JTLS-GO

Version Description Document

June 2026

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.5.0)

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

JTLS-GO 6.4.5.0 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-60 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

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.4.5.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.4.5.0 is a Major delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.5.0 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 6.4.5.0 executes on the Red Hat Enterprise Linux Version 9.4 and Oracle Linux 9.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any 64-bit operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

1.2.2 Unchanged Documents

- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.4.0.0)
- *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 Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.4.3.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.4.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.4.3.0)

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

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.4.5.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 02, Version 6.4.5.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.4.5.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.5.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.4.5.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.4.5.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.5.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.4.5.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.5.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.5.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.4.5.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

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

    Database Configuration Program (DCP)

DDS Client User Interface (DDSC)

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

  The current JODA build number is 234.

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

  Apache Server (APACHE) version 2.4.66

  JTLS XML Serial Repository (JXSR)

  Order Management Authority (OMA)

  Synchronized Authentication and Preferences Service (SYNAPSE)

  XML Message Service (XMS)

  Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service
- Keyhole Markup Language (KML) Operational Interface (KOI)

- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (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 6.4.5.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.5.0 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 6.4.5.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Server (ES) Edition Version 9.4, 64-bit architecture.

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

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

Oracle Linux 9.6 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by Defense Information System Agency (DISA) for use by U.S. Government Agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must be a 64-bit architecture with a Java-enabled web browser. JTLS-GO 6.4.5.0 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 6.4.5.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 492 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 6.4.5.0. 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-2.0HLA Compliance

- KML Operational Interface (KOI)

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

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

1.3.2 JTLS-GO Cybersecurity Compliance

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

One of the DoD requirements is that the software must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. The concept of an expiration date has been removed from the DoD requirement, but the concept of always using the latest version of third-party software remains a strong component of DoD Cybersecurity requirements.

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

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

- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a bug release version (JTLS-GO 6.4.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the bug released versions when distributed.

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

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.4.5.0 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 6.4.5.0 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 6.4.5.0 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 6.4.5.0, 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 6.4.5.0.

2.0 ENGINEERING CHANGE PROPOSALS

No model capabilities were added to JTLS-GO 6.4.5.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

3.0 SOFTWARE TROUBLE REPORTS

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

3.1 JTLS-2026-17677 Cannot See Foreign Ships In Formation

A Blue Formation included Green ships and some Neutral ships. When asking to see the ships in a formation, from the Blue Side WHIP IMT, the user could not see the Green or Neutral ships.

A new IMT panel capable of showing foreign ships was created.

3.2 JTLS-2026-17679 i18n Discrepancies

While demonstrating the i18n translation capabilities, the IMT menu items were not translated, and the WHIP Login Frame was not displaying the number of available WHIPS in the foreign language.

The IMT menu was separated from the standard menu, because it is consistent for all WHIP users. However, the i18n generator tool was not updated to handle the new format and the WHIP was not accessing the correct properties file to build the menu text. Both problems were fixed.

The Login Frame uses the commonui.properties file to determine language translation. However, the properties to specify the number of available WHIPs was in the whipapp.properties file. These resources were removed from whipapp.properties and placed in commonui.properties.

When testing, it was noted that the tooltip for the debug button on the Login Frame was not updated when the language was changed. The update of this text item was added to the source code.

3.3 JTLS-2026-17680 Creating Naval Unit Aircraft Check Messages Bad

When a user creates a new Unit during game execution, two aircraft checks are accomplished:

- If the user specifies a type of aircraft to assign to the new Unit and the Unit prototype does not include any aircraft Combat Systems, a message is generated and the order is rejected.
- If the assigned prototype includes an aircraft Combat System and the user did not indicate the type of aircraft that should be assigned, a message is generated and the order is rejected.

In both situations, the generated message assumed that the Unit being created used a Tactical Unit Prototype (TUP). This is no longer true, because even Naval Units can be assigned aircraft.

The code was changed not to output the index identifier for the prototype, but the name of the prototype. In turn the message format file was changed not to assume that the index provided belonged to a TUP, but simply printed out the name of the prototype that was included in the message data.

3.4 JTLS-2026-17682 MDP Action Button Aesthetics

Some of the buttons on the MDP GUI have a lowercase first character. Though this does not affect the functionality, it is a standard for the first character to be uppercase.

