

JTLS-GO

Version Description Document

August 2026



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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.6.0)**

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

JTLS-GO 6.4.6.0 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

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1.2.1 Obsolete/Outdated Documents.....	1-1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.....	1-8
1.4.3 Standard Repository Changes	1-8
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	1-9
1.6.2 Checkpoint .ev_ascii Files	1-10
2.0 ENGINEERING CHANGE PROPOSALS.....	2-1
2.1 JTLS-2026-17776 MARITIME TRAFFIC GENERATOR	2-1
3.0 SOFTWARE TROUBLE REPORTS	3-1
3.1 JTLS-2026-17774 SINGLE TEXT DATABASE FIELDS READ INCORRECTLY	3-1
3.2 JTLS-2026-17775 SVP CRASH CONTAINER UNIT SQUADRON HOMEBASE	3-1
3.3 JTLS-2026-17777 UNIT TARGET SUMMARY RANGE RINGS NOT UPDATED	3-1
3.4 JTLS-2026-17778 ELS CRASH ACCESSING CONTAINER UNIT	3-2
3.5 JTLS-2026-17779 CANCEL CAS MISSION ISSUES	3-2
3.6 JTLS-2026-17780 COLLECTING SEALANE NODES IN PROPER RECTANGLE	3-3
3.7 JTLS-2026-17781 SVP NON-ROAD PATH EXISTENCE WARNINGS	3-3
3.8 JTLS-2026-17782 SET BARRIER TYPE CONSTRAINT INCORRECT	3-3
3.9 JTLS-2026-17783 MANAGE PIPELINE ORDER FOR FOREIGN PIPELINES	3-4
3.10 JTLS-2026-17784 CANNOT ASSIGN ROES TO VISIBLE OPAREAS	3-4
3.11 JTLS-2026-17785 SVP CRASH WITH NO PERSONNEL COMBAT SYSTEMS	3-4
3.12 JTLS-2026-17788 SET MIN TIME TO 0 IN WHIP ORDER PANELS	3-5
3.13 JTLS-2026-17789 OPAREA CANNOT HAVE ASSOCIATED ACM CHANGED	3-5
3.14 JTLS-2026-17790 FORTIFY FINDING FIXES	3-5

3.15 JTLS-2026-17791 NON-SQUADRON AIRCRAFT DIS CODES NOT CORRECT	3-5
3.16 JTLS-2026-17792 DIS CODE CONTROLLER CHANGE NOT SHOWN IN ENTITIES ...	3-6
3.17 JTLS-2026-17793 ATOT SETS CAS ALLOWED TO FALSE FOR ALL MISSION TYPES	3-6
3.18 JTLS-2026-17794 CRASH DELETING IADS NODE	3-6
4.0 REMAINING ERRORS.....	4-1
APPENDIX A. ABBREVIATIONS AND ACRONYMS	A-1
APPENDIX B. Version 6.4.6.0 DATABASE CHANGES	B-1
APPENDIX C. Version 6.4.6.0 REPOSITORY CHANGES	C-1

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.4.6.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) software suite. JTLS-GO 6.4.6.0 is a Major delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.6.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations called “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 6.4.6.0 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 DDS User Guide* (JTLS-GO Document 06, Version 6.4.5.0)
- *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 WHIP Training Manual* (JTLS-GO Document 10, Version 6.4.4.0)

- *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 DoD Architecture Framework* (JTLS-GO Document 22, Version 6.4.3.0)

1.2.3 Updated Documents

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.6.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.4.6.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.6.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.6.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.6.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.4.6.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.4.6.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)
- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)
- JTLS Symbols Application (JSYMS)
- Database Development System (DDS)

Database Configuration Program (DCP)

DDS Client User Interface (DDSC)

