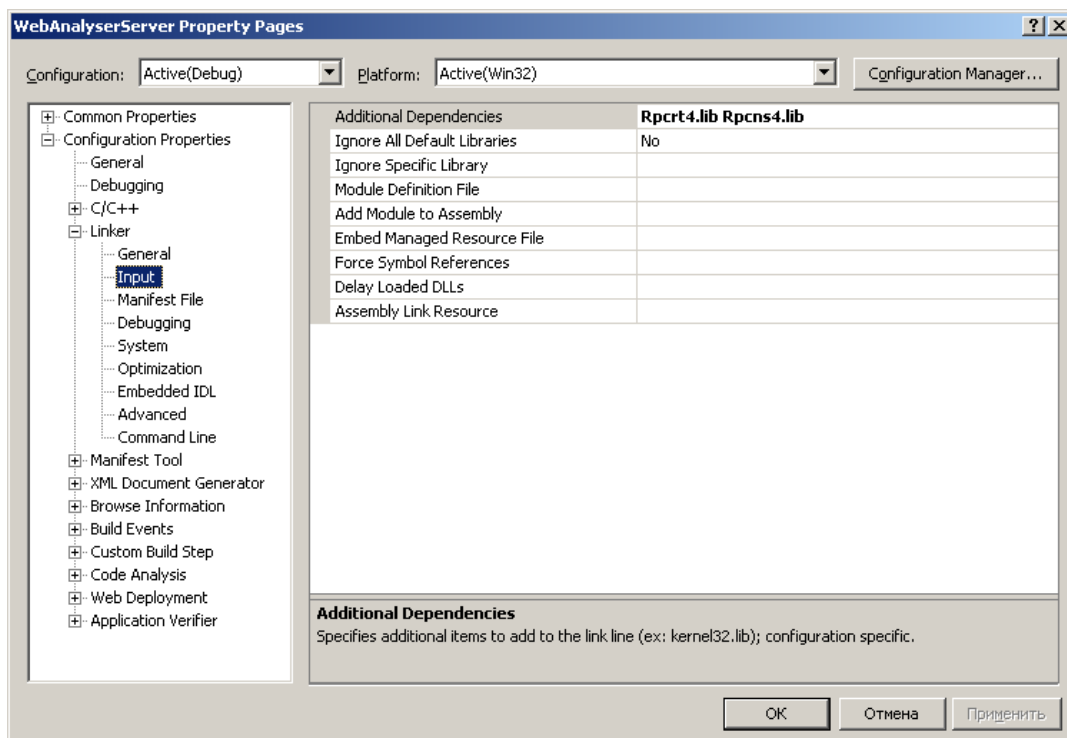


Рис. 3. Диалоговое окно настроек проекта

Установка

Предполагается, что распределенное приложение будет функционировать в рамках одной локальной сети.

Опишем особенности установки распределенного приложения:

- Развернуть публичное хранилище файлов (файловый сервер). Для этого достаточно сделать одну из папок любого компьютера общедоступной для чтения, в рамках сети. В эту папку следует скопировать все необходимое данные для анализа.
- Установить клиентское приложение на обычную рабочую станцию.
- Установить серверное приложение на каждый компьютер (сервер приложений), чьи вычислительные ресурсы предполагается задействовать для распределенных вычислений – анализа лог файлов web-сервера. Отметим, что предпочтительным является высокоскоростное сетевое соединение между серверами приложений и файловым сервером.

Отметим, что физически, клиентское и серверное приложения могут находиться на одном компьютере. Такой подход удобен на этапе разработки приложения.

При запуске клиентского приложения следует указать IP адреса и порты серверных приложений.