The lowercase first characters were changed to uppercase, which matches the majority of the buttons on the MDP GUI.

3.5 JTLS-2026-17683 OTH-Gold Terminal Console Aesthetics

When connecting to an OTH-Gold Message Service with a JODA console and requesting help, one of the help items has a description of "set attribue for named object", misspelling the word attribute.

The offending help text was corrected.

3.6 JTLS-2026-17684 Protect Save Checkpoint Files From Deletion

When modifying and running the savchkpt.tmplt file to backup checkpoints, the resultant tar files are left with read/write permissions. To protect them from accidental deletion, they should be changed to read only.

The script was modified to adjust the permissions after the tar file is created.

3.7 JTLS-2026-17685 Add Recipient To MDP Log

The MDP log records the recipient of sent, e-mail or socket, and saved to file messages, but this information is not visible in the GUI.

The text within the GUI log viewer was expanded to include the name of the recipient for e-mail and socket sent messages, and the file name for saved messages.

3.8 JTLS-2026-17687 Individual Target OPM Pages Missing HTML Extension

When opening an Online Player Manual (OPM) page for an individual Target from the WHIP, the URL for the target was missing the ".html" extension, causing the load of the page to fail. Each Target has an OPM page.

The ".html" extension was added to the URL.

3.9 JTLS-2026-17689 JDSP String Duplicate Routine Memory Copy

The routine set_STRING() allows the copying of memory from the source to the destination when the source and destination are the same memory location. For Linux, this is undefined and may not copy properly, which resulted in garbage in the destination string.

The set_STRING() routine was modified to first check if the source and destination were the same. If so, the copy is unnecessary.

3.10 JTLS-2026-17690 JODA Peer Client Reports Peers It Does Not Have

The JODA uses the JDSP library of routines and variables to manages peers and peer requests. Two JDSP routines, peer_db_list_known_peers() and peer_db_list_known_peers_table(), print each from the array of peers whenever the user request a list of peers of the JODA from which the peers are connected. The list is maintained by all JODAs connected to the Primary JODA, so it contains a complete list of all peers in the tree of JODAs, starting with the Primary one. The Logging JODA are excluded.

The list should only show the peers connected to the JODA from which the list was requested.

The routines that print the list do not consider the parent JODA of each peer. Therefore, all peers in the list are displayed from the interface of any JODA (Logging JODA excluded).

These routines were modified to print from a JODA's interface only the current and former peers which are/were connected to this JODA or EODA. Any of the remaining peers will be printed from the JODA or EODA to which they are connected.

3.11 JTLS-2026-17691 Whip Notification Tool Updates

The WHIP Notification Tool (WNT) incorrectly displays Super WHIPs. The WNT displays all WHIP logins in a tree like structure where each WHIP is listed under its corresponding side. Super WHIPs have multiple Sides associated with them, and are displayed in the WNT only under one Side. This makes finding Super WHIPs difficult, as you need to search under each possible side the Super WHIP is assigned to in order to find it.

Additionally, the WNT does not have search capabilities for searching for WHIP logins.

The WNT now displays duplicate Super WHIP entries under each corresponding Side. When a user updates the state of the Super WHIP entry, any duplicate logins will also update accordingly.

A search bar was added to allow users to dynamically search for WHIP login names. The tree will update per keystroke.

3.12 JTLS-2026-17694 Orbit Paths With Identical Termination Points

Orbiting air missions with very close orbit path points (including start and end points) cause the model to perform air mission movement at a very high frequency. This causes a strain on the model.

If given orbit paths are too close to each other, either convert the order to Orbit Point or reject the order. The criteria for determining a close orbit path is all distances between each points within a given orbit path (including start and end points) being less than the defined threshold. The order is converted to Orbit Path if no given Orbit Path exists, and only start and end points are given.

The order is rejected if the Orbit Path is given, as the user's intention is more ambiguous in this case.

3.13 JTLS-2026-17697 Red Background in Error Fields Are Too Harsh

The red background box color for error text fields make the foreground text difficult to see, especially for black text color.