- ATO Translator Service (ATOT)
- ATO Generator Service (ATOG)
- ATO Retrieval Program (ATORET)
- JTLS Convert Location Program (JCONVERT)
- Count Critical Order Program (CCO)
- JTLS HLA Interface Program (JHIP)
- After Action Review Client (AARC)
- Scenario Data Client (SDC)
- Order Entry Client (OEC)
- Order Verification Tool (OVT)
- JTLS Object Distribution Authority (JODA)

The current JODA build number is 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 6.4.6.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.6.0 is not necessary. No other upgrade beyond installation of the compressed TAR files or CD is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called “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 6.4.6.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Server (ES) Edition Version 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 6.4.6.0 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 6.4.6.0 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 1.8.4.
- 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 6.4.6.0. 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.
- 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. 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 6.4.6.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

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

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the \$JGAME/data/hla directory. Federation testing of JTLS-GO with CAE's GESI model 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 6.4.6.0 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 6.4.6.0 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 6.4.6.0, but end-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

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

1.4.3 Standard Repository Changes

R&A has continued to improve and expand the unclassified data repository, which has been renamed to “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 6.4.6.0.

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 show 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 Parms 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*.

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 with HRU Traffic events.

This applies for any checkpoints with 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      3
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 1.000000000000000000
[more data...]

```

Figure 1.3 Current <scenario_name>.ev_ascii File Format

2.0 ENGINEERING CHANGE PROPOSALS

One model capability was added to JTLS-GO 6.4.6.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2026-17776 Maritime Traffic Generator

Summary of Model Change Request (MCR)

Customers have requested the representation of maritime shipping background noise, similar to how civil air traffic is currently handled. The new desired capability should meet the following requirements:

- The automatically-generated civilian maritime traffic should move along Sealanes.
- Use a set of pre-defined ships for the generated traffic. The data for these ships should include the ship name, the country flag under which the ship sails, and other unique ship identifiers, such as Maritime Mobile Service Identity (MMSI).
- Naval ports built into the database will be used as departure and arrival locations.
- The user will be able to define geographic areas in which to generate traffic.
- Ability to specify the number of ships to generate.
- Support for designating dynamic no-sail zones, causing ships to reroute automatically.
- The operator needs to maintain the ability to take control of any ship for Main Event List or Main Incident List (MEL/MIL) injects.

As an example, the user story could be specified as: “I want 100 ships sailing in the Mediterranean, flying flags from these nations.”

Design Summary

A new order, called Generate Maritime Traffic, has been created, specifying the desired traffic parameters. Once submitted, the model will automatically generate Enter Port orders for the ships that meet the generation parameters.

The requesting customer’s desire is to use this new capability in an upcoming exercise that will be using the JTLS-GO 6.4 series. The design calls for new database parameters, which cannot be directly entered in JTLS-GO 6.4, because the database format cannot change within a two digit Configuration Managed series. This problem will be bypassed by using what is known as the Non-Configuration Managed data file, in which the needed new data is uploaded from a hand-edited file.

The full capability of this ECP will be delivered in JTLS-GO 7.0, and the design is provided in the *JTLS-GO 7.0 Design Plan*. The unique changes needed to implement this same capability in JTLS-GO 6.4.6.0 are described in [Section 1.6](#) and in the *JTLS-GO Player Guide*.

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-2026-17774 Single Text Database Fields Read Incorrectly

