

Installing and Building the IA-32 Execution Layer

Contents

- [Before You Begin](#)
- [About This Document](#)
- [About the IA-32 Execution Layer \(IA-32 EL\)](#)
- [IA-32 EL Components](#)
- [Hardware Requirements](#)
- [Software Requirements for Running IA-32 EL](#)
- [Software Requirements for Building `libia32x.so` and `suid_libia32x.so`](#)
- [Preparing to Install the IA-32 Execution Layer](#)
- [Installing IA-32 EL](#)
- [Enabling IA-32 EL](#)
- [Enabling IA-32 EL Manually](#)
- [Enabling IA-32 EL Automatically](#)
- [Checking the Status of IA-32 EL](#)
- [Writing a Utility to Check the Status of IA-32 EL](#)
- [Implementing the `is_ia32el_enabled` Utility](#)
- [Checking the Status of IA-32 EL with a Control Script](#)
- [Installing IA-32 EL to a Custom Path](#)
- [Changing the Installation Path in the Source Code](#)
- [Building `libia32x.so` and `suid_libia32x.so`](#)
- [Upgrading IA-32 EL](#)
- [Disclaimer and Legal Information](#)

Before You Begin

Installing the IA-32 Execution Layer to the recommended path can be a straightforward process involving copying a few files and rebooting your machine. You may, however, need to update your kernel and/or modify the source code in order to facilitate installing the IA-32 Execution Layer to a path other than the recommended one. For an experienced Linux* administrator, these code modifications should not add a significant amount of time to the recommended installation process.

About This Document

This document describes how to:

- Install the default version of the IA-32 Execution Layer provided in this package
- Install the IA-32 Execution Layer to a path other than the recommended one
- Enable the IA-32 Execution Layer
- Check if the IA-32 Execution Layer is enabled

This document also describes and provides examples of how to update the kernel, if necessary, to run IA-32 EL.

About the IA-32 Execution Layer (IA-32 EL)

The IA-32 Execution Layer (IA-32 EL) is a software package that enables the user to run IA-32 applications on Itanium®-based platforms.

This distribution adds support for IA-32 EL to Linux distributions that do not have it built in.

Download the IA-32 Execution Layer for Linux systems from www.intel.com/software/products.

IA-32 EL Components

IA-32 EL includes the following components:

- **ia32exec.bin**
The main component of the IA-32 Execution Layer. It translates IA-32 code to Itanium architecture code. This component is only delivered in binary form.
- **libia32x.so**
The IA-32 Execution Layer operating system interface library. This component is delivered in binary form, as well as in source code under the GNU Lesser General Public License (LGPL).
- **suid_libia32x.so**
The IA-32 Execution Layer operating system interface library for running 32-bit binaries with `setuid` permission. This component is delivered in binary form, as well as in source code under the LGPL.

Hardware Requirements

A system based on the Itanium processor.

Software Requirements for Running IA-32 EL

This distribution of IA-32 EL requires the following:

- Linux kernel 2.6.0 or newer
- Kernel support for misc binary format (binfmt_misc)

Software Requirements for Building `libia32x.so` and `suid_libia32x.so`

GCC version 3.2.3 or later is required.

Preparing to Install the IA-32 Execution Layer

Unpacking the IA-32 EL Components

This section discusses how to prepare to install IA-32 EL, and describes the contents of the IA-32 EL packages.

To unpack the IA-32 EL packages, follow these steps:

1. Download `ia32el_intel_component.bin.tgz` and `ia32el_lgpl_component.tgz`.

2. In a directory to which you have write access, unzip these files by entering the following:

```
<prompt> tar -xzf ia32el_intel_component.bin.tgz
<prompt> tar -xzf ia32el_lgpl_component.tgz
```

The directory `ia32el_lgpl_component/` is ready for installation. Continue with the following steps to prepare the `ia32el_intel_component`.

3. Run the self-extracting archive `ia32el_intel_component.bin` by entering the following:

```
<prompt> ./ia32el_intel_component.bin
```

4. Read the license and accept it when prompted to do so.
5. Specify where you want to extract the files when prompted to do so, or press **Enter** to accept the default location specified.

The self-extracting archive places the extracted files in the location you specified.

IA-32 EL Package Contents

The following table describes the contents of the `ia32el_lgpl_component/` directory.

