JTLS-GO

Version Description Document

December 2025

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.2.0)

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

JTLS-GO 6.4.2.0 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 6.4.2.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.4.2.0 is a Major delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.2.0 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 6.4.2.0 executes on the Red Hat Enterprise Linux Version 9.4 and Oracle Linux 9.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any 64-bit operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

1.2.2 Unchanged Documents

- *JTLS-GO Air Services User Guide (*JTLS-GO Document 02, Version 6.4.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.4.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.4.1.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.4.1.0)
- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 6.4.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.4.1.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.4.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.4.0.0)

- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.4.0.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.4.0.0)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.4.2.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.2.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.4.2.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.4.2.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.4.2.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.4.2.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.2.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.2.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.2.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.4.2.0 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 233.

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

  Apache Server (APACHE)

  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 6.4.2.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.2.0 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 6.4.2.0 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 6.4.2.0 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 6.4.2.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 472 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 15.13 database server and the full PostgreSQL installation. PostgreSQL 15.13 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 6.4.2.0. There are several alternative methods available for obtaining the PostgreSQL 15.13 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 6.4.2.0 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 6.4.2.0 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 6.4.2.0 and the previous JTLS-GO 6.3 series database structure. APPENDIX B. Version 6.4.2.0 DATABASE CHANGES 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 6.4.2.0, 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 15.13 required to operate JTLS-GO 6.4.2.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.4.2.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2025-17419 Specify External Missions Call Sign ELS Order

Summary of Model Change Request

The Entity Level Server (ELS) required the capability to specify a call sign for external air missions. The call sign or aircraft tail number will be specified in an ELS order sent by the external model.

Design Summary

The CEP already supported the setting of call sign for external missions. Changes were made to propagate these changes through orders sent from the ELS. A keyword was added to the ELS order used to manage external missions. The ELS received the order and compiled the information needed in the subsequent order sent to the CEP. New values for the call sign were properly reflected in the aggregate air missions.

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-2025-17394 Refuel Mission Quick Order Refillable External Tanks

The CEP was enhanced to allow external "tip tanks" to be refueled in the air if the aircraft type was capable of doing so. The existing aerial refueling logic was updated appropriately, but the Refuel Mission quick order was overlooked. As a result, the calculated refuel amount excluded the refillable tank capacity.

The Refuel Mission logic was corrected to include the capacities of refillable external tanks (if present) plus the internal (intrinsic) fuel capacity of the aircraft type when calculating the amount of fuel to obtain from the tanker.

3.2 JTLS-2025-17396 Tanker Move To Next Orbit Deleted Refuel Chits

All refuel chits belonging to an orbiting air refueling mission were deleted when the mission moved to its next planned orbit. In this situation, the refuel chits should have been retained.

Code was added to retain refuel chits when the tanker leaves orbit to fly to the next orbit location. If the mission is leaving its final (or only) orbit, the refuel chits are deleted. However, if the tanker is leaving orbit because it is running low on fuel (and therefore Heading Home), the refuel chits are not deleted, because the Controller may magically add fuel and return the mission to orbit.

3.3 JTLS-2025-17397 OVT Crash AAR Reports With Graphic Utility Field

A utility field was added to the AAR Supply Usage Report that provided the points for a geographic region. This utility was also listed in the graphic section of the order, so that if the utility was filled out, the geographic region would appear on the Map while the order was up.

However, this situation caused a crash in the Order Verification Program (OVP) when it attempted to open and verify the utility file in the $JGAME/data/orders/ directory instead of the $JGAME/data/aar/orders/ directory.

The code was modified so that the location the OVP attempts to open the utility file from is based on the type of the order using it - a regular order, an AAR report, or a DDS report.

3.4 JTLS-2025-17398 AAR Engagement Data Collection Crash

The recent change in which an air mission can intercept and kill a cruise missile was not properly tested with the AAR on. Code was centralized to get the location of the defending object, but this newly centralized code was called without properly setting the of object for which its location was being requested.

The code was corrected to properly set the type of object for which its location was desired.

3.5 JTLS-2025-17400 Supply Usage Report Filtering By Country

The AAR Supply Usage Report had the capability to filter by Force Side and Faction. However, if all of the relevant units within a database belong to one Side and one Faction, these filters will not be useful for filtering by country.

The database has proper Political Countries assigned to the various units, allowing filtering by Political Country to fulfill the required goals. The CEP was modified to report initial assignment and changes to the Political Country of aggregate units. The AAR Supply Usage Report was modified to allow filtering based on this unit attribute.

Air missions, convoys, HRUS, and targets that receive, dispatch, or consume supplies look at the Political Country of their owning/associated aggregate unit to determine if the data should be included in the report.

3.6 JTLS-2025-17403 Order Constraint Mixed Case Vocabulary Entries

A constraint was added to the Damage Combat System order to only allow the "Always Evacuated" flag for damaged Combat Systems to be applied to Combat Systems that were Personnel.

Using the vocabulary entry "Always_Evacuated" would pass the OVP, but would not perform the restriction in the Order Management Authority (OMA). Instead, the value of the flag (4) had to be used in order to get the OMA to properly work. However, using an integer in the order caused the OVP to issue an error about the format of the order.