If a text field consists of only one character and is the last field on a data record, the read routine puts an extra space after the single character. This was found when a user's database had a single character C4I NAME for a ground unit. When read in, the C4I NAME ended up two characters long, the extra last character being a space character.

This specific situation was encountered when a user's database had assigned a value of "1" to a ground unit's C4I NAME. The Version Conversion Program read in "1", but output a "1 ", a numeral one followed by a space. The resulting converted database could not be loaded into PostgreSQL because C4I NAME is not allowed to contain spaces.

The read routine now correctly reads in one character Text Strings even if the field is the last field of a data record.

3.2 JTLS-2026-17775 SVP Crash Container Unit Squadron Homebase

A database was created with the JLVC Fusion Tool (JFT) that had squadrons homebased at a Container Unit within JTLS-GO. The DDS restricts a squadron homebase to being an Airbase, FARP, or Naval Unit. When the SVP was run for this scenario, it crashed.

Even though this is an impossible situation when solely using the DDS, there are other legitimate ways to get the data into a database and the database itself does not prevent this situation. It is the responsibility of the SVP to catch and report the error without crashing in the process.

The crash happened while checking squadron-specific data, specifically while attempting to access the Tactical Unit Prototype (TUP) name of the container unit. Because Container Units do not have a TUP, the SVP crashed.

A conditional check was placed around the access of the TUP name to prevent the crash. Once the error of an improper homebase type is generated, the code now exits the routine for Container Units instead of attempting to perform other checks where the TUP name is required.

3.3 JTLS-2026-17777 Unit Target Summary Range Rings Not Updated

A SAM/AAA Target with a Fire Control Sensor having multiple ranges (such as Air Search, Surface Search, and Fire Control) was not included in the Target Summary menu, which prevented the different range rings from being displayed on the WHIP map.

Additionally, when a Controller submitted the Set Sensor Type Data order to change the range of a Sensor Target, the range ring displayed on the map did not reflect the new range.

New logic was added to include multi-role Fire Control Sensor ranges in the Target Summary menu, which were previously omitted because Fire Control Sensors belong to Targets, not Units/HRUs.

There was an error in the CEP checkpoint saved data for Unit Target Summary Range Rings that threw away some of the Unit Target Summary Range rings data and created new data with new receiver numbers when the initial JDS download was accomplished after coming up from a checkpoint. This error was corrected.

New logic was added to prevent the Controller from changing the range of a Fire Control sensor used by a SAM/AAA target to zero. A new message is generated to notify the Controller that the change request is ignored.

While testing these code changes, airborne sensor and jammer range rings were retested. When the sensor ranges were adjusted using the Controller Set Sensor Type Data order, the rings automatically updated. When the jammer ranges were adjusted, the rings did not update, unless they were turned off/on. A subroutine call to rebuild the jammer range rings was missing for jammers. The missing call was added to correct the problem.

3.4 JTLS-2026-17778 ELS Crash Accessing Container Unit

The ELS crashed attempting to access a Container Unit. The unit did not exist within the ELS data structures.

Container Units were not incorporated into the ELS when the JODA download was processed. This was not a problem for most scenarios because the Container Units were located at the highest level of the hierarchy. In this case, the Container Unit was placed at a middle level, where the parent and child unit relationships needed to be maintained.

Code was added to properly handle all Container Units and this resolved the crash.

3.5 JTLS-2026-17779 Cancel CAS Mission Issues

The logic that cancels a Close Air Support (CAS) request did not allow for active CAS missions originating from a Ground or Support Unit. Consequently, those missions could not be canceled using the Cancel CAS Request order.

The Cancel CAS logic omitted Ground and Support Units when looking for CAS missions to cancel from those unit types. The missing unit types were added to the search criteria.

While testing this change, it was discovered that aircraft belonging to a Ground or Support Unit could not be placed On Alert at their home unit. The model rejected the order with an incomplete Player message.

By design, only Airbases, Naval Units, or FARPs may be specified in the order as an Alert. However, if the Alert field is left blank, the home unit (no matter the type) is treated as the alert location by design. If the home unit happened to be a Ground or Support Unit, the logic failed to accept that alert location and rejected the order. The code that validates the alert location was changed to accept a mission that is from a Ground or Support Unit.