The background color for those boxes was changed to light red for easier visibility of the text. The following components have been changed:

- WHIP: error text field search for IMT screens, Command Hierarchy, ORBAT search
- DDS: error text field search for ORBAT search, Command Hierarchy, Logistics Hierarchy, DJINN database search
- ICP: cell editor error message that pops up in the lower screen

3.14 JTLS-2026-17699 ATOT Repeats Error 218

Whenever the ATO contains a mission having more than 99 or less than zero aircraft, the Translator needlessly repeats the error for this condition many times. The error, "Number of aircraft for mission is invalid", is written too many times into the errors and warnings report.

The ATOT correctly checks for this error condition but is doing the check in the wrong location of a code loop. The check is inside a loop for computing aircraft tasking for each squadron.

The check for error aircraft count for each mission was moved to the appropriate location.

3.15 JTLS-2026-17701 DDSC ICAO Fields Header Titles

The header title for the ICAO fields in the Squadron and the FARP unit DDSC tables should be capitalized to "ICAO" to be consistent with the other table header titles.

The header title for the ICAO fields in the Squadron and FARP unit DDSC tables was capitalized.

3.16 JTLS-2026-17703 Leftover Line Grids Logic Error

A "Left Over Line Grids" logic error was continuously generated by the CEP.

When a Naval unit attempts to enter a sealane network, the logic finds the closest network node as an entry location. During this process, the logic calls a routine that determines whether a path to a node is feasible. However, that routine was not emptying the data used to determine the path to a node that is being tested for feasibility, but the routine was expecting an empty data set. To stop the logic error, code was added to empty the data set after each call.

Note this correction also applied to Air Missions attempting to enter a flight path network.

3.17 JTLS-2026-17704 Sensor Max Range Zero Crashed Model

The CEP entered Simdebug when a zero (0.0) maximum range was entered in the Set Sensor Type Data order.

The legal minimum for the maximum range of a sensor type was changed from 0.00 to 0.001 in the Sensor Range List utility.

Update: *This change was reversed*. The STRN.xml utility was changed back to accept 0.00, pending further investigation.

3.18 JTLS-2026-17706 ICP Whip Panel Has No Searching Capability

Searching for a WHIP name in the ICP WHIP panel can be difficult when there are many WHIPs in the scroll panel.

A search bar was added to the lower panel to allow dynamic searching that updates the WHIP's table view per user keystroke.

3.19 JTLS-2026-17708 Update WHIP Profile For ATO Operations

There should be a WHIP profile definition file for an ATO processing WHIP that has access to the Manage Air Control Mean Area order.

Added a new WHIP definition file menu_atot_air.xml that has access to both ATO modules and the MCAM order.

3.20 JTLS-2026-17709 JODA Not Updated Removing Naval Support Unit

The JODA is not being updated if a WHIP operator changes the routine support unit for a Naval Unit to None.

The code that updates this JODA attribute was not being skipped in the new Support Unit was None. This error was corrected.

3.21 JTLS-2026-17711 Crash Magic Moving HRU On Grid Corner

The model crashed when a Controller submitted a Magic Move order for an HRU that was placed exactly on a grid corner. The crash occurred because the model was attempting to remove the HRU from the wrong grid.

This problem has happened before and was corrected, but the Magic Move routine was missed and did not have the correction that was previously implemented in other places when an HRU is being removed from a grid set.

The routine that caused the crash was missed because the code to remove an HRU from a grid was not centralized. Instead of correcting this one routine, the decision was to centralize the code and always use one routine to correctly remove an HRU from the grid to which it belongs.

3.22 JTLS-2026-17713 Changing SLP Convoy Limits Changes Attributes

A user wanted to change the attribute for a Barge Convoy, indicating that it should have a minimum distance of 1000 KM, rather than the current database value of 20 KM. When changing that parameter, the end result was that the SLP Barge parameter indicating that the Minimum Unit Size was blank. The model did not crash, but OPMs could not be generated.

The problem was that the code to interpret the Convoy limits did not check if the user had in fact changed the Minimum Unit Size parameter. This caused the Minimum Unit Size attribute to be set to "zero" because the order field was not filled.

Several issues were found as a result of fixing this issue:

- The Utility that passed the requested Supply Limit changes was changed to an "initialize" Utility. This results in the utility order being filled with negative 1, when the user does not fill in an optional field.
- The code to interpret the utility records was changed to use the coding standard of .ORDER.INITAIIZATION.VALUE indicating to everyone accessing the code to expect -1s in unfilled fields.
- The Show SLP parameter did not properly identify which Convoy type was being changed if the SHOW ALL option was set to No.

