

JTLS-GO

Version Description Document

April 2026



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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.4.0)**

[Blank Page]

ABSTRACT

The Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) is an interactive, computer-based, multi-sided wargaming system that models air, land, naval, Special Forces, 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 6.4.4.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.4.4.0 is a Major release of the JTLS-GO 6.4 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as major model functionality improvements implemented as Engineering Change Proposals (ECPs), summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) will be described in Chapter 3 in future releases - because this is the first release of the JTLS-GO 6.4 series, there have been no STRs to correct. Known, 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 Director of Development
Valkyrie Enterprises LLC
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940
United States
jtlsgo@valkyrie.com

Copyright 2026 - Valkyrie Enterprises LLC - All Rights Reserved

[Blank Page]

TABLE of CONTENTS

ABSTRACTiii

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 1.2.1 Obsolete/Outdated Documents..... 1-1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 1-4

 1.3.1 Support Software 1-4

 1.3.2 JTLS-GO Cybersecurity Compliance 1-6

 1.3.3 JTLS-GO High Level Architecture Compliance 1-7

 1.4 DATABASE MODIFICATIONS 1-8

 1.4.1 JTLS-GO Using Legacy Default Symbol Set 1-8

 1.4.2 JTLS-GO Using New Default Symbol Set..... 1-8

 1.4.3 Standard Repository Changes 1-9

 1.5 INSTALLATION 1-9

2.0 ENGINEERING CHANGE PROPOSALS..... 2-1

3.0 SOFTWARE TROUBLE REPORTS 3-1

 3.1 JTLS-2026-17589 WIA/REMAINS REMOVED FROM MEDICAL UNITS 3-1

 3.2 JTLS-2026-17590 RADAR VOCABULARY LIST DOES NOT INCLUDE SAR 3-1

 3.3 JTLS-2026-17592 ATOT CHIT ORDERS FOR RESIDUAL MISSIONS 3-1

 3.4 JTLS-2026-17594 CRASH PROCESSING JAMMED UNIT SITREP 3-1

 3.5 JTLS-2026-17596 ICP EXCEPTION CLICKING ON TREE ROOT NODE 3-2

 3.6 JTLS-2026-17598 SUPPORT UNIT HAS NOT ARRIVED IN THEATER CRASH 3-2

 3.7 JTLS-2026-17600 ORDER GROUP EDITOR EXTERNAL UPDATE NAMES 3-2

 3.8 JTLS-2026-17602 CRASH READING IN CHECKPOINT WEAPON USAGE DATA 3-2

 3.9 JTLS-2026-17604 ELS CRASH RESTARTING FROM CHECKPOINT 3-3

 3.10 JTLS-2026-17606 UPDATE POSTGRESQL DATABASE CLIENT 3-3

 3.11 JTLS-2026-17608 CANNOT CHANGE PORT FLAG 3-3

 3.12 JTLS-2026-17610 SQL KEYWORDS NOT FLAGGED AS OEC COLUMN NAMES 3-3

 3.13 JTLS-2026-17612 CRASH CREATING UNIT AFTER GAME TIME 99 DAYS 3-4

 3.14 JTLS-2026-17614 IMT HELP MENUS DISPLAY "NULL" 3-4

 3.15 JTLS-2026-17627 ORDER REDIRECT CRUISE MISSILE CRASH 3-4

 3.16 JTLS-2026-17629 WHIP EXCEPTIONS ON EMITTER OBJECTS 3-4

 3.17 JTLS-2026-17632 INCORRECT FIX FOR OTH RADARS 3-5

 3.18 JTLS-2026-17634 FRAME TITLE SPACING 3-5

3.19 JTLS-2026-17636 ICP PANELS DO NOT SCROLL TO POSITION 3-5

3.20 JTLS-2026-17637 RESTING UNIT REJECTS SEQUENTIAL MOVE ORDER 3-5

3.21 JTLS-2026-17640 ICP OTH-GOLD COUNTRY CODE DEFAULT SETTING 3-6

3.22 JTLS-2026-17642 REJECT ORDER STATUS NOT SET WHEN UNIT IN FORMATION 3-6

3.23 JTLS-2026-17645 SVP JAVA PANEL DOES NOT UPDATE CORRECTLY 3-6

3.24 JTLS-2026-17648 RESTART ELS SPECIFY CHECKPOINT NUMBER 3-6

3.25 JTLS-2026-17651 NAVAL MOVE CRASH 3-6

3.26 JTLS-2026-17652 GIS TOOL - TERRAIN GRID MISCLASSIFICATION 3-7

3.27 JTLS-2026-17654 AIRBASE GATHER ORPHAN SQUADRONS 3-7

3.28 JTLS-2026-17657 UAVS NOT AUTO-LAUNCHED WITH SITUATION CHANGE 3-7

3.29 JTLS-2026-17658 IMPROVE NAVY DEBUG OUTPUT 3-8

3.30 JTLS-2026-17660 OPM AIRCRAFT WEIGHT ATTRIBUTE 3-9

3.31 JTLS-2026-17661 CONTAINER UNITS IN LOGFAS PROFILE CRASH 3-9

3.32 JTLS-2026-17662 JOI MODULES DO NOT SUPPORT CONTAINER UNITS 3-10

