JTLS-GO

Version Description Document

February 2026

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.3.0)

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

JTLS-GO 6.4.3.0 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

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.4.3.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.4.3.0 is a Major delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.3.0 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 6.4.3.0 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 WHIP Training Manual* (JTLS-GO Document 10, Version 6.4.2.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.4.0.0)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.4.3.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 02, Version 6.4.3.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.4.3.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.3.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.4.3.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.4.3.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.4.3.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.4.3.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.3.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.4.3.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.4.3.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.3.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.3.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.4.3.0)
- *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)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.4.3.0 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 6.4.3.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.3.0 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 6.4.3.0 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 6.4.3.0 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 6.4.3.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 482 package. Both the ICP and DCP have the option for an organization to increase the maximum memory heap for the WHIP and DDSC. For large scenarios and databases, an organization should consider increasing the maximum heap size.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.4.

- JTLS-GO database tools require a certified PostgreSQL 15.16 database server and the full PostgreSQL installation. PostgreSQL 15.16 that has been compiled under Linux 9.6 is bundled with the JTLS-GO 6.4 release tar files. It is not necessary to use the delivered solution, but it is the easiest method to meet the requirements of JTLS-GO 6.4.3.0. There are several alternative methods available for obtaining the PostgreSQL 15.16 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.

- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of the supported versions of Red Hat Linux ES.

- The Perl script language is used by the JTLS-GO system and game setup scripts. The version of Perl included with the supported versions of Red Hat Linux ES is sufficient. The Perl program is typically located in the /usr/bin directory. If Perl is installed in a another location, a link should be created from the /usr/bin directory to this program.

- SIMSCRIPT III (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the

SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code.

- ANSI C Compiler: It is not necessary to use a C compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO component programs. The C Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.

- C++ Compiler: It is not necessary to use a C++ compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO HLA component programs. The C++ Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.

- The JTLS-GO DDS application uses these open source libraries:

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

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

  Commons-math3-3.0.jar, licensed under Apache Software Foundation (Apache License, Version 2.0) http://www.apache.org/licenses/LICENSE-2.0HLA 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 6.4.3.0 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 6.4.3.0 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 6.4.3.0 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.

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 6.4.3.0, but end-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

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

1.4.3 Standard Repository Changes

R&A has continued to improve and expand the unclassified data repository, which has been renamed to "repository64". The DDS comparison and synchronization function can be used to

determine if any of the changes delivered are of use to a JTLS-GO user organization. Specifically, significant effort has been applied to ensuring that all important Targetable Weapons have a unique Supply Category from the weapon should be drawn. This results in the model managing a detailed weapon count of all used weapons.

1.5 INSTALLATION

The *JTLS-GO Installation Manual,* a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery, It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 15.16 required to operate JTLS-GO 6.4.3.0.

2.0 ENGINEERING CHANGE PROPOSALS

No model capabilities were added to JTLS-GO 6.4.3.0 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-2026-17512 DDS/WHIP Obsolete Email Address

The DDS and WHIP still displayed the old JTLS-GO help desk email address (jtlsdev@rolands.com) in the stack traces and various parts of the application.

All of the help desk email addresses were updated to the new help desk email address (jtlsgo@valkyrie.com).

3.2 JTLS-2026-17513 OVT Crashes When Scenario Argument Provided

The Order Verification Tool (OVT) allows a number of optional arguments to be specified on the command line. One of these allows the specification of a scenario (–scenario <scenario>), which allows the user to bypass the scenario selection dialog. However, using the argument would crash the OVT.

scenario could also be obtained by specifying a configuration file withe the -cf argument. When this argument was used, the OVT failed to come up without producing an error message.

Both problems were identified and fixed in the source code.

Additionally, a shortcut was added to the script that starts the OVT such that if a single argument is specified it should be treated as a scenario. If the scenario is valid the argument is passed to the program as the proper "–scenario <scenario>" argument. This aligns the starting of the OVT with other JTLS-GO programs, such as the Web Services Manager (WSM).