All three issues were fixed.

3.23 JTLS-2026-17715 SDC Disk Space When New Filter File Deployed

An SDC was being run for each Side in the scenario. A new filter file was created for every SDC. The SDCs recognized the existence of the new file and attempted to load it. This load failed and the SDCs went into a loop writing out error messages to their logs. Because the SDC was busy

with the error messages, it did not respond to the WSM requests and appeared to have crashed. The SDCs quickly filled up the available disk space, which caused other issues.

The SDC code attempted to add and subtract from the existing JODA subscription when a new filter file was loaded. This does not appear to be a capability of the JODA libraries.

This code was removed, and instead the SDC produces an error message every 60 seconds informing Tech Control that it should be stopped and restarted to use the new filters.

3.24 JTLS-2026-17716 Additional Filters for IMT TPFDD Panel

The IMT TPFDD panel shows all Units with a future TPFDD arrival. On the Controller WHIP this includes Units on all Sides, and there is no ability to filter the Units by Side.

Standard filters to filter by Unit type and Unit Side were added to the IMT TPFDD screen.

3.25 JTLS-2026-17717 Require DIS Code for Naval Unit Creation

The Create Unit order does not have a field to specify the DIS Code for new Naval Units. This results in a newly-created unit that has a empty string for the value, which when written for a checkpoint results in a bad Naval Unit file that cannot be read back into the game.

A new mandatory field was added to the Create Unit order for Naval Units. This field should check that the format of the entry matches the basic format.

It has also been noted that the Set Unit Parameter order allows the Controller to change the DIS Code of a unit. If an invalid format for a code is entered, the CEP will refuse the change. This checking of the entry format should be done at the WHIP and OMA level. A pattern was added to the DIS Code field in the Set Unit Parameter order to ensure the entry is of the proper format.

3.26 JTLS-2026-17718 Conflicting Mission Canceled Reports

When an air mission with a resource task that was delayed over its maximum delay time is canceled, the user received two conflicting error messages:

"The mission was canceled by a Player" and "The Mission was Canceled. Its Maximum Launch Delay Time was Exceeded. It was Delayed Due to Aircraft Availability."

The first message was incorrect and should not have been generated. This has been corrected.

3.27 JTLS-2026-17720 IMT Region Filters Do Not Work Across the 0E/0W

If a Geo Region filter is drawn that crosses the zero longitudinal line when it is used in the IMT, all objects outside the region, but within the latitudinal band, are retrieved and no objects within the region are retrieved. This issue exists because the JXSR converts all negative longitudes to

positive longitudes from the GMT line, so 10W becomes 350E. This removes any issues with Geo Regions going over the International Data Line, but introduces the issue for the GMT line.

The JXSR code was improved to determine if the GMT is being crossed, and if so slightly different checks are used to determine if an object is within the Geo Region. Logic was also written to handle geographic regions that cover the poles.

While fixing this change, some programming level debug statements were also removed from the JXSR debug log.

3.28 JTLS-2026-17722 DDSC DRM Help Link Not Functioning

The DDSC includes a function that is intended to display the associated PDF file of the Data Requirements Manual. However, this help link to the DRM is currently not functioning.

An automated process that was used to split the Data Requirements Manual PDF file into separate files inadvertently deleted the "dds" documents subdirectory. This occurred due to a change in the name of the data_requirements_manual.pdf file. As a result, the JTLS-6.4.4.0.documents.tar.bz2 and JTLS-6.4.4.0.documents.zip files did not contain the "dds" documents subdirectory. The related cron job has been corrected, and the "dds" documents directory has been restored from the previous JTLS-GO 6.4.3.0 release.

3.29 JTLS-2026-17723 Assess Weapon Damage Event CEP Crash

The CEP crashed while executing an Assess Weapon Damage event. The reason for the crash was an incorrect entity type for an attribute.

The event was attempting to assess damage on a ground mission, but the code was trying to access attributes for an HRU. The problem was that the routine TRANSLATE DATA TYPE did not have a case for ground missions, so it defaulted to HRU. A case was added to account for ground missions and the problem was resolved.

