

Разработка Java приложений для Lotus Notes

Разработка Java приложений для Lotus Notes с помощью пакета Lotus Domino Toolkit for Java/CORBA.

Дмитрий Томко

The Lotus Domino Toolkit for Java/CORBA представляет собой набор компонент и классов, включая документацию и примеры к ним, которые позволяют разрабатывать:

- Java программы, которые используют объекты Domino для Java для локального и удаленного доступ к данным и сервисам;
- Java программы с доступом к реляционным СУБД;
- Java и C++ программы, которые используют технологии CORBA и IIOP для доступа к удаленным Domino объектам;

Пакет объединяет в себе мощь технологий Java, Domino и CORBA, позволяя разрабатывать локальные и удаленные апплеты и приложения, не требуя специально предустановленного клиента Notes — для этого в состав пакета входят библиотеки классов Notes.jar и NCSO.jar. Набор классов поддерживает популярные среды и пакеты разработки для Java, в частности Sun JDK 1.1.6, IBM VisualAge for Java 3.0 (Win32) и версию сервера Notes выше 5.0.3. Мои тесты касались платформ Netware и Windows и версии сервера Notes 5.0.3.

Потенциально существует несколько подходов написания Java программ, работающих с Domino: агенты (разработанные с помощью средств Domino Designer программы, хранящиеся в базах Domino, которые могут быть запущены несколькими путями и задаются с помощью предопределенных действий, @-формул, LotusScript и Java кода), апплеты, приложения, и сервлеты.

- Также доступны следующие механизмы доступа:
- Объекты Domino для Java — позволяют манипулировать объектами Domino,

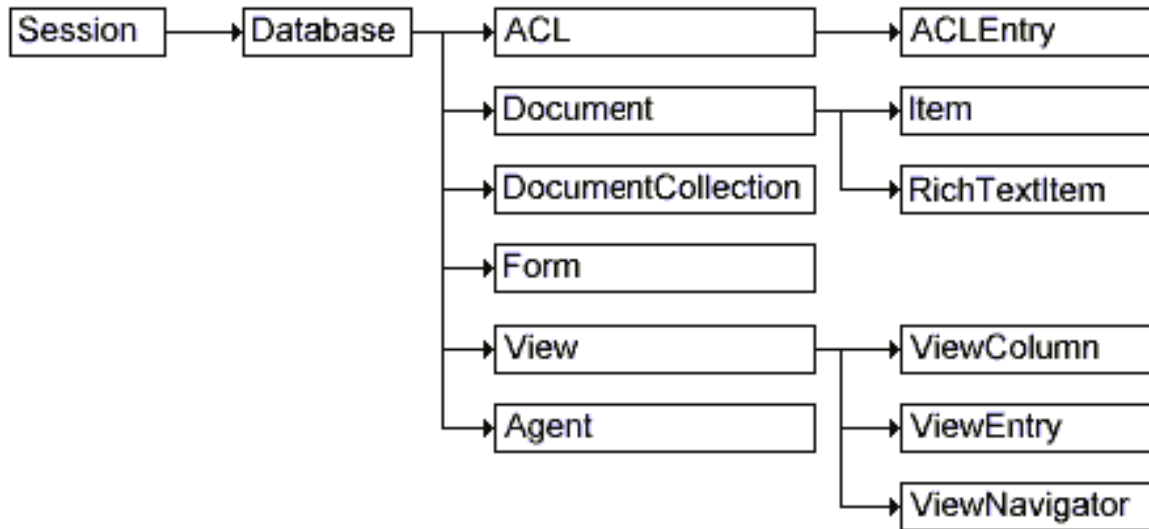
включая компоненты, базы данных, сервисы и т.д.

- Язык описания интерфейсов (Domino IDL).
- Java Native Interface (JNI) — используется для вызова C или C++ API функций и программ. Возможно использование API третьих фирм для доступа к объектам Domino
- Пакет LSX (пакет расширения Lotus Script) Java адаптер (Adapter). Пакет LSX позволяет разработчикам создавать объекты на C++ и использовать их из Lotus Script, а затем через Java Adapter вызывать эти объекты из Java
- Драйвера Lotus Domino для JDBC,
- JDBC драйвера третьих фирм (через использование пакета `java.sql.*`) — используются для доступа к базам данных, отличных от Domino.
- Библиотека классов Lotus Domino Connector, предоставляющая разработчику средства для доступа к данным, отличным от формата Domino.

В статье коснусь специфики написания удаленных Java программ для Lotus Notes, которые позволят пользователям, не имеющим клиента Notes получить доступ к данным на сервере Domino. Запуск удаленной Java программы похож на запуск локально, однако реально объекты находятся и запускаются на сервере Domino. Для этого ей предоставляются объекты `stub` (название заглушки для клиента) и `проху`. В отсутствие локального программного обеспечения Notes Domino объекты используют объектную архитектуру запросов CORBA. Объекты `stub` взаимодействуют (через объектный брокер запросов) с Domino сервером по протоколу IIOP.

