

Windows

Prerequisites

OS: Windows XP SP2, Windows Vista, Windows 7, Windows 8, Windows 10

IDE: MS Visual Studio 2013, MS Visual Studio 2015

Instructions

1. Download and install following dependencies:

1. [wget](#) and [libarchive](#).
2. [cmake 3.5.1](#)
****NOTE:**** Make sure that you chose one "Add CMake to system path..." for all users or for current user on the "Install options" step
3. [Python 2.7\(amd64\)](#) or [Python 2.7\(i386\)](#) depending on your system architecture.
4. [Boost 1.60](#).
5. 7-Zip from <http://www.7-zip.org/> . Add the directory you installed 7-Zip into to your PATH (Start -> Control Panel -> System -> Advanced -> Environment Variables)

2. Create a separate directory (for example, C:\build_kaa), further in text KAA_BUILD_DIR.

1. Place `env.bat` in `KAA_BUILD_DIR`. (File `env.bat` located in the `tools` folder in C++ SDK)
2. Place `build_sdk_thirdparty.bat` in `KAA_BUILD_DIR`. (File `build_sdk_thirdparty.bat` located in the `tools` folder in C++ SDK)

3. Edit `env.bat`:

1. Update the `BUILD_PLATFORM` variable. It must be equal to the target architecture.
3. Update the `BOOST_ROOT` variable. It must point to the Boost installation directory (see the Download and install Boost step).
4. Update the `GNUWIN32_ROOT` variable. It must point to the `wgen` and `libarchive` installation directory (see the Download and install `wget` and `libarchive` step).
5. Update the `AVRO_ROOT` variable. It must point to the directory where Avro binaries and includes will be installed.
6. Update the `BOTAN_ROOT` variable. It must point to the directory where Botan binaries and includes will be installed.
7. Update the `ZLIB_ROOT` variable. It must point to the directory where zlib binaries and includes will be installed.

4. Open the command line terminal [Developer Command Prompt](#)

and proceed as follows:

1. Go to `KAA_BUILD_DIR`:

```
$ cd KAA_BUILD_DIR
```

2. Build thirparty 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
```

3. To build the Kaa C++ SDK, proceed as follows:

1. Download and untar an appropriate C++ SDK tar.gz archive from Admin UI.
2. Open the command line terminal [Developer Command Prompt](#) and run the following commands:

```
$ KAA_BUILD_DIR\env.bat
    $ avrogen.bat
    $ md build
    $ cd build
```

3. To build Kaa C++ SDK with nmake run:

```
$ cmake -G "NMake Makefiles" -DCMAKE_INSTALL_PREFIX="C:\KaaSdk"  
-DKAA_DEBUG_ENABLED=1 ..  
    $ nmake  
    $ nmake install
```

For additional cmake options, see the [page](#).

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:

```
$ KAA_BUILD_DIR\env.bat  
$ build.bat deploy
```