Archived:Java no Maemo

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

Acredita-se que este artigo ainda seja válido no contexto original (quando ele foi escrito)

Resumo

Este artigo descreve como usar Java no Maemo.

Introdução

A Nokia não disponibiliza java oficialmente na sua plataforma maemo, ainda. Claro que é tecnicamente possível desenvolver qualquer tipo de aplicações em C/GTK. Mas habilitar o *Internet Tablet* para executar programas Java oferece possibilidades além da linguagem de programação: Java é a linguagem mais usada para desenvolver aplicações móveis de terceiros, e a plataforma Java é a número um, sem rivais, para o desenvolvimento de aplicações corporativas do lado servidor.

Habilitar o *Internet Tablet* para executar Java significa habilitar a larga audiência de desenvolvedores Java a espalhar suas idéias para *Internet Tablets* e usar a mesma tecnologia para o desenvolvimento de aplicações clientes como para servidores.

Este artigo aborda o desenvolvimento para maemo usando a plataforma Jalimo. Além de Jalimo, existem outras abordagens para trazer a plataforma Java para maemo. Em contraste a outras, uma característica única de Jalimo é não apenas concentrar em um ambiente de tempo de execução para o dispositivo, mas também suportar o ciclo completo de vida e desenvolvimento de aplicações. Resumindo, o objetivo de Jalimo é prover tudo que um desenvolvedor Java precisa para se sentir confortável ao desenvolver para Linux embarcado e móvel.

Acesso ao dispositivo

Se você começa a desenvolver para seu *Internet Tablet*, primeiro você deve conseguir acesso ao *shell*. Para sua conveniência pode acessar o pequeno maemo tutorial How Do I Become Root. Após isso, você está hábil a se conectar no dispositivo usando ssh (por exemplo, usando PuTTY no Windows). Com ssh, você também tem uma solução muito confortável para troca de arquivos com o dispositivo, usando o comando scp (veja WinSCP no Windows).

Para ter controle total sobre todos os pacotes instalados no seu dispositivo, você deve habilitar o Red Pill mode do gerenciador de aplicações. Isto não é necessário se você prefere usar o comando *apt-get* ao invés de ssh.

Jalimo

O ponto central para acessar informações sobre o Jalimo é www.Jalimo.org. Além de maemo 3 e maemo 4, Jalimo também suporta OpenMoko e outras distribuições móveis e embarcadas. Para cada distribuição suportada existe um repositório com pacotes pré-compliados.

Os pacotes Jalimo contém a máquina virtual cacao, junto com *GNU Classpath*. Esta combinação se alinha com capacidades de Java 1.5 SE, mas até o momento não provê *Swing/AWT* no maemo. Até o momento, a melhor solução *GUI* Java disponibilizada no maemo é Eclipse SWT, portanto somente isto é coberto neste artigo. A escolha por *SWT* é porque ela é muito poderosa e seus pontos GTK rápidos, que se juntam com o *look and feel* da plataforma. Além disso, é uma das soluções mais portáveis para Java (executando em Linux, Windows, bem como Windows Mobile e Symbian S80 na sua variante eSWT).

Instalando os Pacotes

Para instalar os pacotes e verificar que eles estão funcionando corretamente, simplesmente aponte seu *Internet Tablet* para este **link de instalação** da aplicação exemplo Jalimo. Alternativamente você pode fazer isso manualmente, seguindo esses passos:

Adicione o repositório de pacotes Jalimo, usando o gerenciador de aplicações:

Repositório:	*http://jalimo.evolvis.org/repository/maemo/*
Distribuição:	*maemo3.2* ou *chinook* para maemo 4.x

(C) Copyright Nokia 2014. All rights reserved.

