

# JTLS-GO

## Version Description Document

June 2022



DEPARTMENT OF DEFENSE  
JOINT STAFF J7  
116 LAKE VIEW PARKWAY  
SUFFOLK, VA 23435-2697

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS  
(JTLS-GO 6.1.6.0)

[Blank Page]

## ABSTRACT

The Joint Theater Level Simulation - Global Operations (JTLS-GO<sup>®</sup>) is an interactive, computer-based, multi-sided wargaming system that models air, land, naval, and Non-Governmental Organization (NGO) functions within a combine joint and coalition environment.

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.1.6.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.1.6.0 is a Maintenance release of the JTLS-GO 6.1 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as a major model functionality improvement implemented as Engineering Change Proposals (ECPs). This ECP is summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

This publication is updated and revised as required for each Major or Maintenance version release of the JTLS-GO model. Corrections, additions, or recommendations for improvement must reference specific sections, pages, and paragraphs with appropriate justification and be forwarded to:

JTLS-GO Development Team Leader  
ROLANDS & ASSOCIATES Corporation  
120 Del Rey Gardens Drive  
Del Rey Oaks, California 93940 USA  
jtlsgo@rolands.com

Copyright 2022 - ROLANDS & ASSOCIATES Corporation - All Rights Reserved

[Blank Page]

## TABLE of CONTENTS

ABSTRACT .....	iii
1.0 INTRODUCTION .....	1-1
1.1 SCOPE .....	1-1
1.2 INVENTORY OF MATERIALS .....	1-1
1.2.1 Obsolete/Outdated Documents.....	1-1
1.2.2 Unchanged Documents .....	1-1
1.2.3 Updated Documents.....	1-2
1.2.4 New Documents.....	1-2
1.2.5 Delivered Software Components .....	1-2
1.2.6 Released Databases.....	1-4
1.3 INTERFACE COMPATIBILITY .....	1-5
1.3.1 Support Software .....	1-5
1.3.2 JTLS-GO Cybersecurity Compliance .....	1-7
1.3.3 JTLS-GO High Level Architecture Compliance .....	1-8
1.4 DATABASE MODIFICATIONS .....	1-8
1.4.1 JTLS-GO Using Legacy Default Symbol Set .....	1-9
1.4.2 JTLS-GO Using New Default Symbol Set.....	1-9
1.4.3 Standard Repository Changes .....	1-9
1.5 INSTALLATION .....	1-9
2.0 ENGINEERING CHANGE PROPOSALS.....	2-1
2.1 JTLS-2022-15742 GLASSFISH COMMUNICATION WITH POSTGRESQL VIA SSL .....	2-1
3.0 SOFTWARE TROUBLE REPORTS .....	3-1
3.1 JTLS-2022-15693 CRASH PUSHING CHANGE MANUAL PAIR FLAG .....	3-1
3.2 JTLS-2022-15695 MAKE SEALANE OPTIMIZATION CONSISTENT .....	3-1
3.3 JTLS-2022-15698 NETWORK LOGIC ERROR INSUFFICIENT INFORMATION .....	3-1
3.4 JTLS-2022-15699 IMPROVE SVP WARNING 1277 .....	3-1
3.5 JTLS-2022-15701 CONFUSING SVPR LANGUAGE .....	3-2
3.6 JTLS-2022-15703 UPDATE SVP WARNING 1108 .....	3-2
3.7 JTLS-2022-15705 SVP WARNING 1457 NUMERICAL VALUES .....	3-2
3.8 JTLS-2022-15707 CONVOY TRANSPORTED UNIT NAME .....	3-2
3.9 JTLS-2022-15709 I18N PROPERTIES GENERATOR TRANSLATION .....	3-2
3.10 JTLS-2022-15711 SVP WARNING 1333 CONFUSING .....	3-3
3.11 JTLS-2022-15713 DDS AUTO DISPLAY NEW CHECKBOX .....	3-3
3.12 JTLS-2022-15715 CRASH FLOATING POINT VIOLATION .....	3-4
3.13 JTLS-2022-15717 AIR TRANSPORT SUPPLIES OFFLOAD NEVER FINISHED .....	3-4
3.14 JTLS-2022-15719 DUPLICATE SVP WARNINGS IDENTIFY SAME PROBLEM .....	3-4
3.15 JTLS-2022-15722 BATTERY PERCENT FOR NON-NAVAL UNIT CRASH .....	3-5
3.16 JTLS-2022-15724 CRUISE MISSILES NOT DISPLAYING WITH TRACK NAME .....	3-5
3.17 JTLS-2022-15725 WHIP ORDER MULTIPLE GROUP AUTO-SELECT FIELD .....	3-5

- 3.18 JTLS-2022-15726 REMOVE OLD SAMPLE E-MAIL ADDRESSES ..... 3-6
- 3.19 JTLS-2022-15727 FORCED DOWNLOAD JODA WEAPON COUNT ERRORS ..... 3-6
- 3.20 JTLS-2022-15729 PERIODIC REPORT AIR MISSION MESSAGE FORMAT ERROR .. 3-6
- 3.21 JTLS-2022-15730 CHANGE CLASSIFICATION INCORRECT PROCEDURES ..... 3-6
- 3.22 JTLS-2022-15732 MERGING FORMATION WITH GROUND WAIT SHIPS CRASH ..... 3-6
- 3.23 JTLS-2022-15734 INACTIVE FORMATION STARTS TO PATROL CRASH ..... 3-7
- 3.24 JTLS-2022-15736 MISSION REPORT PRIMARY TARGET NO LOCATION ..... 3-7
- 3.25 JTLS-2022-15738 JTLS ORDER SCHEMA MODIFIED ..... 3-7
- 3.26 JTLS-2022-15739 WHIP LARGE NETWORK DISPLAY MAP IMPROVED ..... 3-7
- 3.27 JTLS-2022-15740 CONVOY SUPPLIES NOT ALWAYS ON JODA ..... 3-8
  
- 4.0 REMAINING ERRORS..... 4-1
  - 4.1 DDSC – TMU LINE MODE CHANGES MULTIPLE GRIDS ..... 4-1
  - 4.2 DDSC – MULTIPLE TYPES IN DDS HISTORY TABLE ..... 4-1
  - 4.3 WHIP - PIPELINE NOT SHOWN ON IMT ..... 4-1
  - 4.4 DDSC/WHIP/JOBE - CADRG MAP ZOOM ..... 4-1
  
- APPENDIX A. ABBREVIATIONS AND ACRONYMS .....A-1
  
- APPENDIX B. VERSION 6.1.0.0 DATABASE CHANGES ..... B-1
  
- APPENDIX C. VERSION 6.1.0.0 REPOSITORY CHANGES ..... C-1

## 1.0 INTRODUCTION

### 1.1 SCOPE

This *JTLS-GO Version Description Document (VDD)* describes Version 6.1.6.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO<sup>®</sup>) software suite. JTLS-GO 6.1.6.0 is a Maintenance delivery for the JTLS-GO 6.1 series of releases.

