JTLS-GO

Version Description Document

June 2025

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.3.6.0)

[Blank Page]

ABSTRACT

The Joint Theater Level Simulation - Global Operations (JTLS-GO®) is an interactive, computer-based, multi-sided wargaming system that models air, land, naval, Special Forces, and Non-Governmental Organization (NGO) functions within a combine joint and coalition environment.

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.3.6.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.3.6.0 is a Maintenance release of the JTLS-GO 6.3 series that includes fixes to uncovered software issues, an updated repository of standard data, and a demonstration scenario based in the western Pacific. This release includes one new model enhancement, described in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

This publication is updated and revised as required for each Major or Maintenance version release of the JTLS-GO model. Corrections, additions, or recommendations for improvement must reference specific sections, pages, and paragraphs with appropriate justification and be forwarded to:

JTLS-GO Director of Development
Valkyrie Enterprises LLC
1200 Piedmont Ave, Suite A
Pacific Grove, California 93950-5157
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 6.3.6.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.3.6.0 is a Maintenance delivery for the JTLS-GO 6.3 series of releases.

JTLS-GO 6.3.6.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario, called “wespac63”, based on the Western Pacific theater of operations. There were no database format modifications between this Maintenance release and the original JTLS-GO 6.3.0.0 version. Appendix B of the original *JTLS-GO 6.3.0.0 Version Description Document* summarized the database format changes made between the JTLS-GO 6.2 series and this JTLS-GO 6.3 series of the software system.

Detailed description of the new Engineering Change Proposals (ECPs) is provided in Chapter 2.0. Chapter 3.0 summarizes the Software Trouble Reports (STRs) that have been corrected and are delivered with this version of JTLS-GO 6.3.

JTLS-GO 6.3.6.0 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 6.3.4.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.3.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.3.4.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.3.5.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.3.4.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.3.5.0)

- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.3.4.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.3.4.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.3.4.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.3.4.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.3.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.3.5.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.3.0.0)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.3.6.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.3.6.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.3.6.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.3.6.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.3.6.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.3.6.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.3.6.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)
- Geo-Spatial Service (GSS)
- 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 214.
- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:

  Apache Server (APACHE) version 2.4.62

  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)

  JTOI_ICC302 - Used to feed NATO Integrated Command Control (ICC) Version 3.0.2 system.

  JTOI_ICC320 - Used to feed NATO ICC Version 3.2.0 system.

  JTOI_ICC340 - Used to feed NATO ICC Version 3.2.0 system.

  JTOI_ICC350 - Used to feed NATO ICC Version 3.2.0 system.

  JTOI_NECCCIS - Used to feed NATO Northern European Command, Command Control Information System (NECCCIS).

  JTOI_TBMCS - Used to feed US Theater Battle Management Core System (TBMCS).

- 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.

Instructions for installing JTLS-GO 6.3.6.0 are provided in the *JTLS-GO Installation Manual*. Compared to the JTLS-GO 6.2 series, the JTLS-GO 6.3 series uses a significantly different version of PostgreSQL and the Linux operating system. If an organization has not already upgraded to the JTLS-GO 6.3 version, ensure special attention is given to following the documented operating system and PostgreSQL installation procedures. 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 “repository63”. 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 “wespac63”, which is suitable for training and demonstrations. The scenario has been updated to use the newest version of engineering level data.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.3.6.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

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

  JTLS-GO 6.3 has been tested with the following versions of Linux 9:

  RedHat Linux 9.4 - this operating system license must be purchased.

  Oracle Linux 9.4 - 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 6.3.6.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 9.4

  Oracle Linux 9.4

  Windows 10, which can be used only if the workstation is an external HTTP client of the simulation network.

- JTLS-GO 6.3.6.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 452 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 6.3.6.0 is being delivered PostgreSQL 15.13 that has been compiled under Linux 9.4 and is bundled with the tar files for this release. This version of PostgreSQL is the latest security patch release of PostgreSQL and is being delivered in accordance with US Department of Defense Cybersecurity requirements. It is not necessary to use the delivered solution, but it is the easiest method to meet the requirements of JTLS-GO 6.3.6.0. 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.