Two errors were uncovered while investigating this issue:

- The first was that vocabulary entries were being converted to upper case before searching for a match.
- The second was that if a match was not found, a value of zero was being returned, which is a valid value for some vocabulary entries and therefore was not generating an error in the OMA on startup.

Converting the vocabulary items to uppercase was removed from both the OVT Library and the AARC, which had similar code, and the return code for not finding a matching vocabulary entry was switched to -1, which is not used as a value for any existing vocabularies. Once the return code was changed it was necessary to check all programs that used the vocabulary call and update the associated error handling code to expect -1 on failure.

3.7 JTLS-2025-17405 Additional Corrections To JVLC Fusion Tool

A number of corrections/modifications to the JLVC Fusion Tool (JFT) were identified:

1. The security classification needs to be displayed on the JFT window.

2. When launching the JFT from the command line, there are command line options that allow the user to immediately go to the import window. There are no equivalent command line options to get to the export window.

3. There are several processes that can take the JFT a long time to execute. Because the user cannot interact with the GUI during these periods, it would help the user if the JFT GUI provided visual cues to indicate when these processes are executing and when they have ended.

Modifications to the JFT are summarized below.

1. If the user is using the JFT to convert a JLVC OBS file to JTLS-GO scenario files, the JFT now reads the security classification provided in the OBS file metadata and displays the classification on the JFT window border.

   If the JFT is being used to convert JTLS-GO scenario files to an OBS file, the classification is obtained from the scenario's .glo global data file.

2. The JFT introduces two command line arguments, -i and -x. These indicate import (OBS to JTLS-GO translation) and export (JTLS-GO to OBS translation), respectively. The arguments are used as follows:

   jft -i -s <JTLS-GO scenario name> -f <JLVC OBS file name> -d <JTlS-GO DIS Code Report file name>

   jft -x -s <JTLS-GO scenario name>

3. The standard arrow cursor is replaced with an hourglass icon when a unit hierarchy is being created, the JFT is converting from OBS to JTLS-GO (importing), or when the JFT is converting from JTLS-GO to OBS (exporting). The arrow cursor is restored when these processes end.

3.8 JTLS-2025-17407 End Google/KOI Support In JTLS-GO 6.4

Google Earth and Keyhole Markup Language (KML) Operational Interface (KOI) service are no longer supported in 6.4. The new JTLS-GO 6.4 map server capability fulfills the same functions as the former KOI.

The Google and KOI tabs have been removed from the Interface Configuration Program (ICP) and Web Services Manager (WSM). as well as the corresponding Java source code and resources for them.

3.9 JTLS-2025-17412 SVP Indicates Max Object In Grids Skipped

Within the SVP, the user can indicate which Warnings should be skipped. As the SIP accomplishes the SVP checks, it skips those warnings it was told to skip. This skipping process is accomplished silently. There is no indication that the warning is being skipped during execution. There is one warning that the SIP notifies the user during execution that the warning is being skipped. This is the check that warns the database builder that there are too many objects in a single grid.

This inconsistency caused confusion leaving the user wondering if there is something special happening with that one warning. For that reason, the print statement telling the user that the Warning was being skipped was removed from the code base. The maximum objects per grid warning is now executed or skipped based on the input from the user. As with all skipped warnings, the final SVP report indicates which SVP Warnings were skipped.

3.10 JTLS-2025-17414 WHIP/DDS Unit Colors Inconsistent

The map.colors file mapped WHITE as "gray50" and CONT as "white". This leads to inconsistencies on the WHIP map when creating white range rings or ACMs, as it displays them as being gray.

Also, in the Map object filter panel, the object table column header for units with unknown sides is displayed as black. However, units with unknown sides on the map display as gray (or NEUTRAL in the map.colors file).

The map.colors file was changed to map WHITE as "white" and CONT as "gray50". This ensures the map will display true white when creating and displaying objects and selecting their color as white.

Also, changing the Controller side to "gray50" ensures that CONT and WHITE will not produce duplicate colors.

Also, the Map filter header column for units with unknown sides has now been changed to gray (specifically NEUTRAL in the map.colors file) instead of black. This ensures the header color for objects with unknown sides matches the unit color on the map who have unknown sides (currently set as gray or NEUTRAL).

3.11 JTLS-2025-17415 New SVP Warning - Self Propelled Combat System

A number of Combat Systems drawn from Supply Categories with a Shipment Type of "S_P", for "self-propelled", had CS.Fuel.Per.KM values of 0.

This is not an Error, but a Warning. It is expected for self-propelled Combat Systems to have a value greater than 0 for CS.Fuel.Per.KM to obtain proper fuel consumption data.

New SVP Warning 1165 is generated by the SVP to inform the database user of this situation.

3.12 JTLS-2025-17418 Skip Creation Of Unused TUPs

The JFT was used to import an OBS XML file to create a JTLS-GO scenario. During this process, the tool created Tactical Unit Prototypes (TUPs) for every unit in the XML file. Many of these units were rolled up to create larger units, which were more appropriate for a operational-level model. As a result, many of the generated TUPs were not used by the resulting JTLS-GO scenario.