3.33 JTLS-2026-17663 AIR DEBUG GOING TO SCREEN NOT TO FILE 3-10

3.34 JTLS-2026-17666 DEBUGGING CRUISE MISSILES DID NOT WORK 3-10

3.35 JTLS-2026-17667 CONTAINER UNITS IN DDS 3-10

3.36 JTLS-2026-17668 CONTROLLER POLITICAL COUNTRY UPDATE 3-11

3.37 JTLS-2026-17670 CRASH SPLITTING ALL SHIPS IN FORMATION 3-11

3.38 JTLS-2026-17672 EXERCISE LOG - RESTRICT IMAGE UPLOAD FILETYPES 3-11

3.39 JTLS-2026-17673 ICP WHIP SIDE COLUMNS DIFFICULT TO SEE 3-11

3.40 JTLS-2026-17674 PASSWORD RESET PROMPT WINDOW TOO SMALL 3-12

4.0 REMAINING ERRORS..... 4-1

APPENDIX A. ABBREVIATIONS AND ACRONYMSA-1

APPENDIX B. Version 6.4.4.0 DATABASE CHANGES B-1

APPENDIX C. Version 6.4.4.0 REPOSITORY CHANGES C-1

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document (VDD)* describes Version 6.4.4.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) software suite. JTLS-GO 6.4.4.0 is a Major delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.4.0 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 “wespac64”. Database modifications that were accomplished to upgrade the previous JTLS-GO database format to this current version are summarized in this chapter, as well as [APPENDIX B](#). Detailed descriptions of the Engineering Change Proposals (ECPs) implemented for this release are provided in [Chapter 2.0](#).

JTLS-GO 6.4.4.0 executes on the Red Hat Enterprise Linux Version 9.4 and Oracle Linux 9.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP[®]) user workstation interface can be executed on any 64-bit 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 a 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 Air Services User Guide* (JTLS-GO Document 02, Version 6.4.3.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.4.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.4.3.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.3.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.4.3.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.4.3.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.4.3.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.4.3.0)

- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.4.3.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.3.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.4.3.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.4.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.4.3.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.4.3.0)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.4.4.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.4.4.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.4.4.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.4.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.4.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.4.4.0 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 234.

- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:

Apache Server (APACHE) version 2.4.66

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 (JOBED)
- 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.

VCP62 - Converts a JTLS-GO 6.1 database to a JTLS-GO 6.2 formatted database.

VCP63 - Converts a JTLS-GO 6.2 database to a JTLS-GO 6.3 formatted database.

VCP64 - Converts a JTLS-GO 6.3 database to a JTLS-GO 6.4 formatted database.

Instructions for installing JTLS-GO 6.4.4.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.4.0 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 “repository64”. 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 “wespac64”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.4.4.0 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 9.4, 64-bit architecture.

JTLS-GO 6.4 is compatible with the following versions of Linux 9:

RedHat Linux 9.6 - this operating system license must be purchased.