- JTLS-GO 6.3.6.0, using the JODA service, allows connections and data exchange with customer client programs. The customer client programs are linked with a set of JTLS-GO-provided API libraries that permit a TCP/IP connection between the JODA and the client program. These API libraries, called JDSP libraries, are built for Linux and Windows and allow customers to built client applications on either of these operating systems. Below are the development environments under which each of the JDSP libraries are built:

  RedHat Linux 9.4 using gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-3.0.1)

  Windows 10 using Visual Studio 2017 version 15.9.60 and Visual C++ 00369.60000.00001-AA807

- JTLS-GO 6.3.6.0 runs using OpenSSL version 3.0.7, which is a required RPM for execution of JTLS-GO. These libraries are required for the use of PKI certificates and for verification of Secure Shell Protocol (SSH) logins.

JTLS-GO 6.3.6.0 runs using OpenSSL version 3.0.7. These libraries should not be upgraded to a later version.

To verify the version of OpenSSL that you are using, you can use the **openssl version** command. If the openssl command fail to run, you will need to install OpenSSL 3.0.7. This specific version can be installed using the following command:

dnf install openssl-3.0.7-28.0.1.el9_4

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 keeps 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 6.3.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.

The JTLS-GO 6.3 series has been issued an Exit Gate letter and certification from the JS/J7 Cybersecurity branch. Please contact the U.S. Government Program Manager, Mr. Douglas Failor (douglas.l.failor.civ@mail.mil) to obtain the completed Cybersecurity paperwork.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.3.6.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

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

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the $JGAME/data/hla directory. Previous federation testing of JTLS-GO with CAE's GESI model has been accomplished using this FOM. The reader should note that the JTLS-GO Development Team, to date, has not been able to

test this federation using this maintenance release of JTLS-GO. 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 6.3.6.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation (www.pitch.se). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

Significant database structure differences exist between the JTLS-GO 6.3 series and the previous JTLS-GO 6.2 series database structure. Appendix B of the *JTLS-GO 6.3.0.0 Version Description Document* has a summary of all database changes. To upgrade your JTLS-GO 6.2 scenario to JTLS-GO 6.3 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

Due to changes made in support of STR JTLS-2025-17088 Missile Capability Does Not Have Torpedo, the record has been added to the JTLS-GO 6.3.5.0 Configuration Managed tw_missile_capability lookup table. This is not a database format change, but to make this new record available for the database building effort, users should **unload** their JTLS-GO 6.3 series scenarios prior to installing JTLS-GO 6.3.5.0, and then **load** them following installation. Please refer to the *JTLS-GO 6.3.5.0 Version Description Document* for more details.

Upgrading earlier JTLS-GO versions to the JTLS-GO 6.3 format will automatically correct the problem. No additional action is required.

Due to the mandated Cyber-Security improvements made to the Database Configuration Program (DCP) and the Database Development System Client (DDSC) for JTLS-GO 6.3.4, each PostgreSQL database scenario needs to be reconfigured. If you skipped moving in JTLS-GO 6.3.4, please execute the procedure in the following note for each JTLS-GO 6.3 scenario:

- Go to the $JTLSHOME/data/scenario/<scenario_name>/config directory.
- Delete the entire directory and its subdirectories.
- Start the new DCP and create the needed DDSCs.
- Each organization can decide whether they want to use the new encrypted password capability and the new DDSC privilege capability. Refer the *JTLS-GO Database Development User Guide* for information concerning the new DCP interface.

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 6.3.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.3.0.0 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

Users are reminded that organizations should not make any modifications to the Default Symbol Set delivered with JTLS-GO. End-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to their scenarios to meet specific organizational needs.

The JTLS-GO Default Symbol Set has not been changed since August 2020. At that time, some new symbols were created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed. If any existing scenarios have not moved to this new Default Symbol Set, this means that the user can easily move in this new symbol set. Please follow the steps outlined in the *JTLS-GO DDS Users Guide*, Section B.13, Updating Scenario Symbol Set.

1.4.3 Standard Repository Changes

The JTLS-GO Database Team has continued to improve and expand the unclassified data repository, which has been renamed to “repository63”. 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 started to represent additional Combat Systems to more closely match the Combat Systems recognized by the Joint Live Virtual Constructive (JLVC) federation of models. This effort is expected to be an ongoing effort for the next three to five months.