JTLS-GO 6.1.6.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations called “wespac61”. There were no database format changes made to support this maintenance release. Database modifications that were accomplished to upgrade the previous JTLS-GO database format to the JTLS-GO 6.1 series format were summarized in the VDD for Version 6.1.0.0 delivered in the documents directory.

This release is being provided on a short notice requirement because of a Joint Staff/J7 security concern that both the Database Development System (DDS) and the After Action Review (AAR) Glassfish server could not be configured to communicate with the PostgreSQL server via a Secure Socket Layer (SSL). The correction to this problem has been labeled as an Engineering Change Proposal (ECP) and is described in detail in [Chapter 2.0](#). A summary of all Software Trouble Reports (STRs) corrected in this Maintenance delivery are summarized in [Chapter 3.0](#).

JTLS-GO 6.1.6.0 executes on the Red Hat Enterprise Linux Version 8.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP<sup>®</sup>) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

### 1.2 INVENTORY OF MATERIALS

This section lists documents and software that are relevant to JTLS-GO. All JTLS-GO documents included in this delivery are provided in PDF format within the documents subdirectory.

#### 1.2.1 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

#### 1.2.2 Unchanged Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.1.2.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 02, Version 6.1.2.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.1.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.1.0.0)

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.1.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.1.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.1.4.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.1.2.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.1.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.1.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.1.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.1.2.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.1.2.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.1.0.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.1.2.0)
- *JTLS-GO ATO Services User Guide* (JTLS-GO Document 24, Version 6.1.0.0)

### 1.2.3 Updated Documents

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.1.6.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.1.6.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.1.6.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.1.6.0)

### 1.2.4 New Documents

No new documents are required for this version of the software.

### 1.2.5 Delivered Software Components

JTLS-GO 6.1.6.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)
- Scenario Initialization Program (SIP)

- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)
- JTLS Symbols Application (JSYMS)
- Database Development System (DDS)
  - Database Configuration Program (DCP)
  - DDS Client User Interface (DDSC)
- ATO Translator Service (ATOT)
- ATO Generator Service (ATOG)
- ATO Retrieval Program (ATORET)
- JTLS Convert Location Program (JCONVERT)
- Count Critical Order Program (CCO)
- JTLS HLA Interface Program (JHIP)
- After Action Review Client (AARC)
- Scenario Data Client (SDC)
- Order Entry Client (OEC)
- Order Verification Tool (OVT)
- JTLS Object Distribution Authority (JODA)
  - The current JODA build number is 173.
- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:
  - Apache Server (APACHE)
  - JTLS XML Serial Repository (JXSR)
  - Order Management Authority (OMA)
  - Synchronized Authentication and Preferences Service (SYNAPSE)
  - XML Message Service (XMS)

### Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service
- Keyhole Markup Language (KML) Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBIE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool
- JTLS Version Conversion Program (VCP)

VCP60 - Converts a JTLS-GO 5.1 database to a JTLS-GO 6.0 formatted database.

VCP61 - Converts a JTLS-GO 6.0 database to a JTLS-GO 6.1 formatted database.

Instructions for installing JTLS-GO 6.1.6.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.1.6.0 is not necessary. No other upgrade beyond installation of the compressed TAR files or CD is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

#### 1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called “repository61”. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space.
- The scenario “wespac61”, which is suitable for training and demonstrations.

## 1.3 INTERFACE COMPATIBILITY

### 1.3.1 Support Software

JTLS-GO 6.1.6.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Server (ES) Edition Version 8.4, 64-bit architecture.

JTLS-GO 6.1 has been tested with the following versions of Linux 8:

RedHat Linux 8.4 - this operating system license must be purchased.

Oracle Linux 8.4 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

CentOS 8.4 - a free version of Linux 8 that has not been approved by DISA for use by U.S. Government agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.1.6.0 has been tested on the following operating systems:

Red Hat Linux Enterprise Edition Version 7.9 and 8.4

CentOS Linux Version 7.9 and 8.4

Windows 10, which can be used only if the workstation is an external HTTP client of the simulation network.

- JTLS-GO 6.1.6.0 is delivered with the AdoptOpenJDK 1.8 Update 332 package, which is equivalent to the current version of OpenJDK. Using AdoptOpenJDK provides two benefits:
  - a. Only the JTLS-GO account on the system servers access this version of Java. An installation site can use the JTLS-GO servers for programs other than JTLS-GO without impacting the version of Java used by other programs.
  - b. Security releases of AdoptOpenJDK software are produced on the same schedule as the Oracle OpenJDK security release procedure. An organization can expect to receive a bug release version of JTLS-GO within two-weeks of a new Java 1.8 security release. As long as a user organization installs all of the JTLS-GO bug releases, JTLS-GO can guarantee that the latest Java security release is being used on the servers. JTLS-GO no longer depends on system administration for implementing proper Java security update procedures.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.4.
- JTLS-GO database tools require a certified PostgreSQL 11.13 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.1.6.0. There are several alternative methods available for obtaining the PostgreSQL 11.13 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.
- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of the supported versions of Red Hat Linux ES.
- TCP/IP is required for inter-process communication between the JODA data server and all user interface programs. The version of TCP/IP included with the supported versions of Red Hat Linux ES is sufficient.
- The Perl script language is used by the JTLS-GO system and game setup scripts. The version of Perl included with the supported versions of Red Hat Linux ES is sufficient. The Perl program is typically located in the `/usr/bin` directory. If Perl is installed in a another location, a link should be created from the `/usr/bin` directory to this program.
- SIMSCRIPT III (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc.
- ANSI C Compiler: It is not necessary to use a C compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO component programs. The C Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.
- C++ Compiler: It is not necessary to use a C++ compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO HLA component programs. The C++ Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.
- The JTLS-GO DDS application uses these open source libraries:

JFreeChart, licensed under a GNU Lesser General Public License (LGPL) by Object Refinery Limited, <http://www.object-refinery.com>

JCommon, licensed under LGPL2.1 (GNU Lesser General Public License version 2.1 or later) by Object Refinery Limited, <http://www.object-refinery.com>

Commons-math3-3.0.jar, licensed under Apache Software Foundation (Apache License, Version 2.0) <http://www.apache.org/licenses/LICENSE-2.0> OHLA Compliance

- KML Operational Interface (KOI)

The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth™. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using an operational Command, Control, Communication, Computer Information (C4I) systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

Chapter 3 of the *JTLS-GO C4I Interface Manual* describes requirements and procedures for using the KOI capabilities.