The JFT created TUPs for every unit to facilitate the counting of equipment in every unit. For units which were rolled up into larger units, the TUP data were not needed for the subordinate units.

Code changes were made to omit the data for these TUPs in the JTLS-GO scenario. The resulting scenario only included those TUPs which were used by units in the scenario.

3.13 JTLS-2025-17421 ICP Configures Exercise Log Late

When saving the ICP after activating the Exercise Log, the Exercise Log configures after the Glassfish domain is already shut down, which is too late in the save process.

The Exercise Log configuration now takes place earlier when saving the ICP, after the application as been deployed and before the ping connection to the Exercise Log database has been executed.

3.14 JTLS-2025-17422 Satellite Damage Logic Errors With No Report

The Assess Weapon Damage order was submitted to damage a satellite. Logic errors were produced by the model by this order. No Player damage report was generated.

Logic errors occurred due to missing code for satellites and were corrected. There was also no code to generate a Player damage report when a satellite was damaged; new code was added to generate the report.

3.15 JTLS-2025-17424 SVP Error 342 Auto-Corrections

SVP Error 342, which checks that Targetable Weapons representing hypersonic missiles have appropriate ranges during their boost phases, did not have any auto-corrections to assist the database user.

Additionally, the model team has a rule to never access an attribute of an entity more than once in a routine. Simscript checks for every access, and this checking takes time. There were several attributes of the Targetable Weapon entity that were redundantly being accessed in the routine CHECK.TARGETABLE.WEAPONS.

To give the database user more clarity, two new auto-corrections, to either increase the minimum range or decrease the speed of the missile, have been added, as well as a third correction informing the user to decrease the number of boost phases for the Targetable Weapon.

The routine CHECK.TARGETABLE.WEAPONS has also been updated to reduce the number of redundant checks when the attribute of an entity is accessed more than several times throughout the routine.

3.16 JTLS-2025-17426 Duplicate Execute Mission Task Events

If the user delays a mission's primary task while the mission is in a Pre-Launch posture, the model automatically returns the mission's assets to the home squadron, places the mission back into a Scheduled posture and schedules a new resourcing event to occur at the appropriate time. This is done to ensure that the mission does not unnecessarily hold onto assets during the delay period.

However, the event used to tell the mission to launch is not canceled under this circumstance. A mission that has two Mission Task Execute events can cause major model issues.

The code was corrected to properly delete the event that had been scheduled to have the mission begin.

3.17 JTLS-2025-17428 Stop Duplicate Targets On New Unit Create

It is possible that when the Controller creates a new unit during game play that the new unit's Prototype Owed Targets (POT Targets) will create a target name that already exists in the game. The end result is that there are two targets with the same name in the game, which will cause numerous problems.

When creating a new unit that has POT Targets, the model now check if the POT target will end up with a unique name. If not, the POT target is not created, and a Logic Error is generated.

3.18 JTLS-2025-17430 When Creating New Unit Dead Unit UIC Feasible

When creating a new unit during game play, the model ensures that the UIC assigned to the new unit is not already being used. The model only checks if the UIC is being used by a unit in the Conflict Set, which means the unit is alive or still waiting to arrive in theater. It does not check if the UIC was previously used by a unit that is currently dead. Because the Controller can resurrect a dead unit, the Create Unit UIC check should also ensure that the UIC was not used by a dead unit.

The UIC of a unit must be unique. The primary reason for this rule is that in the real world, UICs are unique identifiers for units. Within the JTLS-GO modeling world it needs to be unique because the UIC is used to automatically name POT Targets. A POT Target name is created by taking the owning unit's UIC and concatenating it onto the POT Target’s Base Name, which is a database parameter. If two units used the same UIC, it is possible that their POT targets may end up with the same name.

When creating a new unit, the model now checks the UICs for not only active units, but dead units as well. This is necessary because when a unit dies, its stationary targets do not die, but

are simply left behind. The new check is needed to ensure the model does not end up with two targets that have the same name.

The code to check UIC was centralized but is called under numerous circumstances. Under one circumstance, when creating a Strategic Lift Manifest, the desire was to find an active unit by its UIC. In this one case, the dead unit list was not supposed to be checked. For this reason, the centralize routine was changed to pass in an argument indicating with the dead unit list should or should not be checked.

3.19 JTLS-2025-17432 UAVs Should Be Detected By Own Side

A UAV is being controlled from the ground, so there should be no reason to not have the mission automatically detected by its own Side.

There are now four reasons why an air mission will be self-detected by its own Side:

1. The mission's aircraft are Link 16 self-reporters.
2. The mission is normally in communication with the ground or is at its assigned orbit location, so the ground knows where it is.
3. The mission's aircraft are equipped with a GPS tracking capability, such as Blue Force Tracker.
4. The air mission is an Unmanned Aerial Vehicle.

3.20 JTLS-2025-17434 ICC JTOI NATO Digraph Country Codes

The ICC JTOI by default used the country codes that the scenario database was built with. ICC expects country codes in NATO Digraph, so if the scenario database is using a different format, ICC JTOI updates will fail.

The ICC JTOI was modified to always provide country codes to ICC in the NATO Digraph format.