Архитектура классов Domino объектов основывается на концептуальной контейнерной модели. Объект-контейнер всегда используется для доступа к содержащимся в нем объектам. Полная диаграмм классов достаточно громоздка, поэтому приведу основной каркас:

Разработка Java приложений для Lotus Notes



Тестовая программка, реализующая описанный подход:

```
//
// PROGRAM: LotusCorbaApp
// SYNTAX: LotusCorbaApp <host> <database> [<username> [<password>]]
// host The internet hostname or ip address of the server,
// for example, myserver.lotus.com
// database The database name, intro.nsf
//
// username Specify a Domino username (optional)
//
// password Specify the user's internet password, if any
import lotus.domino.*;

public class LotusCorbaApp implements Runnable
{
    // N&A Server record -
    // Net Address for TCPIP Port (or IP address)
    String host = null;
    String dbName = null;
    // N&A Person record - Short name and/or Internet address
    String userName = "";
    // N&A Person record - Internet password
```

```
String password = "";

public LotusCorbaApp(String argv[])
{
    host                = argv[0];
    if (argv.length >= 2) dbName    = argv[1];
    if (argv.length >= 3) userName = argv[2];
    if (argv.length >= 4) password = argv[3];
}

public static void main(String argv[])
{
    if(argv.length < 1)
    {
        System.out.println
        ("ERROR: You must supply server
         TCPIP Net (or IP) Address.");
        return;
    }
    if(argv.length < 2)
    {
        System.out.println
        ("ERROR: You must supply database filename.");
        return;
    }
    LotusCorbaApp CorbaApp = new LotusCorbaApp(argv);
    Thread nt = new Thread((Runnable)t);
    nt.start();
}

public void run()
{
    try
    {
        Session s = NotesFactory.createSession
            (host, userName, password);
        Name serverName = s.createName(s.getServerName());
        System.out.print("Connected to server " +
            serverName.getAbbreviated() + " as ");
        System.out.println(s.getCommonUserName());
    }
}
```

Разработка Java приложений для Lotus Notes

```
Database db = s.getDatabase(s.getServerName(), dbName);
System.out.print ("Title of database " +
                  db.getFilePath() + " is ");
System.out.println(db.getTitle());
}
catch (NotesException e)
{
    System.err.println
        (e.getClass().getName() + ": " + e.text);
    e.printStackTrace();
}
}
```

После запуска увидим следующее:

```
Connected to server proxyipa/ipa as Dmitry Tomko
Title of database homepage.nsf is homepage
```

Программа достаточно проста — создание сессии с удаленным сервером, связь с базой данных, вывод заголовка, плюс обработка исключений. Интересно, что механизм работы CORBA сокрыт достаточно хорошо и прозрачен для использования.

Усложним приложение, добавив доступ к почтовому файлу пользователя и чтение его почты — этакий прототип Notes клиента, реально работающий на любой JAVA платформе — а это уже кое-что.

```
//
// PROGRAM: LotusCorbaApp
//
// PURPOSE: Minimal Notes CORBA application
//
// SYNTAX: LotusCorbaApp [<username> [<password>]]
//
// host The internet hostname or ip address of the server,
// for example, myserver.lotus.com
//
// username Specify a Domino username (optional)
//
// password Specify the user's internet password, if any
import lotus.domino.*;
```

```
public class LotusCorbaApp implements Runnable
{
    // N&A Server record -
    // Net Address for TCPIP Port (or IP address)
    String host      = null;
    // N&A Person record - Short name and/or Internet address
    String userName = "";
    // N&A Person record - Internet password
    String password = "";

    public LotusCorbaApp(String argv[])
    {
        host          = argv[0];
        if (argv.length >= 2) userName = argv[1];
        if (argv.length >= 3) password = argv[2];
    }

    public static void main(String argv[])
    {
        if(argv.length < 1)
        {
            System.out.println
            ("ERROR: You must supply server
             TCPIP Net (or IP) Address.");
            return;
        }
        LotusCorbaApp t = new LotusCorbaApp(argv);
        Thread nt = new Thread((Runnable)t);
        nt.start();
    }

    public void run()
    {
        try
        {
            System.out.println();
            Session s = NotesFactory.createSession
                (host, userName, password);
            Name serverName = s.createName(s.getServerName());
        }
    }
}
```

Разработка Java приложений для Lotus Notes

```
System.out.print("Connected to Lotus Notes server " +
    serverName.getAbbreviated() + " as ");
System.out.println(s.getCommonUserName());
// (Get Mail Directory)
DbDirectory dir = s.getDbDirectory(s.getServerName());
Database db = dir.openMailDatabase();
DocumentCollection dc = db.getAllDocuments();
System.out.println("Mail database: " +
    db.getTitle() + " is " +
    ((int)(db.getSize()/1024)) + "KB long and has " +
    dc.getCount() + " documents");

//Get Documents
Document doc = dc.getFirstDocument();
while (doc != null)
{
    System.out.println();
    System.out.println("Your MailBox:");
    System.out.println("Subject: "+
        doc.getItemValueString("Subject"));
    System.out.println("Content: "+
        doc.getItemValueString("Body"));
    doc = dc.getNextDocument();
    System.out.println();
}
}
catch (NotesException e)
{
    System.err.println
        (e.getClass().getName() + ": " + e.text);
    e.printStackTrace();
}
}
```

Результаты работы программы:

```
Connected to Lotus Notes server proxyipa/ipa as Dmitry Tomko
Mail database: Dmitry Tomko is 6144KB long and has 1 documents
```

Your MailBox:
Subject: Tesing your Lotus
Content: Test Letter from Lotus Notes.

1. Заключение

Применение технологий Java, Domino Object for Java, CORBA позволяет объединить мощь Java — приложений с средствами групповой работы и почтовыми сервисами, предоставляемыми Lotus Notes — лидером на рынке средств groupware. Возможные приложения данных технологий обеспечивают им широкий рынок применения благодаря поддержке большого количества платформ и разнообразию механизмов разработки.