

JTLS-GO

Version Description Document

October 2026



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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.8.0)**

[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 6.4.8.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.4.8.0 is a Maintenance release of the JTLS-GO 6.4 series that includes an updated repository of standard data and a demonstration scenario based in the western Pacific. No model functional improvements are included in this maintenance release. Code modifications needed to correct identified Software Trouble Reports (STRs) are described in Chapter 3. All 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
1200 Piedmont Avenue, Suite A
Pacific Grove, CA 93950
United States
jtlsgo@valkyrie.com

Copyright 2026 - Valkyrie Enterprises LLC - 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1.2.1 Obsolete/Outdated Documents.....	1-1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	1-4
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.....	1-9
1.4.3 Standard Repository Changes	1-9
1.5 INSTALLATION	1-9
1.6 CHANGES RELATED TO AUTOMATIC MARITIME TRAFFIC GENERATOR	1-9
1.6.1 Scenario .ncm File for Automatic Maritime Traffic	1-10
1.6.2 Checkpoint .ev_ascii Files	1-11
2.0 ENGINEERING CHANGE PROPOSALS.....	2-1
3.0 SOFTWARE TROUBLE REPORTS	3-1
3.1 JTLS-2026-17805 Generative Weapon Does Not Regenerate	3-1
3.2 JTLS-2026-17806 HRU Patrol Route Blocked Issues	3-1
3.3 JTLS-2026-17807 SVP Crash For Federation Checks	3-2
3.4 JTLS-2026-17808 Targetable Weapon Checkpoint File Does Not Load In DDS	3-2
3.5 JTLS-2026-17809 Ghost Update Message Does Have Enough Information	3-3
3.6 JTLS-2026-17810 Inconsistent Definition Of USMTF CINT Mission Type	3-3
3.7 JTLS-2026-17811 Resting Unit Creates Multiple Withdraw Tasks	3-3
3.8 JTLS-2026-17812 Fire Mission Delay Message Has Time Format Error	3-4
3.9 JTLS-2026-17813 Crash Attempting to Cancel Close Air Support	3-4
3.10 JTLS-2026-17819 ATOT Locked Computing Mission To Squadron Distribution	3-4
3.11 JTLS-2026-17823 SVP Crash If Air Defense Type Has No Fire Control Radar	3-4
3.12 JTLS-2026-17827 Mark Duplicate DIS Code As SVP Error	3-5
3.13 JTLS-2026-17828 C4I Name For Automatically Created Civilian Squadrons	3-5
3.14 JTLS-2026-17829 Container Unit Owns HRU	3-5
3.15 JTLS-2026-17830 ICP Whip Panel Column Sorting Broken	3-6

3.16 JTLS-2026-17831 Crash Sending A Ship To Location ON/OE	3-6
3.17 JTLS-2026-17832 Aircraft Unit Check Minor Issues	3-6
3.18 JTLS-2026-17833 Crash Mixing Launch Time And Impact Time Fire Missions	3-7
3.19 JTLS-2026-17835 ATOT Not Properly Filling Wait Flag For Mission Instructions ...	3-7
3.20 JTLS-2026-17836 ATOT Should Set First Mobility Mission Stop As Dropoff	3-8
3.21 JTLS-2026-17837 SVP Check For Illegal Characters	3-8
4.0 REMAINING ERRORS.....	4-1
APPENDIX A. ABBREVIATIONS AND ACRONYMS	A-1
APPENDIX B. Version 6.4.8.0 DATABASE CHANGES	B-1
APPENDIX C. Version 6.4.7.0 REPOSITORY CHANGES	C-1

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.4.8.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) software suite. JTLS-GO 6.4.8.0 is a Maintenance delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.8.0 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](#). No Engineering Change Proposals (ECPs) were implemented for this release, but for consistency [Chapter 2.0](#) normally used to summarize model improvements have been included in this document.

JTLS-GO 6.4.8.0 executes on the Red Hat Enterprise Linux Version 9.6 and Oracle Linux 9.6 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 Analyst Guide* (JTLS-GO Document 01, Version 6.4.5.0)
- *JTLS-GO Air Services User Guide* (JTLS-GO Document 02, Version 6.4.5.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.4.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.4.5.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.7.0)
- *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 Installation Manual* (JTLS-GO Document 09, Version 6.4.7.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.4.4.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.6.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.4.3.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.4.5.0)
- *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 6.4.6.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.4.3.0)

1.2.3 Updated Documents

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.4.8.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.8.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.8.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.4.8.0 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 235.