Directory/File Name	Description
LICENSE	The license agreement in ASCII text format
bin/	Contains the binary versions of <code>libia32x.so</code> and <code>bin/suid_libia32x.so</code>
bin/libia32x.so	IA-32 EL component
bin/suid_libia32x.so	IA-32 EL component
src/	Contains all source files necessary to build IA-32 EL
src/Makefile	Makefile for building <code>libia32x.so</code> and <code>suid_libia32x.so</code>
src/COPYING	LGPL License
src/COPYING.LIBC	LGPL License reference
src/ia32toia64/	Contains source code for translating 32-bit system calls to Itanium® architecture system calls
src/ia32x/	Contains source code for the Linux* operating system interface
src/ia32x_utils/	Contains source code for a libc substitute

The following table describes the files that are extracted from `ia32el_intel_component.bin`.

Directory/File Name	Description
ia32exec.bin	IA-32 EL component

ia32el	Sample control script for enabling and checking the status of IA-32 EL. NOTE: This sample script is written for Red Hat Linux* systems. You need to edit this script to ensure compatibility with your system.
IA32EL-LICENSE	The license agreement in ASCII text format
INSTALL.htm	This document

Installing IA-32 EL

This section describes how to install all IA-32 EL components.

NOTE: You need root permissions in order to install IA-32 EL to the recommended path.

NOTE: If you install IA-32 EL to any path other than the recommended one, you need to follow the steps in [Installing IA-32 EL to a Custom Path](#).

To install IA-32 EL, follow these steps:

1. Copy the IA-32 EL components to the installation directories. The recommended installation directory for each component is listed in the following table.

Component Name	Recommended Installation Path
ia32exec.bin	/usr/local/bin/ia32exec.bin
libia32x.so	/usr/local/bin/libia32x.so
suid_libia32x.so	/usr/local/bin/suid_libia32x.so

2. Verify that execution permission is set for **owner**, **group** and **others** for `libia32x.so` and `suid_libia32x.so`.
3. Verify that `setuid` permission is set for **owner** for `suid_libia32x.so`.
4. Enable IA-32 EL as described in [Enabling IA-32 EL](#). If IA-32 EL fails to start, you may not meet the minimum system requirements. See [Software Requirements for Running IA-32 EL](#).

Enabling IA-32 EL

This section describes how to enable IA-32 EL.

You have to enable IA-32 upon booting the machine, because the `binfmt_misc` module does not maintain its state when the system shuts down or reboots. You can enable IA-32 EL manually at any time, or you can write an initialization script to automatically start IA-32 EL when the operating system starts.

Running an initialization script upon rebooting the machine has the following benefits:

- It automatically enables IA-32 EL.

- You can ensure that the system starts IA-32 EL before starting any IA-32 processes, so that they run using the newly installed version of IA-32 EL.

Enabling IA-32 EL Manually

To enable IA-32 manually follow the procedure defined in the function `activate_ia32el()` in the sample control script, `ia32el`, provided with the IA-32 EL package.

Enabling IA-32 EL Automatically

The sample control script, `ia32el`, provided with the IA-32 EL package enables IA-32 EL. It is written for Red Hat Linux systems. You need to edit this script to ensure compatibility with your system.

NOTE: You need root permissions to enable IA-32 EL.

To automatically enable IA-32 EL, follow these steps:

1. Edit the sample script `ia32el` provided with the IA-32 EL package, ensuring that the following are true:
 - the function `activate_ia32el()` is written correctly for your system
 - the script references the correct IA-32 EL installation path. If you have installed IA-32 EL to the recommended installation directory, the script should already reference the correct path. Otherwise, see [Changing the Installation Path in the Source Code](#).
2. Copy the control script to `/etc/init.d/`.
3. Integrate the `/etc/init.d/ia32el` script into the early stages of the system's boot process, before any IA-32 process executes. For example, on Red Hat Linux, enter the following command line:

```
<prompt> /sbin/chkconfig --add ia32el
```

4. Reboot the machine.

Every subsequently invoked IA-32 process runs using IA-32 EL.

NOTE: Enable IA-32 EL before starting any IA-32 processes to ensure that they run using the newly installed version of IA-32 EL.

Checking the Status of IA-32 EL

To track problems with IA-32 applications, it is recommended to use a utility that checks whether IA-32 EL is enabled.

This section discusses the necessary steps for writing and using such a utility.

Writing a Utility to Check the Status of IA-32 EL

This section describes how to write a utility that checks whether IA-32 EL is enabled. The following sections refer to this utility by the name `is_ia32el_enabled`, but you can use any name you prefer.

Check whether IA-32 EL is enabled by running the 32-bit version of the CPUID instruction to check the value of CPUID. If the value of "family" is 0xF and the value of "extended family" is 0x2, then IA-32 EL is running. The following code prints 1 if IA-32 EL is running, and prints 0 if it is not running. You can use this code to write your own `is_ia32el_enabled` utility.