While fixing this issue, a problem was found where the Apache tab was not providing information about the Apache server. This was traced to the OVT making HTTP requests (instead of HTTPS) for the required information. This issue was also fixed.

3.3 JTLS-2026-17515 Windows JDSP Library Network Activity Delay

The network library for Windows clients to the JODA behaves differently than the Linux network library, which uses the same parameters. The "select" function from the Windows system is used to determine when a network socket needs to be serviced. This function is delaying too long when it is called by the JDSP library, compared to Linux clients.

The Windows system library function "select" is used to determine when a write or read is available on the desired network socket. It uses an optional timeout time, which specifies how long it should delay for the detection.

The corresponding wrapper routine from the Windows JDSP library was modified to set the timeout time to zero, improving speed.

3.4 JTLS-2026-17517 Facility Target Partial Damage Inconsistency

When a JDPI Target is attacked and damaged, any linked Facility Target is also damaged to the same amount, which can vary anywhere from 0% to 100%.

However, when a Controller Change Target order is submitted to damage a Facility Target, the model allows only 0% (damaged) or 100% (undamaged) as the percent capable, with no variation between those values. It was impossible to damage a Facility Target to 50%, for example.

To remove this inconsistency, the logic was changed to permit Facility Targets to accept any percent capable value between 0% and 100%, as specified in the Controller Change Target order.

3.5 JTLS-2026-17518 Map Shape File Customization UI Improvements

Selecting a custom map shape file from the Interface Configuration Program (ICP) or Database Configuration Program (DCP) displays a dialog that allows the user to choose a new map shape file or use the default one. However, the user can only choose files from the $JGAME/data/maps/alternate directory, and the UI for selecting the new file is confusing.

The following changes have been implemented:

- Allow the user to choose files from both the $JGAME/data/maps/alternate and $JGAME/data_site/maps/alternate directories.
- Change the "Add", "Edit", and "Delete" buttons to "Alter" and "Default". "Alter" will prompt the user to select a file. "Default" will use the default map file.
- Change the title of the dialog frame to "Customize Map Shape File", to be consistent with the dropdown menu button from the ICP/DCP.
- In the file chooser menu, change the "Add" button text to "Select", and change the title of the file chooser menu to "Choose Customized Shape File".
- Add tooltip text popups when hovering over the "Alter" and "Default" buttons, explaining what each button's functionalities will accomplish.

3.6 JTLS-2026-17520 Add Joint Fusion Tool To JTLSmenu

The Joint Fusion Tool (JFT) can only be started by executing the "jft" script.

The Joint Fusion Tool was added to the JTLSmenu and Javamenu. It is now located under Option 1, "Prepare or Alter a Scenario Database".

In order to fit the new JFT option, the previous options to Combine Symbols and Edit Symbol Glyphs were combined into a new Symbols Menu under Option 1, "Prepare or Alter a Scenario Database", and the other options were renumbered.

3.7 JTLS-2026-17521 JFT Colors Tab Issues

The following problems were reported in the Colors tab of the JFT:

- When trying to assign a JTLS-GO color in the drop-down menu, clicking out of the menu before selecting a color causes the whole tab to crash and results in an empty drop-down menu.
- Duplicate colors would display in the JTLS-GO color column drop-down menu if the user tried to consecutively reassign colors, one after another, in the same drop-down menu.

The Colors tab will no longer crash if the user clicks out of the drop-down menu. The empty drop-down menu was also fixed to no longer display an empty menu.

There are no longer duplicate colors displaying in the JTLS-GO colors drop-down menu when clicking consecutive colors.

3.8 JTLS-2026-17526 Fire Mission Not Updated When Victim Altered

When a missile from a fire mission reaches its impact location, the assigned victim may not be there, or may not be detected. When this happens, the missile is assigned another victim in the vicinity or is changed to a coordinate impact area. The attributes of the fire mission are not altered to reflect this change. This can cause a crash when creating the damage reports for the mission.