### 1.3.2 JTLS-GO Cybersecurity Compliance

Because of recent incidents of intrusions into software systems, the United States Department of Defense (DoD) has implemented a strong and strictly enforced Cybersecurity program. JTLS-GO, as software that executes on DoD systems, must comply to the mandates of the program, along with all of the third party software used by JTLS-GO, such as PostgreSQL and Java.

One of the DoD requirements is that the software must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. After a certain date, all Java related programs stopped working, whether connected to an open network or not. All JTLS-GO releases were closely linked to the Java expiration date.

OpenJDK™ has not implement an expiration date. In order to fulfill this DoD Cybersecurity requirement, JTLS-GO has moved to AdoptOpenJDK, a full OpenJDK Java environment with licensing alternations allowing an application to deliver the software. The following procedure has been established and approved by the JS/J7 Cybersecurity branch:

- Within days of an Oracle Java security release, AdoptOpenJDK produces an equivalent version using infrastructure, build and test scripts to produce pre-built binaries of the OpenJDK class libraries. All AdoptOpenJDK binaries and scripts are open source licensed and available for free.

- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a bug release version (JTLS-GO 6.1.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the bug released versions when distributed.

Contact the U.S. Government Program Manager, Mr. James Blank (james.e.blank.civ@mail.mil) to obtain the completed Cybersecurity paperwork and a current Gate completion certificate.

### 1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.1.6.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

The solution combines JTLS-GO with CAE's GESI constructive tactical entity-level simulation system. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The GESI system is used to represent a virtual battlefield, including weapons, vehicles, aircrafts, ground forces and more.

Combining JTLS-GO and GESI brings together operational and tactical level constructive simulations to prepare commanders and staff to make timely, informed and intelligent decisions across the full spectrum of operations, including conventional combat, disaster relief, and operations other than war.

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the \$JGAME/data/hla directory, Federation testing of JTLS-GO with CAE's GESI model have been accomplished. Future plans include expanding the capabilities of the GlobalSim federation.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.1.6.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation ([www.pitch.se](http://www.pitch.se)). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

## 1.4 DATABASE MODIFICATIONS

Significant database structure differences exist between the JTLS-GO 6.1 series of the software and the previous JTLS-GO 6.0 series database structure. There are no database format changes between JTLS-GO 6.1.6.0 and the initial JTLS-GO 6.1.0.0 version of the software.

To upgrade your JTLS-GO 6.0 scenario to JTLS-GO 6.1 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

STR JTLS-2022-15607 Grid Lower-Left Location Not Exact On Restart was included in JTLS-GO 6.1.2.0, released in March 2022.

This STR required a change to the ASCII checkpoint format. Any saved checkpoints from JTLS-GO 6.1.0.0 and JTLS-GO 6.1.1.0 will not work with JTLS-GO 6.1.3.0.

#### 1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 6.1.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.1.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario\_name>.gs and databases symbol.scf file.

#### 1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.1.6.0, but end-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

This means that the user can easily move in this new symbol set. Please follow the steps outlined in the *JTLS-GO DDS Users Guide*, Section B.13, Updating Scenario Symbol Set.

#### 1.4.3 Standard Repository Changes

R&A has continued to improve and expand the unclassified data repository, which has been renamed to “repository61.” The DDS comparison and synchronization function can be used to determine if any of the changes delivered are of use to a JTLS-GO user organization.

### 1.5 INSTALLATION

The *JTLS-GO Installation Manual*, a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery. It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 11.13 required to operate JTLS-GO 6.1.6.0.



## 2.0 ENGINEERING CHANGE PROPOSALS

This section describes the one new capability, known as an Engineering Change Proposal (ECP), that is delivered with this release.

### 2.1 JTLS-2022-15742 Glassfish Communication With PostgreSQL Via SSL

#### Summary of Model Change Request

As required by the Joint Staff / J7 (JS/J7) Cyber-security team, JTLS-GO needed to implement a new capability in which the Database Development System (DDS) Glassfish instance and the After Action Review (AAR) Glassfish instance could, if desired, communicate with the PostgreSQL server via a Secure Socket Layer (SSL).

In previous versions of JTLS-GO, client software, such as the Database Development System Client (DDSC) and the Web Hosted Interface Program (WHIP), can be setup to communicate with Glassfish via a secure connection. The issue is that Glassfish cannot communicate with the PostgreSQL server using SSL. The purpose of this ECP is to solve the noted issue.

#### Design Summary

This is an extensive change to the manner in which Glassfish communicates with the PostgreSQL server. Complete instructions are included in the *JTLS-GO Installation Manual*, the *JTLS-GO DDS User Guide*, and the *JTLS-GO Technical Controller Guide*.

- To be in full compliance with the JS/J7-mandated use of JTLS-GO, all US Department of Defense (DoD) users should follow the instructions to implement the SSL communication layer between each Glassfish instance and the PostgreSQL server.
- A non-DoD user agency is under no obligation to use the new SSL communication layer between Glassfish and the PostgreSQL server, but insider threat protections are compromised without the use of this newly implemented capability.

#### Non-SSL Option Between Glassfish and PostgreSQL

If SSL is not to be used, no changes in procedures or data files are required after installing this version of JTLS-GO. The following summarizes the suggested steps that should be taken to ensure the new Glassfish operates:

- All JTLS-GO programs should be stopped, including Glassfish.
- The new software should be installed.
- The new ".cshrc" should be moved in with all of your system definitions. The easiest way to do this is to compare our ".cshrc" file with your existing file.

- The directories `dds_as` and `aar_as`, found under the `$JGAME/web_active` directory, should be deleted.
- The Database Configuration Program (DCP) for each of your active scenarios should be started, and a save should be performed to deploy a new domain.
- The Interface Control Program (ICP) for each of your active scenarios, that use the AAR capability and have an AAR Glassfish marked as active, should be started and a save should be performed to deploy a new domain.