- 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

- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- 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 6.4.8.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.8.0 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 6.4.8.0 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 9.6, 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 6.4.8.0 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 6.4.8.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 502 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 1.8.8.
- JTLS-GO database tools require a certified PostgreSQL 15.18 database server and the full PostgreSQL installation. PostgreSQL 15.18 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 6.4.8.0. There are several alternative methods available for obtaining the PostgreSQL 15.18 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.
- JTLS-GO uses GlassFish 5.1.0.
- 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

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. The JTLS-GO 6.4 series Exit Gate Certificate expired on 8 August 2026. The JTLS-GO 6.4 series software cannot obtain a current Cyber-Security Certificate due to the use of GlassFish Version 5.1.0, which is no longer supported software. The JTLS-GO 6.4 series has an extension to operate on U.S. Government computers until 31 October 2026.

The JTLS-GO 7.0 series of JTLS-GO is nearing completion and its Cyber-Security tests. This new version of JTLS-GO has moved to the most current and supported version of GlassFish, which required that JTLS-GO move away from Java 1.8 to Java 21. JTLS-GO 7.0 will be officially released prior to the expiration of the JTLS-GO 6.4 series extension.

The current plans are to continue supporting the JTLS-GO 6.4 series until June 2027, even though it will be unavailable for use by U.S. Government organizations after 31 October 2026.

One of the purposes of this release of the JTLS-GO 6.4 series is to prepare for a new JTLS-GO 7.0 series database character restriction. Please refer to [JTLS-2026-17837 SVP Check For Illegal Characters, Section 3.21](#)

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.4.8.0 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 6.4.8.0 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 the JTLS-GO 6.4 series 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, a database function was changed in the JTLS-GO 6.4.4.0 maintenance release. This STR fixed issues with attempting to set a Faction Leader to a Container type Unit and in the code to change a unit to a Container type Unit. If an organization is still using a JTLS-GO 6.4 series version that was release prior to the JTLS-GO 6.4.4.0 maintenance version, the following procedure needs to be followed after installing this maintenance release.

Users must **unload** their JTLS-GO 6.4 series scenarios prior to installing this version of JTLS-GO, and then **load** them following installation.

If this process was followed during the installation of the JTLS-GO 6.4.4.0 through JTLS-GO 6.4.7.0 maintenance releases, 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 6.4.8.0, 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

The JTLS-GO Database Team 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 15.18 required to operate JTLS-GO 6.4.8.0.

1.6 CHANGES RELATED TO AUTOMATIC MARITIME TRAFFIC GENERATOR

The ability to automatically generate maritime traffic, in order to simulate realistic background civilian shipping that is desired to be represented in a scenario, has been added to JTLS-GO. This is similar to the method used to generate realistic background air traffic. The user will establish a list of ships and ports that will be used for automatic maritime traffic, and the model will automatically send Enter Port orders to those ships, using those ports as destinations, according to the parameters of the individual order establishing the traffic. More than one order can be issued at one time.

The full capability for automatically generated maritime traffic is being delivered in JTLS-GO 7.0.0.0, but there is a desire to deliver this capability in the JTLS-GO 6.4 series, so that it may be used by customers in exercises this year. Because the JTLS-GO database cannot be changed within the 6.4 series, an alternate method of setting up the automatic maritime traffic capability has been created and the necessary steps are described here.

Additional instructions, from a WHIP user perspective, are provided in Chapter 6 of the *JTLS-GO Player Guide*. The complete design for the automatic maritime traffic capability is provided in the *JTLS-GO 7.0 Design Plan*.

1.6.1 Scenario .ncm File for Automatic Maritime Traffic

All of the code to support the automatic maritime traffic capability is included in this version of JTLS-GO. However, the user will need to create, by hand, a special file within the **data/scenario/****<scenario_name>** directory to support the capability, because the database schema cannot be changed in this maintenance release.