If the Assess Weapon Damage routine needs to change the victim of the missile, the fire mission data structures are not updated to reflect the automated model change.

3.9 JTLS-2026-17529 Emission Control Order Transmission Jammers

The Emission Control order did not consider Transmission Jammers.

The model was fixed to properly turn on and turn off Transmission Jammers that affect Satellite communications.

3.10 JTLS-2026-17531 Aircraft Intercepting Missile Crash

The CEP crashed when an aircraft was instructed to intercept a cruise missile with a change of ROE specified on the intercept order. The cause was the part of the response message related to the ROE, which did not make a distinction between the type of intercepted track (air mission or missile).

The code was modified to check the type of object being intercepted and extract the needed data for the message as appropriate.

3.11 JTLS-2026-17533 Destroyed MUSE Mission Crash

A MUSE mission was passed back to the model by the ELS. When determining whether fuel was needed the model crashed. The mission should never have been passed back to the CEP, because it was destroyed, but the CEP needs to protect against such incidents.

The external mission pass-back procedure was improved so that the fuel consumption of the mission is always calculated, whether the mission will be accepted by the CEP or not.

3.12 JTLS-2026-17535 Reduce Ship Passive Sonar Detection Calculations

Ships that are in port should not be subject to passive sonar detections. Additionally, if a ship's location is already known, an additional passive detection is not necessary. Removing these passive detection situations will improve model performance.

The code was modified to implement these two processing-saving steps.

3.13 JTLS-2026-17537 HRUs Being Detected Too Quickly

A code error was found in the calculation for determining the time to detect an HRU. The number of foreign HRUs and ARUs within detection range of the covert HRU is central to the algorithm, with more objects covering the area decreasing the time until detection of the HRU. The error was that all assets covering the HRU, regardless of Force Side, contributed to any individual Side's detection time. This meant that even units and HRUs on the same Side as the covert HRU would help a foreign Side detect the covert HRU faster.

The code was modified so that only objects on the same Side as the Side attempting to detect the covert HRU are included in the detection algorithm.

3.14 JTLS-2026-17539 Unit With No Combat Systems Crash

A fully functioning detachment unit with all Combat System TOE values set to zero was introduced into the game when an airlift of a unit was aborted. As the unit was processed into the game, it went through logic to issue its combat systems from supplies. As part of this process an array is reserved for the number of combat systems in the unit. This was zero resulting in a crash for reserving a zero-length array.

A logic error was introduced to check if the unit has any Combat Systems. If it does not, the issue of Combat Systems is completed and exited without any Combat Systems being issued.

3.15 JTLS-2026-17541 BE History Should Hold SUP Instead Of Class

The model keeps a history of units, targets, air missions, and convoys detected within a Basic Encyclopedia (BE) Facility. For ship objects, the history was recording the Ship Class Name instead of the Ship Unit Prototype (SUP). The SUP is a more descriptive item and should be what is saved in the BE history.

The code was modified to save the SUP name instead of the Ship Class for detections of naval vessels.

3.16 JTLS-2026-17543 OTH Radar Missile Detection Terrain Masking

The algorithm to detect missiles was applying terrain masking and earth curvature to detections by Over the Horizon (OTH)-capable sensors. This restriction should not be applied to these sensors.

The missile detection code was modified to remove the restriction of terrain masking and earth curvature for sensors that are marked as being OTH-capable.

While debugging the situation, it was recognized that for non-OTH sensors, the terrain masking algorithm did not consider the antenna height of the sensor. It assumed that the sensor was at the elevation of the grid in which it was located. This problem was also solved.

Finally, for satellite sensors, JTLS-GO 6.4 has the correct computation for satellite altitude, but this was not being used. This error has little to no impact on the jamming of the satellite sensors, but for consistency the correct sensor altitudes were used.