### SSL Option Between Glassfish and PostgreSQL

If SSL between Glassfish and PostgreSQL is to be used, besides installing this new JTLS-GO software release, the following steps must be taken.

- Step 1: Alter the `.pgpass` file in the `$JGAME/data_site` directory using any text editor. Each scenario should be specified to use SSL. The following is an example of a scenario entry in this file. The text highlighted in "Red" is the text that needs to be added to each line in the `.pgpass` file and indicates SSL communication should be used between Glassfish and the PostgreSQL server.

```
radius:5432:jtls_db:wespac61:jtls61:sslmode=true
```

- Step 2: Alter the `pg_hba.conf` file that exists under the PostgreSQL account. The file is located in the directory defined by the `$PGDATA` parameter, which is normally set within the PostgreSQL account to be `$HOME/pgsql/data/pg11`. Because JTLS-GO is delivered with a containerized version of PostgreSQL, only the JTLS-GO account is accessing the database server. For this reason, the `pg_hba.conf` file is greatly simplified and can easily be set to insist that all JTLS-GO connections to the PostgreSQL server use SSL.

Only one line is required in the `pg_hba.conf` file. This entry specifies that the SSL encryption uses the Defense Information Security Agency (DISA) required `scram-sha-256` algorithm. It also indicates that JTLS-GO only support SSL in the `verify-full` mode. The suggested `pg_hba.conf` entry for use when running with SSL is as follows:

```
hostssl jtls_db +jtls_role samenet scram-sha-256 clientcert=1
```

If your organization is not using the containerized version of PostgreSQL delivered with JTLS-GO, your database manager will need to update the `pg_hba.conf` file to indicate that all JTLS-GO connections will be accomplished via SSL. More details can be found in the *JTLS-GO Installation Manual*.

- Step 3: Tell the JTLS-GO account that only the `verify-full` mode of SSL is supported. There is a commented-out line in the `.cshrc` file that reads:

```
# setenv PGSSLMODE verify-full
```

It should be uncommented, so it will be executed each time the ".cshrc" script is executed. The line should read as follows:

```
setenv PGSSLMODE verify-full
```

- Step 4: Create needed SSL certificates, so the SSL connection can be established. R&A has developed a script to help with this process. To run the certificate creation script, enter the following command while in the \$JHOME directory of the JTLS-GO account:

```
create_ssl_certificate -H <PGServerName> -N <FullPGServerName> -d /tmp
```

For example, at the R&A simulation center, the name of our PostgreSQL server is "rapgsq1", and we initiate the certificate generation script by entering the following command:

```
create_ssl_certificate -H rapgsq1 -N rapgsq1.rolands.com -d /tmp
```

- Step 5: The script generates three files in the /tmp directory: <PGServerName>.cert, <PGServerName>.key, which must be moved to the PostgreSQL account. The third file <PGServerName>.key.pk8 is not needed and should be ignored.

Move the two needed certificate files over to the \$PGDATA directory under the PostgreSQL account. Ensure the files are owned by the PostgreSQL account and have the following access codes: **0600**.

- Step 6: JTLS-GO is delivered with several general purpose client certificates in the JTLS-GO account \$JGAME/data/certificates directory: These certificates are jtls\_client.crt, jtls\_client.key, and jtls\_client.key.pk8. Using these general purpose certificates eliminates the need to create certificates for every JTLS-GO server machine. Within the delivered JTLS-GO ".cshrc" file, there are three commented-out lines that read:

```
# setenv PGSSLKEY $JGAME/data/certificates/jtls_root.crt  
# setenv PGSSLCERT $JGAME/data/certificates/jtls_client.crt  
# setenv PGSSLKEY $JGAME/data/certificates/jtls_client.key
```

These three lines need to be uncommented so the code will be executed each time the ".cshrc" script is executed. These three lines should read as follows:

```
setenv PGSSLKEY $JGAME/data/certificates/jtls_root.crt  
setenv PGSSLCERT $JGAME/data/certificates/jtls_client.crt  
setenv PGSSLKEY $JGAME/data/certificates/jtls_client.key
```

- Step 7: The PostgreSQL account also needs access to these general purpose client certificates. Place a copy of the `jtls_root.crt`, `jtls_client.crt`, and `jtls_client.key` certificates into the `$PGDATA` directory in the PostgreSQL account. Again they should be owned by the PostgreSQL account and have an access code of 0600. In addition, add the following environment variables in the `$HOME/.bash_profile` file of the PostgreSQL account:

```
export PGSSLROOTCERT=$PGDATA/jtls_root.crt
export PGSSLCERT=$PGDATA/jtls_client.crt
export PGSSLKEY=$PGDATA/jtls_client.key
```

- Step 8: In the same `$PGDATA` directory is a file named `postgresql.conf`. This file also needs to be changed to tell the PostgreSQL server that the password encryption algorithm being used is the DISA required `scram-sha-256` algorithm.

```
#md5 or scram-sha-256
```

It should be changed to read:

```
password_encryption = scram-sha-256
```

- Step 9: Within the same `postgresql.conf` file, there is a section of commented-out code labeled "SSL", indicating SSL is not being used for communication with the server. It appears as follows:

```
# - SSL -
#ssl = off
#ssl_ca_file = "
#ssl_cert_file = 'server.crt'
#ssl_crl_file = "
#ssl_key_file = 'server.key'
#ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL' # allowed SSL ciphers
#ssl_prefer_server_ciphers = on
#ssl_ecdh_curve = 'prime256v1'
#ssl_dh_params_file = "
#ssl_passphrase_command = "
#ssl_passphrase_command_supports_reload = off
```

This section of the file needs to read as follows. Only the lines requiring changes are highlighted in “Red”, and the certificate names generated in Step “4” and described in Step “5” are shown in “Blue”. The “Blue” entries should be replaced with the proper certificate names generated in Step "4".

```
# - SSL -
ssl = on
ssl_ca_file = 'jtls_root.crt'
```

```
ssl_cert_file = 'rapgsql.rolands.com.crt'  
#ssl_crl_file = "  
ssl_key_file = 'rapgsql.rolands.com.key'  
ssl_ciphers = 'TLSv1.2:!aNULL'  
ssl_prefer_server_ciphers = on  
#ssl_ecdh_curve = 'prime256v1'  
#ssl_dh_params_file = "  
#ssl_passphrase_command = "  
#ssl_passphrase_command_supports_reload = off
```

- Stop the database server using your normal procedures.
- Sign out of your PostgreSQL account.
- Re-login to your PostgreSQL account and restart the PostgreSQL server.