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.

Reminder: Ensure existing JTLS-GO Version 6.3 databases held in the PostgreSQL table-space are unloaded prior to installation and reloaded after installation, as explained in Section 1.4.

2.0 ENGINEERING CHANGE PROPOSALS

One Engineering Change Proposal (ECP) was delivered with this release.

2.1 JTLS-2025-17221 Parse Spreadsheet of Simple ATO Missions

Summary of Model Change Request

JTLS-GO should have the ability to translate an ATO, having only basic information for a small set of missions, into a corresponding ATO of missions. The information for each mission would be provided in spreadsheet format, and the resulting set of ATO missions would be produced in a standard XML format usable by the ATO Translator.

Design Summary

A Parser will used for extracting mission information from the provided ATO spreadsheet. The Parser will build an XML ATO for the ATO Translator, which gives the user the capability to change mission data. The ATOT will be able to generate mission orders, including the ATO ID order for the period.

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-2025-17174 Non-Squadron Unit Aircraft Surge Computation

The number of surge sorties over the last three days is not computed correctly for non-squadron units that own aircraft.

The event ADJUST SURGE FACTOR only computed the surge data parameters for squadrons, and not non-squadron units that own aircraft. The event now computes the surge sorties for all aircraft-owning units.

3.2 JTLS-2025-17175 Runaway Task Stops When Mission Moves Into OPAREA

Operations Areas (OPAREAs) can have a Special Flag set, which sets the C4I Special Interest Flag for air missions when they enter the OPAREA.

A mission was running away from an interceptor, and it moved into an OPAREA with the Special Flag set. The model crashed indicating the victim mission has no existing route points.

When the victim mission started to execute its next move, it moved in an OPAREA with the Special Flag set. When the mission entered the OPAREA, all interceptors heading towards the mission were canceled. Because the mission was no longer being intercepted, the mission had no reason to continue running away, so its Run Away task was canceled and its route emptied out.

The issue was the mission was in the middle of its next move. When the logic asked what the next route point is, the model crashed because the mission no longer had a next route point. The logic was changed to catch this situation and cancel the current ongoing move for the mission. The mission then automatically picked up its next task and properly continued on its way.

Because there is no way to guarantee that there is not another set of unusual circumstances that can cause this problem, a check has been placed in the code. If the mission tries to access its next route point and there is no route point, the model now generates a Logic Error and exits the move routine.

3.3 JTLS-2025-17176 Set Sensor Range Unowned Sensor Target Crash

If the Controller changes the range of a sensor, and the database includes a non-owned Sensor Site Target that uses the sensor for which the range change is being made, the model crashes.

The crash happens when the model is rebuilding the range rings for the Target. The Target that caused a problem was associated with a Unit, but the code incorrectly checked if the sensor was

owned by an HRU. This check was not formatted properly, and it incorrectly came to the conclusion the Target was owned by an HRU. This caused the model to crash.

The check's formatting issue was corrected, as well as the incorrect code assumption.

If this crash occurs, the Snap Checkpoint is good and can be used.

3.4 JTLS-2025-17177 Logic Error Changing Join Formation Task

A user entered an order to change the formation a unit was ordered to join, and the model generated a Logic Error.

The logic error was caused by the model asking for the perceived location of the new formation. The code was attempting to update the perceived location on the naval unit's Pending Task, but Formations do not have a perceived location.

The model was programmed to decide whether or not to update the task with its current location, based on the characteristics of the original task. If the original task had a location, the altered task will have the location of the formation. If the original task did not have a location, the altered task would also not be given a location.

3.5 JTLS-2025-17178 Non-Squadron Units With Aircraft Wrong RIC Code

NATO uses Routing Identification Codes (RICs) with their Logistics Functional Area Services (LOGFAS). In JTLS-GO, Combat Systems, Aircraft, and various other objects are assigned RIC codes in the database. JTLS-GO uses this data to interface with LOGFAS.

Allowing non-squadron Units to own aircraft is a relatively recent change. Aircraft from non-squadron Units were not reporting the proper aircraft RIC codes.