The user must create a **<scenario_name>.ncm** file within the **data/scenario/<scenario_name>** directory. This file should be created as a simple text file. This file will contain data parameters that establish:

- Which Support Units are considered to be Naval Ports.
- What is the access location for that Naval Port, which Naval Units will use as their destination. This is meant to represent the geographical location where the ships actually enter the port, rather than the location of the Support Unit, which may represent the geographical center of the port, a headquarters for the port, or some other location.
- Which Naval Units will be eligible for selection by the model for automatic maritime traffic generation.

The format of the file is shown in [Figure 1.1](#).

```
PORT_FLAG CHEJU.NB_KR
PORT_FLAG JEJU.NB_KR
PORT_FLAG CHANGJON.NB_KP
PORT_FLAG GROTON.NB_US
PORT_ACCESS CHANGJON.NB_KP 38.737796783 128.204437256
PORT_ACCESS GROTON.NB_US 41.344356537 -72.084739685
AUTO_ASSIGN LEE.EOKGI_KR
AUTO_ASSIGN LEE.SUNSIN_KR
AUTO_ASSIGN NA.DAEYONG_KR
```

Figure 1.1 <scenario_name>.ncm File Format

- Each Support Unit to be considered a Naval Port is identified by a line beginning with “PORT_FLAG”, followed by the Support Unit’s short name.
- Each Naval Port may have a Port Access Location provided, which will be used as the actual destination for the generated Enter Port orders. This field is optional - if left blank, the model will use the location of the Support Unit itself as the destination.
- Each Naval Unit to be designated as eligible for selection for the automatic maritime traffic is identified by a line beginning with “AUTO_ASSIGN”, followed by the Support Unit’s short name.

Once this file is established, the user may continue to change the parameters throughout the game by using the Set Unit Data order to change the Port Flag and Port Access Locations of Support Units, or the Change Auto Traffic Parameters order to set the Auto Traffic Flag for Naval Units.

Actual generation of automatic maritime traffic is a Player order, Generate Maritime Traffic, and is described in the *JTLS-GO Player Guide*.

If the user has a checkpoint that was created using a JTLS-GO 6.4 series version that was released prior to JTLS-GO Version 6.4.6.0 and the desire is to restart from that checkpoint using this version of JTLS-GO, a simple modification must be made to the checkpoint's `<scenario_name>.ev_ascii`. Please follow the instructions in [Section 1.6.2](#).

1.6.2 Checkpoint .ev_ascii Files

In order to support the development of the Automatic Maritime Traffic Generator capability for JTLS-GO 6.4.6.0, the event GENERATE.TRAFFIC.HRU was restructured to GENERATE.TRAFFIC, and now handles both HRU and Maritime traffic. A text keyword was added to the event ASCII data to indicate the specific type of traffic. Users must manually modify the `<scenario_name>.ev_ascii` file in their existing checkpoints to continue using them. Since in JTLS-GO Version 6.4.6.0 was the first version to allow Maritime traffic generation, and Traffic events in a checkpoint created with a previous JTLS-GO 6.4 series must represent HRU Traffic events.

If HRU Traffic events are not used in saved checkpoints from which the user plans to restart their scenario, then these changes are not required.

For each HRU Traffic event in the `<scenario_name>.ev_ascii` file, modify the first line. After its start time, add a space, then the keyword “HRU” (see the examples in the following images):

```
1 GENERATE_TRAFFIC_HRU EVENTS
+8.317179118535E-002
Unit_Name +4.166666666667E-002      2      :
Unit_Name +6.500000000000E-001
Unit_Name +1.000000000000E+000
1_50000000000000000000
2_65000000000000000022
3 1.000000000000000000
[more data...]
```

Figure 1.2 Previous `<scenario_name>.ev_ascii` File Format

```
1 GENERATE_TRAFFIC_HRU EVENTS

+8.317179118535E-002 HRU
Unit_Name +4.166666666667E-002      2      3
Unit_Name +6.500000000000E-001
Unit_Name +1.000000000000E+000
1 .50000000000000000000
2 .65000000000000000022
3 1.00000000000000000000
[more data...]
```

Figure 1.3 Current <scenario_name>.ev_ascii File Format

2.0 ENGINEERING CHANGE PROPOSALS

No new Engineering Change Proposal (ECP) capabilities were added to JTLS-GO 6.4.8.0.