Oracle Linux 9.6 - 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 Defense Information System Agency (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 be a 64-bit architecture with a Java-enabled web browser. JTLS-GO 6.4.4.0 is compatible with the following operating systems:

Red Hat Linux Enterprise Edition Version 9.6

Oracle Linux 9.6

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

- JTLS-GO 6.4.4.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 482 package. Both the ICP and DCP have the option for an organization to increase the maximum memory heap for the WHIP and DDSC. For large scenarios and databases, an organization should consider increasing the maximum heap size.
- 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 15.16 database server and the full PostgreSQL installation. PostgreSQL 15.16 that has been compiled under Linux 9.6 is bundled with the JTLS-GO 6.4 release tar files. It is not necessary to use the delivered solution, but it is the easiest method to meet the requirements of JTLS-GO 6.4.4.0. There are several alternative methods available for obtaining the PostgreSQL 15.16 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.
- 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.

- **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>HLA 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. The concept of an expiration date has been removed from the DoD requirement, but the concept of always using the latest version of third-party software remains a strong component of DoD Cybersecurity requirements.

The following procedure has been established and approved by the JS/J7 Cybersecurity branch to meet the software update requirement:

- 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 6.4.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.

Please contact the U.S. Government Program Manager, Mr. Douglas Failor (douglas.l.failor.civ@mail.mil) to obtain the completed Cybersecurity paperwork. It is expected that a current Gate completion certificate will be available with four to five weeks of this initial release. Due to time and funding considerations, the JTLS-GO 6.4.0.0 project has not obtained a Checkpoint Gate certificate.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.4.4.0 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 has been accomplished. The reader should note that the JTLS-GO Development Team, to date, has not been able to test this federation. If there is interest in running this federation, please contact the JTLS-GO Help desk at jtlsgo@valkyrie.com.

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 6.4.4.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation (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 JTLS-GO 6.4.4.0 and the previous JTLS-GO 6.3 series database structure. Appendix B of the *JTLS-GO 6.4.0.0 Version Description Document* has a summary of all database changes.

To upgrade your JTLS 6.3 scenario to JTLS-GO 6.4 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

Due to changes made in support of STR JTLS-2026-17667, the database has changed between JTLS-GO 6.4.4.0 and previous versions of JTLS-GO. Users must **unload** their JTLS-GO 6.4 series scenarios prior to installing JTLS-GO 6.4.4.0, and then **load** them following installation.

Upgrading earlier JTLS-GO versions to the JTLS-GO 6.4 format will automatically correct the problem. No additional action is required.

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 6.4.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.4.0.0 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 6.4.4.0, 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 “repository64”. 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. Specifically, significant effort has been applied to ensuring that all important Targetable Weapons have a unique Supply Category from the weapon should be drawn. This results in the model managing a detailed weapon count of all used weapons.

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 15.16 required to operate JTLS-GO 6.4.4.0.

2.0 ENGINEERING CHANGE PROPOSALS

No model capabilities were added to JTLS-GO 6.4.4.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

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-2026-17589 WIA/Remains Removed From Medical Units

When a convoy has delivered supplies, it looks around to determine if there are any Remains or Wounded in Action (WIA) that need to be evacuated. The design indicates it should not try and evacuate personnel that are located at a medical-capable unit which represents a trauma center. The check to exclude such units was not implemented correctly.

The check to determine if a unit was a medical-capable unit was not correct. The code was corrected to match the manner in which a medical-capable unit is specified.

3.2 JTLS-2026-17590 Radar Vocabulary List Does Not Include SAR

The SIP and the CEP generate a vocabulary list for the permanent entities, such as Sensor Types. There is a vocabulary list called Radar_ST, that should include all of the sensors that are considered radars. The list does not include Synthetic Aperture Radars (SAR). For the purposes of this list, SAR Sensors should be considered radars.

The SIP and CEP routines used to create these data vocabulary lists were corrected to include SAR sensors in the Radar_ST vocabulary list.

3.3 JTLS-2026-17592 ATOT Chit Orders For Residual Missions

When the ATO contains missions that have their Residual Mission Indicator set to "R", the ATOT will not generate orders for these missions when the option to Process Residual missions is deselected in the ATOT module. However, in this case the ATOT is still incorrectly generating the refuel chits for these missions, and they will not pass the group checking.