3.30 JTLS-2026-17724 Login Build Default Context-Sensitive Menu Option

The Login Build order should be one of the default options in the Context Sensitive Menu on the Controller WHIP. Set Unit Disease Failures should be replaced, as it is rarely used.

The Set Unit Disease Failures order was replaced with Login Build in the Unit Context Sensitive Menu for Controller WHIPs.

3.31 JTLS-2026-17727 No Weapons Listed In Fire Artillery Order

A Fire Artillery order was selected for a Unit that had indirect fire combat systems. The order had no weapons in the Projectile Type drop-down list, even though the Unit had the correct ammunition supply categories, with non-zero amounts on hand.

This behavior was reproduced by creating a new artillery Unit while the game was executing. After the new Unit was TPFDDed into the game, the Fire Artillery drop-down list did not have any weapons. The model failed to create the list of usable weapons for the newly-arrived Unit.

The existing Unit arrival process built weapon lists for database Units arriving at game start but did not build lists for new Units created during execution. Because both situations execute the same Finish Unit Arrive routine, the routine call that builds the usable weapons list was moved from the Complete Unit Initialization routine (used only at start) to the Finish Unit Arrive routine.

3.32 JTLS-2026-17729 Incorrect Satellite Type Printed

A Controller created several satellites using the JTLS-GO Satellite Service (JSAT). All of the satellites were created with a particular Aircraft Target Class (ATC).

The satellites were then engaged using an ADA site. One of the satellites was destroyed and the Air Defense Report was generated. Part of this message describes the ATC for the satellite. In this case, the message reported it having the wrong ATC.

The Air Defense Report referenced the wrong data item. The data item was changed to match the output variable location in the write statement in the underlying routine.

3.33 JTLS-2026-17731 Change Fuel Type Crash

A Player sent a Change Mission Parameter order to change the fuel type of an air mission that was in Fuel Delay status, which caused a model crash.

A mission in Fuel Delay posture does not have an assigned fuel type. A request to change fuel type from the Change Mission Parameter order cannot be executed before the mission's initial fuel type has been assigned. The logic was changed to reject a change fuel type request when the mission is in Fuel Delay posture with an appropriate player message. This change is consistent with existing logic which also rejects a change fuel type request if the mission has already started (i.e. in the air).

3.34 JTLS-2026-17732 Projected Force Strength AAR Report

The user does not have a good way to view current force strength within an Operations Area versus enemy strength, nor to estimate the future force strength versus enemy strength.

A new AAR report was generated to generate the weighted strength of friendly and enemy units within a geographic area, along with the forward-estimated friendly weighted strength based on TPFDD events, and compare the strengths.

The user specifies time intervals for an evaluation and the friendly estimated strength at each time step is compared to the friendly estimated final strength. The friendly estimated strength at each time step is also compared to the current enemy strength. This provides the user with a way

to see the build up of friendly forces against the current enemy, and the status of that build up versus a projected end state.

While implementing the report an error was discovered with the display of durations within the report stylesheets. This issue was also fixed.

3.35 JTLS-2026-17733 Create Unit Rejection Message Names

A unit was created with a long name and a short name. Another user then tried to create the same unit using the same long name and a different short name. A rejection message was received, saying the short name and the long name were already in use by another unit.

Creating a unit A, then creating a unit A^1 with the same long name but different short name, did not reproduce the message that both short and long name are already in use. The message correctly highlighted just the long name.

A further investigation revealed that it is possible for the new unit A^1 to have shared the short name with a completely different unit B. The message does not specify whether the unit that overlaps with the short name is the same as the unit that overlaps with the long name.

A new line was added to specify how many units overlap with the short and long names of the new unit. This detail removes the confusion on how many names overlap.

3.36 JTLS-2026-17736 Combined Aircraft Load Not Destroyed

In the routine Check Load Validity, a combine aircraft load is created which combines a user-specified load with the aircraft default load. This combined load is checked for validity that it will fulfill the requirements of the mission type. Once the check is done, the combined load is no longer needed. It was only destroyed and its memory released if the decision was that the load was valid. If the load was invalid, the load was not released.

The code was changed to always destroy the combined load, whether it was or was not valid.