The logic to determine the proper RIC code was centralized to ensure that it is used by all objects, whether they are squadrons or other types of Units.

While testing the new code a crash situation, not caused by the code change, was discovered in preparing a LOGFAS RIC Code Report. The crash situation was corrected.

3.6 JTLS-2025-17179 National Boundary Overrides Delay Mission

When a user inputs National Boundary overrides as part of an air mission order, the National Boundary Override Tasks are given a start time equal to the primary task's desired execution time. This causes the mission to stop and wait until it is time to execute the Alter National Boundary Override Tasks, causing the mission to be late for its primary task.

The National Boundary Override Tasks are properly placed in the mission's task list, after the mission is told to leave and before the mission's first primary task. Because it is properly placed

in the task list, it does not need an execution time. The times have been removed from the National Boundary Override Tasks.

3.7 JTLS-2025-17180 Negative Unit Detection Time Not Handled Correctly

Units on the map that show a detection time calculated to be in the future (negative calculated time) would display the detection bar as "low strength" (indicating an old detection). This is because negative calculated detection times were not being handled properly, and would default to "low strength".

Detection times calculated to be in the future now display the corresponding unit detection bars with full strength (indicating a recent detection time).

3.8 JTLS-2025-17181 Mission Route Fuel Computation Incorrect

An air mission was continually leaving its orbit location to get fuel from a tanker at exactly the same location. The first error found was in the routine that computed how much fuel was needed for the remainder of the mission's planned pending execution tasks (PETs). This computation took into account the distance the mission needed to travel to get to the PET location and the time it needed at the PET location (if the PET included a duration or an end time). If the mission had more than one task that had a duration, the task's duration time was being accumulated over all tasks.

Several issues were found in the routine:

1. If a PET has no location, that means that the mission does not need to travel to get to execute the task. The check to determine if the PET had a location was incorrect. It asked if the location was greater than almost zero. This check would not work for things in the southern or western hemisphere. The check was changed to ask if either the latitude or longitude were not equal to zero, and if so, the fuel needed to get to the task location should be computed.

2. The duration of a task was not computed correctly. The duration of a task should be calculated based on the following things:

    a. The time the task should start (the expected arrival time at the task location).

    b. The duration of the task (if a duration is specified).

    c. The task end time (if no duration was specified).

3. Finally, the total duration of a task was reset to zero for each task computed. This was corrected.

3.9 JTLS-2025-17182 Cancel Submarine Depth Change Creates Logic Error

The Change Depth task is a Standalone Task, which means it has an Object Take Action Event scheduled to make the task execute. When the Player cancels a Standalone Task, the model was not canceling the associated Object Take Action Task.

A new routine was written to check if the Standalone Task being canceled is the first Standalone Task in the object's Task Set. If it is, then the Object Take Action Event is canceled, and if appropriate a new Object Take Action Event is scheduled for the next Standalone Task on the object's task list.

3.10 JTLS-2025-17183 Loitering Munition Used For CAS Crash

When a Loitering Munition aircraft is assigned to fulfill a Close Air Support (CAS) request, the model crashes attempting to find the best weapon to fire.

The one and only weapon that a Loitering Munition aircraft carries usually has a range of zero, indicating the aircraft and weapon need to be directly over the target when fired. The logic used to determine the best range weapon did not pick a zero-range weapon. The code was corrected to allow a zero-range weapon to be selected.

3.11 JTLS-2025-17184 Polyarc ACM Orbiting Crash

The model crashed when a mission arrived at its assigned orbit location, which was a Polyarc Air Control Mean (ACM) area.

The polygon used to represent the Polyarc was not properly saved on the missions Orbiting Task. The code was changed to ensure that the polygon coordinates used to represent ACM Polygons, Radarcs, and Polyarcs are properly saved on the Orbiting Task.

3.12 JTLS-2025-17185 PostgreSQL Database Server Upgrade Documented

The PostgreSQL Database Server needed to be upgraded from one interim version to a newer interim version (primarily for security and bug fixes). This update process needs to be documented in the JTLS-GO Installation Manual.

JTLS-GO 6.3 was distributed with the latest PostgreSQL version 15 client. JTLS-GO 6.3 database tools were tested against the matching PostgreSQL Database Server. Users need to update their PostgreSQL Database Server if they are still using an older interim version.

