

USER'S GUIDE

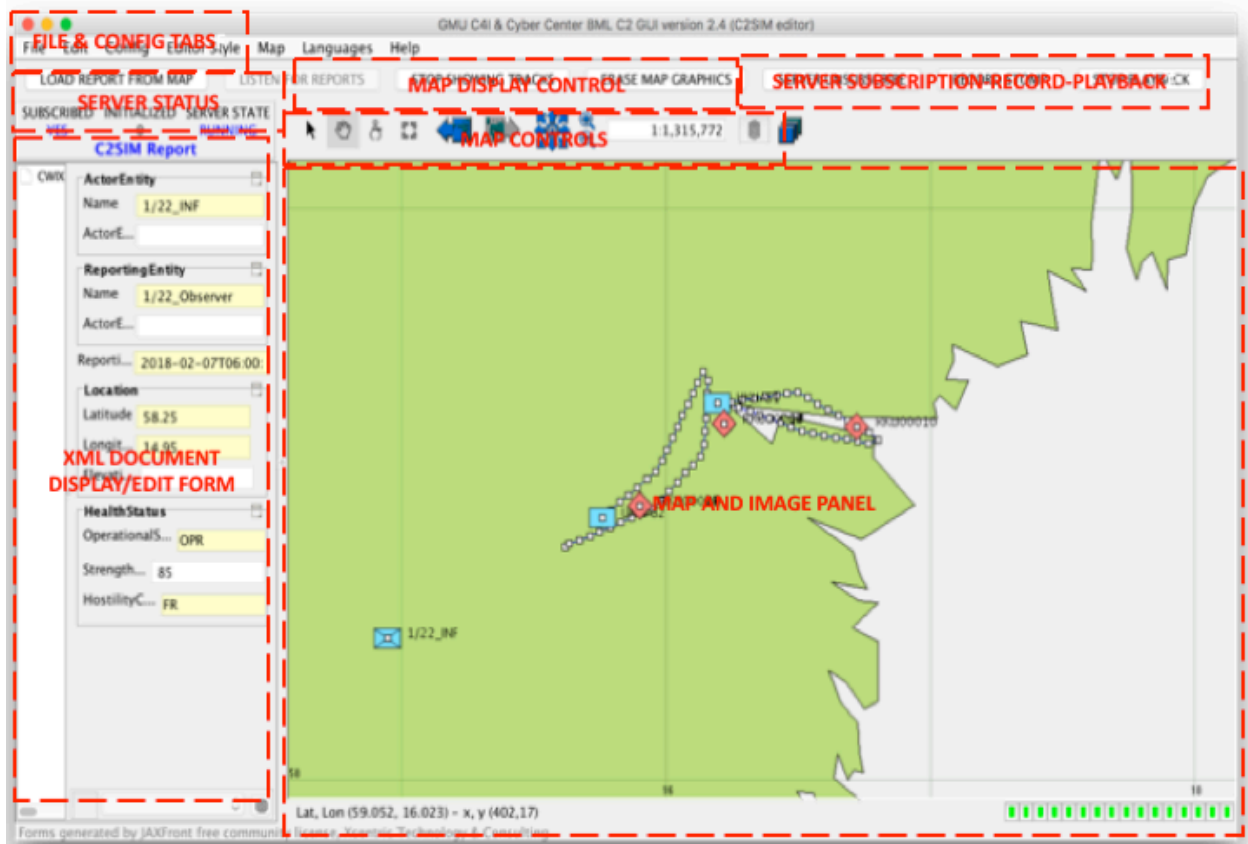
BML C2 GUI Version 2.5 C2SIM

The BML C2 GUI editor was developed in 2009-2010 by Lt Col Mohamamad Ababneh, Jordanian Air Force, as graduate student work during his PhD program at George Mason University. The GUI is written in Java for portability. It went through a major revision to version 2.0 in 2013 to include the STOMP and REST protocols for use with the WISE-SBML server in NATO MSG-085. Recently it has been upgraded to version 2.3, including the draft SISO Command and Control-Simulation (C2SIM) standard for NATO MSG-145 use in CWIX. It is expected to be further upgraded, incorporating the final balloted version of C2SIM, when standard drafting is complete. This document supersedes the original User's Guide by Dr. Ababneh; however the original will continue to be included with the BMLC2GUI open source for continuity since it contains information not included in this document.

Purpose of the BML C2 GUI

The GUI is open source software patterned after the C2LG GUI, similar software developed by Fraunhofer-FKIE that is not available open source. Like to FKIE GUI, BMLC2GUI is intended to be a tool that will interact with a C2SIM server by creating and/or editing XML Order and Report files, pushing such files into the server, subscribing to receive such files, and displaying their tactical graphics (unit icons and related graphical control measures) on a map or image.