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-2026-17805 Generative Weapon Does Not Regenerate

Air missions with a regenerative weapon load do not automatically generate new weapon counts.

The code was fixed to properly regenerate weapon counts when using regenerative weapons. The code was also reorganized to remove such code from routines identifying satellites and generalize the code.

3.2 JTLS-2026-17806 HRU Patrol Route Blocked Issues

An HRU was patrolling and encountered an uncrossable barrier. The HRU stopped moving because it could not find an alternate route when one was feasible.

The HRU Patrol task allows the Player to select a single pass route, a repeating route, or a random route within a polygon/OPAREA to traverse. The Patrol route type is a major factor that determines how the HRU reacts when an uncrossable barrier or impassable terrain is encountered. Testing revealed several issues:

- Single Pass Route - The HRU halted at the barrier and continued to execute successive Ground Waits and Move tasks using the Player-specified single pass route, which was not viable.

The logic was changed to automatically cancel the Patrol task in this situation.

- Repeated Route - The HRU halted at the barrier, then moved back to the start of the Player-specified route, whereupon it restarted the Patrol task from the beginning as expected. However, the HRU instantly jumped to the end of the route on the other side of the barrier.

This problem occurred because the task route had been reversed at the start of the Patrol in preparation for the return trip (by design), but when the HRU was blocked and moved back to the start position, the reversed route caused the HRU to jump over the barrier to the last route location.

To correct this problem, new code was added to reverse the route back to the original sequence when the HRU encounters a barrier. This change enables the HRU to follow the Player-specified route from the original start location up to the barrier and then repeat.

- Polygon/OPAREA - The HRU moved along successive random routes until eventually encountering the barrier. The HRU canceled its Patrol order instead of looking for a new random route that did not intersect the barrier.

This situation occurred when the logic created a new random route with a starting location on the other side of the barrier. The check location logic concluded the HRU needed to move to the start location to begin the new Patrol route. However, there was no feasible route and the Patrol was automatically canceled (by design).

The parameter passed to the check location routine was corrected to use the polygon/OPAREA route type which allowed the logic to conclude the HRU could begin the Patrol from its current location. This correction caused the logic to recognize the route was blocked and create a different route instead of canceling the Patrol when it could not move to the new start location. This change enables the HRU to continue random route patrolling within the polygon or OPAREA without canceling on its own.

Given the above changes, there is no need for a tactical alert when the HRU encounters an impassable barrier.

3.3 JTLS-2026-17807 SVP Crash For Federation Checks

The SVP crashed while attempting to check federation-related data. The crash occurred in the routine READ TUP TEMPLATES.

This crash occurred because the program was attempting to open a file using a file identifier (integer) which was already in use. A file identifier was added for use with the reading the ELS templates. The read code was modified to use the new number to access these data files.

3.4 JTLS-2026-17808 Targetable Weapon Checkpoint File Does Not Load In DDS

It is possible to take a closing checkpoint from an exercise and load it into PostgreSQL so it can be the starting point for a future exercise. The checkpoint version of the Targetable Weapon file does not load into the DDS.

Two Targetable Weapon attributes, TW.MISSILE.CAPABILITY and TW.BOOST.INTERDICTION TYPE are both Non-Null DDS parameters. Each of these parameters have a legal value of NO. When the ASCII file for the Targetable Weapon was written, the code wrote out a value of NONE for the weapon attributes. Within the CEP, NO and NONE are identical; therefore, writing a value of NONE and reading in the value when restarting from a checkpoint worked fine.

In the DDS, NO and NONE are not equivalent. A value of NONE tells the DDS to leave the field as NULL. Since this was not allowed, the file could not be loaded into the DDS.

The CEP now properly writes out the keyword "NO".

3.5 JTLS-2026-17809 Ghost Update Message Does Have Enough Information

When a collection asset collects information at a location and the Force Side expected to see a unit at that location, the asset creates a special type of update known as a ghost update. The US Government has not indicated how we should handle ghost updates; therefore, they are currently skipped. A message is written to the terminal window indicating that a ghost update was ignored. The issue is that if any other unrecognized update type is found, the model also reports the fact that the update will be ignored. These two warning messages have different significance.