The update process is now documented in Chapter 6 of the *JTLS-GO Installation Manual*.

3.13 JTLS-2025-17186 Air Mission Routes Stuck On Display

When opening an air order panel with an established air route or utility route displayed on the map, exiting out of the order panel too soon will cause the order route to be stuck on the Map, even after the window has closed.

This was caused by a race condition, where the order route was still being drawn on the Map and, because the order panel was closed too quickly, the WHIP thinks the order panel is still open.

Exiting an air order panel before the route was drawn now overrides the route drawing procedure, causing it to exit and not draw the route onto the Map.

3.14 JTLS-2025-17187 AAR Reports Save As HTML Action Does Not Work

When attempting to save AAR reports that contained supply amounts and Units of measure (UOMs), the resulting file would say that a transformation error had occurred.

When requesting that a report be saved as HTML, the WHIP/TRIPP was sending a request to the Glassfish server to transform the report into the displayed HTML. This was a different action than when a report is viewed in the Report Browser, where the WHIP/TRIPP performs the transformation internally. Performing the task internally makes all the user preferences, including supply UOMs, available to the transformation.

Because the supply UOMs had not been set up on the Glassfish server, it would crash attempting to access those user preferences. This issue was affecting saving the report as HTML, e-mailing the report, and printing the report, all of which were passing the transformation request to the Glassfish server. The code for all three actions was modified to perform the transformation internally to the WHIP/TRIPP and then save, send, or print the results.

3.15 JTLS-2025-17189 Refueled Mission Returns To Orbit After End Time

An orbiting air mission required fuel to continue its tasking. The logic selected a tanker that was some distance away. By the time the mission refueled and returned to its orbit location, the off-station time had passed. The mission then headed home later than originally scheduled. In this situation, that tanker should not have been selected to refuel the mission.

The logic that determines the feasibility of a tanker to refuel a mission was expanded to consider the off-station time. A new criterion was added to exclude a tanker (or airbase) if the flight time to or from the tanker would cause the mission to return to orbit after its orbit end time. If there is no other feasible refueling source available, the mission simply heads home due to low fuel remaining, as usual.

3.16 JTLS-2025-17190 OPM MSC Page Has Incorrect Field Title

The Minefield Search Capability (MSC) Online Player Manual (OPM) page listed a database parameter field for "Clear Rate". The data presented was actually the Detection Probability. The label is wrong.

The column label was corrected and the Data Requirements Manual (DRM) reference page link was also corrected.

3.17 JTLS-2025-17191 Backhaul Supplies Not Showing On IMT

When a supply convoy is told to backhaul (transport from a supported Unit back to the Support Unit) causalities and remains after a delivery, the convoy properly executes the tasks, but the WOUNDED or KIA personnel supplies loaded on the convoy are not sent to the JODA.

The code as corrected to update the convoy supplies after the backhauled supplies are loaded.

3.18 JTLS-2025-17202 Mission Selected Tanker Beyond Control Range

An air mission, consisting of an aircraft type that had a control distance restriction, attempted to refuel from a tanker that was beyond the mission's maximum control distance. The mission moved toward the tanker and stopped upon reaching the maximum control distance. In this situation, that tanker should not have been chosen by the refueling logic.

New logic was added to reject refuelers (tankers or airbases) that are beyond the aircraft type's maximum control distance.

3.19 JTLS-2025-17203 Random Number Seeds For ELS Mismatch

Six random number seeds were used exclusively by the Entity Level Server (ELS). They were not used in the main functions of the Scenario Initialization Program (SIP) nor by the Combat Events Program (CEP).

The Names of the seeds, and the initial values for them, were stored in the Database Development System (DDS). The names held by the DDS did not match the named expected by the SIP and CEP. This mismatch causes the programs to crash while starting.

In several places in the SIP and CEP code, the names, indexes, and values of the random number seeds were changed to be consistent with the DDS. These changes resulted in no further crashes by the SIP and CEP.

3.20 JTLS-2025-17204 Airborne Mission Cannot Receive External Tank Fuel

JTLS-GO represents an aircraft's internally held fuel, and depending on the Aircraft Load selected, any fuel held externally in tanks. If an aircraft is air-to-air refuelable, JTLS-GO currently

does not represent the ability to refuel the mission's assigned external fuel during an air-to-air refuel operation.