The ATOT was not checking either the residual mission indicator for each mission or the desire for Processing Residual Missions while building the chit orders. This caused the ATOT to withhold the residual mission orders but still produce the refuel chit orders for the missions.

The ATOT was modified to include these two checks during the chit processing.

3.4 JTLS-2026-17594 Crash Processing Jammed Unit SITREP

The model crashed when a unit is being jammed and periodically wants to send a Situation Report reminding the Player that it is being jammed. The automatically-generated order was missing an assignment statement indicating that it was a single unit report.

The missing assignment statement was added to the automatic model-generated report.

3.5 JTLS-2026-17596 ICP Exception Clicking On Tree Root Node

When clicking on the scenario name/root tree node in the ICP, the ICP will throw a Java exception in the terminal. Functionality is not affected and the ICP behaves normally.

The ICP will no longer throw an exception when clicking on the root node/scenario name.

3.6 JTLS-2026-17598 Support Unit Has Not Arrived In Theater Crash

In the SVP, there is a warning that indicates that a unit's assigned support unit arrives later than the unit that may need supplies. This is a warning because it should not cause a crash. It does cause a crash.

The SVP-reported issue is considered a warning because support units may arrive before the requisitioning unit within the initialization database, but the Controller could easily TPFDD the unit out of theater and cause the same situation. Therefore, the problem was solved by ensuring that a unit's logistics requisition order does not get delivered to a support unit that is no longer alive or no longer in theater.

3.7 JTLS-2026-17600 Order Group Editor External Update Names

The WHIP Order Group Editor is showing all saved External Update orders as "EU_NONAME", even though we can retrieve the saved orders and see them in the actual order panel.

The External Update order did not have a <queue> element fully defined. Defining the element corrected the problem.

3.8 JTLS-2026-17602 Crash Reading In Checkpoint Weapon Usage Data

A checkpoint crashed being read into the Shadow Game. The problem was pinpointed to the artillery weapon usage data for a specific unit. Instead of a number, the number of rounds used was output as four stars (**).**

A Simgscript integer write statement specifies the number of digits that should be printed to the file. If the data parameter is larger than the number of digits allowed by the write statement, the compiler outputs a series of stars (****).

The user had been giving an artillery unit a large number of Fire Artillery orders. The rounds fired was greater than 10,000 rounds over a 48-hour period. The Simgscript write statement only allowed for four digits; therefore, the model output the four stars (****).

The code was changed to write up to ten digits, which allows for any normal game usage.

3.9 JTLS-2026-17604 ELS Crash Restarting From Checkpoint

The ELS crashed while it was restarting from a checkpoint. The program was trying to destroy an entity-level convoy object because the associated aggregate convoy no longer existed in the game. This situation occurred because the CEP had advanced in time from the time when the ELS checkpoint was saved. This happened when the game time was advanced for several hours while the ELS was not running.

The ELS checkpoint data contained a convoy which no longer existed in the CEP. The code was attempting to clean up the associated entity-level data. An error in the code resulted in a situation where the aggregate convoy was deleted, but the entity level objects still existed in the ELS. The code to delete the entity-level objects was prematurely exiting before they were cleanly removed. Changes were made to correct this problem.

3.10 JTLS-2026-17606 Update PostgreSQL Database Client

Minor PostgreSQL updates are essential for improving system stability, ensuring security, and fixing bugs. PostgreSQL 15.16 has been released and needs to be delivered as part of the next maintenance release.

The PostgreSQL Client, which is distributed with JTLS-GO, has been updated to version 15.16. Also, a custom-created PostgreSQL 15.16 database server .tar file has been included with the JTLS-GO release. Detailed security information for PostgreSQL version 15.16 can be found at:

<https://www.postgresql.org/support/security/15/>

3.11 JTLS-2026-17608 Cannot Change Port Flag

The Controller has the ability to alter the flag that indicates that a unit represents a naval port, but it does not work.

The code to have the Controller change this flag was accessing the incorrect tab of the order; therefore, the code did not see the request to alter the flag. This has been corrected.