- An ignored ghost update is simply informational.
- An unrecognized type of update is serious and needs to be thoroughly investigated.

Two separate messages are needed to distinguish between the seriousness of the warning.

Two different messages are now created to indicate the seriousness of the unrecognized update.

3.6 JTLS-2026-17810 Inconsistent Definition Of USMTF CINT Mission Type

The Air Tasking Order Translator (ATOT) uses a file called the `ato_constants.xml` file. Among other things, this file links all ATO Message Missions Types With JTLS-GO missions types. The ATO Mission Type was inconsistently defined. One attribute indicates that the Mission Type should become a Reconnaissance Mission, but other attributes indicate it should become an Electronic Combat Mission.

The data file was changed to indicate that the ATO Mission Type CINT should become a Communications Interdiction mission meaning it should jam communications.

3.7 JTLS-2026-17811 Resting Unit Creates Multiple Withdraw Tasks

A ground unit's strength was reduced sufficiently in combat to force a withdrawal. The Withdraw task was immediately postponed and a Resting task was generated. A few minutes later, another Withdraw task was created because the unit's strength was still below the Withdraw threshold level. The second Withdraw task also was postponed allowing the Resting task to continue. This behavior repeated and generated a large number of superfluous Withdraw tasks until the Resting task completed.

The Resting unit was still in combat at its current location. The Assess Combat event executed every 30 minutes. Because the unit strength remained below the 45% Withdraw threshold, the logic canceled the Rest task and created another Withdraw task. When the new Withdraw task attempted to execute, the Resting task generated again because the unit hadn't rested long enough and the new Withdraw task was also postponed. As a result, a new Withdraw task was added to the unit's pending execution task list every 30 minutes.

A code change was made to prevent a new Withdraw task route from being created if the unit was currently executing a Resting task.

3.8 JTLS-2026-17812 Fire Mission Delay Message Has Time Format Error

If a Fire Mission cannot be fired, a message is generated indicating why the Fire Mission has not executed. This message lists what is known as the Not Later Than Time (NLT). If the user did not specify an NLT time, the model automatically sets the NLT time to Ten Years from the start of the game. The resulting Fire Mission Delay message incorrectly indicates that the NLT time held by the Fire Mission has a message formatting error.

The message was set up to indicate that no NLT time had been entered for the Fire Mission, but the model logic did not properly support this message capability, when formatting the message. The issue was corrected. The new message will indicate that no NLT time had been entered on the order.

3.9 JTLS-2026-17813 Crash Attempting to Cancel Close Air Support

The CEP crashed while attempting to process an order to cancel Close Air Support (CAS). The reason for the crash was a subscript out of range.

As part of the process to cancel CAS, the model was looping through all of the units in the game to find units with the correct aircraft type. This loop did not filter out units which had no aircraft at all. The crash occurred when it attempted to access a property for a non-existent aircraft type. The code was modified to only consider units which held aircraft systems.

3.10 JTLS-2026-17819 ATOT Locked Computing Mission To Squadron Distribution

When the ATO Translator is executed for an ATO that contains too many missions for a tasking unit to satisfy, it becomes stuck while computing the arrangement of missions to assign to the available squadrons at that tasking unit.

The ATO Translator attempts to assign an optimum arrangement of missions to the available squadrons of a tasking unit, which should minimize the number of missions that may be unassigned. The routine for doing this sets up a long list of mission permutations and compares the score for each arrangement. However, when the number of missions exceeds 8, the number of permutations becomes too long to efficiently compute in a reasonable time. Therefore, the routine was changed to limit the number of permutations so the computed result provides a good arrangement without taking an unreasonable length of time.

3.11 JTLS-2026-17823 SVP Crash If Air Defense Type Has No Fire Control Radar

Although unusual, the current database rules are that an Air Defense Type does not need to have a Fire Control Sensor specified. If a user creates an Air Defense Type without a Fire Control Sensor, the SVP will crash.

The check that caused the crash was fixed to check if the database specified a Fire Control Sensor before attempting to check if the Fire Control Sensor has an Air Search capability.

3.12 JTLS-2026-17827 Mark Duplicate DIS Code As SVP Error

In development for JLVC Light Federation, DIS Code is used as a unique identifier of an object. No two objects should share the same DIS Code.