The database parameter AC AIR REFUEL CAPABLE is used to indicate whether the Aircraft Class is capable of participating in an air-to-air refuel operation. This database parameter previously held a value of either **Yes** or **No**. This was changed to be a three-way flag that has the following meaning:

- **No** means the aircraft cannot participate in an air-to-air refuel operation.
- **Yes** means the aircraft can participate in an air-to-air refuel operation, but only internal fuel can be refueled.
- **External** is a new value that can be assigned to this database parameter. It means the aircraft can participate in an air-to-air refuel operation, and both internally and externally-held fuel can be transferred as part of the operation.

The following logic was changed within the model:

- When determining the maximum amount of fuel that can be transferred in an air-to-air refuel operation, if the AC AIR REFUEL FLAG is set to:
  a. **No**, then no fuel can be transferred.
  b. **Yes**, the maximum fuel that can be transferred is held in the database parameter AC FUEL.
  c. **External**, the maximum fuel that can be transferred is held in the database parameter AC FUEL plus the AL.FUEL database parameter for the Aircraft Load assigned to the Air Mission.
- When determining the maximum amount of fuel that can be transferred in a base refuel operation, the AC FUEL plus AL FUEL amount is always used, regardless of the AC AIR REFUEL FLAG setting.
- When indicating that an Air Mission's fuel should be topped off using the Magic Air Ops Order, the mission will end up with its current fuel set based on the AC AIR REFUEL FLAG value:
  a. **No** - the mission will end up with AC FUEL
  b. **Yes** - The mission will end up with AC FUEL
  c. **External** - The mission will end up with AC FUEL plus AL FUEL for the Aircraft Load assigned to the Air Mission.

3.21 JTLS-2025-17205 Improve Aircraft Load Assignments

The Aircraft Load Assignment link under the Lethality OPM menu takes a long time to load. The Load Assignment link for Aircraft in the Aircraft Target Category menu is linked to the same Load Assignment page, which also causes long delays.

Two improvements were made:

- The Load Assignment link in the Aircraft table was corrected. It now takes the user to the Load Assignment page specific to the Aircraft Type.
- The Aircraft Load Assignment link under the Lethality OPM menu was rewritten to display only records where the Aircraft, Target Type, and day/night combination contains a load assignment. NULL entries are not displayed.

3.22 JTLS-2025-17206 Crash Change Tasks For Delayed Mission

An air mission was in Load Delay and a user entered an order to cancel one of its tasks. The last thing the code does is check if any of the changes made by the user would require a change into the scheduled launch time of the mission. This procedure crashed the model, because the mission was in a Delay State.

The logic first asked whether the mission had already started. It had not, because it could not launch due to a capability problem or an asset unavailability problem.

The logic then recomputed a new launch time. Because the mission was delayed, this logic sequence could not complete. There is no launch event scheduled for a delay mission, because the model has no idea when the delay situation will be solved.

To correct this error, an additional check was added. If the mission is in a delay state, then the entire code sequence used to recompute a new launch time is skipped. When the delay problem is solved, the model will automatically recomputes a new launch time.

3.23 JTLS-2025-17209 Cancel Mobility Mission Returns Doubled Supplies

An Air Transport Mobility mission carrying supplies landed at a destination airbase and began to offload. Before the mission had completed the offload task, a Player canceled the mission. When the mission returned home, double the supply amount was restocked at the home location.

When an Air Transport Mobility mission offloads supplies, the Offload task holds the list of supplies to remove from the mission aircraft. The estimated time to offload the supplies is calculated and a task completion event is scheduled for each aircraft in the mission. As each task completion event occurs, the supplies are removed from the mission and the task offload list is decremented accordingly.

If the mission (or the offload task itself) was canceled by the Player, the logic was incorrectly adding the remaining items in the offload list back to the mission, which caused a doubling of the amount on board. This incorrect subroutine call was removed. As a result, the loaded supplies remain on the mission and any offloaded supplies remain at the receiving unit as expected.