3.21 JTLS-2025-17436 WHIP's Order Group Editor Bug Fixes

Sharing groups in the order group editor panel with other users sometimes results in the shared groups being invisible to other users even if they have read permission.

An order group viewer frame from the order group editor only opens the first time. After closing it, trying to reopen it again does not work, and the frame will not show up.

Users that have groups shared with them in the order group editor with the correct read permissions can how see the shared groups in the shared group folder panel.

Users are now able to close and reopen the order group viewer frame multiple times.

3.22 JTLS-2025-17437 MIDB Tool Target Locations Supporting MGRS Format

Target locations in the MIDB migration tool do not support MGRS formats, but JDPIs do support MGRS.

The Java help documentation in the MIDB tool should reflect the supported formats but it currently does not.

Target locations now support MGRS format.

The Java help documentation now reflects the supported MGRS location formats for Targets and JDPIs.

3.23 JTLS-2025-17439 JFT Command Level Matching Too Slow

In the command level matching tab of the JFT, unknown JTLS-GO command levels are set to Unknown values at the bottom of the drop-down list. This makes command level matching slow, because you have to scroll all the way up to find the matching command level.

For unknown command levels, the JFT will now automatically set the value of the JTLS-GO command level drop-down menu to the closest JLVC echelon leveling. This makes the searching and matching process must faster.

3.24 JTLS-2025-17440 Crash Canceling Delayed Mission

The CEP crashed while canceling an air mission which had a posture of Supply Delay. The mission was unable to launch because it was deficient in some of the required supplies.

The model crashed because it was attempting to access the list of supplies held in the aircraft load. The load had not been defined in the code, so the model crashed while dereferencing a null variable. Changes were made to assign the aircraft load to the local variable before it was used to evaluate supply levels.

3.25 JTLS-2025-17443 Crash Repairing Unowned Equipment Shelter

The CEP crashed while attempting to repair an unowned equipment shelter.

The crash occurred because the code was attempting to access an attribute of the owning unit. Because the equipment shelter was not owned by a unit, the model attempted to dereference a null value. Code was added to skip the repair process for unowned targets.

3.26 JTLS-2025-17444 CADRG Map Layer Displays On Whip Startup

When starting a WHIP, the Map's CADRG layer may display on startup even though it is disabled.

A race condition was present where the CADRG map layer visibility was being set too early, depending on the system's performance. This issue has been resolved.

3.27 JTLS-2025-17446 CEP Crash Pushing Large Orders With AARC

When the CEP has packets to send to the AARC, and the AARC is not yet connected, the CEP stores the packet info as strings in an internal queue. This is common on a game start where Units and Targets are created before the CEP connects to the JODA, and therefore does not have a connection to the AARC. This also happens on a CEP restart with a push, where the pushed orders are read and processed before the JODA connection is established.

In such a situation, when the AARC does connect, the CEP will cycle through the queue, reconstructing the original packets from the strings, and send the packets to the AARC. During one exercise, some of the orders were more than 4000 characters long, which is the buffer size for reading the packet tokens when converting the string back into a packet. As the CEP code was reconstructing the order from the string, the buffer overflowed, causing heap corruption and a crash.

The code to reconstruct the packet from a string was modified so that the buffer will not be exceeded, limiting a token to the size of the buffer (4000 characters).

This limit will cause the truncation of the order string being sent to the AARC, but this would happen anyway when the AARC receives the packet, because the database column size for the order is only 4000 characters.

3.28 JTLS-2025-17449 Inconsistent LOGSTIREP Column Headers

The Combat System IMT, Unit/HRU Logistics Reports, and roll-up Unit Logistics Report had different column headers for the number of Combat Systems available for operations.

The column headers in the individual Unit/HRU Logistics Report and roll-up Unit Logistics Reports were adjusted to match the usage of "Oper" for the number of available Combat Systems in the other four.

The Combat System IMT is a separate media from the printed reports, and therefore the use of "Avail" is acceptable and has been for years.

Some minor clean-up was also accomplished.

3.29 JTLS-2025-17450 Web Login Page Still Shows KOI Column

The web login page still displayed the KOI column. This caused the row entries after the KOI column to be misaligned with their corresponding headers.

The KOI column has been removed from the web login page.

3.30 JTLS-2025-17452 Controller Change Naval Unit Strength

A Controller Change Unit order was submitted to reduce a naval unit's strength to 60%. Then, the order was submitted again to reduce the ship to 50%. The strength dropped to 0% and the ship started sinking. The order was submitted a third time to bring the ship back up to 100%, but the strength remained at 0%, despite the Player message indicating the ship had been increased to 100%.

The problem was traced to a relatively new routine PROCESS.CONTROLLER.UNIT.STRENGTH. The local variable NEW.FRACTION was not initialized before the new strength calculations were made for a naval unit. The zero value resulted in the illogical behavior observed.

The error was corrected by moving the initialization of NEW.FRACTION from later in the routine to just before the subroutine call that calculates the new naval unit strength.

3.31 JTLS-2025-17453 RIC Codes In IMT Screens

Reportable Item Codes (RIC) were missing from Combat System, Target supply, and Unit supply IMT tables in the WHIP.

