

Windows

Prerequisites

OS: Windows XP SP2, Windows Vista, Windows 7, Windows 8

IDE: MS Visual Studio 2013

Instructions

1. Install cmake 3.2.2 from <http://www.cmake.org/download/>
2. Install Python 2.7 from <https://www.python.org/downloads/release/python-279/>
3. Download and install 7-Zip from <http://www.7-zip.org/>
4. Add the directory you installed Python into to your PATH (Start -> Control Panel -> System -> Advanced -> Environment Variables).
5. Add the directory you installed 7-Zip into to your PATH (Start -> Control Panel -> System -> Advanced -> Environment Variables).
6. Download and install Boost 1.55 from http://sourceforge.net/projects/boost/files/boost-binaries/1.55.0-build2/boost_1_55_0-msvc-12.0-32.exe/download
7. Download Avro 1.7.5 from <http://archive.apache.org/dist/avro/avro-1.7.5/avro-src-1.7.5.tar.gz>
8. Download Botan 1.11.28 from <https://github.com/randombit/botan/archive/1.11.28.tar.gz>
9. Create a separate directory (for example, C:\build_kaa), further in text *CAA_BUILD_DIR*.
 - a. Place archives avro-src-1.7.5.tar.gz and Botan-1.11.28.tgz to *CAA_BUILD_DIR*.
 - b. Place env.bat to *CAA_BUILD_DIR*(. (File env.bat located in the 'tools' folder in C++ SDK)
 - c. Place build_sdk_thirdparty.bat to *CAA_BUILD_DIR*. (File build_sdk_thirdparty.bat located in the 'tools' folder in C++ SDK)
10. Edit env.bat:
 - a. Update the MSVS_HOME variable. It must point to MS Visual Studio installation directory.
 - b. Update the BOOST_ROOT variable. It must point to the Boost installation directory (see the *Download and install Boost* step).
 - c. Update the AVRO_ROOT_DIR variable. It must point to the directory where Avro binaries and includes will be installed.
 - d. Update the BOTAN_HOME variable. It must point to the directory where Botan binaries and includes will be installed.
11. Open the command line terminal ([Developer Command Prompt](#)) and proceed as follows:
 - a. Go to *CAA_BUILD_DIR*:

```
> cd CAA_BUILD_DIR
```
 - b. Build thirdparty components. Execute the build_sdk_thirdparty.bat script.

```
> build_sdk_thirdparty.bat
```

NOTE: By default, the debug configuration is used. To build release versions, use the 'release' argument:

- ```
> build_sdk_thirdparty.bat release
```
12. To build the Kaa C++ SDK, proceed as follows:
    - a. Download and untar an appropriate C++ SDK tar.gz archive from Admin UI.
    - b. Open the command line terminal ([Developer Command Prompt](#)) and run the following commands:

```
> CAA_BUILD_DIR\env.bat
> avrogen.bat
> md build
> cd build
> cmake -G "NMake Makefiles" -DKAA_INSTALL_PATH="C:\KaaSdk" -DKAA_DEBUG_ENABLED=1 ..
> nmake
> nmake install
```

For additional cmake options, see [the build configuration page for C++ endpoint SDK](#).

## Demo applications

To build and run a Kaa C++ demo, proceed as follows:

1. Download and untar appropriate C++ demo sources from Kaa Sandbox.
2. Open the command line terminal ([Developer Command Prompt](#)) and run the following commands:

```
> CAA_BUILD_DIR\env.bat
> build.bat deploy
```