

Hello World!
i.MX35 PDK Linux Application Note

by Multimedia Applications Division
Freescale Semiconductor, Inc.
Austin, TX

This document shows how to create and load a simple Hello World application into the tree directory for the packages used in the i.MX35 PDK Linux distribution.

Note that the distribution contains a Hello World application; however, the procedure described in this document explains how to create a new Hello World application.

Contents
1 Installation and Setup .....2
2 Creating a New Application.....2
3 Building the Application.....3
4 Adding the new Application to the Package Directory Tree.....4
5 Running the Application .....5
Appendix A. Source Files .....6
hello.c.....6
hello.spec .....6
Makefile.....7



# 1 Installation and Setup

The following elements are required to create and load a new application:

- Software Development Kit, SDK version 2009.12, installed as explained in the *i.MX35 PDK Linux User's Guide*
- NFS file system set to work with the PC host
- Serial interface between the target and the host PC (minicom for Linux or HyperTerminal for Windows) configured to operate at 115200 bps

## NOTE:

For additional information, see the instructions for the building process and NFS procedures in the *i.MX35 PDK Linux User's Guide*

# 2 Creating a New Application

In order to create the application at least the following files are needed:

- **Source file** (c file): The application in C, and the associated headers if the application requires them.
- **Makefile**: Each piece of code in Linux that requires a build process has a Makefile. The Makefile provides the rules, flags, includes and other elements that determine how the sources are built.
- **Spec file**: A file with the extension `.spec`: This is a file specification from the Linux Target Image Builder (LTIB) package that determines the instructions that the builder and installer use when building installing, unpacking and even patching the package. The spec file usually calls the make function that the Makefiles use from each Package.

In some circumstances only the source file is needed and the information provided by the Makefile is included in another file with the extension `.spec`, which is also created when making a new application.

See Appendix A for the complete source files needed for the new Hello World sample application.

The process to create and load a new application is to obtain the source files, clean them, and then create a package. The tarball compression format can be `gz` or `bz2`. The example in this document uses `gz`. It is important to provide a version to the folder name and to the tarball as well.

The procedure to create a new application is depicted below:

1. Enter the following command in the Linux host:

```
mkdir hello-1.0
```

2. Copy the .c file, the Makefile into the hello-1.0

```
cd hello-1.0
make clean
cd ..
tar zcvf hello-1.0.tar.gz hello-1.0
```

3. Copy the created package in the location where all the LTIB packages are stored. By default the packages are stored in /opt/Freescale/pkgs:

```
mv hello-1.0.tar.gz /opt/freescale/pkgs/
```

4. Create the spec file for this package. All the packages have a spec file and they are stored in <ltib location>/dist/lfs-5.1. There is a special folder where a template of the spec file is stored.

```
cd <ltib location>/
mkdir dist/lfs-5.1/hello
cp dist/lfs-5.1/template/template.spec dist/lfs-5.1/hello/hello.spec
```

After the hello.spec file is created, some editing should be made in order to make it work properly. Any text editor can be used to make the changes. Table 1 describes some of the fields to be defined in the spec file

**Table 1 spec file fields**

Field	Description
Summary	Brief summary of what the package is/does
Name	Name of the package (usually from the tarball name)
Version	Package version (usually from the tarball directory)
Release	Begin at 1 and add a revision each time you change the spec file
License	For example: GPL, LGPL, BSD. Find it in the package files
Group	If this exists on an rpm-based machine, copy from rpm -qi. If not, select from /usr/share/doc/rpm-/GROUPS
%Build	Apply Make
%Install	Copy the executable to the usr/bin directory

## 3 Building the Application

To build the application use the following procedure:

1. Unpack the package that is stored in /opt/Freescale/pkgs:

```
cd <ltib location>/
./ltib -m prep -p hello.spec
```

This command unpacks the hello -1.0.tar.gz inside the <ltib location>/rpm/Build/

2. Build the package:

```
./ltib -m scbuild -p hello.spec
```

3. Install and deploy the package. (this is done as specified in the %install section within the hello.spec file, see Appendix A):

```
./ltib -m scinstall -p hello.spec  
./ltib -m scdeploy -p hello.spec
```

The executable package is installed in the NFS root filesystem area (`rootfs`). We specified in the spec file that it needs to be installed in `rootfs/usr/bin`.