3.6 JTLS-2026-17780 Collecting Sealane Nodes In Proper Rectangle

When establishing a Sealane Route for a ship to follow, the model looks in a box around the start location for a feasible Sealane Node at which the ship can enter the Sealane Network. Similarly, the model finds a Sealane Node from which the ship can leave the Sealane Network and head to its designated destination. The model incorrectly computed the box that was supposed to be used to find the Sealane Start Node and Sealane End Node.

The ship's current location and the destination location were each supposed to be in the center of the node search box. They were not. The computation of the rectangle that defined the search box was corrected.

The same problem existed in the Scenario Initialization Program (SIP). Each program was corrected. Although no error was found in a subordinate routine in the SIP, the subordinate routine did not match the same routine in the CEP. The SIP code was changed to use the identical algorithm used by the CEP to compute the initial Great Circle Bearing from one point to another. This change was made for code consistency and ease of maintenance.

3.7 JTLS-2026-17781 SVP Non-Road Path Existence Warnings

One of the checks done by the SVP is to determine if there is a path between a Support Unit and all of the Units it has been designated to support. If no Road Network is available between the two Units, the SVP checks for a straight-line travel path. The implemented algorithm can indicate that the straight-line path exists when in fact it does not exist.

The SVP algorithm was not checking whether empty grids along the straight-line path would cause a travel problem. This meant a totally empty ocean grid along the straight-line path was not being tagged as a problem for trucks. The SVP came to the conclusion that a straight-line path existed.

The model was correct and was stopping trucks from traveling from the Support Unit to the supported Unit along the straight-line path, but because the SVP did not report the warning, the terrain could not be corrected prior to game execution. This problem was corrected in the SVP.

3.8 JTLS-2026-17782 Set Barrier Type Constraint Incorrect

The Set Barrier Type Order has an incorrectly-formatted constraint. The constraint is designed to ensure that if the user requests a Changes Only returned message, that the user submits an

order with a change to at least one Barrier Type attribute. The constraint designed to check for this situation did not do the job properly.

The constraint check was corrected, and the failure message was improved to properly indicate that no Barrier Type data was being changed. The new failure message reads: "To submit a Changes Only order, at least one Barrier Type attribute must be changed."

3.9 JTLS-2026-17783 Manage Pipeline Order For Foreign Pipelines

The Manage Pipeline Support order allowed pipeline operations to be stopped for a specified pipeline, even if the pipeline was currently operated by a foreign Side unit. Only pipelines currently operated by a unit from the player's own Side should be allowed to cease operations.

The Manage Pipeline Support order panel was modified to limit the selectable pipelines to only those pipelines belonging to the Player's Force Side when the Cease Operations option is selected. By implication, a unit from the Player's Side must be currently operating the pipeline in order to shut it down. Note that a pipeline's side is determined when a unit starts operating it. If there is no currently-operating unit, the pipeline has no Side.

Similarly, the selectable pipeline for the Begin Operations option was modified to allow only pipelines that belong to the Player's Side or have no Side. This change permits the Player to replace the unit currently operating the pipeline or begin operating an un-owned pipeline, thereby claiming it for the Player's Side.

A new check was added to prevent the Player from directing a foreign pipeline shutdown if the order was entered with a future execution time and a foreign unit began operating the pipeline in the meantime. A new Player message is generated to inform the Player.

3.10 JTLS-2026-17784 Cannot Assign ROEs To Visible OPAREAs

An OPAREA is owned by a specific Force Side, but can be made visible to multiple Sides. Each Side for which the OPAREA is visible can establish their own Rules of Engagement for the OPAREA, that should be used by their Side's assets. The ROE order incorrectly only allowed the OPAREA-owning side to establish ROEs for itself.

The order panel was changed, to allow any Side for which the OPAREA has visibility to create their own set of unique ROEs.

3.11 JTLS-2026-17785 SVP Crash With No Personnel Combat Systems

The scenario database was changed so it had no personnel systems. The SVP crashed.

The SVP was corrected not to crash under this circumstance. Once fixed, the SVP was already configured to give several Errors indicating that the serious problem needed to be solved. Thus, the only change that needed to be made was to ensure the situation did not cause an SVP crash.