3.37 JTLS-2026-17738 OPM Unit Page Link 16 Mission Blocks Link

A Unit's mission block data was missing from the OPM Unit page, as well as the corresponding list of Link 16 Mission Blocks page.

An OPM page for Unit mission blocks was added. A row was also added to the Unit page that links to the Unit's mission block.

3.38 JTLS-2026-17740 ATOT Uses Squadrons With Zero Aircraft TOE

Whenever a squadron in the Tasking Unit mapping of the Translator has a zero (0) TOE for aircraft, the Translator still uses the squadron for the assignment of missions from the Tasking Unit for which the squadron is mapped.

The logic used by the Translator for selecting the appropriate squadron for each mission was not considering the number of aircraft at the squadron. A squadron in the Tasking Unit's mapping list was just as likely to be selected as any of the others mapped to the Tasking Unit.

The logic was changed to generate all permutations of missions vying for the squadrons of the specified aircraft type at the Tasking unit. It then computes the number of missions able to gain the required aircraft from the squadrons at this permutation. The permutation yielding the highest number of missions with their required aircraft is ultimately used by the Translator for writing the air mission orders.

3.39 JTLS-2026-17741 Contamination Status Incorrect

The model establishes the status of Airbases, FARPs, and Naval Units to report to C4I systems. One of the checks is whether the unit is within a contamination area. The distance between the unit and the weapon explosion was not properly checked.

The distance check was corrected.

3.40 JTLS-2026-17743 Inefficient End Contamination Logic

When nuclear or chemical contamination ends, the model does one last contamination assessment for the unit. After the assessment is complete, the model need to determine if the unit is covered by other contaminations of the same type. This determination is inefficient and ends up cycling through each end contamination event in the game. It only needs to check those end contamination events that are covering the unit.

The logic was changed to check only contamination events that are currently covering units that were covered by the one contamination event that just finished.

3.41 JTLS-2026-17745 Preamble Overlap

In the CEP Preamble, two values for the GE.TYPE entity overlapped. Although in 6.4 it does not produce an error, it can induce an error for a select case statement in 6.5.

One of the overlapping values was changed.

3.42 JTLS-2026-17746 Air Movement Report Missing Lifted Missions

An Air Mobility mission was tasked with carrying several UAV missions for insertion. Unlike ground units, HRUs, or supplies, the Air Movement Report did not list the carried air missions. Also, the pending task list mislabeled the associated missions as "Associated ACM" in the Insert task.

The report template and underlying routine were corrected to display all mission names, aircraft types, and mission types carried on the lift mission.

Also, the label was corrected in the pending Insert/Extract/Pickup/Dropoff tasks to "Associated Mission".

3.43 JTLS-2026-17747 End Contamination Does Not Save Radius

Event attributes are saved with values passed in during event scheduling. The radius of a contamination area is passed in on every scheduling instance, but the event routine END.CONTAMINATION itself did not have it defined as its parameter.

The event routine was using the Targetable Weapon's radius instead. The Manage Contamination Area order allows a user-defined radius of a contamination area, making the weapon radius incorrect. The parameter for event attribute radius was added to the END.CONTAMINATION routine. Redundant lines for manually setting event attributes were deleted. In JDS routine, the line for accessing weapon radius was replaced with EC.Radius.

Also, there was an error in which a checkpoint restart would not initialize more than 1 dispersal object tied to END.CONTAMINATION event. This was due to its reference number being passed in as a pointer, instead of a value. This was fixed.

3.44 JTLS-2026-17749 JDS Update Object Loses Weapon Count

If the user performs a JDS Update on a Unit, the Unit loses all of its Weapon Count data.

The routine JDS.UNIT.REINITIALIZATION is responsible for deleting any sub-entities of a Unit and the Unit itself before initializing everything in JODA. It did not delete Weapon Count objects, so they existed in the CEP and JODA. The JXSR does not link the existing Weapon Counts when a Unit is created after them.

New Weapon Counts associated to the Unit are now created after the Unit is created. The reinitialization routine was also updated. No other changes were required as the routine already takes care of the weapon initialization.