3.12 JTLS-2026-17610 SQL Keywords Not Flagged as OEC Column Names

An order was created where one of the fields was given an oec-name attribute of "group". The oec-name attribute is then used as a column name within the Order Entry Client (OEC) table and matching code.

Because "group" is a reserved keyword in SQL, this caused problems in the table creation script and the source code to manage the table. This error was only found during the OEC compile, but should have been flagged as an issue by the order verification process.

The Order Verification Program (OVP) already had code to check and flag SQL keywords being used as database column names. This list of keywords also already included "group".

However, the OVP was checking the field's label to determine if it violated the keyword constraint instead of the oec-name attribute. This code was modified to use the oec-name attribute instead of the label.

3.13 JTLS-2026-17612 Crash Creating Unit After Game Time 99 Days

The model crashed when the user attempted to create a unit at a game time greater than 99 days into the simulation. The model had a hard-coded unit Time Phased Force Deployment Data (TPFDD) time of 99 days. This caused the model to attempt to schedule the Unit Arrive event at 99 days, which was a decrease in simulation time.

The model wants to schedule the unit arrival a long time in the future to give the Controller a chance to establish the remainder of the unit arrival parameters. The code was changed to have the unit arrive not at day 99, but 99 days in the future.

3.14 JTLS-2026-17614 IMT Help Menus Display "null"

The WHIP's IMT help menus, inside the Help -> "This Screen" option, would display the string "null" at the beginning of the text description. This occurs across all IMT tables.

The "null" string was removed, so that it no longer appears in the help menu's text.

3.15 JTLS-2026-17627 Order Redirect Cruise Missile Crash

A player submitted an order to redirect a cruise missile. The order checked the box indicating a cancellation of the current orbit. No new route for the missile was specified. The order included only a new attack location.

The crash occurred in the code where the model was attempting to re-use the receiver number for the missile's route. The receiver number specified the unique identifier for the route on the JODA. Because there was no existing route, the model crashed when trying to get the old receiver number by dereferencing a null pointer. Code was modified to give the route a new receiver number if there was no existing route

3.16 JTLS-2026-17629 WHIP Exceptions On Emitter Objects

The WHIP was found to be periodically throwing exceptions on the Emitter objects in the game. Although it had no negative effect on the objects, Emitters are not currently reading the TARGET_SUBCATEGORY attribute.

The TARGET_SUBCATEGORY attribute was added to the Emitter object so that value does not throw a read exception.

3.17 JTLS-2026-17632 Incorrect Fix For OTH Radars

In JTLS-GO 6.4.3.0, STR JTLS-2026-17543 fixed the problem in which Over The Horizon (OTH) Radars should not be limited by terrain masking. In implementing the fix for this issue, an error was introduced to non-OTH radars which made it almost impossible for the non-OTH radars to successfully detect foreign missions.

A check for terrain masking was being called for only non-OTH Radars as a result of the JTLS-GO 6.4.3.0 STR. The terrain masking routine switched the answer on whether there was terrain masking that would stop detection. The inadvertent switch in terrain masking results was corrected. This issue affected detection of aircraft and missiles by Air Search radars.

3.18 JTLS-2026-17634 Frame Title Spacing

The frame titles for the DDS, DCP, and ICP were missing some spaces after the tag lines. For example, the DDS frame title bar displayed "Scenario:scenario_name" when it should be "Scenario: scenario_name".

The spacing issues were fixed in the DDS, ICP, and DCP title bars.

3.19 JTLS-2026-17636 ICP Panels Do Not Scroll To Position

Inside the main scroll panel in the WHIP ICP tab, clicking on a WHIP entry will also update the left scroll panel to locate and automatically scroll to the corresponding WHIP entry. This does not work in vice versa (upon clicking on a WHIP entry in the left scroll panel, the main scroll panel will locate and highlight the corresponding WHIP entry, but it will not scroll to it).

The main panel in the ICP was fixed, so that it scrolls to the selected position from the left scroll/tree panel.