3.17 JTLS-2026-17544 Changing DCA Mission To Protect ACM Crash

The model crashed when the user submitted a Change Mission Parameter order to use an ACM as its protection area.

JTLS-GO only allows DCA missions to protect a circular area, an OPAREA polygon, or a list of specific air missions. Those are the only three protection types modeled within JTLS-GO. A four protection alternative was available on the Change Mission Parameter order panel. This was an error.

The Protect ACM option was removed from the Change Mission Parameter order. There are no plans to add this capability to the model.

3.18 JTLS-2026-17546 Convoy Assets Not Removed From JODA

When a convoy is removed from the game, the JODA entities called Convoy Assets are not removed from the JODA

The JODA objects are now destroyed and removed.

3.19 JTLS-2026-17548 Update Call Sign From External Model

The Call Sign for a mission was not updated within the CEP when assigned by an external model.

The Call Sign provided by the external model is now represented in JTLS-GO

3.20 JTLS-2026-17550 No Federate ID Crash

The model crashed when linked to MUSE and an externally-owned unit did not have a federate identifier.

The circumstances which caused the crash could not be replicated, so a patch has been put in place to ensure the model does not crash if the situation happens again.

3.21 JTLS-2026-17552 Intercepting Aircraft Immediate Head For Fuel

An intercepting aircraft immediately stopped intercepting and headed for fuel

The issue was the check for whether the DCA missions was suitable to intercept the select foreign mission. The fuel check assumed that the interceptor would conduct the intercept at its normal cruise speed. Due to a recent change in the logic, this is no longer true. The mission now intercepts at its maximum speed, unless ordered to travel at a different speed. As soon as the mission started to intercept, its fuel consumption rate increased, and the mission believed that it would need fuel.

The interceptor feasible check now determines if the mission has enough fuel at its expected intercepting fuel consumption rate.

3.22 JTLS-2026-17555 Rescheduling Pre-Launch Event Crash

The model crashed attempting to find a delayed mission event.

The exact cause of the crash was not found, so a patch was incorporated into the model to ensure the model will not crash.

3.23 JTLS-2026-17556 ATO Parser Incorrectly Sums Aircraft

The new C2Core system provides multiple Mission Aircraft ("MSNACFT") records for some missions. The ATO Parser for this ATO format is tabulating all the aircraft in the provided MSNACFT records of the mission and uses the total for the count of aircraft in the JTLS-GO mission. It is also using the call sign from the first record as the call sign for the mission in JTLS-GO. This is not correct.

The Parser should be considering each MSNACFT record and using the count of aircraft and call sign from the MSNACFT record having the lowest call sign postfix number.

The Parser was modified to find the MSNACFT record having the lowest call sign number and only use the count of aircraft for the mission having that record. It will also use the call sign specified in that record as the call sign for the associated mission in JTLS-GO.

3.24 JTLS-2026-17558 ATOT Mission Available Time Scenario Start Time

When a mission is suppose to start its first tasking at a time that occurs before the start time of the scenario, the ATO Translator (ATOT) is not including any time value as the mission available time in the mission's order.

The ATOT uses a routine to determine the mission duration from all tasks included for the mission. This routine also computes the mission available time for the mission.

When the start time for the first task falls before the start of the scenario start, the ATOT did not compute the mission available time. This available time was then left empty in the order. Although this situation is unique to the very first ATO for an exercise, it can happen if a user, for example, shifts the ATO period of an older ATO using the ATO Parser.

The routine for computing the mission available time was modified to allow the early task time, even when it may fall before the scenario start time. The simulation will handle these early times by adjusting them to the scenario start time.

3.25 JTLS-2026-17560 ELS Event Timing Problems

The ELS experienced problems with the timing of events. In some cases, when the ELS was restarted from a checkpoint, the program would successfully download objects on the EODA but would never advance in time. On other occasions, the ELS would run for a period of time and then crash with a reason that it attempted to decrease the simulation time.

