

	 DAVE s.r.l. www.dave.eu		
		VERSION:	1.0.0
		DOCUMENT CODE:	DNZELK001
		NO. OF PAGES:	7

Zefeer



DN-ZELK-001

Java native applications on Zefeer



Printed in Italy

Trademarks

Ethernet® is a registered trademark of XEROX Corporation

All other trademarks are the property of their respective owners

Copyright

All rights reserved. Specifications may change any time without notification.

Company Address

DAVE S.r.L.

Via Forniz 2

33080 Porcia (PN) – Italy

Phone: +39 0434 921215

e-mail: info@dave.eu

URL: www.dave.eu

Technical Support (available to registered kit owners only)

e-mail: support-zefeer@dave.eu

<i>History</i>		
1.0.0	June 2009	First release

1 - Introduction

This design note illustrates how to build a Java native application on the Zefeer platform.

The tool used to compile the application is gcj (<http://gcc.gnu.org/java/>), a free software compiler which is part of the GNU Compiler Collection (GCC, <http://gcc.gnu.org/>). One of the most important features of gcj is the possibility to compile Java sources directly to native machine code.

2 - Building the cross-toolchain

The key point is the cross-toolchain. Let's suppose that the host system is a x86 Linux environment, while the target is Dave's Zefeer (ARM 9) platform. One of the most used methods to build a cross-toolchain is through crosstool (<http://www.kegel.com/crosstool/>), which is a set of scripts that automatically download, patch, build, and test binutils, gcc, and glibc cross-toolchains.

The steps for building the cross-toolchain are the following:

1. Choose the components versions
2. Configure the environment variables for crosstool
3. Apply the necessary crosstool patches
4. Launch crosstool scripts

Step 1 is crucial: not all the versions of gcc, glibc, binutils and linux kernel headers can build a working toolchain. The developer must check the crosstool build results matrix (<http://kegel.com/crosstool/crosstool-0.43/buildlogs/>) and select the components versions. In this case, we choose gcc-4.0.2 and glibc-2.3.6.

Step 2 consists in setting the following environment variables:

```
export MY_GCC_VERSION=4.0.2
export GCC_LANGUAGES=c,c++,java
export GCC_EXTRA_CONFIG='--disable-libgcj-debug --disable-java-awt
--disable-interpreter --with-softfloat-supp --without-fp'
export GLIBC_EXTRA_CONFIG='--without-fp'
export SRC_DIR=/home/gcj/build/$MY_GCC_VERSION/gcc-src
export GCC_HOST=
export BUILD_DIR=/home/gcj/build/$MY_GCC_VERSION/gcc-$MY_GCC_VERSION-
build
```



```
export PREFIX=/home/gcj/build/$MY_GCC_VERSION/gcc-install
export RESULT_TOP=/home/gcj/build/$MY_GCC_VERSION/gcj/gcc-install
export TARBALLS_DIR=/home/gcj/build/$MY_GCC_VERSION/src/gcc-src
```

Step 3 consists in copying all the necessary patch files to the crosstool `patches/gcc-4.0.2` directory

Step 4 consists in launching the crosstool script:

```
eval `cat arm-softfloat.dat gcc-4.0.2-glibc-2.3.6.dat` sh all.sh --notest
```

Once the crosstool scripts execution is complete, the cross-toolchain can be find in the `$MY_GCC_VERSION/gcc-install` directory.

Please note that the cross-toolchain is a softfloat toolchain.

3 - Compiling a Java application

To compile a Java application, developer must configure the cross-compiling environment:

```
export PATH=/home/gcj/build/4.0.2/gcc-install/bin:$PATH
```

Then, to natively compile the java source code, launch the following command:

```
arm-softfloat-linux-gnu-gcj -msoft-float src/dave/test/HelloWorld.java  
--main=dave.test.HelloWorld -o HelloWorld
```

The resulting executable file is dinamically linked to the following libraries:

```
/gcj/gcj-test$ readelf -d HelloWorld
```

```
Dynamic section at offset 0x1018 contains 25 entries:
```

Tag	Type	Name/Value
0x00000001	(NEEDED)	Shared library: [libgcc_s.so.1]
0x00000001	(NEEDED)	Shared library: [libgcj.so.6]
0x00000001	(NEEDED)	Shared library: [libm.so.6]
0x00000001	(NEEDED)	Shared library: [libpthread.so.0]
0x00000001	(NEEDED)	Shared library: [libdl.so.2]
0x00000001	(NEEDED)	Shared library: [libc.so.6]

Please note the `libgcj` library, which is part of the cross-toolchain with gcj support.