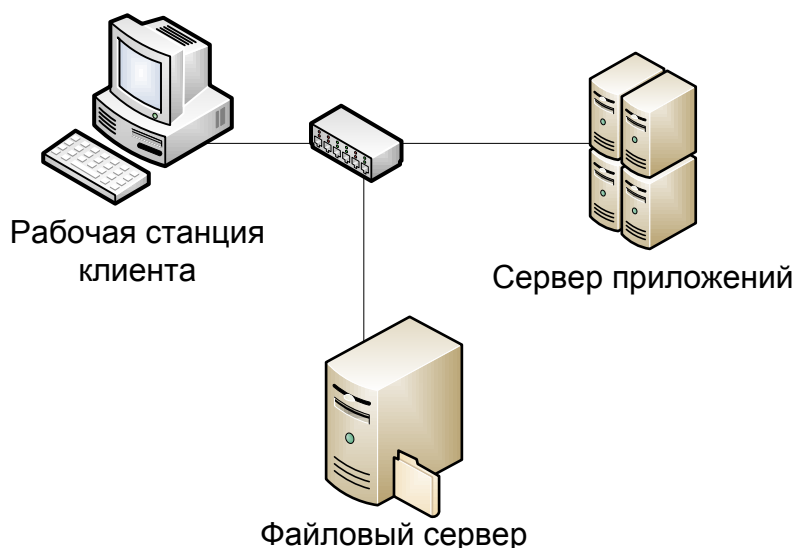


Рис. 4. Диаграмма размещения приложений

При установке распределенного приложения следует проверить наличие всех необходимых компонентов, которые перечислены в статье [11].

Безопасность

По умолчанию, в ОС Windows XP, начиная с Service Pack 2 присутствуют ограничения на удаленные вызовы RPC. Для включения возможности удаленных вызовов следует воспользоваться статьей [4]. На рис. 5 приведено окно редактора системного реестра Windows, где показан новый раздел RPC, который следует добавить, а также значения его содержимое – новые ключи со значениями: RestrictRemoteClients EnableAuthEpResolution. Более детальную информацию о настройках безопасности можно почерпнуть в статье [4].

Для ориентации в системных службах и сетевых портах Windows предлагается ознакомиться со статьей [13].

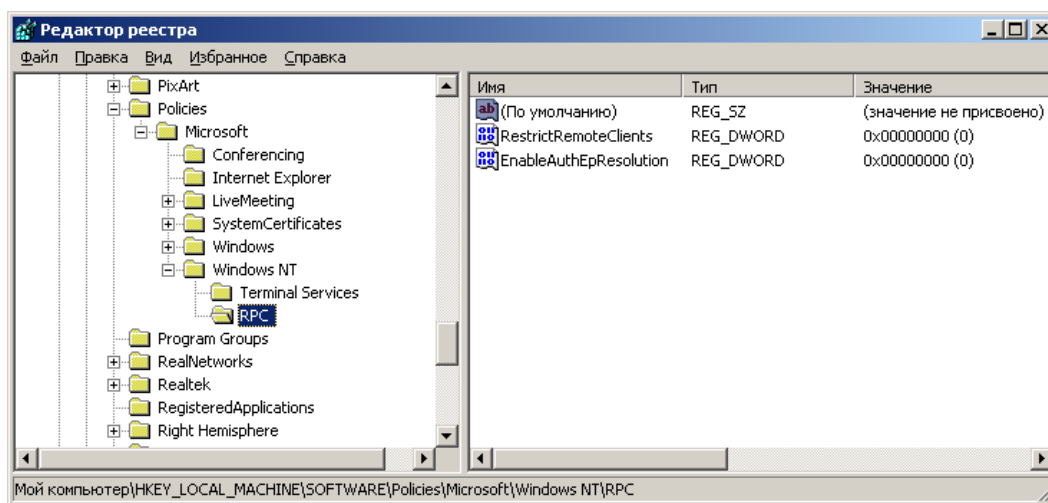


Рис. 5. Раздел RPC в реестре, а также значения атрибутов

Чтобы начать процесс настройки диапазона динамических портов RPC, загрузите средство настройки RPC (RPCCfg.exe) [9], а затем скопируйте его на рабочую станцию или сервер, который подвергнется изменениям. Описание процесса настройки подробно описан в работе [8].

Используйте политику протокола IPsec или брандмауэра, чтобы заблокировать доступ к уязвимым портам на соответствующем компьютере [8].

Для создания политики IPsec загрузите средство политик безопасности протоколов Интернета (Ipsecpol.exe) [10], а затем скопируйте его на рабочую станцию или сервер, который подвергнется перенастройке. Описание процесса настройки подробно описан в работе [8].

Совсем недавно Microsoft опубликовало бюллетень об уязвимостях связанных с реализацией механизма RPC [14], при реализации приложения следует учитывать информацию данного бюллетня.