The Hello World application file is placed in the `rootfs/usr/bin` directory.

## 4 Adding the new Application to the Package Directory Tree

To add the Hello World package to the list of available packages when configuring LTIB, follow these steps:

1. Delete the package that was unpacked in `<ltib location>/rpm/BUILD/`.  

```
cd <ltib location>/  
cd /rpm/BUILD/  
rm -r hello-1.0/
```
2. Go to `<ltib directory>/config/userspace/`:  

```
cd <ltib location>/  
cd config/userspace/
```
3. Use a text editor (such as *vi*, *emacs*, or *gedit*) to open the file `packages.lkc`. The sections in this file are ordered alphabetically, look for the “H” section. In front of the section `PKG_HELLOWORD` add the following:  

```
config PKG_HELLO  
bool "hello"
```
4. Save the changes.
5. Go to `pkg_map` file that is located in the same directory (`config/userspace`).
6. Open `pkg_map` with a text editor.
7. Add the following entry anywhere (it does not need to be in alphabetical order).  

```
PKG_HELLO      =    hello
```

The application can now be added from the directory tree.

## 5 Running the Application

To run the application, follow these steps:

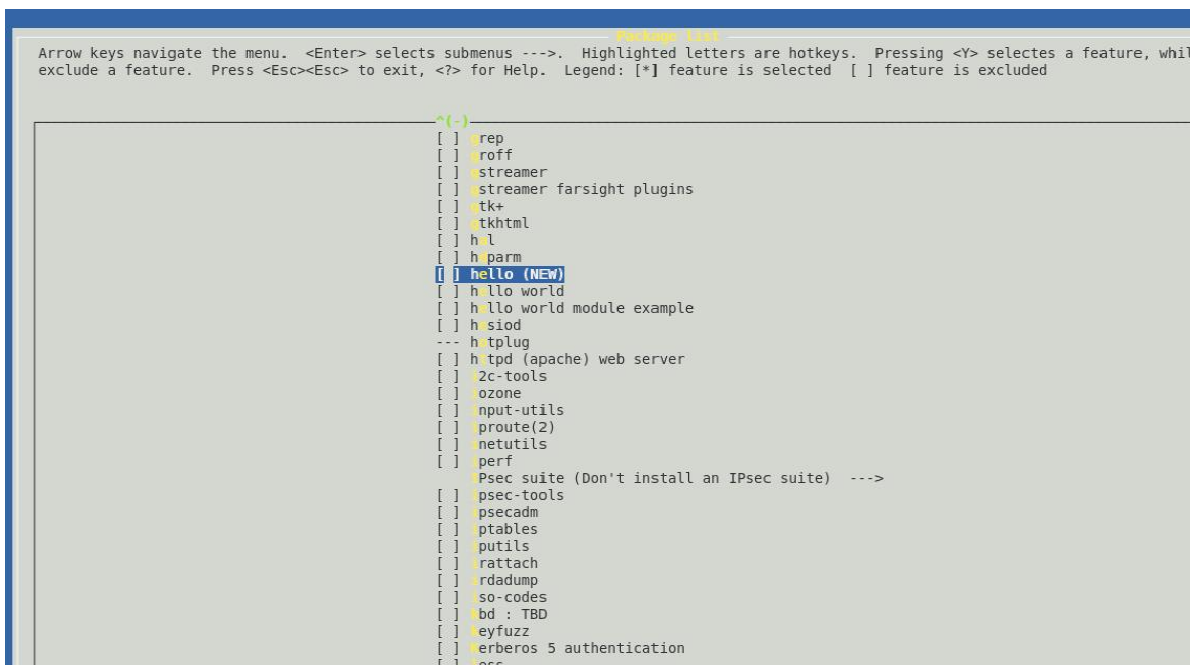
1. Move to the location where the LTIB is installed on the host PC and run the LTIB configuration:

```
cd <ltib location>/
./ltib -c
```