3.12 JTLS-2026-17788 Set Min Time To 0 In Whip Order Panels

In the WHIP order panels, users could input any time in the time fields that occurred before the game start time.

The game start time is now the minimum start time in order time fields. If a user tries to set a time that occurs before game start time, the calendar will automatically set the time to game start time.

3.13 JTLS-2026-17789 OPAREA Cannot Have Associated ACM Changed

An OPAREA can either be a Player-specified polygon or an Air Control Measure polygon. The vertices of an OPAREA polygon can be changed by either specifying a polygon or specifying a different ACM. Modification of an existing polygon by specifying a different ACM that should be used was not processed by the model.

The code only altered the vertices of the OPAREA if the Player specified a unique polygon as part of the order. If an ACM was entered, the change was ignored.

The code was in place to alter the OPAREA based on the ACM specified, but due to the code error, the alter location of the OPAREA never got called. The alter OPAREA location code is now called whether a new polygon is specified or a new ACM is specified.

3.14 JTLS-2026-17790 Fortify Finding Fixes

Correctable privacy violation vulnerabilities were identified in the latest Fortify code inspection. These violations converted the game's Apache password to a string when in use, and converted a string password to make a connection to an Oracle database when Oracle database connections are no longer supported.

The Apache password is no longer converted to string and is now securely used and cleared after its use, and the legacy Oracle connection code where a string password was used has been removed.

3.15 JTLS-2026-17791 Non-Squadron Aircraft DIS Codes Not Correct

The ELS published entities for individual aircraft in the game. When the aircraft were owned by non-squadron units, the published DIS Codes did not correctly reflect the type of the aircraft.

When creating the disaggregated entities, the ELS was not considering the case where aircraft were owned by Unit types other than squadrons. Code was modified to correctly use the DIS Code for the appropriate aircraft type for all types of Units.

3.16 JTLS-2026-17792 DIS Code Controller Change Not Shown In Entities

When a Controller changed the DIS Code of a Combat System or an Aircraft Class, the Combat System entities published by the ELS did not have the new DIS Codes.

DIS Code changes were not properly propagated through to the entities in the ELS. The DIS codes were updated for the Combat Systems and Aircraft Classes, but they were not correctly changed on the entities. Code changes were made to apply these changes.

3.17 JTLS-2026-17793 ATOT Sets CAS Allowed To False For All Mission Types

The ATO Translator module in the WHIP frequently reads and saves the XML linking data that is used for translating various ATO mappings. One configurable parameter that is read and saved in the linking data is the option to allow CAS tasking for specific mission types. However, when the Mission Types mapping page is displayed, all missions types have "false" for the CAS parameter.

The linking data uses XML tags to identify parameter values in this data. The tag for the CAS parameter was incorrect in the ATOT module, so the value was getting lost when reading the CAS value. Without a value for the CAS parameter the Missions mapping page simply showed "false" for the CAS parameter for each mission type.

The value was also written incorrectly, so the ATOT service would not get the proper value for the CAS parameter when writing orders.

The tag name was changed to the proper value in the ATOT module so the CAS parameter is communicated properly between the ATOT module and the ATOT service.

3.18 JTLS-2026-17794 Crash Deleting IADS Node

The Controller sent an order to remove a link between two targets on an IADS network. The model determined that one of the nodes within the link was no longer needed (no other arcs were connected to it). The model attempted to delete the node without removing it from the network set or the grid set, causing a crash.

A routine exists that is designed to clean up and delete a network node, including removing it from sets. Instead of just deleting the node, the code was modified to call this routine. The routine itself needed some minor modifications to handle an IADS node, but otherwise performed all necessary cleanup.

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 6.4.6.0 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
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

ICPLogin	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 6.4.6.0 DATABASE CHANGES

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

APPENDIX C. Version 6.4.6.0 REPOSITORY CHANGES

No significant changes have been made to the structure of the JTLS-GO 6.4.6.0 repository.