Устранение неполадок

При устранении неполадок при развертывании распределенного приложения, следует воспользоваться набором инструментов Windows Resource Kit [6], информация по использованию инструментария доступна [5].

Перечень возможных неполадок, а также способов их устранения приведен в [7].

Для облегчения процесса отладки предлагается включить режим отладки распределенных приложений, как описано в [12], соответствующее диалоговое окно приведено на рис. 6.

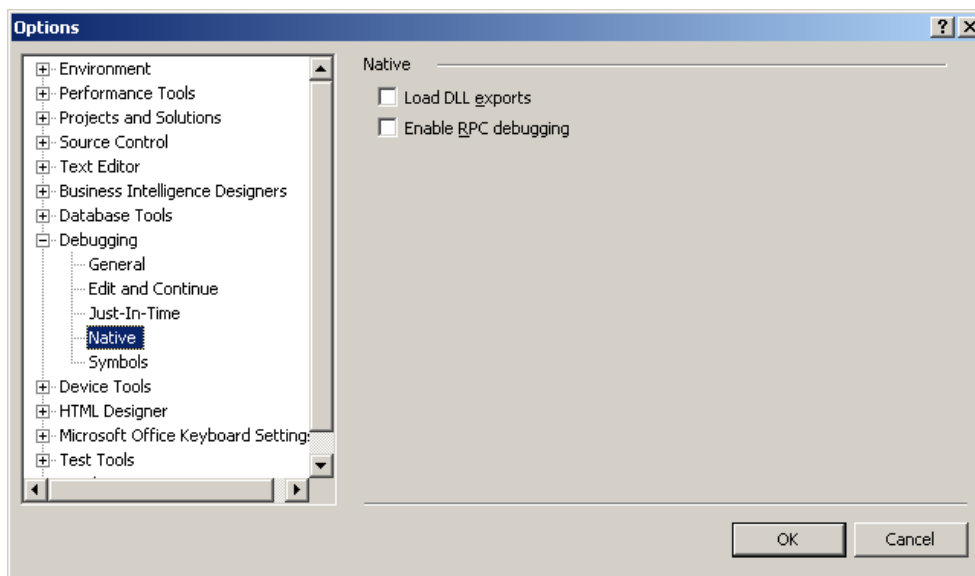


Рис. 6. Диалоговое окно настроек отладки в Microsoft Visual Studio

Для отладки приложений следует использовать возможность установки параметров командной строки через диалог настроек проекта, см. рис. 7.

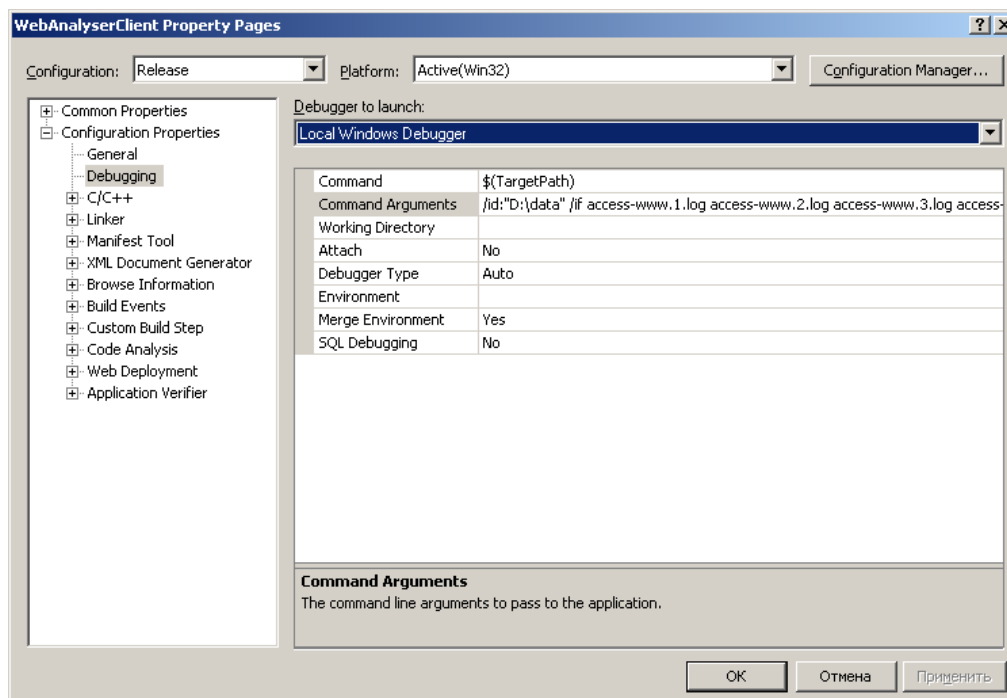


Рис. 7. Диалог настроек проекта

Например для перечисленных проектов распределенного приложения предлагается использовать следующие командные строки:

- WebAnalyser: /id:"D:\data" /if access-www.1.log access-www.2.log access-www.3.log access-www.4.log access-www.5.log access-www.6.log access-www.7.log access-www.8.log access-www.9.log access-www.10.log /o:report.txt
- WebAnalyserClient: /id:"D:\data" /if access-www.1.log access-www.2.log access-www.3.log access-www.4.log access-www.5.log access-www.6.log access-www.7.log access-www.8.log access-www.9.log access-www.10.log /sa localhost /sp:5051 /o:report.txt
- WebAnalyserServer: /sp:5051 /cc:10

Литература

1. Microsoft Developer Network. URL: <http://www.msdn.com>

2. Remote Procedures Call (RPC). URL: <http://msdn.microsoft.com/en-us/library/aa374358%28v=VS.85%29.aspx>
3. Формирование документации к исходному коду с помощью средства doxygen. URL: www.nrjetix.com/r-and-d/lectures
4. List of Remote Procedure Call (RPC) fixes in Windows XP Service Pack 2 and in Windows XP Tablet PC Edition 2005. URL: <http://support.microsoft.com/kb/838191>
5. Использование программы RPC Ping для устранения неполадок, связанных с установкой подключений компонентом «Exchange через Интернет» в Outlook 2007 и Outlook 2003. <http://support.microsoft.com/kb/831051/ru>
6. Windows Server 2003 Resource Kit Tools. URL: <http://www.microsoft.com/downloads/details.aspx?FamilyID=9d467a69-57ff-4ae7-96ee-b18c4790cffd&DisplayLang=en>
7. Устранение неполадок с установкой подключения, имеющих отношение к параметрам системного реестра, которые используются для управления протоколом RPC. URL: <http://support.microsoft.com/kb/325930/ru>
8. Настройка RPC для использования определенных портов и защита этих портов с помощью IPsec. URL: <http://support.microsoft.com/kb/908472/ru>
9. RPsccfg.exe URL: <http://www.microsoft.com/downloads/details.aspx?FamilyID=0f9cde2f-8632-4da8-ae70-645e1ddaf369&DisplayLang=en>
10. Ipsecpol.exe URL: <http://www.microsoft.com/downloads/details.aspx?displaylang=ru&FamilyID=7D40460C-A069-412E-A015-A2AB904B7361>
11. Доступные компоненты RPC. URL: <http://msdn.microsoft.com/ru-ru/library/bb499144%28WinEmbedded.51%29.aspx>
12. Включение режима отладки в Microsoft Visual Studio. URL: <http://msdn.microsoft.com/ru-ru/library/kf71skzd.aspx>
13. Службы и сетевые порты в серверных системах Microsoft Windows. URL: <http://support.microsoft.com/kb/832017/ru>
14. Бюллетень по безопасности Microsoft MS09-026 — существенный. Уязвимость в службе RPC делает возможным несанкционированное получение прав (970238). <http://www.microsoft.com/rus/technet/security/bulletin/MS09-026.msp>
15. Introduction to RPC – Part 1. URL: <http://www.codeproject.com/KB/IP/rpcintro1.aspx>
16. Introduction to RPC – Part 2. URL: <http://www.codeproject.com/KB/IP/rpcintro2.aspx>
17. Microsoft Interface Definition Language. URL: <http://msdn.microsoft.com/en-us/library/aa367091%28VS.85%29.aspx>
18. The Application Configuration File (ACF). URL: <http://msdn.microsoft.com/en-us/library/aa378704%28v=VS.85%29.aspx>
19. General Build Procedure. URL: <http://msdn.microsoft.com/en-us/library/aa373926%28v=VS.85%29.aspx>