The RIC codes were added as a new column to the corresponding IMT screens.

3.32 JTLS-2025-17455 Add Evacuations To AAR Maintenance Info

The AAR Maintenance Report does not display evacuations of WIA personnel.

The EVACUATED Maintenance Action category was added to the report, and the CEP now sends these actions to the AAR system. An Evacuated column was added to the Maintenance Report and Object Event Report.

3.33 JTLS-2025-17457 Order/Utility Air Routes Difficult To Distinguish

Utility air routes and order air routes are difficult to distinguish on a WHIP Map when both routes are displayed together through an air order panel. This is because they are both painted on the Map as gold.

Change utility air routes to magenta. This makes order and utility air routes easier to distinguish when they are displayed together.

3.34 JTLS-2025-17459 WHIP/DDS/TRIPP Login Rephrasing

The login frames for the WHIP, DDS, and TRIPP all say "Log into the ..." and "Logging into server", which is incorrect grammar.

The login frames for the mentioned applications have been updated to "Log in to the ..." and "Logging in to server".

3.35 JTLS-2025-17461 Improve AAR Report Parameter Table

Every AAR Report starts with a table to list the parameters and values on the original order that generated the report. These seldom appeared in the same order as they were displayed on the WHIP, and the parameters would use the field keyword instead of the label. This made it difficult to compare the table to an order to quickly determine what was set.

The displaying of the parameter table is done by a single stylesheet that is imported by all the reports. A new version of this stylesheet was developed that checks the actual report order files to determine the proper sequence to list the parameters and to access the labels for groups and fields to display them instead of the keyword.

The new stylesheet also adheres to the WHIP user's preferences for displaying field values, namely locations. This requires the importation of the format-util.xsl stylesheet be added to all reports that did not already import it.

3.36 JTLS-2025-17464 Document Mission Name Length Check

The developer wanted to disable the OMA check for the mission name length, which is used on orders creating a new air mission. The restriction limits the name length to 8 characters, allowing the CEP to add on a hyphen and unique numeric identifier to the mission name. This prevents duplicate mission names from being sent by the WHIP. If a new mission name with a length greater than 8 characters reaches the WHIP, the WHIP does not add on the unique identifier.

The developer wanted to submit mission names via a Read Order File (ROF), with more than 8 characters so that follow on orders could be prepared in a ROF without having to determine any CEP adjustments to the mission name. The Web Services Manager (WSM) and Order Verification Tool (OVT) show checks with a toggle box next to them indicating whether the checks are turned on or off, but this is for display purposes only. The developer could find no documentation instructing them how to turn off the mission name length check.

Checks can be enabled/disabled via an OMA console session as outlined in both the WSM on-line help and the *JTLS-GO Technical Coordinator's Guide*. However, the inoperable toggle buttons that are displayed on both the WSM and the OVT are confusing. The table with toggle buttons and check names are delivered in HTML format from the OMA to both the WSM and OVT. The function providing the information was modified to specify that the checks could be enabled/disabled via an OMA console.

3.37 JTLS-2025-17465 Airlift Unit Mission Crash With Transit Route

The model crashes when the user enters a Mobility Mission order in which the mission is given a route to follow between the pickup and dropoff locations.

An incorrect routine was called when processing the order's specified transit route. The code was rearranged and the correct routine is now being used.

While making this correction, duplicate code found in two related routes was removed and placed in a callable routine.

3.38 JTLS-2025-17467 Off-Road Land Units Slow Down

When an aggregate land unit moves off-road, the code is incorrectly applying the terrain movement multiplier to the unit's speed with each move. This drastically slows the unit down.

The error was a result of the new Container Unit code. The JTLS-GO 6.3 unit speed algorithm was put back into the JTLS-GO 6.4 code base. The restrictions for Container Units were implemented in a different manner that does interfere with the unit attainable speed algorithm.

3.39 JTLS-2025-17468 Transport Of Air Missions In ELS

With the introduction of swarming air missions, the CEP was changed to allow air missions to be transported by other air missions. The ELS needed to be changed to support this change.

When an air mission was transported by another mission, the mission being carried needed to be marked as mounted. When mounted, the transported mission was not displayed as an independent mission by an external model. This was the same rule as for other entities which were on board other objects, such as crew on a ship or crew on a vehicle.

The ELS code was modified to mount air missions which were transported by other missions.

3.40 JTLS-2025-17469 ELS Air Mission Types And Postures Changes

The accepted types of air missions and their postures were changed in the CEP. To agree with the aggregate model, the equivalent changes were required to be made in the ELS.

The number of air mission postures increased to a total of 34, and the number of mission types was reduced to 13 from 18 original types. The associated ELS code was modified to incorporate these changes.

3.41 JTLS-2025-17471 Ground Route Does Not Always Attempt Direct Route

When a ground object attempts to find a route from one location to another, there are three different algorithms available:

- Optimized movement - This algorithm is available to Units and HRUs, and using the algorithm is specified as part of the movement order from the Player. The model attempts to find a road network for the specified move.
- Direct movement - This algorithm is always used for convoys and Units in an attack, delay or withdraw posture. In addition, the Player can indicate that the direct movement algorithm should be used as part of the movement order.