All of these steps are described in more detail in Chapter 6 of the *JTLS-GO Installation Manual*.



### 3.0 SOFTWARE TROUBLE REPORTS

Software Trouble Reports (STRs) describe software code errors that have been discovered by JTLS-GO users or developers and have been corrected.

#### 3.1 JTLS-2022-15693 Crash Pushing Change Manual Pair Flag

The model crashed while re-running a portion of an exercise in which a Change Mission Parameter order was submitted via the JTLS-GO Push capability. The re-run portion of the exercise resulted in a mission being destroyed prior to the receipt of the Change Mission Parameter order to change the mission's Manual Pair Flag. The model attempted to change the flag for the destroyed mission, and the model crashed.

The code properly checked to determine if the mission still existed or not. In this case, the model correctly determined that the mission did not exist, but the model incorrectly did not pass over the change Manual Pair portion of the logic. The code logic error was corrected.

#### 3.2 JTLS-2022-15695 Make Sealane Optimization Consistent

The individual ship sealane optimization algorithm is executed when a direct great circle path will run into land. This algorithm has no distance or time limits imposed under the assumption that if a ship cannot travel in a straight line, a sealane is the best and only choice. The formation version of this algorithm did impose a distance and time limit, which is inconsistent.

The formation algorithm was altered to remove the distance and time limitations for finding a sealane route. The two algorithms are now consistently implemented.

#### 3.3 JTLS-2022-15698 Network Logic Error Insufficient Information

A logic error was generated that indicated a road arc had a distance of zero. Once the logic error is generated, it is not possible to determine which road arc was causing the problem; therefore, Technical Control cannot find or solve the issue.

The code was changed to provide the information. When the logic error is found again, the underlying issue can be found and solved.

#### 3.4 JTLS-2022-15699 Improve SVP Warning 1277

The wording for SVP Warning 1277 is confusing, and an option to display the HUP entire combat system list needed to be added.

The current wording of the SVPR correction list was reviewed and some modifications made, and the option to display the entire HUP combat system list for review and modification was added.

### 3.5 JTLS-2022-15701 Confusing SVPR Language

**The phrase "Automatically Set..." and similar phrases in SVP Error and Warning messages can be confusing for some of our foreign customers.**

The language for Warning 1279 was changed from "Automatically Set" to "Set". A spelling error in Warning 1360 was corrected.

### 3.6 JTLS-2022-15703 Update SVP Warning 1108

**SVP Warning 1108 needs the capability to set the AIR OPS CAPABLE flag to NO.**

The capability to set the AIR OPS CAPABLE flag to NO was add to Warning 1108.

### 3.7 JTLS-2022-15705 SVP Warning 1457 Numerical Values

**The numerical values in SVP Warning 1457 are confusing.**

Warning 1457 computes the actual Basic Load value need by the prototype for a specific supply category, but the displayed value is slightly higher than the precisely computed value. When using the automatic correction capability associated with Warning 1457, the rounded up value is used to update the database. This slight round-up is needed to prevent the continued generation of the warning since comparing two computed real values is not always accurate portrayed within the computer memory. The wording of the warning and its automatic corrections were modified to properly display and explain the two different values.

### 3.8 JTLS-2022-15707 Convoy Transported Unit Name

**A convoy completed delivery of a transported Unit. The Supply Run IMT continued to display the name of the transported unit as the empty convoy returned to its origination.**

The routine that updated the JTLS Object Data Authority (JODA) was incomplete. The name of the transported Unit was displayed while loaded on the convoy in the Supply Run IMT, but did not remove the name after the Unit was unloaded. The missing JODA Data System (JDS) update was added.

### 3.9 JTLS-2022-15709 i18n Properties Generator Translation

**The user wanted to translate all files listed by the i18n Properties Generator, except DDS-related files. The language selected was Chinese - Taiwan (zh\_TW). The i18n started the translation with the WHIP order files. The translation proceeded for a time, but eventually every line fed to the i18n for translation resulted in an exception, with the message "[google-api-translate-java] Error retrieving translation".**

When the i18n Properties Generator receives a line of text to translate, it creates and runs a thread. This allows multiple lines in a given file to be translated concurrently. Some files will have

hundreds, and even thousands, of lines to be translated. Having a large number of concurrent threads all making translation requests to the Google Translate API may overwhelm the API. The API may start throwing exceptions with the "Error retrieving translation" message. This was not an issue when the i18n was initially developed. It may be possible that Google has either recently introduced an additional quota limit, or put more restrictions on an existing quota - it is not clearly documented.

Although the exception and its message does not indicate what the problem is, we believe that one of the following two quotas has been reached:

- The number of translation requests per unit of time.
- The number of characters to translate per unit of time.

Based on these quotas, the i18n no longer creates and runs a thread for each line in the file. All lines are now translated sequentially on a single thread. If an exception with the "Error retrieving translation" message is now thrown, an attempt is made to translate the line again after a randomly drawn delay time.

Testing indicates that the delay for message (.mdf) files can be from 20 to 80 seconds, and for all other files this can range from 20 to 30 seconds. Up to five attempts can be made per line. This should be sufficient - testing showed that when multiple attempts had to be made, the number of attempts never exceeded two.

### 3.10 JTLS-2022-15711 SVP Warning 1333 Confusing

**SVP Warning 1333 indicates that an Aircraft Type holds a Aircraft Load within the Load Assignment Array used to attack a specific group of target types, but the Aircraft Load contains no weapons. The Warning referenced an outdated CRUISE load entry in the Load Assignment Array. This resulted in the generation of a totally meaningless Warning.**

The Warning was rewritten with proper variables. The SVPR correction messages were also rewritten to provide the user more information for correcting the problem.

The same problem existed for Error 317, where Warning 1333 is for aircraft types that are unused by any squadrons, while Error 317 identifies aircraft types that are assigned to squadrons. This problem was also corrected.

### 3.11 JTLS-2022-15713 DDS Auto Display New Checkbox

**The DDS showed the Auto Display New checkbox in the Area Filter Panel. This feature was a WHIP and TRIPP feature only.**

A flag was added to the WHIP/DDS shared code. This new flag prevents the creation of the Auto Display New checkbox to the DDS.

