eSpeed C/C++ Application Program Interface

INSTALL AND BUILD GUIDE

System Version: 1.4

Revision Date: October 19, 2005

Document Version: ESPD – 1

Table of Contents

1. Introduction

1.1 Overview

This document describes how to install the eSpeedAPI on the development platform and how to build your application with the C/C++ eSpeedAPI.

Other documents relating to the eSpeedAPI:

eSpeedAPI Reference Guide describes the C interface and the message flow of the eSpeedAPI.

eSpeedAPI Release Notes contains the changes for a particular release version of the eSpeedAPI.

The eSpeedAPI is supported on a number of Windows and Solaris platforms, see section Supported platforms for details.

The procedure to install the eSpeedAPI software development kit on a specific platform is described in section Installation.

For instruction on how to build the client application with the eSpeedAPI, see section Building and Running eSpeed API Applications.

1.2 Supported platforms

The eSpeed API is available as an installable component for the following platforms:

Platform	Operating System	Compiler	Build Type
Sun SPARC	Sun Solaris 8 and later versions. Tested with Solaris 8 Solaris 9	Sun WorkShop Compiler C++ 5.3 and later versions. Tested with compiler 5.3, 5.4 and 5.5	Dynamic/shared library built with multi-threading and STL
Linux 2.4	SuSE 8 on x86 Red Hat AS3 on x86	GCC 3.2.3	Dynamic/shared library built with multi-threading and STL
Win32, MSVC60	Microsoft Windows NT 4.0 (SP6) Tested with Microsoft Windows 2000 (SP4)	Microsoft Visual C++ 6.0 (SP6)	DLL built with multi-threading and STL
Win32, MSVC71	Runtime compatible from Microsoft Windows NT 4.0 (SP6) Development require at least Microsoft Windows 2000 (SP4)	Microsoft Visual Studio 2003, version 7.1.	DLL built with multi-threading and STL

1.3 Required patch levels and service packs

1.3.1 Operating Systems

The following table lists the patch level and other specification details required of each operating system, when building applications with the eSpeed API. The same patches are recommended when running applications.

OS	OS Specification
Solaris 8	Patch levels: Generic_108528-21, SUNWlibC 108435-12
Linux 2.4	SuSE Linux kernel version 2.4.19, glibc-2.3.2-9 Red Hat AS3 Linux kernel version 2.4.21-4, glibc-2.3.2-95.3
WinNT4	Service pack: SP6
Win2000	Service pack: SP4
Win XP	Service pack: SP2

1.3.2 Compilers

The following table lists the patch levels and other compiler specifications recommended when building applications against the eSpeed API.

Compiler	Compiler Specification
SUNPRO53	Sun WorkShop 6 update 2 C++ 5.3. The libESPD is built with Sun CC 5.3. When building the application with Sun CC 5.3, the following compiler patch is required: 111685-16 Sun's STL library Cstd is also required
GCC32	gcc 3.2.2 (on Linux only) Configured with: ../configure --enable-threads=posix --libdir=/usr/lib --enable-languages=c,c++,f77,objc,java,ada --enable-libgcj --with-gxx-include-dir=/usr/include/g++ --with-slibdir=/lib --with-system-zlib --enable-shared --enable-__cxa_atexit i486-suse-linux Thread model: posix gcc version 3.2 glibc-2.3.2-9 libstdc++.so.5 libc.so.6
MSVC60	Service pack: SP5
MSVC71	Microsoft Visual Studio .NET 2003 Version 7.1.

1.4 Customer Support

Customer Support is available between 7:00am and midnight U.S. Eastern time, Monday through Friday, excluding holidays. When calling, please assist us by being ready with your product and account information.

The eSpeed Customer Support group has a series of phone numbers and e–mail addresses to meet various user needs. They are as follows:

- If you are a Customer with questions about possible trading scenarios or specific trading features (e.g. price improvement) please contact your account representative.

- If you are a Customer experiencing technical difficulty, have questions on how to use the system, please call or write:

  eSpeed Call Center (US) — (+1) (212) 610-2300 or support@espeed.com
  eSpeed Help Desk (Europe) — (0)20-7894–8600 or support@espeed.co.uk

- If you are a Customer or Salesperson requesting information on how to make changes to your electronic account, or require a new access account, please call or write:

  eSpeed Customer Access (US) — (212) 610-2300 or customeraccess@espeed.com
  eSpeed Customer Access (Europe) — (0)20-7894-8886 or customeraccess@espeed.co.uk

- If you are a Customer with questions regarding API development issues, downloading the latest version of the SDK or JNI, or testing a trade feed or market–making application, please call (between the hours of 9:00am and 6:00pm EST) or write:

  eSpeed Customer Integration — (+1) (212) 610-3560 or customerintegration@espeed.com

- If you are a Customer with questions regarding specific trades, verification of a trade, or are experiencing delivery problems with a trade, please call:

  eSpeed Trade Support (US) — (212) 610-2300
  eSpeed Trade Support (Europe) — (0)20-7894-8600

2. Installation

The steps required to complete the installation are listed here for each target platform. The eSpeedAPI version and OS version are just for the example.