- Gridded optimization - This algorithm is a computationally heavy algorithm and it is employed when the optimized or direct route algorithms fail to find a route.

When Optimized movement is selected and fails, the model should have attempted to find a Direct movement option before moving on to the gridded optimization algorithm. This will improve model efficiency.

The model now works as specified above.

3.42 JTLS-2025-17474 Feasible Network Restriction Too Restrictive

When attempting to find a road network to more from one location to another, a rectangular region is formed to limit where the model should look for an appropriate route. The Road Network algorithm ensures that the selected roads do not go outside the database-established rectangular area.

The code also made some distance checks within the algorithm. These distance checks were unnecessary because the rectangle was established for this very purpose. Some convoys refused to use a fully acceptable road route because of these unnecessary distance checks.

The additional distances checks were removed. This makes the road selection algorithm run faster and ensures that the database-established rectangular region is responsible for ensuring only reasonable routes are selected.

3.43 JTLS-2025-17475 Container Unit Establish Communication Level Low

A Container Unit has data indicating what Communication Quality Rating (CQR) should be used by the Unit. The model assumed that the Unit's communication capability was at the lowest level for the assigned CQR. It should be at the highest level for the assigned CQR.

The unit is now set at the highest capability level for the assigned CQR.

3.44 JTLS-2025-17476 Magic Move Container Units Created During Game

One of the initial design decisions for Container Units was that Container Units would always be in the game (all database Container Units would arrive at game start). Obviously, when the Controller creates a Container Unit during game execution, this is not true. A Controller-created Container Unit is automatically scheduled to arrive at Day 99, and it is up to the Controller to decide when the Unit should arrive in the game.

Several errors were found for Container Units created during game execution. Among these errors were the inability to Magic Move the Unit. Although it appeared to be in the game, it was not completely active.

To fix this problem, the first step was to determine whether the model should restrict the Container Unit from having a TPFDD arrival at a future time. The decision was to make Container

Units created during game execution operate in the same manner as all other Unit types created by the Controller during game execution. There were three primary reasons that this decision was made:

1. Database Container Units go through the Unit Arrive event like all other units. The only difference is that Container Units have no database-specified TPFDD Time. Because database Container Units go through the Unit Arrival process, Container Units created during game execution should as well.

   During testing, it was noticed that TPFDD event data was not being sent to the JODA for the game-created Units. This problem was fixed under this STR.

2. One of the options for the Manage TPFDD orders is to TPFDD in a Command Hierarchy of units. It seems reasonable that a Controller decides to create a Container Unit and place several real units subordinate to the Container Unit and, when needed, TPFDD in the Container Unit and all subordinates. This could not be done, if all Container Units created during game execution arrived as soon as they are created.

3. All game-created Units are automatically placed subordinate to the top Unit on their Force Side. Filling out and submitting the Manage TPFDD order is the method used to place the new created Units in their proper Command Hierarchy position. If Container Units arrive as soon as they were created during game execution, they would always arrive subordinate to the top Unit. A Player would need to submit a Change HQ order to get them into their correct hierarchy position. This cannot be done by the Controller, meaning there would need to be coordination between the Controller and Player when creating Container Units during game execution. The Controller should be able to accomplish this task without coordination.

Once the decision was made to have Container Units created during game execution go through the Unit Arrive event, a few changes needed to be made specifically for Container Units, such as ensuring their posture showed OUT_OF_GAME prior to arrival, and their active flag attribute was managed appropriately.

Container Units now process in a consistent manner with all other Units created during game play.

3.45 JTLS-2025-17477 SDC Leaks Memory During Processing

The SDC was observed gaining memory while running for several hours. The SDC was restarted in order to reinitialize and possibly reduce its memory consumption. As a result, a significant amount of memory was dropped when it completed initialization and processing of the same objects it was handling before the restart.

A memory checking utility was used to wrap the SDC while running in a game of 900 air missions for 24 hours. This uncovered a leak related to processing object updates and creates. When the

object is processed, the SDC must hash it into a local storage structure. During the hash process, the SDC obtains the mask of attributes that are set when the object container is received. A small amount of memory is acquired when the mask structure is used but not returned. This happens frequently for every object, so the cumulative memory loss continues to grow during a multi-day run.

The two routines for processing object updates and creates were modified. The memory used for the mask structure is returned to the system when processing is complete.

3.46 JTLS-2025-17481 Player Cannot See OPAREA Special Flags

Besides Rules of Engagement (ROE), an Operations Area (OPAREA) can be told to mark missions as special missions within generated Link 16 messages when the mission is in the OPAREA. The Manage Operations Area order is used to set or unset these special mission flags, but there is no method available to a Player to ask for the current special flag status for an OPAREA.

Currently, a Player, using the Manage Operations Area order can create a new OPAREA, delete an OPAREA, or modify any of an existing OPAREA's related information, except for the assigned OPAREA's ROE.

A fourth option has been added, called "Describe OPAREA". Submitting a "Describe OPAREA" order will generate a message that is sent to the requesting WHIP that contains all of the current OPAREA data (except the OPAREAs ROEs). This message includes the current status of the Foreign Side mission special flag instructions, and the Own Side Mission Type special flag instructions.