### 3.12 JTLS-2022-15715 Crash Floating Point Violation

The model crashed on game start during the routine CALCULATE.AREA.RATIO. The database contained a Combat Arms Type (CAT) for which there was no Command Control Prototype (CCP) density data. This presumably happened when a user did a copy of a CCP instead of a Deep Copy to create a new CCP. Therefore, there were no records in the CCP\_CAT\_BASE\_DENSITY, CCP\_CAT\_TT\_DENSITY and CCP\_CAT\_UP\_DENSITY tables. When the model was calculating the area ratios for the Tactical Unit Prototype/Command Control Prototype combinations, there was a divide by zero crash.

The database contains a default definition of 1.0 for each of the density parameters. When the model was reading in the data, since there were no density records associated with the Command Control Prototype, the density data was initialized to zero. Zero is not a legal value for the density data. The model was changed to initialize all density records to the default value of 1.0. Thus any missing density records will contain legal values within model and the crash is no longer possible.

### 3.13 JTLS-2022-15717 Air Transport Supplies Offload Never Finished

An Air Transport order was submitted, with a list of supplies to pick up from an airbase and deliver to a FARP. One of the supply categories in the list did not exist at the airbase. The mission loaded the supplies that were present and flew to the FARP as expected. This was confirmed in the Air Movement Report.

The supplies carried by the mission were offloaded at the FARP, but the mission remained in offload status well past the End Time displayed in the Air Tasking Order IMT display. The Manage Air Mission Tasks order was required to cancel the Offload task so that the mission could return home.

The logic that offloads each supply category from a mission uses two lists: the list of supplies carried by the mission and the list of supplies requested to be offloaded. When each supply category is offloaded, that category is removed from both the mission's list and the requested list. However, if the supply category was not present on the mission list, that category was not removed from the requested list. That logic flaw caused the offload process to continue indefinitely because there was always a supply category remaining in the requested list that could not be offloaded.

The error was corrected by adding code to remove the supply category from the requested list if no such category was carried by the mission. This solution now causes the offloading task to terminate normally.

### 3.14 JTLS-2022-15719 Duplicate SVP Warnings Identify Same Problem

SVP Warnings 1254 and 1476 both identify the same problem concerning available aircraft fuel at airbases.

When the capability to represent multiple fuels types within JTLS-GO was implemented, the code that checks Airbases and Naval units was split out from the original code and put into unit type-specific checks. The problem was encountered because the original checks were not removed from the code, so the same problem was being identified in two different routines. This was corrected.

While doing so, the Warning number was also changed and moved from the 1200 series, which are reserved for prototype warnings, to the 1400 series, which is reserved for unit warning. This change was appropriate since units can now have alternative initial supply data within the database. The majority of supply checks are now done at the unit level, vice the prototype level.

### 3.15 JTLS-2022-15722 Battery Percent For Non-Naval Unit Crash

**The Set Unit Data order did not have any constraints on the type of unit that could be input for changing a unit's Remaining Battery Percent. It was possible to send an order setting the value for an Airbase or any other type of unit. This caused a crash in the CEP if the Tactical Unit Prototype (TUP) index for the type of unit was greater than the number of Ship Unit Prototypes.**

The order contained numerous fields that had constraints to limit filling them by the type of unit involved. The Percent Battery Remaining was just another example of this type of constraint, but instead of putting it in as a constraint, the order was re-worked so all those constraints became auto-select groups, thus limiting the user's ability to enter the wrong field data in the first place.

At the same time it was noted that the Controller could change the reason for a unit attachment from Reconstitution to Task Organization or vice-versa. It was felt that this should not be permitted since it would alter any TOE numbers if the value was changed and the unit was detached. This field was removed from the order.

### 3.16 JTLS-2022-15724 Cruise Missiles Not Displaying With Track Name

**Cruise missiles are appearing on the WHIP without an associated real world or track name. This makes it impossible to select the missile and receive any Situation Report (SITREP) information.**

The problem was tracked down to the initialization routine for cruise missiles to the JODA. A check to see if the last location time was greater than zero was being made to determine if a Side should know about the cruise missile. During creation of the cruise missile the last location time is set to zero if the cruise missile is known and less than zero if it is not. The check in question should have checked if the last location time was greater than or equal to zero. This was appropriately changed.

### 3.17 JTLS-2022-15725 WHIP Order Multiple Group Auto-Select Field

**Designating an individual order field to be auto-selected from within more than one group of fields was found to not work.**

The orders were changed so that any individual order field can be assigned to multiple groups as an auto-select option for each group.

### 3.18 JTLS-2022-15726 Remove Old Sample E-Mail Addresses

**The sample e-mail addresses delivered with JTLS-GO no longer exist.**

The sample e-mail addresses were altered to match real-world e-mail addresses.

### 3.19 JTLS-2022-15727 Forced Download JODA Weapon Count Errors

**The JODA registered several Weapon Count errors when forcing a JODA download.**

The issue was the Weapon Count initialization routine incorrectly assigned a new receiver number to Weapon Count entities when forcing a download. Instead the model should have used the existing receiver numbers.

While investigating this problem we noted several routines that did not following programming standards. The code was not wrong, but the routines were fixed to follow the proper programming standards.

### 3.20 JTLS-2022-15729 Periodic Report Air Mission Message Format Error

**Message number 9040 is used to generate the Periodic Report Air Mission. An error was found in the format of this message while using the Message Delivery Program (MDP) to deliver messages to external files. An error message was visible on the Web Services Manager, which indicated that an invalid Mission Type was accessed in the scenario vocabulary data.**

This error was associated with a mismatch between the message data which was output by the CEP and the data format in the message definition file. The CEP was not sending any mission type data, but the format was expecting to receive that information. The message definition file was corrected to match the CEP output.

### 3.21 JTLS-2022-15730 Change Classification Incorrect Procedures

**The documentation in the Technical Controller Guide for changing a game's classification markings during game play were not accurate, and included a step that could have major negative consequences.**

The documentation was corrected and the corrected procedures tested.

### 3.22 JTLS-2022-15732 Merging Formation With Ground Wait Ships Crash

**The model crashed when Formation 1, which had been ordered to move to a land location, was merged with Formation 2. Formation 1's ships were in a Ground Wait state, waiting for the formation to leave its land location, so they could move to an appropriate stationing location. The**

merge was processed, but the first time Formation 2 moved, the newly added units crashed because they were no longer in a Ground Wait state, but held a Ground Wait task.