3.20 JTLS-2026-17637 Resting Unit Rejects Sequential Move Order

A Move order with a sequential Execution Time option that specified a sequence number 1 was submitted to a ground unit currently executing a Resting task (sequence number 0). The order was not accepted and a non-compliance player message was generated. The sequential Move order should have been accepted and then executed after the Resting task was complete.

The existing logic properly rejected an attempt to order a resting unit to move. However, the logic failed to make an exception for a Move order with a specified sequence number. Note that the sequence number is restricted in the Move order panel to a value greater than zero (the currently executing task). The logic was corrected to accept a Move order with a Sequential Execution Time option.

While investigating this issue, the logic was found to reject Move orders with a Start or Arrival time that was later than the end time of the resting task. This oversight was also corrected.

3.21 JTLS-2026-17640 ICP OTH-GOLD Country Code Default Setting

The JOI panel in the ICP defaults the outgoing country code for OTH-GOLD to ISO-3166-1-TRIGRAPH, when it should be set to OTH-GOLD as standard format.

Set the OTH-GOLD outgoing country code to OTH-GOLD as the default setting.

3.22 JTLS-2026-17642 Reject Order Status Not Set When Unit In Formation

Every order that is rejected should have a rejection status. When an order is rejected because a user did not indicate the ship had permission to leave a formation, it did not set any rejection status in PLAYER.ORDER entity.

In all such cases, the rejection status is now set.

3.23 JTLS-2026-17645 SVP Java Panel Does Not Update Correctly

The SVP Java window (started from the Javamenu) does not update the checkboxes upon switching the scenario after the initial startup.

The logic has been fixed to where the checkboxes update properly upon switching to multiple scenarios.

3.24 JTLS-2026-17648 Restart ELS Specify Checkpoint Number

In the past, the ELS would restart from the same checkpoint number as was used by the CEP. This was not always the best checkpoint to use for the ELS if the CEP had been running for a long period of time.

The ELS code was modified to allow the user to specify which checkpoint should be used during a restart. The restart process remained the same, except that the checkpoint number was included as an argument when restarting the ELS. The javamenu and jtlsmenu were also updated to allow the user to specify which ELS checkpoints to restart from.

3.25 JTLS-2026-17651 Naval Move Crash

A Naval Move order containing a very long distance move was submitted to a Formation. The route traversed a large terrain grid that did not contain any objects of any kind. The model crashed when it attempted to determine if any barriers in the grid would cause a delay.

Because of the large number of terrain grids in a JTLS-GO scenario database, each terrain grid is assigned a set owner ONLY if there is at least one object in the grid. The set owner is assigned when the first object enters the grid. If there are no objects, then no set owner is established for that grid.

The routine that checks for barrier delays exited early because there was no set owner in the empty grid. The logic therefore bypassed the creation of a temporary great circle arc that would have been used to determine if the Formation crossed a barrier. At the exit point, the logic attempted to destroy the non-existent great circle arc, which caused the model to crash due a null pointer.

To correct this problem, a check was added to destroy the arc only if it exists.

3.26 JTLS-2026-17652 GIS Tool - Terrain Grid Misclassification

Several fictional islands were added to the database. When the user created a GIS Tool project, created terrain layers and grids, and then ran the tool's Calculate Grid Parameters, most of the grids on the islands were classified as Ocean when they clearly should have been classified as some other terrain type (Open, Forest, etc).

When the GIS Tool tried to process a polygon list object, it only processed one polygon in the list, not the entire list. This has been fixed so that the method processes every polygon in a polygon list.

3.27 JTLS-2026-17654 Airbase Gather Orphan Squadrons

An Airbase was magic moved to a location where a fixed-wing Squadron was located without an assigned home base. The Airbase and Squadron were from the same Force Side. Also, the Squadron was located within the Airbase radius after the magic move. The Squadron properly adopted the Airbase as its home base.

However, a code review revealed that there was no logic to Airbase the airbase if it happened to be from a different Force Side or if the Squadron was located outside the Airbase radius. Either of these situations could have caused complications during subsequent model processing.

The missing criteria was added to the routine to prevent a Squadron from adopting a home base from a different Force Side or located beyond the Airbase radius. This logic also applies during the unit arrival process if an Airbase arrives after the Squadron.