3.47 JTLS-2025-17483 OPAREA ROEs Can Be Changed By Any Side

If an OPAREA is visible to several Sides, the ROE order allowed any Side with visibility of the OPAREA to change the OPAREA's ROEs.

This was not the intent of making an OPAREA visible to multiple Sides. Only the Side that owns the OPAREA should have the ability to set and change the ROEs for the OPAREA. The ROE order was changed to allow only the owning Side the ability to change an OPAREA's ROEs.

It should be noted that in August 2025, STR JTLS-2025-17296 was implemented to stop users from creating OPAREAs owned by the Controller as part of the database. This issue was not uncovered at that time.

This time a thorough review of all orders that reference an OPAREA was conducted. there were basically three types of orders that were considered for change:

- The ROE order was changed so only the owning Side can submit ROE changes for an OPAREA.

- The Manage Operations Area was also changed to limit who can modify, delete or view the settings for an existing OPAREA. Only the owning Side is allowed to accomplish these manage tasks for an existing OPAREA.

- All orbiting air mission orders have the ability to tell a mission to orbit in a known OPAREA. The Design Team decided not to change these orders. Any air mission that has visibility on an OPAREA can reference the OPAREA and have its mission orbit within the OPAREA.

3.48 JTLS-2025-17485 ROE Order Does Not Accept Container Unit

Container Units cannot have ROEs, because they contain no assets, but the design indicates you should be able to specify a Container Unit in those orders that follow the command chain down. The ROE order is one such order, but it was not setup to handle a Container Unit.

The order and code were changed to fully support the Container Unit design. If a Container Unit is specified as the top Unit for an ROE order, the user must indicate that the intention is to follow the hierarchy down the command chain.

3.49 JTLS-2025-17488 Unit Sensor Range Rings No Longer Available

If a Unit owns any Sensor Targets, a range ring for the maximum range of each of the Sensor's usage types should be available from the Unit icon on the Map, but those range rings are no longer available.

A Sensor Target has two ranges:

- The engineering capability of the type of Sensor.

- The Target's capability, which can never be larger than the engineering capability, but may be smaller based on individual Target limitations (such as minor maintenance issues or reducing power due to the interference of the signal on civilian assets).

  Sensors, Jammers, Communication Sites, Surface-to-Surface Missiles, and Air Defense Sites can have individual Target limitations. Few databases implement the representation of individual Target limitations, but the ability to do so exists.

A code review uncovered the problem that a Target's individual limitation was not considered when creating the range rings available on a WHIP. The correction to this problem was properly implemented for all types of Targets except Sensor Targets. This error resulted in the Unit's Summary Target Range Rings for Sensors to all be set to zero and therefore not created.

The code for Sensor Targets was corrected and it now properly considers both the engineering range capability of a Target and the Target's individual limitations if such a limitation exists in the Target's database definition.

3.50 JTLS-2025-17490 Tech Tool Copying Files To Remote Host Too Slow

When trying to copy a large number of order groups with many orders, or a large number of map slides at a time to/from a remote host using the Tech Tool, the time to copy takes too long (for example, an order group with 1000+ unique orders can take up to 10 minutes).

The existing copying methodology for copying a large group of files was simplified, significantly reducing the copying speed.

3.51 JTLS-2025-17492 ATOT ACMs Ingress/Egress Route Names

When an ACO contains an Air Control Mean (ACM) Area with spaces in its name, the ATO Translator is careful to substitute an underscore for the space in all references to the ACM in both the ACO and the ATO.

However, for missions in the ATO that use ACM corridors for ingress or egress routing, the substitution is not done. In this case the mission order fails to validate because the corresponding ACM corridor exists, but it has underscores in its name and does not match a corridor name having spaces.

The ATO Translator was modified to perform the substitution on ACM names used for ingress and egress routing, just like it does for ACMs used for tasking.

3.52 JTLS-2025-17494 OPM Tables - Toggle All Columns On/Off

Toggling table columns on/off in the OPM can only be done one-by-one.

Table columns can now be toggled on/off with a single click.

3.53 JTLS-2025-17495 MDP Display In 10-Meter MGRS

The Message Delivery Program (MDP) only allows a user to choose location formats as either being in Text Lat/Lon or 5 digit (1 meter) MGRS. There was a need to display locations, via the MDP, in 4 digit MGRS.

The WHIP allows location displays in either text Lat/Lon, decimal Lat/Lon, 5 digit MGRS, 4 digit MGRS, 3 digit MGRS, 2 digit MGRS, or 1 digit MGRS.

The MDP configuration/monitoring program was modified to allow all the same location formats as the WHIP.

3.54 JTLS-2025-17497 MDP Directed Message WHIP Selection Not Working

The MDP configuration allows the selection of directed messages by message type and recipient WHIP. Numerous recipient WHIPs can be specified in the configuration, and an actual message may have one or numerous recipient WHIPs. For messages with multiple recipients, the MDP was

not properly determining the match and processing the message even if there was a configuration match.