The cause of the issue was not in the formation move logic. The problem occurred when the merge procedure was executed. When the ships were removed from Formation 1, they should have cleared their wait station status. The code actually cleared the status, but too late in the logic sequence. The clearing of the ship's station wait status was moved above the logic that attempted to cancel the ship's wait task.

### 3.23 JTLS-2022-15734 Inactive Formation Starts To Patrol Crash

An inactive formation, with no ships, was given a Patrol order, and the model crashed as it was gathering noise data for each ship in the formation. There were no ships in the formation, and the model was reserving an array by the number of ships in the formation. You cannot reserve an array with length 0.

This was not a model code problem. The problem was that the Formation Patrol order allowed the user to send a Patrol order to an inactive formation. This should never have been allowed. The order panel was fixed, and the user will no longer be allowed to send a Patrol order to an inactive formation.

### 3.24 JTLS-2022-15736 Mission Report Primary Target No Location

The mission report always shows the location of the primary target as zero, zero.

An old variable was being used in the message. The new correct variable is now accessed and properly set the first time the mission drops or fires weapons. The old variable was removed from the model.

### 3.25 JTLS-2022-15738 JTLS Order Schema Modified

One of the attributes in the JTLS orders schema was set to the wrong value, causing an error when the checkorder script was run for the SET.UNIT.DATA order.

The minOccurs attribute was set to the proper value in the JTLS orders schema.

### 3.26 JTLS-2022-15739 WHIP Large Network Display Map Improved

After increasing the number of representative arcs and nodes for the sealane network, WHIP Map performance was severely reduced when sealanes were shown on the Map. The reduced performance was caused by the sealane map display layer populating the graphics for the network. The layer was also unnecessarily reprocessing of each of graphics' display setting when the sealane layer received periodic data updates, regardless whether or not the update contained updates to the current sealane data.

The sealane network layer was changed to only process the graphics contained within the Map's projected view for display. This significantly reduces the amount of work and load the layer has to perform in order to render the sealane graphics. The periodic update was corrected to avoid reprocessing the graphics when there is no update to the current data.

To help improve a world view of the sealane network, the nodes are set to be off until the user decides to turn on the display of node names. The same improvements are applied to all other Map network layers.

### 3.27 JTLS-2022-15740 Convoy Supplies Not Always On JODA

**A convoy carrying known supplies did not have all of its supplies displayed on the IMT. The record for the supplies was not in the JTLS Object Data Authority (JODA).**

The issue was tracked down to a forced download of current status data from the CEP to the JODA. The code was not initializing convoy supply categories that had a very small value for on hand. If a convoy is just starting to load supplies, it may have a small amount of supplies on board and so the forced download code skipped the supply category and did not send the object to the JODA. As supplies were loaded on the convoy, the proper convoy supply updates were being sent to the JODA, but were rejected because the object did not exist in the JODA.

## 4.0 REMAINING ERRORS

Every effort has been made to correct known model errors. All reproducible errors that resulted in CEP catastrophic software failures (crashes) have been corrected. Other corrections were prioritized and completed according to their resource cost-to-benefit relationship.

As JTLS-GO 6.1.0.0 represents a major release of new functionality, all outstanding errors have been reviewed. If the error could not be reproduced, it was considered obsolete and no longer relevant to JTLS-GO. These errors have been removed from consideration for correction at this time. This chapter contains the remaining errors that could be reproduced. As development team assets become available, these remaining known issues will be corrected.

In future maintenance releases, newly uncovered outstanding errors related to JTLS-GO will be listed in this chapter, along with information regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

### 4.1 DDSC – TMU Line Mode Changes Multiple Grids

When using the line mode in the TMU, more grids than the ones the line passes through are changed. This can also cause a warning about trying to change multiple layers to appear.

### 4.2 DDSC – Multiple Types In DDS History Table

If records for more than one table type are selected in the DDS History table, “History Details” will display details for only one type.

### 4.3 WHIP - Pipeline Not Shown On IMT

A pipeline being operated by a non-detected unit is not shown in the pipeline IMT.

### 4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

When using the CADRG map projection, if the width of the map is less than the height, the zoom tool does not work correctly.



## APPENDIX A. ABBREVIATIONS AND ACRONYMS

Terms are included in this Appendix to define their usage in JTLS-GO design, functionality, and documentation.

AAA	Anti-Aircraft Artillery
AADC	Area Air Defense Commander
AAL	Air-to-Air Lethality
A/C	Aircraft
ACP	Air Control Prototype
ADA	Air Defense Artillery
AEW	Airborne Early Warning
AFB	Air Force Base
AG	Air-Ground (Air-to-Ground)
AI	Air Interdiction
AIM	Air Intercept Missile
AIREF	Air Refueling
AKL	Area Kill Lethality
AMMO	Ammunition
AO	Area of Operations
AOC	Air Operations Center
APC	Armored Personnel Carrier
ARECCE	Armed Reconnaissance
ARTE	Air Route
ARTY	Artillery
ASC	Automatic Supply Calculation
ASCII	American Standard Code for Information Interchange
ASW	Anti-Submarine Warfare
ATC	Aircraft Target Category
ATGM	Anti-Tank Guided Missile
ATK	Attack
ATO	Air Tasking Order
ATORET	Air Tasking Order Retrieve Program
ATOT	Air Tasking Order Translator
AWACS	Airborne Warning And Control System
AZ	Altitude Zone

---

BADGE	Bilateral Air Defense Ground Environment (used by Japan Defense Agency)
BAI	Battlefield Air Interdiction
BDA	Battle Damage Assessment
BDE	Brigade
BN	Battalion
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CA	Civil Affairs
CADRG	Compressed ARC Digitized Raster Graphics
CAP	Combat Air Patrol
CAS	Close Air Support
CAT	Category
CCF	Central Control Facility
CCP	Command Control Prototype
CCU	Controller Change Unit
CEP	Combat Events Program
CMDR	Commander
COP	Common Operational Picture
CP	Combat Power
CS	Combat System
CSP	Combat System Prototype
CTAPS	Contingency Tactical Air Planning System
CTG	Commander Task Group
CTRL	Control keyboard command
DCA	Defense Counter Air
DCL	Digital Command Language
DDS	Database Development System
DEMSDB	Demonstration Standard Database
DISA	Defense Information Systems Agency
DIV	Division
DMA	Defense Mapping Agency
DoD	Department of Defense
DOS	Days of Supply