3.45 JTLS-2026-17752 Add Orbit Task To Air Mission Rejected

A Patrol air mission was guiding off a Formation. The Player submitted a Manage Air Mission Tasks order to add a new orbit task at a different bearing and range. The order was rejected with a peculiar message:

MANAGE.AIR.MISSION.TASKS order named ORBIT cannot be executed:

The specified HRU MH60R-060003 does not exist.

The rejection error was caused by an overloaded variable in the Manage Air Mission Tasks order. The Formation name and a Unit name used the same variable, which was then accessed by the underlying routine. The logic assumed that the variable was a Unit name, when in fact it was a Formation name. The logic attempted to accept the order as a Unit and when it determined no

such Unit existed, it assumed the Unit was actually an HRU. The logic then determined that the HRU did not exist and rejected the order.

The problem was corrected by assigning the Formation name to its own variable in the Manage Air Mission Tasks order. Also, the routine was changed to look for the Formation name in the correct order field when processing a Manage Air Mission Tasks order.

3.46 JTLS-2026-17760 Satellite Fire Weapon Multiple Received Orders

A Satellite Fire Weapon order was submitted to an orbiting satellite that was at the end of its programmed route and no longer moving. The CEP window filled with repeated Received Order events.

When the satellite first received the Satellite Fire Weapon order, it was not within range of a satellite communications facility and the owning force side did not have a global satellite communications capability. By design, the Satellite Fire Weapon order was rescheduled to be received after the satellite's next move in case it acquired communications. Hours later, the satellite reached the end of its route and stopped moving without ever acquiring communications. The Satellite Fire Weapon order event was then rescheduled every few seconds because the satellite's next move time was forever frozen, and therefore earlier than the current advancing game time.

New logic was added to cancel the Satellite Fire Weapon order when a satellite stops moving and communications are not transmittable to the satellite. A new message was added to inform the Player of the situation.

Another message was added to inform the Player the first time (only) a Satellite Fire Weapon order is submitted and there are no available communications to the satellite. In this situation, the Player is informed that the model will keep rescheduling the order receipt after each move until the satellite has communications and can fire.

3.47 JTLS-2026-17761 Update PostgreSQL Database Client

Minor PostgreSQL updates are essential for improving system stability, ensuring security, and fixing bugs. PostgreSQL 15.18 has been released and needs to be delivered as part of the next maintenance release.

The PostgreSQL Client, which is distributed with JTLS-GO, has been updated to the latest 15.18 security version. Also, a custom-created PostgreSQL 15.18 database server tar file has been included with the JTLS-GO release.

The detailed security fix information for 15.18 version can be found at:

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

3.48 JTLS-2026-17763 CEP Routine Unit Checks

The ACCEPT.ORDER routine performs a series of Unit checks within the CEP. The routine does not always select all Units that require checks.

The routine was modified to enable it to select all Units that require the relevant checks.

Additionally, the JUST.HRU.CHECKS routine was corrected to select the correct attribute for the name of an HRU.

3.49 JTLS-2026-17764 DDSC Command Hierarchy Change Unit Type Update

The DDSC Command Hierarchy did not update appropriately after the user changed a unit's type between a Container Unit or another type. The unit also disappeared from the Hierarchy and can only be redisplayed after the user reloads the data.

The new Container Unit type was not properly handled by the Command Hierarchy. The code has been fixed to properly handle the new unit type.

3.50 JTLS-2026-17765 DDSC/WHIP Map Filter Container Units

Container Units were missing from the DDSC and WHIP Map components' Command Hierarchy filters.

The Map filter was not processing the new Container Units. This was common code used by the DDSC, JOBE, and WHIP, so all of their Command Hierarchy map filters were missing Container Units. The oversight has been resolved.

3.51 JTLS-2026-17766 DDSC Command Hierarchy Collapse After Change

When the user changed a Unit's type, if the Command Hierarchy was expanded to show or select the Unit, this expansion and selection was reset.

The code has been fixed so that the old node's expansion and selection state are preserved after a change of a Unit's type.

3.52 JTLS-2026-17767 Mine Laying Mission Incorrect Load Delay

A Mine Laying Mission order was submitted with the required Targetable Weapon (TW) specified. The specified TW was the same as the TW in the defined Mine Laying Mission load for the aircraft type. The mission went into Load Delay, even though the Load existed for the Mine Laying Mission in the database, and the weapons were present on the airbase.