Major Features of the GUI



The GUI is shown above with labels on its features. The features are:

- File and Config tabs: these select the File and Config dropdown menus; File provides different tabs for each message format because the XML must be associated with a schema when it is loaded
- Server Status: shows subscription status, initializations pending, and server running/paused/stopped/uninitialized status
- Map Display Control: triggers one-time import of a report from the map; turns on/off listening to STOMP for reports; turns on/off display of tracks; erases map graphics
- Server Subscribe Control: subscribes to/unsubscribes from server
- Record and Playback of the incoming STOMP XML file stream can be started from the GUI.
 - Recording is in the same format as that made by the server.
 - Both Recording and Playback require the BMLC2GUI to be subscribed to server STOMP. Recording captures STOMP from the server, while Playbacks are sent to the server via REST and then displayed on the map when returned through STOMP.
 - There a timescale factor in Config that speeds up play rate (0 plays all documents at once) and a timelimit factor in Config that automatically stops the player after a preset time.
 - Playing automatically turns on LISTEN FOR REPORTS.
- XML Document Form: auto-drawn form from JAXFront (under free community license) that displays and updates Config, Cyber, Initialization, Order and Report data
- Map Controls: rosette pans map; displays layers; manipulates map graphics (hand icon enables mouse-drag panning but takes away cursor arrow; to get it back click on arrow)
- Map and Image Panel: displays a map or image file with superimposed layers and graphics; has buffer for 500 Orders and/or Reports; when full it clears itself; when 80% full it warns

REST and STOMP in the GUI

Representational State Transfer (REST) is the new protocol for input to Web services, replacing the less efficient Simple Object Access Protocol (SOAP). Since version 2.0, the server uses only REST for input, so the GUI uses only REST to provide input to the server.

Stream Oriented Message Protocol (STOMP) listens to messages distributed by the server. The server has a capability to filter the message streams by Topic; the GUI listens to all topics. It displays on the map icons for all reports in C2SIM, IBML09, and C-BML formats. When a new report is posted for a unit, the old one is not removed; thus, the display shows the track of the unit. The map buffer holds up to 100 reports; when the buffer is full the icons are erased automatically. They also may be erased using the ERASE MAP GRAPHICS button.

File Menu of the GUI

The File dropdown menu controls input/output of Initialization messages, Orders, and Reports from the host computer file system and the XML display/edit panel and the map. When an Order or Report is read from the file system, it is automatically displayed in text on the JAXFront panel. Orders and Reports also are displayed graphically on the map when they are read. Options exist for C2SIM, IBML09 and C-BML message formats. (At present the C2SIM format deals with only the CWIX 2018 draft of C2SIM.)

The options allow creation of a new Initialization message, Order or Report; reading of Initialization message, Order, or Report from the file system; pushing the last Initialization message, Order or Report that was read to the server via REST; and saving an Order, Report or Initialization message from the editing form to the file system.

When subscribed to the server (which happens automatically on startup if so configured in the configuration panel) and LISTEN FOR REPORTS is selected, one copy of any incoming C2SIM Report will be displayed on the map. The map auto-centers to the first report received; thereafter it is possible for received reports to lie beyond the part of the map shown – use zoom to see them. There is a button LOAD SERVER REPORT in the map control panel that will open the latest report received in this way into the display/editing form.

The File menu also includes the ability to send control messages to the C2SIM Reference Implementation: SHARE+STOP, SHARE, STOP, START, PAUSE, RESET, STOP_RESET. A password is required (see Config Menu below). These controls are important for assembling and distributing C2SIM_MilitaryOrganization messages, which take the place of MSDL under C2SIM. Just below the File menu tab there are status displays for the server connection and the status of the coalition of C2 and simulation systems supported by the server.

JAXFront Form Use With the GUI

JAXFront provides an open source capability to draw input/output forms based on an XML Schema (XSD) file. The BMLC2GUI uses this to produce a form automatically for Initialization Messages, Orders and Reports created new or read from the file system. For C2SIM format these can be edited and saved to the file system. There are two subpanels; the left one uses the

top-level structure to control what part of the message is shown in the form, while the right one provides the form itself. To reset either of these, click on the small box on the vertical border and drag the box left or right. It may be necessary to click on elements within the form or “New” to display the form that is needed. After creating or editing a form it can be saved by selecting “Save JAXFront” under the “File” dropdown.

Config Menu of the GUI

The Config dropdown menu uses JAXFront to display GUI options in a form and allow them to be updated. Use Load to display the options; after editing them use Save to persist the values. The contents of the form are set by files in the BMLC2GUI/Config folder and must be coordinated with load/save of values in module BMLC2GUI.java. Here are some of the most important options:

- Map Scale: at startup
- Center Unit: center on the map the first unit loaded from file system or subscription
- Start Subscribed (causes GUI to subscribe to the STOMP server at startup)
- CBML, IBML and C2SIMns: namespace to be used in interpreting messages of these types
- Order Server: IP address of the server where the GUI will push orders
- Report Server: IP address of the server will subscribe for reports
- AutoDisplayReports: C2SIM, IBML or CBML will show reports of that type when LISTEN FOR REPORTS is selected
- Server Password: required for server control messages.
- Init Map Lat and Lon: Center of the map at startup

Maps in the GUI

Zoom within the map can be achieved using the mouse wheel. Panning can be done only by using the pan icon (arrowheads at NSEW) no using mouse. Layers, if present can be turned on using the layers icon (multiple rectangles stacked). In principle the map can be replaced by .shp (“shape”) files and graphics such as TIFF. (See *The Open Map Developer’s Guide*.) In practice, we have not been successful with this.