3.24 JTLS-2025-17211 Fixed-Wing Aircraft Zero Landing Length Crash

JTLS-GO currently assumes that a fixed-wing aircraft requires some type of runway and must have a positive runway takeoff and landing length specified in the database. If a fixed-wing aircraft needs to "crash land" on a runway that is not longer than its landing length, and the aircraft's landing length is zero, the model will crash.

The model now checks whether the user has somehow made a fixed-wing aircraft have a zero landing length. If this is true, a Logic Error is generated, and the model indicates that no aircraft were lost when the mission crash landed.

An SVP warning was added to ensure that all fixed-wing aircraft have a landing and takeoff length greater than zero.

Several holes were also discovered in the Controller's ability to change data during execution. Code was improved to catch the following situations:

- If the Controller changes an aircraft class from Rotary to Fixed Wing, the runway lengths could end up as zero. If the user attempts to change an aircraft class to Fixed Wing when the landing length is zero, the model will accept the order but inform the user that the change is illegal and internally update the AC.LAND.LENGTH to 1 ft.

- If the Controller sets the landing length to zero for a fixed-wing aircraft. The Set Aircraft Data order has been updated to refuse the order if the user attempts to change the landing length to zero if the aircraft class if Fixed Wing.

3.25 JTLS-2025-17212 Make Unit Its Own Higher Headquarters

A user submitted a Move Order and indicated that after the move, the unit should report to a new Higher HQ. The order indicated that the moving unit should become its own new Higher HQ. This caused the model and the XML Message Service (XMS) to go into an infinite loop.

The model correctly ensures that a command chain loop will not be created when changing a unit's Higher HQ. The logic did not check for the concept that the user would attempt to change its Higher HQ to itself. The code now makes this check.

3.26 JTLS-2025-17215 Manage ACM Sets Altitude To -1

After submitting a Modify ACM Shape order, an ACM was given a Default Altitude of -1.

The Default Altitude parameter does not apply to the ACM Track Shape. The code ensured that a Modify ACM order did not change the Default Altitude of an ACM Track Shape, but it failed to ask whether the user had even entered a value to modify the Default Altitude.

The Modify ACM order initializes all order parameters to negative 1. Because the code failed to ask if the user had specified a new Default Altitude, it was possible to set the Default Altitude of a non-track ACM to negative 1. The code now only changes the Default Altitude of an ACM shape if the user specified a change to the ACM attribute.

3.27 JTLS-2025-17217 Cannot Fire Missile From HRU-Owned SSM

A High Resolution Unit Prototype (HUP) was modified to own a Surface-to-Surface Missile (SSM) Target and a pair of Sensor Targets. A number of HRUs were created using this HUP. A Stop Checkpoint was then taken and the game restarted.

When the Fire Missile order was brought up and eligible units selected, the HRUs were not listed or highlighted. Selecting the specific missile for the SSM target made no difference in displaying the HRUs as eligible shooters. A JDS Object Update of the HRU also did not list it as an eligible shooter.

This error could not be reproduced. However, the WHIP menus have been updated to include the Fire Missile order in the context sensitive menu for an HRU.

3.28 JTLS-2025-17218 Trusted User Setup Script For Non-Directory Files

For Trusted User setup, all non-directory files, except executables, should have group write permissions added.

The Trusted User setup script was modified so that all non-directory files, except the executables, have group write permissions as well.

3.29 JTLS-2025-17219 Database Schema Should Have Build Number

There is no build number in the database schema to determine if a database change was implemented between the interim or minor releases of JTLS-GO. This is also true during the internal development cycle. The database schema should have a build number.

The new schema_description table to host the built number was created.

3.30 JTLS-2025-17220 Target Range Ring Not Updated For New Range

The range ring for a Sensor Site Target was not updated when the range was changed using the Set Target Data order. The range ring displayed on the WHIP never changed, but the range held in the model was updated with the new value.

The code to process the Set Target Data order was properly assigning a new value to the range for a specific Target. However, the code to update the range ring held by the JODA was not processed correctly. Modifications were made to incorporate the new value for the Target range such that the range ring was updated and subsequently displayed on the WHIP.

3.31 JTLS-2025-17222 Roll-Up Naval Unit Strength Computation Error

An error existed in the CEP code used to compute the rolled-up unit strength for Naval Units, causing a crash.