3.28 JTLS-2026-17657 UAVs Not Auto-Launched With Situation Change

If the situation changes for a Squadron, such as receiving more supplies, the model automatically determines if there are any air missions waiting for the supplies so they can launch. This check is currently only done for Squadrons, and not for other types of units that may own Unmanned Aerial Vehicles (UAVs).

The code only called the routine CAN YOU LAUNCH when supplies arrived at an Airbase, FARP, Naval Unit or Squadron. It did not call the routine for other types of units that could own aircraft. This was fixed.

The code was thoroughly reviewed to determine if there were any other situations in which non-Squadron units owning UAVs did not check whether waiting air missions could now launch because of a situation change. The following additional situations were found and fixed:

- If an Aircraft Class characteristic, such as usable fuel type, changes such that would make it not possible to launch the air mission.
- If the owning Unit was magic moved into an area that has needed supplies.
- If a portion of the Unit was placed on the ground as a result of an airlift or amphibious operation.
- If the Controller magically gave the Unit more available aircraft.
- If a Unit arrives in the game or is returned to the game, and there are already missions waiting to launch.
- If the weather is changed over a Unit with air missions waiting to launch.
- If one Unit attaches to another Unit, providing the surviving Unit more aircraft.

3.29 JTLS-2026-17658 Improve Navy Debug Output

When JTLS started development, the Simscript compiler did not have a debugging capability. To help with debugging, a Debug File was created for each type of object, Ground Unit, Naval Unit, Air Mission and Supply Run. Simscript has subsequently incorporated a debugger to help with debugging tasks.

The old Navy Debug file is not useful because it is easier to get the information through the improved Simscript debugging capability. What is needed is a Navy Debug file that tracks submarine detection calculations.

The old Navy Debug output was replaced with details concerning the computation of when a submarine will be detected. These are the steps needed to start the detection computation debug output:

1. From a Controller WHIP, turn on the Navy Debug Flag.
2. From a Controller WHIP, submit a Debug Object Order for the submarine whose detection time computation should be tracked.
3. Place the model in the debugger and set a breakpoint for SCHEDULE.COVERT.DETECTION at line 2. The model will indicate that there is no code at line 2 and will place the breakpoint at the first executable line. As of this writing that is line 53, but it may change if code changes to the routine are made.

4. Set game speed as desired.
5. When the model enters the debugger in SCHEDULE.COVERT.DETECTION, submit the following two commands:

sv SAVED.DEBUG.FLAG #, where '#' is the side number of the detecting Force Side for which you are interested. For example, in the wespac64 scenario, Side 6 is the US side. If I wanted output for how Side 6 is computing its detection times for the ship named in Step 2, I would enter the command: **sv SAVED.DEBUG.FLAG 6**

db 1, which will remove the breakpoint, so the model no longer enters the debugger each time it enters the routine SCHEDULE COVERT DETECTION.
6. View the results as the model is executing by going to the \$JGAME/<scenario_name>/debug directory. The computation debug statements are held in the files labeled **SIMU53<TimeStamp>.debug**.

Note this file will be closed and a new file opened each time a checkpoint occurs.

3.30 JTLS-2026-17660 OPM Aircraft Weight Attribute

The Online Player Manual (OPM) has the ability to change the Unit Of Measure (UOM) of any data value based on the type of value.

In JTLS-GO 6.4, a new Aircraft Class attribute was added to the database. This attribute has a default UOM for Dry Weight. However, the OPM user could not change the viewed UOM for this database parameter.

The SIP code that built the page was corrected to indicate that AC WEIGHT represented a Dry Weight. The OPM User can now view that database parameter using any desired Dry Weight UOM.

3.31 JTLS-2026-17661 Container Units In LOGFAS Profile Crash

If a database includes a Container Unit in a LOGFAS profile, both the SVP and the LOGFAS XML generation process crash during execution.

The code used to check LOGFAS Profiles and the generation of the LOGFAS XML file did not expect the existence of Container units. The code was corrected.

While reviewing the code, it was noted that the symbol names of Container Unit did not include the Container unit Echelon. The model should treat Container Units in the same manner that it treats Ground Units. The code was changed in both the SIP and the CEP for consistency.

