



Yet Another APRS Client

Status: Beta
Brought to you by: apavlin

Building and Modifying YAAC



Authors:

Building and Modifying YAAC

To modify and rebuild YAAC with your modifications, you will need the following:

- a source code distribution of YAAC, either from the YAACSrc.zip file, or (preferably) by Subversion checkout from the SourceForge-hosted repository.
- an installed copy of the Java Development Kit (JDK), Standard Edition, of at least Release 7, for your platform.
- the Apache Ant build tool,
- copies of these third-party libraries used by YAAC. Note that both the binary and source code distributions of YAAC include binary libraries of these products, but to obtain source code and exercise your rights under their open-source licenses, you will have to obtain the source code from the various products' distributors:
 - [JSSC 2.9.2](#)
 - [OpenMap 5.0](#)
 - [JavaHelp 2.0](#)
 - [Apache Commons Compress](#)

If you plan on making changes and submitting them back to the YAAC user community, getting your source code from the SourceForge-hosted Subversion repository is preferred. This will also make it easier for you to merge in updates made by other YAAC developers. This is available on the Code page of this project. Submit a ticket on SourceForge to arrange to be added to the authorized developers' list.

To compile YAAC, change directory to the top level directory of the source distribution. This directory is the one contain the "YAAC" and "lib" subdirectories and the main "build.xml" file (among other files). Invoke Ant in the appropriate manner for your platform, specifying the build target of interest. The build targets currently included in YAAC are:

- MAIN (the default target) - compiles everything by invoking the other targets recursively.
- JAVAC - compile only the source code of YAAC itself; does not compile the sample plugin.
- CLIENT_JAR - collect the compiled class files into a properly structured JAR file for execution. The YAAC.jar file will be located in the "classes" subdirectory.
- SAMPLEPLUGIN_JAVAC - compile the source code of the sample plugin.
- SAMPLEPLUGIN_JAR - collect the sample plugin's files into a properly structured JAR file and install it in the plugins directory.
- JAVADOC - generate the Javadocs of the base YAAC application, suitable for providing information to the developers of YAAC plugins.

Other subdirectories are for the source code and build.xml files for various plugins and utilities available with YAAC, including:

the ADS-B receive plugin,
the Bluetooth port driver plugin,
a utility for creating YAAC map data from the [Burning Man](#) city planning data,
the callsign database plugin,
a utility for converting the Multiple Sclerosis Society's cue sheets for the annual City-to-Shore 150 fundraiser bike ride event into YAAC map overlay data,
the repeater finder plugin,
the YAAC-customized version of the RXTX library's Java and C source code (supporting only the legacy RXTX plugin for users who, for some reason, cannot use JSSC),
the sample plugin,
the sounds plugin, and
the telemetry alarm plugin.

Most of these subdirectories have their own build.xml file and any special third-party libraries required by the particular sub-module. They should be compiled after YAAC is built, as most of them depend on linking to YAAC API methods (as documented for writing plugins).

Once YAAC has been completely compiled and stored into JAR files, the PACKAGE Ant target can be used to collect all the components for a binary distribution into a YAAC.zip file.

Related

[Wiki: Downloading YAAC](#)

[Wiki: Home](#)

**Get latest updates about
Open Source Projects,
Conferences and News.**

Sign Up

No, Thank you