When the ELS was restarted, it would schedule the existing events, but it would never set the time for the next ELS event. This prevented the ELS from advancing in time because it never reached the portion of the code which incremented the game time. In the case of the crash associated with a decrease in time, the code was inadvertently incrementing the game time by an amount which exceeded the time of the next event. A code correction was made to prevent either of these problems from happening.

3.26 JTLS-2026-17562 ATO Parser Incorrect Takeoff Task Start Time

The C2Core service provides "AIRMOVE" records for each of the stops for Mobility missions. However, the first record listed is typically missing a start time in the record (the stop time is always provided). The C2Core service is representing the home takeoff location in the first AIRMOVE record, so a start time for this location would not be necessary. In fact, a stop location for the home takeoff location is also not necessary.

The ATO Parser discovers the stop time associated with this first AIRMOVE record, but is unable to extract the associated start time. The Parser stores an unassigned time for this tasking and the ATOT later converts this time to 01 Jan 1970 (start of epoch).

The ATO Parser was modified to exclude the stop location held by the AIRMOVE record as the home takeoff location. This stop is not necessary.

3.27 JTLS-2026-17564 ELS Crash Computing HRU Radius

The ELS crashed while attempting to compute the radius of an HRU.

The code was attempting to compute the radius, using the type of entity when it was expecting an object type. The code was modified to use the correct object type in the called routine.

3.28 JTLS-2026-17566 ELS Crash Deleting Active Engagement

The ELS crashed while attempting to delete an active engagement. These structures are used to hold combat engagements which are bundled by the CEP according to their engagement identifier.

The ELS stored active engagements in hash buckets to make their access more efficient. When an active engagement was deleted, the object was not properly removed from the hash bucket. This caused some corruption in the memory and led to an ELS crash. The code was modified to remove the engagement from its associated hash bucket.

3.29 JTLS-2026-17568 JSXR Responses Empty For Georegions Crossing IDL

The JXSR uses special filtering to discover simulation objects contained within a polygon area. This is used primarily for IMT requests made by the WHIP. When an object falls within the polygon, the unit is listed in the reply back to the requesting WHIP. However, when any portion of that polygon crosses the International Date Line (IDL), the reply is empty, even though objects exist within the polygon.

The JXSR stores the latitude/longitude vertex values for one or more polygons in a georegion. When any object is checked to see if it is within the georegion, the JXSR employs a formula to determine whether the coordinates of the object falls within any of the polygons of the filter. The formula fails when vertices of a polygon are on opposite sides of the IDL.

The georegion filter was changed to only store positive longitudes, in effect transforming the range of longitudes for vertices from (-180,+180) to (0.0,+360). Then, temporary coordinates for each object in the comparison are transformed in this same way and applied to the formula. This coordinate transformation allows the formula to properly function.

3.30 JTLS-2026-17571 Is Air Defense Site Link 16 Capable Crash

The model crashed when a dual-capable Fire Control Sensor and Acquisition Radar detected a missile. The model was attempted to determine if the detecting target was Link 16-capable.

The code incorrectly assumed that the detecting object was a Sensor Site - in fact, it was a SAM/AAA Target with a dual-capable Fire Control Sensor and Acquisition Radar. The crash occurred because, in the specific database that was being used for testing, there were more types of SAM/AAA Targets than Sensor Sites. The code was wrong, but because a SAM/AAA Target had a sub-type index greater than the number of Sensor Sites, the error became a crash.

The code was corrected to specifically ask the type of Target that accomplished the detection.

Database builders should note that if they want the detections made by the dual-capable Fire Control Sensors and Acquisition Radars to be Link 16 detections, that appropriate track blocks need to be assigned to the unit that owns or is associated with the SAM/AAA Site.