SVP Error #5019 is created to check against all DIS Code owning objects for duplicates. In total of 24 types are checked, including 20 permanent entities and 4 temporary entities.

3.13 JTLS-2026-17828 C4I Name For Automatically Created Civilian Squadrons

The Civilian Air Traffic Order automatically creates squadrons, if needed, to accomplish the order of transferring an aircraft of a given type from one airport to another. The automatically created squadrons do not have an assigned C4I name. This is a mandatory field and needs to be filled.

The C4I name is set to the same of the newly automatically created squadron. The model also checks to make sure this name is unique.

While making this correction, several other issues concerning C4I Name were also noted and corrected.

- There was an inconsistency whether spaces were or were not allowed in the C4I name. The correct design is that C4I names should have spaces allowed.
- The C4I name entered on a Controller Create Unit Order was not being honored by the model.

3.14 JTLS-2026-17829 Container Unit Owns HRU

The Scenario Verification Program crashed when a Container Unit was specified as the parent unit for a High Resolution Unit (HRU)

Container units should not be allowed to be the parent of an HRU. The crash situation is caught. In addition, several other related SVP checks were added:

- Error 505 Indicates that an HRU has a Parent Unit that is a Container Unit
- Error 469 Indicates that a Container Unit was specified as the owner for a database target.
- Error 470 Indicates that a Container Unit was specified as the associated unit for a database target.

3.15 JTLS-2026-17830 ICP Whip Panel Column Sorting Broken

Sorting the WHIP panel table column by WHIP name in the ICP would sort abnormally. Only this column in the table was broken, the other columns sorted normally.

The WHIP table was mistakenly using two implementations of a table sorter. This has now been simplified into one. The WHIP name column now sorts normally again by alphabetical order.

3.16 JTLS-2026-17831 Crash Sending A Ship To Location ON/OE

A ship was sent to ON/OE resulting in a game crash with a floating point exception.

Traceback indicated that the crash occurred in the routine CONSTRUCT GREAT CIRCLE ARC. There are 4 arguments passed to this routine (LATITUDE1, LONGITUDE1, LATITUDE2, LONGITUDE2). These are the endpoints of the great circle arc to be created. There are checks to determine whether the endpoints share the same latitudes or the same longitudes. When the latitudes or longitudes are equal the code is simplified and avoids a possible divide by zero.

What caused the crash was that LATITUDE1 and LATITUDE2 had nearly identical values. Because the check for equal latitudes is an absolute test for equality, the code flow followed a path that resulted in an intermediate value that should have been extremely small, but due to rounding errors, resulted in a value of zero. This intermediate value was then used in a following equation where it served as a divisor, resulting in a division by zero.

To avoid this problem, the test for equal latitudes and longitudes has been modified to check if the absolute value of their difference is within one meter. If it is less than one meter, the latitudes, or longitudes, are considered “equal”.

3.17 JTLS-2026-17832 Aircraft Unit Check Minor Issues

A user organization contacted the Help Desk with a crash in the CEP. The crash turned out to be a unit that owned a Combat System Aircraft, but had no aircraft type specified. The SVP catches this error, and it was unclear whether they can cleared all SVP errors prior to starting the model.

A review of the code was conducted to determine if there were any existing logic errors that would have caused the problem from not being found. No issues were found concerning the report of the error, but during the review three small issues were uncovered:

- Error 416 assumed that the name of the Combat System Aircraft was AIRCRAFT. Although this is true for all Development Team scenarios, it is not a requirement. A user could assign any desired name of the Combat System that represents aircraft.
- Warning 1426 had several automatic corrections that would not work if the unit that owned the aircraft was a Naval Unit.

- Warning 1229 had several automatic corrections that were not named following the standard naming correction. This did not result in any negative consequence, but could lead to some debugging issues in the future.

Each problem was corrected.

3.18 JTLS-2026-17833 Crash Mixing Launch Time And Impact Time Fire Missions

If a single unit is given a mixture of Launch Time Fire Missions and Impact Time Fire Missions while a unit is moving, the model can crash when recomputing the desired launch time for the Impact Time Fire Missions.

Specifically, this was the exact situation.