A window is displayed with the available configuration options, look for the section labeled Package selection and select Package list

2. Figure 1 shows the package list configuration menu that is displayed. Look for the new hello package, select it and exit saving the changes.

The hello package is built automatically and the binary file is placed in rootfs/usr/bin/.



**Figure 1. Package List**

3. Enable the NFS server and connect the i.MX35 PDK to the serial terminal at 115200 bps
4. Run the file in the target using the following commands

```
cd /usr/bin
hello
```

The application runs and messages are displayed in the serial console. All of the applications will follow this procedure. The differences are in the complexity of the Makefile, sources, or spec file.

## Appendix A. Source Files

### hello.c

```
#include <stdio.h>

int main()
{
    int i;

    printf("hello world\n");

    for ( i = 0; i < 10 ; i++ ) {
        printf("loop count = %d\n", i);
    }

    printf("hello this is the end\n");

    return 0;
}
```

### hello.spec

```
%define pfx /opt/freescale/rootfs/{_target_cpu}

Summary      : hello application for appnote
Name         : hello
Version      : 1.0
Release      : 1
License      : xxxx
Vendor       : Freescale
Packager     : User
Group        : MAD
URL          : http://xxxx
Source       : %{name}-%{version}.tar.gz
BuildRoot    : %{_tmppath}/%{name}
Prefix       : %{pfx}

%Description
%{summary}

%Prep
%setup

%Build
make

%Install
rm -rf $RPM_BUILD_ROOT
mkdir -p $RPM_BUILD_ROOT/%{pfx}/usr/bin
cp hello $RPM_BUILD_ROOT/%{pfx}/usr/bin/
```

```
%Clean
rm -rf $RPM_BUILD_ROOT

%Files
%defattr(-,root,root)
%{pfx}/*
```

## Makefile

```
EXEC = hello
OBJS = hello.o

all: $(EXEC)

$(EXEC): $(OBJS)
    $(CC) $(LDFLAGS) -o $@ $(OBJS) $(LDLIBS$(LDLIBS_.$@))

romfs:
    $(ROMFSINST) /bin/$(EXEC)

clean:
    -rm -f $(EXEC) *.elf *.gdb *.o
```

## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **Web Support:**

<http://www.freescale.com/support>

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor  
Technical Information Center, EL516  
2100 East Elliot Road  
Tempe, Arizona 85284  
+1-800-521-6274 or +1-480-768-2130  
[www.freescale.com/support](http://www.freescale.com/support)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
Technical Information Center  
Schatzbogen 7  
81829 Muenchen, Germany  
+44 1296 380 456 (English)  
+46 8 52200080 (English)  
+49 89 92103 559 (German)  
+33 1 69 35 48 48 (French)  
[www.freescale.com/support](http://www.freescale.com/support)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku,  
Tokyo 153-0064, Japan  
0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor China Ltd.  
Exchange Building 23F  
No. 118 Jianguo Road  
Chaoyang District  
Beijing 100022  
China  
+86 010 5879 8000  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

### **For Literature Requests Only:**

Freescale Semiconductor Literature  
Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
1-800-441-2447 or 303-675-2140  
Fax: 303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale and the Freescale logo are trademarks or registered trademarks of Freescale Semiconductor, Inc. in the U.S. and other countries. All other product or service names are the property of their respective owners. Microsoft and Windows are registered trademarks of Microsoft Corporation.

© Freescale Semiconductor, Inc. 2009. All rights reserved.