The search was properly identifying a configuration match, but was not properly terminating the search. This allowed the search to continue, which often ended in a determination that there was no match. This bug has been fixed so the search properly terminates once a configuration match is found.

3.55 JTLS-2025-17499 Tech Tool Order Fields In Proper Sequence

One of the capabilities of the Tech Tool program is to search for orders that contain a specified string. The returned orders can then be viewed either in their raw format as sent to the CEP, or in a more user-friendly format that uses the labels and the data appearance of the WHIP.

Because the WHIP does not guarantee the sequence that fields and groups appear in a submitted order string, the Tech Tool's user-friendly format does not match the sequence of fields and groups as they appear on a WHIP order panel. For large orders, this can substantially slow down debugging.

The parser that breaks an order string into field/group keywords and values was reorganized to load the fields and groups into memory, reorder them so they match the appearance of a WHIP order panel, and then write the data out for display.

It is possible for an order to have fields or groups that should not appear in the order string because the group option to which they belong was not selected. The OMA prevents this from happening for orders coming from the WHIP, but there are other ways to submit such bad orders. These fields/groups will still be displayed at the bottom of their section and highlighted to denote an error.

3.56 JTLS-2025-17501 Improve Logistics Rollup Report Personnel Portion

There are two sections in the Logistics Rollup Report that address the current personnel status of the specified Unit and its subordinates. The information in the two sections are not consistent and it is difficult to provide information to support exercise audience requests.

The two sections of the report are:

- The total personnel supplies that exist in the specified unit and its subordinates. This portion of the report is needed to indicate what Wounded In Action (WIA) personnel and Killed in Action (KIA) Remains still need to be evacuated.
- The total personnel Combat Systems that exist in the specified Unit and its subordinates. This portion of the report is needed to provide the exercise audience with a summary of the total personnel still within the specified Unit and its subordinates. This include the cumulative WIA and KIA for each type of personnel Combat System.

These two sections do not match, and it is not easy to provide the exercise audience with a consistent explanation of the personnel status of the specified unit and its subordinates.

Several changes were made to help with this issue.

These changes were made to the Personnel Supply Section:

- One error was found in this section of the report - the WIA and KIA supplies were passed using the wrong format. Thus, when there are only three personnel Combat Systems became WIA supplies, the model indicated that there were 0 WIA supplies. The format was changed so the value 0.3 was output to the message and is now displayed properly.

- Users did not know how to properly interpret this section of the report. There are two important concepts a user needs to keep in mind when reviewing the personnel supply portion of the report:

  The Personnel Combat System Section of the report contains the cumulative number of WIA and KIA for the various Personnel Combat Systems. If some of those WIAs have been evacuated or repatriated, then the Personnel Supply Section of the report will not match. This is intended and the primary reason that the Logistics report includes a Personnel Combat System summary and a Personnel Supply Summary Section.

  When the model determines a person is KIA, the Remains for the KIA may not be recovered. This is a database parameter CSP PROB REMAINS RECOVERED. Thus, the Personnel Combat System section of the report may show 6 KIAs but the supply portion of the report only indicates that 5 personnel Remains are waiting to be repatriated.

Response cell personnel reported that they need more information in the Personnel Combat System Section to easily answer training audience questions. This portion of the table now includes the following information for each Personnel Combat System that exists in the specified unit and its subordinates:

- Total TOE - this is new information.

- Total Operational - this existed in the previous version of the report.

- Maint / Hospital - this existed in the previous version of the report, but the exercise audience did not understand the label. The label has been changed to indicate the personnel are either Sick or Injured.

- WIA - this existed in the previous version of the report.

- KIA - this existed in the previous version of the report.

- Total - this is new information. The report now provides the total number of personnel Operational, Injured, WIA and KIA for each Personnel Combat System

- A footnote indicator (*) if the Total column is greater than the TOE. The footnote in the report indicates that this could happen if replacement personnel had been sent to the unit.

Finally, column totals for this Personnel Combat System table were added to the report. Thus, the user now has immediate access to the Total Personnel TOE, Operational, Injured, WIA and KIAs for the specified unit and its subordinates.

3.57 JTLS-2025-17503 RSP With Double Default Fields

When running the Reformat Spreadsheet Program (RSP) with a control file that specifies default fields, the user is given the opportunity to not generate an order file if an error exists in a piece of data. If the user chooses not to generate an order file, fixes the bad data, and then processes the file again in the same RSP session, duplicate default fields on the orders are created. The resulting order file will fail OMA checks and will need to be generated again with a fresh RSP session.

Orders that passed the first generation attempt, but were not written out, were left in the set of orders. On the second pass, after reading in the corrected orders and adding them to the set of orders, the default fields were added to all orders in the set of orders. This resulted in those orders that passed verification on the first attempt would appear twice, once with duplicate default fields, and once correctly.

The code was changed so that when the user chooses not to generate the order file, all orders are removed from the set of orders. Thus, on the second pass through the RSP the program is starting with a clean slate of orders.

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 6.4.2.0 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 6.4.2.0 DATABASE CHANGES

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

APPENDIX C. Version 6.4.2.0 REPOSITORY CHANGES

No significant changes have been made to the structure of the JTLS-GO 6.4.2.0 repository.