---

DPICM	Dual Purpose Improved Conventional Munitions
DS	Direct Support
DSA	Directed Search Area
DTG	Date Time Group
EC	Electronic Combat
ECM	Electronic Counter Measure
ECP	Engineering Change Proposal
EI	Essential Elements of Information
ELINT	Electronic Intelligence
ELS	Entity Level Server
EODA	Entity Level JTLS Object Data Authority
ETA	Estimated Time of Arrival
FARP	Forward Arming and Refueling Point
FLP	Fire Lethality Prototype
FLOT	Forward Location of Troops
FOL	Forward Operating Location
FWL	Frederick W. Lanchester (originated a differential equation model of attrition)
GAL	Gallon
GCCS	Global Command and Control System
GRTE	Ground Route
GS	General Support
GSR	General Support Reinforcing
GUI	Graphical User Interface
HARM	High-speed Anti-radiation Missile
HE	High Explosive
HELO	Helicopter
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HQ	Headquarters
HRU	High Resolution Unit
HTML	Hypertext Markup Language
HTT	High resolution unit Target Type
HUP	High resolution Unit Prototype
ICM	Improved Conventional Munitions
ICP	Interface Configuration Program

---

ICPLLogin	Interface Login Program
ID	Identifier
IFF	Identification Friend or Foe
IIP	Intelligence Information Prototype
IMT	Information Management Tool
INFO	Information
INTEL	Intelligence
JCATS	Joint Conflict And Tactical Simulation
JDA	Japan Defense Agency
JDPI	Joint Desired Point of Impact (formerly DMPI: Desired Mean Point of Impact)
JDS	JTLS Data System
JDSP	JTLS Data System Protocol
JEDI	JODA Entity Data Identifier
JMCIS	Joint Maritime Combat Information System
JMEM	Joint Munitions Effectiveness Manuals
JODA	JTLS Object Distribution Authority
JOI	JTLS Operational Interface
JPL	Jet Propulsion Laboratory
JRSG	Joint Rapid Scenario Generation (formerly JIDPS: Joint Integrated Database Preparation System)
JSDF	Japanese Self-Defense Force
JTLS	Joint Theater Level Simulation
JTLS-GO	Joint Theater Level Simulation - Global Operations
JTOI	JTLS Transaction Operational Interface
JXSR	JTLS XML Serial Repository
KIA	Killed In Action
KM	Kilometer
KNOTS	Nautical miles per hour
LA	Lethal Area
LAN	Local Area Network
LAT	Latitude
LB	Login Build (JTLS order type)
LDAP	Lightweight Directory Access Protocol
LDT	Lanchester coefficient Development Tool
LOG	Logistics

---

LOGIN	Logistics Input
LOGREP	Logistics Report
LONG	Longitude
LOTS	Logistics Over The Shore
LR	Long Range
M&S	Modeling and Simulation
MAPP	Modern Aids to Planning Program
MB	Megabyte
MCP	Mobility Counter-mobility Prototype
MCR	Model Change Request
MG	Machine Gun
MHE	Material Handling Equipment
MIP	Model Interface Program
MOGAS	Motor Gasoline
MOPP	Mission-Oriented Protective Posture
MOSAIC	NCSA user interface software
MOTIF	X Window System graphical interface
MP	Maneuver Prototype
MPP	Message Processor Program
MSC	Major Subordinate Command
MSG	Message
MTF	Message Text Formats
MUREP	Munitions Report
MUSE	Multiple Unified Simulation Environment
NCSA	National Center for Supercomputing Applications (University of Illinois)
NEO	Noncombatant Evacuation Operations
NFS	Network File Server
NGO	Non-Governmental Organization
NIS	Network Information Service or Network Information System
NM	Nautical Mile
NTSC	Naval Telecommunications System Center
OAS	Offensive Air Support
OBS	Order of Battle Service (formerly UGU: Unit Generation Utility)
OCA	Offensive Counter-Air

---

OJCS	Organization of the Joint Chiefs of Staff
OMA	Order Management Authority
ONC	Operational Navigation Chart
OPM	Online Player Manual
OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
SDB	Standard Database
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)

---

SIP	Scenario Initialization Program
SITREP	Situation Report
SLP	Sustainment Log Prototype
SOF	Special Operations Forces
SP	Survivability Prototype
SQL	Structured Query Language
SR	Short Range
SRP	Start/Restart Program (a JTLS component)
SRTE	Sea Route
SSM	Surface-to-Surface Missile
STR	Software Trouble Report
SUP	Ship Unit Prototype
SVP	Scenario Verification Program
SYNAPSE	Synchronized Authentication and Preferences Service
TADIL	Tactical Digital Interface Link
TCP/IP	Transmission Control Protocol/Internet Protocol
TEL	Transporter Erector Launcher
TG	Target entity attribute prefix
TGS	Terrain Generation Service (formerly TPS:Terrain Preparation System)
TGT	Target
TMU	Terrain Modification Utility
TOE	Table of Organization and Equipment
TOT	Time Over Target
TOW	Tube-launched Optically-tracked Wire-guided missile
TPFDD	Time-Phased Force Deployment Data
TTG	Target Type Group
TTL	Target Types List
TUP	Tactical Unit Prototype
TW	Targetable Weapon
UBL	Unit Basic Load
UIM/X	GUI builder tool
UNIX	POSIX-compliant operating system
UNK	Unknown
UOM	Unit Of Measure

---

USA	United States Army (U.S. and U.S.A. refer to United States and United States of America)
USAF	United States Air Force
USCG	United States Coast Guard
USMC	United States Marine Corps
USMTF	United States Message Text Format
USN	United States Navy
UT	Unit entity attribute prefix
UTM	Universal Transverse Mercator
VIFRED	Visual Forms Editor
VMS	Virtual Memory System
VTOL	Vertical Take-Off and Landing aircraft
WAN	Wide Area Network
WDRAW	Withdraw
WEJ	Web Enabled JTLS
WHIP	Web Hosted Interface Program
WIA	Wounded In Action
WPC	Warrior Preparation Center
WPN	Weapon
WT	Weight
WW	Wild Weasel
XMS	XML Message Service

## **APPENDIX B. VERSION 6.1.0.0 DATABASE CHANGES**

No changes were made to the JTLS-GO 6.1 database.



## **APPENDIX C. VERSION 6.1.0.0 REPOSITORY CHANGES**

No significant changes were made to the JTLS-GO 6.1 repository.