

Development Strategies for Integrating Logbook of the World into Logging Applications

There are several options for the developer of logging software to consider to integrate Logbook of the World (LoTW) compatibility into a logging program. This document discusses briefly what some of the options are.

LoTW uses the TrustedQSL (TQSL) concept to enable digital signing of log records by the amateur submitting them. The digital signing process comprises, at minimum:

- 1) Generating a signing key pair certificate request file and sending it to LoTW.
- 2) Receiving the LoTW certificate back from LoTW.
- 3) Configuring a “station location” that contains information about the station whose log is being submitted.
- 4) Calculating a digital signature for each QSO record submitted, using the key created in step 1.
- 5) Formatting the QSO data and digital signature into a TQSL file in TQSL format.
- 6) Sending the TQSL file to LoTW.

Steps 4 and 5 are repeated for each QSO record, while step 6 is performed once per file. Steps 1 and 2 are performed once per year, and step 3 is performed as needed.

Clearly, there is a lot going on here! Fortunately, ***logging programs don't have to do any of this!*** The logging software developer may, however, choose to do some or all of these steps in order to add value to the logging program.

The TQSL Software

The freely available TQSL software includes a library, *tqsl*, and two application programs, *TQSL* and *TQSLCERT*. The *tqsl* library contains the digital signing, certificate handling, format conversion and utility routines that implement the TQSL system.

TQSLCERT is used to manage the user's digital certificates. It has a certificate-request wizard that can generate a key pair and the associated certificate request file as well as a function to load the LoTW certificate into the system and facilities for managing multiple certificates. That is, *TQSLCERT* handles steps 1 and 2 above, except for the physical sending of the file to LoTW.

TQSL performs steps 3 through 5 above. For step 3, it includes a wizard that allows the user to describe relevant aspects of his station, such as state/county, ITU zone, etc.

TQSL gets the QSO data of steps 4 and 5 by reading in a file that is in either ADIF or Cabrillo format.

Logging Software Interface

Outputting Compatible Files

At the simple end of the complexity spectrum, a logging program developer need only ensure that the logging program can emit ADIF and/or Cabrillo files that are compatible with *TQSL*. This is not especially difficult, and most logging programs probably can do that unaltered. Note, though, that the user will not want to submit the same log data multiple times, so means of outputting only part of the log data are much to be desired.

The remaining choices discussed here all require use of the *tqsl* library. This is provided as a Windows DLL and as a static library on other platforms, as well as in source code form.

Station Location Editing

The user must establish one or more “station locations,” and *TQSL* can be used to do that. This function could also be integrated into the logging program.

In-Place Conversion

The ADIF/Cabrillo conversion routine is wholly contained in the library. That means that a logging program could automate the conversion process that *TQSL* performs by outputting log data to a file and running it through the library's converter, all without the user even knowing that's what is being done.

Implementing Signing

The logging program could also elect to skip the ADIF/Cabrillo process and use the *tqsl* library's interface to directly sign the data. The library API makes this fairly easy, and there are possible advantages to doing this. The ADIF/Cabrillo converter functions (and thus *TQSL*, which relies on them) require that all of the data in a file be signed using the same station location (which includes not only geographic information but also call sign and DXCC entity). This can be inconvenient for some users. By using the API directly, a logging program that stores user-station information with each log record could automatically associate station locations with the log records being signed, transparently to the user.

Implementing Certificate Management

Logging software could also include the certificate management functionality provided by *TQSLCERT*. If desired, all of the necessary functionality of *TQSL* and *TQSLCERT* could then be provided from within the logging program.

Sending the Data

No matter what *TQSL* functionality the logging program provides, it may also make sense to provide a means to automatically send the data to LOTW. Neither *TQSL* nor *TQSLCERT* provide this capability.