A code error in the routine that assigns the mission load prevented the valid database load from being filed in the Load Check Set. The Load Check Set was therefore empty, and no load could be

assigned to the mission, which then entered Load Delay posture. The local Error Code variable was also not re-initialized to zero within the loop that examines the weapon load alternatives.

The code errors were corrected. Parts of the code were rewritten to improve readability and efficiency.

3.53 JTLS-2026-17768 Mine Laying Mission Crash

A Mine Laying Mission order was submitted with both the required TW specified and the Weapons Only option specified. The required TW was the same as the TW in the mine laying load in the database. The Weapons Only option specified a different TW type.

The mission was loaded with the TW specified within the order and flew to the delivery location. The model crashed attempting to deliver the database-specified TW instead, which were not present on the mission.

The mandatory (original) TW specified in the Mine Laying Mission order is the weapon assigned to the mission's delivery task. When the Weapons Only option is also selected, the specified weapon is loaded on the mission. At the delivery location, the logic looks for the original TW assigned to the delivery task on the mission. The original TW was not present because the Weapons Only weapon was loaded on the mission.

The logic recognized the situation but failed to set the Error Code properly by using an AAR status instead of an Order status. There was no CASE for the AAR status in the delivery routine which caused the model to crash. To correct this error, the AAR status was replaced with the Order status error code: .ORDER.STATUS.IMPROPER.WEAPON. Also, the missing CASE was added using the corrected error code.

3.54 JTLS-2026-17769 SIP Warning 1639 Not Properly Skipped

The database builder has the ability to skip the generation of some warnings in the SVP. Warning 1637, 1638, 1639, and 1640 were not being properly skipped:

- Warning 1637 indicates that the targets in a Basic Encyclopedia (BE) Facility were too far apart.
- Warning 1638 indicates that the units in a BE Facility are too far apart.
- Warning 1639 indicates that probability of detection for a target in the BE Facility cannot be detected.
- Warning 1640 indicates that probability of detection for a mobile target in the BE Facility cannot be detected.

The code was fixed to properly skip the warning, if the user indicated that it should be skipped.

3.55 JTLS-2026-17770 Detachment Target Summary Menu Missing

A Player submitted a Detach order that included Targets. The Detachment's context sensitive Target Summary menu did not display. Also, the Parent Unit's Target Summary menu did not update to reflect the transfer of all Targets (of a particular subcategory) to the Detachment.

The establish detachment logic was missing code to create/update the Target Summary context sensitive menu for the Detached unit and the Parent Unit. Likewise, the attach logic did not create/update the Target Summary menu for the Attached Unit and the Parent Unit. The missing logic was added to build the Unit's Target range rings and generate the Target Summary menu when a detachment or attachment occurs.

While testing these corrections, a similar situation was uncovered when the Transfer Target order was submitted. Logic was missing to create/update the Unit's Target range rings and Target Summary menu for both the losing unit/HRU and gaining unit/HRU. The missing logic was added to the routine that changes the Target's ownership.

After making these corrections, it was discovered that HRU-owned Target range rings did not display on the Map even though the Target Summary menu included the Targets. Missing logic was added to the WHIP code to correct this problem.

3.56 JTLS-2026-17773 Patrol Mission Did Not Automatically Attack Sub

A Patrol air mission was directed to orbit near an enemy submarine that was submerged. Eventually, the mission detected the submerged submarine but failed to engage, despite being within weapon range and having correct ROE. The Player was forced to order the mission to attack the submarine using the Manage Air Mission Tasks order.

The logic that determines whether the active sonar onboard the mission aircraft detects the submarine, so that an attack can occur, used the Surface Search attribute of the Active Sonar which, of course, has a range of zero. The logic excluded the submarine from consideration for attacking.

The code error was corrected to use the Active Sonar range attribute, which has a positive value, when the mission was attempting to detect with an Active Sonar and attack the submarine.

Also, it is necessary that the submarine Target Type Group be selected in the Patrol mission order. If not, the mission will not fire after detecting the submarine.

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 6.4.5.0 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 6.4.5.0 DATABASE CHANGES

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

APPENDIX C. Version 6.4.5.0 REPOSITORY CHANGES

No significant changes have been made to the structure of the JTLS-GO 6.4.5.0 repository.