2.1 Unix

The eSpeedAPI for Unix consists of the following files:

```
libESPD-1.4.1.0.tgz              The eSpeedAPI libESPD release

libESPD-1.4.1.0_checksum.txt     The CRC32 checksum of the file archive
```

The content of the libESPD file archive is typically:

```
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/lib/libESPD.so
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/espd/espd.h
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/espd/types.h
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/espd/fields.h
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/espd/consts.h
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/espd/dllexport.h
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/sample/main.cpp
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/sample/sample.cpp
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/sample/io.cpp
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/sample/io.h
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/sample/sample.mk
<installdir>/eSpeed/eSpeedAPI-1.4.1.0/sample/sample.gnu
```

The `<installdir>` can be any directory on the development system.

Follow these steps to install on Unix.

1. Check the package integrity.
   ```
   sum libESPD-1.4.1.0-SOLARIS26.tgz
   ```
   The output checksum should be the same as the sum in
   ```
   libESPD-1.4.1.0-SOLARIS26_checksum.txt
   ```
   If the check sums do not match, download the package again from the eSpeed Website. If the problem persists, contact eSpeed Customer Integration.

2. Unpack the archive from the directory where the eSpeedAPI directory should reside.
   ```
   cd <installdir>
   gzip -d -c libESPD-1.4.1.0-SOLARIS26.tgz |tar xvf -
   ```

2.2 Windows

The Windows version is distributed in an Install Shield package.

```
eSpeedAPI.exe                    The eSpeedAPI libESPD release
```

The libESPD files will be installed in the following directory structure:

```
<installdir>
      └──espd-1.4.1.0
            ├──Sample
```

```
├──Documents
├──DLL
└──espd
```

1. Run `setup.exe`.
2. Respond to all questions raised by the installation program (including selecting directory in which to install the software). If a previous version of the eSpeedAPI is detected on the system, it will be uninstalled the first time the `setup.exe` is run. In this case run `setup.exe` again to install the latest version.

3. Building and Running Applications with the eSpeed API

eSpeed API applications may be developed using the eSpeed API in either C or C++. In order to use the API applications should include the eSpeed API include file `espd.h`. This will ensure that all eSpeed API definitions are available to the application. Specific build instructions for the supported compilers are listed subsections below.

The eSpeed API package provided is not dependent on any 3rd party libraries apart from the STL library. The application implementation can make use of any 3rd party library without causing clashes as long as the build parameters are compatible and the STL library specified for corresponding compiler is used.

The configuration below is working with including the libESPD header files in the C/C++ code with:
```
#include <espd/espd.h>
```

Code for a simple eSpeed API application is installed in
`<installdir>/eSpeed/eSpeedAPI-<release>/sample`.

3.1 Sun WorkShop Compiler C++

This configuration is for Sun compiler 5.3 or later version , on Solaris 8 or later version.

Application using STL and is dynamically linked with libESPD:
Include path: `<installdir>/eSpeed/eSpeedAPI-<release>`
Link path: `<installdir>/eSpeed/eSpeedAPI-<release>/lib`
Compiler flags: `-mt -D_REENTRANT`
Linker flags: `-mt -D_REENTRANT`
Linker libraries: `-lESPD -lsocket –lnsl`

Only the library flag `Cstd`, which is enabled by default, is required for libESPD. Flags like for instance rwtools can be enabled if required by the application, but it does not affect the libESPD.

3.3 Microsoft Visual C++ 6.0

This configuration is for MSVC 6.0 with Service Pack 6 on for WIN32.

Application using STL and is dynamically linked with libESPD:
Include path: `<installdir>/eSpeed/eSpeedAPI-<release>`
Link path: `<installdir>/eSpeed/eSpeedAPI-<release>/DLL/MSVC60`
Compiler flags: /MD
Linker flags:
Linker libraries: `libESPD.lib ws2_32.lib`

Built executables only look in certain directories for a given function library. For any built executable, be sure to have the libESPD DLL installed in one of the following:

- the same directory as the executable,
- %WINDIR%\system,
- any directory listed in the PATH environment variable.

3.4 Microsoft Visual Studio .NET 2003 v7.1

This configuration is for MSVC 7.1 for WIN32.

Application using STL and is dynamically linked with libESPD:
Include path: `<installdir>/eSpeed/eSpeedAPI-<release>`
Link path: `<installdir>/eSpeed/eSpeedAPI-<release>/DLL/MSVC71`
Compiler flags: /MD
Linker flags:
Linker libraries: `libESPD.lib ws2_32.lib`

Built executables only look in certain directories for a given function library. For any built executable, be sure to have the libESPD DLL installed in one of the following:

- the same directory as the executable,
- %WINDIR%\system,
- any directory listed in the PATH environment variable.

4. CFETI to libESPD Migration

- **New header file name**
  The header file with CFETI was included with
  ```
  #include <cfeti.h>
  ```
  With libESPD the include is:
  ```
  #include <espd/espd.h>
  ```

- **All strings (char*) are now made to be const char***
  To avoid a change of the content of a string or buffer passed from the application to libESPD or in the callback libESPD to the application, all char* are now made to be const char*.

  This may require some change of code in the application if the CFETI-structures are used for storing strings or buffers that the application modifies.

  A compiler error will occur if changes are needed.

□