3.31 JTLS-2026-17572 ATO Parser Missing ACM Use Field

When two or more ACO messages are combined by the ATO Parser, the Parser leaves out the "Use" field for each of the ACMs contained in the ACOs that come after the first.

The ATO Parser reads the ACO files in order and combines the ACMs in a set of ACMs obtained from the first ACO. The routine that adds each ACM was not copying the "Use" field of subsequent ACOs whenever this was done.

The ATO Parser was modified to include the "Use" field of each ACM for each ACO following the first.

3.32 JTLS-2026-17574 DDS Repository Tool Command Hierarchy

When bringing up the ORBAT Migration component of the Data Repository Tool, the destination scenario's Command Hierarchy was no longer being built, leaving the Force Side nodes empty. Reloading the destination ORBAT data did not help.

This error was introduced when Container Units were introduced in JTLS 6.4. The collection of Container Units was not registered as a listener for the data. As a result, the destination scenario's Command Hierarchy was waiting for the Container Unit data to finish downloading, which would never happen. The errors were fixed in the code.

3.33 JTLS-2026-17575 Fire Control Range Rings Not Properly Handled

In JTLS-GO 6.4, Air Defense Sites can have a Fire Control Sensor that also has the ability to be used as an Air Search Acquisition Radar. Displaying the Air Search capability for a unit did not fully handle this situation when creating range rings, causing a non-fatal model crash. The system could be simply continued without any ill effects.

The problem was solved, and the non-fatal error is now properly handled.

3.34 JTLS-2026-17576 SVP Warning 1133 Incorrect Wording

There are six different versions of Warning 1133, all of which are similar to one another. Due to a copy and paste error, all six warnings indicated the problem was with a Surface Search sensor, when many sensors did not have any surface search capability.

The code's copy and paste error was corrected. Warning 1133 now reports the correct sensor use type that has a problem.

3.35 JTLS-2026-17578 Prevent MUSE-Owned Air Mission Ownership Transfer

When the link to the MUSE simulation was disconnected, the ELS attempted to regain ownership of the air missions being flown by MUSE. Ownership of the missions should not change, because they are reacquired when MUSE reconnects.

The ELS was incorrectly attempting to gain ownership of the MUSE missions. Code was changed to prevent this from happening in the future.

3.36 JTLS-2026-17580 ELS Attempting To Move Targets To Offset Locations

The ELS was attempting to move an owned Target to a location which was offset from the owning Unit's location. This was occurring when the Unit was not moving.

The ELS was incorrectly trying to reposition an owned Target when the owning Unit was stationary. Code changes were made to stop this from happening.

3.37 JTLS-2026-17582 Issues ELS Publishing Updates To EODA

The ELS was pushing object updates to the EODA during times when the object updates were not yet complete. This caused some objects to be updated before the data should be changed.

The ELS code was modified to update each type of object in a separate routine. This allowed the code to process each type separately.

3.38 JTLS-2026-17584 ELS Not Launching External Missions

A Controller changed the aircraft type of a Ground Unit, but the ELS did not reflect this change. A ground unit was assigned some aircraft but the ELS was not launching missions from that unit.

The ELS was only updating the aircraft type for squadrons and not for all other unit types. The code was modified to update aircraft type for all unit types. The ELS was only launching missions from squadrons. The same was also true for external missions being created in the CEP. This has been corrected.

3.39 JTLS-2026-17587 Unnecessary Execution Of ELS Events At Startup

The ELS restarted very slowly after the CEP had been running for an extended period of time.

When the ELS was restarted, it was scheduling and executing events which had occurred in the past without regard to the current game time. Modifications were made to prevent the scheduling and execution of past events. This allowed the ELS to restart more efficiently.

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 6.4.3.0 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 6.4.3.0 DATABASE CHANGES

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

APPENDIX C. Version 6.4.3.0 REPOSITORY CHANGES

No significant changes have been made to the structure of the JTLS-GO 6.4.3.0 repository.