NOTE: When you compile the code for this utility, be sure to use an IA-32 compiler.

```
int is_ia32el_active;
unsigned CpuidReturned[2], family_number, ext_family_number;
int _eax = 0x1;

__asm__ __volatile__ ("cpuid"
    : "=a" (CpuidReturned[0])
    : "a" (_eax));

family_number = (CpuidReturned[0] >> 8) & 0xF;
ext_family_number = (CpuidReturned[0] >> 20) & 0xF;

is_ia32el_active = ((family_number == 0xF) && (ext_family_number ==
0x2)) ? 1 : 0;
printf ("%d\n", is_ia32el_active);
```

Implementing the `is_ia32el_enabled` Utility

This section discusses the necessary steps for implementing `is_ia32el_enabled`.

To implement `is_ia32el_enabled`, follow these steps:

1. Ensure that the desired installation directory has read permissions set for all users.
2. Copy `is_ia32el_enabled` to the desired installation directory.
3. Verify that read and execution permissions are set for **owner**, **group** and **others**.

The sample script, `ia32el`, provided with the IA-32 EL package includes an example of how to use `is_ia32el_enabled`. Open the script in a text editor and see the function `is_ia32el_activated()`.

Checking the Status of IA-32 EL with a Control Script

This section discusses how to check whether IA-32 EL is enabled when using the example code in [Writing a Utility to Check the Status of IA-32 EL](#) and calling to it as described in the sample script `ia32el`.

Check the status of IA-32 EL by entering the following command line:

```
<prompt> /etc/init.d/ia32el status
```

One of the following status messages appears:

- IA-32 Execution Layer enabled
- IA-32 Execution Layer disabled

Installing IA-32 EL to a Custom Path

This section describes how to install IA-32 EL to a directory other than the recommended one, `/usr/local/bin/`.

To install IA-32 EL to a directory other than the recommended one, follow these steps:

1. Ensure that the desired installation directory has read permissions set for all users.
2. Copy the file `ia32exec.bin` to the desired directory.
3. Edit the source code as described in [Changing the Installation Path in the Source Code](#).
4. Build `libia32x.so` and `suid_libia32x.so` as described in [Building libia32x.so and suid_libia32x.so](#).
5. Copy the new versions of `libia32x.so` and `suid_libia32x.so` to the desired directory.
6. Verify that execution permission is set for **owner**, **group** and **others** for `libia32x.so` and `suid_libia32x.so`.
7. Verify that `setuid` permission is set for **owner** for `suid_libia32x.so`.
8. Enable IA-32 EL as described in [Enabling IA-32 EL](#).

Changing the Installation Path in the Source Code

This section describes the source code edits required to install IA-32 EL to a directory other than the recommended one.

1. In the source code for `libia32x.so` and `suid_libia32x.so`, edit the path to `ia32exec.bin` by following these steps:
 - A. Open the source file `ia32x/btlib_main.c`.
 - B. Find the array `BTGeneric_search_path` and change the path `/usr/local/bin` to the desired installation path.
 - C. Save the edited file.

For information on building `libia32x.so` and `suid_libia32x.so`, see [Building libia32x.so and suid_libia32x.so](#).

2. If you have a control script similar to the sample script `ia32e1` described in [Enabling IA-32 EL Automatically](#), edit the script by changing all occurrences of the path `/usr/local/bin/` to the desired installation path.

Building libia32x.so and suid_libia32x.so

This section describes how to build `libia32x.so` and `suid_libia32x.so`.

1. Change to the directory in which you unpacked the source package.
2. Run `make` using the included makefile by entering the following:

```
<prompt> make
```

Make builds the following two binaries:

```
<working_directory>/ia32x/bin.release/libia32x.so  
<working_directory>/ia32x/bin.release/suid_libia32x.so
```

Upgrading IA-32 EL

To upgrade IA-32 EL, follow these steps:

1. Delete the IA-32 EL components from the installation directories. The recommended installation directories are noted in [Installing IA-32 EL](#).
2. Install the new version by following the steps in [Installing IA-32 EL](#).

Disclaimer and Legal Information

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

This document as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The software described in this document may contain software defects which may cause the product to deviate from published specifications. Current characterized software defects are available on request.

Intel, the Intel logo, Intel Inside, the Intel Inside logo, Pentium, Itanium, Intel Xeon, Celeron, Intel SpeedStep, Intel Centrino, Intel NetBurst, Intel NetStructure, VTune, MMX, the MMX logo, Dialogic, i386, i486, iCOMP, Intel386, Intel486, Intel740, IntelDX2, IntelDX4 and IntelSX2 are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2004, Intel Corporation. All rights reserved.