Componente:	*user*

Após isso, você pode instalar os pacotes de sua escolha (usando o gerenciador de aplicações ou o *shell*). Instalar o pacote *libswt-gtk-3.4-java* também instalará *cacao* bem como *classpath*, visto que depende deles. O exemplo acima está no pacote *jalimo-swt-example*. Para instalar todos juntos em um "shell", simplesmente execute o seguinte como *root*: <source lang=bash> echo "deb http://jalimo.evolvis.org/repository/maemo/ chinook user" >> /etc/apt/sources.list apt-get update && apt-get install jalimo-swt-example </source>

Ou, para seu IT OS 2007: <source lang=bash> echo "deb http://jalimo.evolvis.org/repository/maemo/ maemo3.2 user" >> /etc/apt/sources.list apt-get update && apt-get install jalimo-swt-example </source>

Primeiro Desenvolvimento

Para seu desenvolvimento Java você pode escolher sua *IDE* preferida (por exemplo, Eclipse pode ser uma boa escolha). Obviamente, também é aceitável iniciar com um editor de texto simples e qualquer compilador Java (por exemplo, o *javac* da Sun). Salve o código exemplo em HelloWorld.java

```
public class HelloWorld {
 public static void main(String[] args) {
                System.out.println("Hello Jalimo");
 }
}
```

e compile ele com *javac HelloWorld.java*. Copie o *HelloWorld.class* para seu *Internet Tablet* e execute

```
cacao HelloWorld
```

Aí está! Executando seu próprio código no *Internet Tablet* em apenas alguns minutos, usando uma linguagem de programação moderna sem a necessidade de instalar grandes *toolchains* ou fazer compilação cruzada.

Desenlovimento Sério

Para desenvolvimento sério você vai precisar obviamente de mais do que uma aplicação de console. Jalimo provê diferentes soluções para desenvolvimento de GUI no maemo. Primeiro, existe suporte a SWT, como ilustrado no exemplo seguinte.

Aplicação SWT Simples

```
import org.eclipse.swt.SWT;
import org.eclipse.swt.layout.RowLayout;
import org.eclipse.swt.widgets.*;

public class SimpleSwtSample {

 public static void main(String[] args) {
  Display display = Display.getDefault();
  final Shell shell = new Shell(display);
  shell.setLayout(new RowLayout(SWT.VERTICAL));

  Label label = new Label(shell, SWT.CENTER);
  label.setText("Hello maemo");

  Button button = new Button(shell, SWT.NONE);
  button.setText("close");
  button.addListener(SWT.Selection, new Listener() {
   public void handleEvent(Event arg0) {
    shell.dispose();
   }
```


(C) Copyright Nokia 2014. All rights reserved.

```
  });

  shell.open();
  while (!shell.isDisposed())
   display.readAndDispatch();
 }
}
```

Compile este código simples com as bibliotecas SWT do Eclipse usando qualquer complicador Java. Você pode pegar as bibliotecas da página da SWT para plataformas diferentes. Execute a aplicação no seu computador. Copie todos os arquivos *.class* para seu dispositivo habilitado para Jalimo (por exemplo, usando scp ou WinSCP no Windows) e execute

```
cacao -cp /usr/share/java/swt-gtk.jar SimpleSwtSample
```

Você pode encontrar este exemplo junto com outro um pouco maior no svn. Este exemplo completo contém uma descrição completa de empacotamento e integração de menus.

Empacotamento Debian

Obviamente um aplicação precisa de um empacotamento. Para maemo isto significa empacotamento em arquivos debian (.deb). A maioria de desenvolvedores Java não é familiar com integração de distribuições linux. Portanto, foi criado um plugin de empacotamento para apache maven (o sistema padrão de construção para Java). Para usar este *plugin*, simplesmente crie um projeto maven e adicione o repositório do *plugin evolvis*, bem como a configuração de empacotamento para seu pom.xml. Você pode encontrar um exemplo aqui ou no exemplo SWT acima. Após isso, simplesmente execute

```
mvn pkg:deb
```

no seu projeto para ter um arquivo .deb. Um dos benefícios do *plugin* é a capacidade de traduzir as dependências maven para dependências de pacote debian.

Para ter uma entrada de menu maemo para sua aplicação, simplesmente crie o arquivo .desktop e adicione ele na descrição de recursos na configuração de empacotamento do *plugin*, com o local alvo */usr/share/applications/hildon/*. Se você quer que sua aplicação seja instalada por usuários normais usando o gerenciador de aplicações do maemo, você deve escolher *user/<SUBSECTION>* como seção de pacote debian configurado. Veja o Application Manager How To para mais detalhes.

Leituras Futuras

Obviamente, Jalimo tem outros aspectos como integração com *scratchbox* e SCIO, a biblioteca de comunicação XML para dispositivos pequenos. Dê uma olhada na página do Jalimo para mais informações.

(C) Copyright Nokia 2014. All rights reserved.