- Fire Mission 1 was a Launch Time Fire Mission, but because the unit was moving, it could not fire the Fire Mission. Its assigned SSMS were not deployed while moving.
- Fire Mission 2 was entered with an Impact Time in the future. When the unit moved, the model recomputes Fire Mission 2's launch time because the move may have resulted in the mission either:
 - a. Moving closer to the intended target and thus the launch time could be delayed, or
 - b. Moving further away from the intended target and thus require a sooner launch time.

The model crash because it assumed that all Fire Missions in the unit's pending Fire Mission set had their launch times adjusted by the move. This was not the case because Fire Mission 1 was a Launch Time Fire Mission and did not need its launch time adjusted. The crash occurred because Fire Mission 1 was delayed and its launch time was in the past. When the model attempted to reschedule the next fire event, it attempted to do so for a time in the past - the time of the delayed Launch Time Fire Mission.

The model was corrected to check for this situation.

3.19 JTLS-2026-17835 ATOT Not Properly Filling Wait Flag For Mission Instructions

The ATO Translator is not including the mandatory WAIT flag for mission instructions that are landing or transiting at a point. This new flag is included for OAS missions and all those mission types that use the Mission Instruction (MI) Utility order.

When missions are translated by the ATO Translator the orders are checked by group but these fails because the WAIT flag is missing. The user must open the order utility for each of these missions and set/save this option manually.

The Mission Instructions utility and OAS instructions utility had recently changed to include the wait flag as a mandatory field. The Translator was not including this flag whenever the land or transit option is used in the mission order.

The Translator was changed to set this flag to FALSE for all transit or landing points of the mission utility order during translation.

3.20 JTLS-2026-17836 ATOT Should Set First Mobility Mission Stop As Dropoff

The ATO Translator writes a list of instructions for Mobility missions coming from the ATO. These missions may include landings for pickup or dropoff. The Translator is alternating the pickup and dropoff but, when the first stop includes a pickup, the simulation computes a large loading time for the mission. This loading time moves the on station time for the mission well past the computed on station time for ICC. This results in the ICC and JTLS-GO represent the mission time profile differently. When the first stop includes a dropoff, however, there is no on station time difference between ICC and JTLS-GO.

The ATO Translator was modified to ensure the first stop is always a dropoff. Thereafter the stops may alternate between pickup and dropoff.

3.21 JTLS-2026-17837 SVP Check For Illegal Characters

A user organization inadvertently used some copy and paste techniques when building their database. In doing so, the database contained several non-printable illegal characters which made restarting from checkpoints impossible.

A series of checks have been added to the SVP to find and report the existence of illegal characters held within all Text Variable data in the scenario. Each of these SVP Errors have no automatic correction capability. The Errors should be fixed manually prior to using the scenario. The single Warning does not need to be fixed.

- Error 176 reports that one or more illegal characters were found in the variable name and the text string indicated. The indices of the illegal character(s) are indicated in the message.
- Error 183 reports that one or more illegal non-printable characters were found in the variable name specified. Since the text string includes non-printable characters, the text string cannot be printed in the Error message. This may make it difficult to find. Please refer to the DDS User Guide for this Error to help locate and correct the problem.
- Warning 1166 reports that a semicolon exists in the specified field. Within JTLS-GO 7.0, the Cyber-Security Team has ordered that the semicolon must be added to the illegal character list. The existence of semicolons in the database are labeled as Warnings if they are contained in fields that do not require name uniqueness.

The JTLS-GO 7.0 Version Conversion Program, which converts a JTLS-GO 6.4 database to the JTLS-GO 7.0 format, will automatically replace any semicolons reported in Warning 1166 with a legal character. User organizations that are worried about the Cyber-Security issues concerning the use of the semicolon, should fix the 1166 warnings issues in their 6.4 databases; otherwise, the VCP will correct the problem automatically during the conversion to JTLS-GO 7.0 format.

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.

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
EEI	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 6.4.8.0 DATABASE CHANGES

No database format changes were made as a result of this maintenance release.

APPENDIX C. Version 6.4.7.0 REPOSITORY CHANGES

[JTLS-2026-17837 SVP Check For Illegal Characters, Section 3.21](#) reported issues with illegal characters contained in user databases. The two scenario databases delivered with the JTLS-GO 6.4 series, repository64 and wespac64 were updated to remove all semicolons from the database.