3.32 JTLS-2026-17662 JOI Modules Do Not Support Container Units

The OTH Gold and Link 16 modules in the WHIP do not support sending Container Unit data.

The JOI module panels were modified to support sending out Container Units.

3.33 JTLS-2026-17663 Air Debug Going To Screen Not To File

If the user turns on the Air Debug Flag, some debug information is supposed to be placed in the Air Debug file. The output is not going to the file, it is going to the screen.

The problem was fixed.

While reviewing the code, it was noted that the current air debug information is not very useful, because it prints out debug information for all flying air missions. What really is needed is to print out the debug information for the one air mission that is of interest to the Technical Control team attempting to analyze what is happening with a particular mission. The code was changed to only print out the debug information for the one air mission assigned as part of the DEBUG.OBJECT order.

3.34 JTLS-2026-17666 Debugging Cruise Missiles Did Not Work

Cruise Missile movements was designed to be a part of the Air Debug output. An error in the order file did not allow the Controller to specify a Cruise Missile that should be followed.

The error in the order panel's Cruise Missile specification field no longer worked. It was corrected to use a new methodology that was developed several years ago.

While testing this correction, it was determined that the debug information was not useful for Hypersonic Weapons. The debug information was updated to be more useful.

3.35 JTLS-2026-17667 Container Units In DDS

The database trigger for the fc_leader for the faction_country table was limiting the Unit Type to eight characters, and did not account for the new "CONTAINER" Unit Type.

There was also an issue in the Glassfish "changeUnitType" servlet where it successfully saved the references, but it did not update the new "CONTAINER" unit to search for those references.

The database trigger was modified to include the "CONTAINER" Unit Type, and the servlet was modified to update the new "CONTAINER" unit to look for those references.

3.36 JTLS-2026-17668 Controller Political Country Update

When the Controller updates a Unit's political country, the JODA update for political country was not being sent. Additionally, only the Unit's HRUs were being updated, instead of HRUs, Targets owned by those HRUs, and the Unit's own owned/associated Targets.

Additionally, the symbol name update to the JODA was incorrect: symbol name updates were not being sent for HRUs, and both HRUs and Targets used an update type that was never sent to the JODA due to a missing call.

An additional JODA update for a Unit's political country was added. Two new routines were created to handle necessary updates required for all other associated objects (HRUs and Targets).

The JODA updates for HRUs now pass symbol names, and the update type was changed to correct the missing call for both HRUs and Targets.

3.37 JTLS-2026-17670 Crash Splitting All Ships In Formation

A user ordered all ships in a Formation to split off into a new Formation, crashing the model.

The design and order help text indicated that splitting all ships in a Formation into a new Formation is not legal. The code did not properly check for this order error. The check was added. The Split Formation task is now rejected if the user has ordered all ships to split into a new Formation.

3.38 JTLS-2026-17672 Exercise Log - Restrict Image Upload Filetypes

The Exercise Log implemented loose file type validation on the client side only. Users were able to upload any file type, causing problems during PDF generation of the Log.

The Exercise Log now performs file type validation on the server side when uploading a file. File types are now restricted to .jpg and .png.

3.39 JTLS-2026-17673 ICP Whip Side Columns Difficult To See

The WHIP Force Side columns in the ICP have the header backgrounds set to the Side colors. However, the column backgrounds are set to a default white color. When there are many Sides in a scenario, this can make it difficult to see which Side you are currently changing when checking boxes on the panel.

The column background colors have been changed to match the Side colors, exactly like the column headers.

3.40 JTLS-2026-17674 Password Reset Prompt Window Too Small

The password reset prompt was reported to be too small, as the initialized height of the window cut off the password requirement text.

The problem could not be replicated. As a work around, the initial height of the window was slightly increased.

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 6.4.4.0 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.

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.

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 6.4.4.0 DATABASE CHANGES

The following changes were made to the JTLS-GO 6.4 database:

APPENDIX C. Version 6.4.4.0 REPOSITORY CHANGES

No significant changes have been made to the structure of the JTLS-GO 6.4.4.0 repository.