While computing the rolled-up strength for a Naval Unit, the code was attempting to access properties for the hull of the ship. These parameters were defined by the Ship Unit Prototype (SUP). The CEP code attempted to access a local value for the prototype before that local value had been assigned. The model cannot access an attribute for an unspecified prototype.

A modification was made to access the SUP before it was used later in the code. This change prevented a crash from occurring.

3.32 JTLS-2025-17223 DDSC ASC Dialog Blocks Other User Interactions

When a DDS Client (DDSC) Automatic Supply Calculation (ASC) dialog is up, it prevents the user from interacting with other DDSC windows. This makes it difficult for the user to examine the data change as the result of the supply calculation.

The ASC now allows the user to interact with other DDSC windows while the dialog is up.

Because user interaction can potentially change a parent table's record selection, the code was changed so that the parent table now announces its selection changes, and the ASC dialog will update the relevant changes accordingly.

3.33 JTLS-2025-17224 ASC SUP Consumption Supplies Calculation

The ASC of the SUP’s unit consumption supply did not compute the general supply consumption.

The DDS Glassfish server code was modified to include the calculation of the SUP's general supply consumption.

3.34 JTLS-2025-17225 ASC SUP Artillery Fire Not Working

The ASC of the SUP’s artillery fire always finished without doing anything.

There was a bug in the DDS Glassfish server code when retrieving the relevant data for the calculation. The bug was corrected.

3.35 JTLS-2025-17226 ASC Unit Missile Fire Calculation Added

The ASC of the Unit's missile fire was missing.

The ASC for a Unit's missile fire is now available, along with the TUP, SUP, and HUP calculations.

3.36 JTLS-2025-17227 ATOG Not Returning Temporary Variable Memory

The ATO Generator (ATOG) was recently updated to correct an error with writing routes for air missions in Military Text Format (MTF). The modification to correct the route included adding a time field. While writing the time field, a temporary time variable is instantiated and the memory for this variable is not returned when the variable is no longer needed.

The ATOG was modified to return the memory for the time variable when the routine for writing the associated MTF record has completed.

3.37 JTLS-2025-17228 ATO Parser Uses Memory For Unused Variable

The ATO Parser writes Error and Warning messages when it detects a problem within the MTF of any ATO message it is processing. The ATO Parser was recently modified to centralize the production of the Error and Warning messages.

In one location of the Parser, a routine was partially converted and allowed a pointer variable to consume memory, but was no longer being used for generating an error message. Whenever this MTF error was detected by the Parser, it would create this variable and not return the memory used by the variable, which amounted to a small memory leak.

The ATO Parser was modified to remove this unused variable.

3.38 JTLS-2025-17229 SVP Fixed-Wing Aircraft Takeoff/Land Length Zero

The model crashes if the takeoff or landing length of a fixed-wing aircraft is zero. The SVP needs to be able to catch and fix this situation.

New SVP Errors 174 and 175 were added to check if the takeoff or landing length of a fixed-wing aircraft is zero.

3.39 JTLS-2025-17230 SVP Externally Refuelable Aircraft Without Loads

Aircraft with the Air Refuelable Flag set to "EXTERNAL" should have a at least one load assignment with extra fuel.

New SVP Warning 1164 has been added to identify aircraft that have an Air Refuelable Flag set to "EXTERNAL", but have no loads containing extra fuel.

3.40 JTLS-2025-17231 Update PostgreSQL Database Client

Minor PostgreSQL updates are essential for improving system stability, ensuring security, and fixing bugs. PostgreSQL 15.13 has been released and needs to be delivered as part of the next maintenance release.

The PostgreSQL Client, which is distributed with JTLS-GO, has been updated to the latest 15.13 security version. Also, a custom-created PostgreSQL v15.13 database server tar file has been included with the JTLS-GO release.

The detailed security fixes information for 15.13 version can be found at:

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

3.41 JTLS-2025-17232 SSM Report Logic Difficult To Follow

The routine for determining whether a sensor covers an SSM that is preparing to fire or firing was extremely difficult to follow, included several unnecessary steps, and even included code that was wrong but had no negative effects on the outcome of the desired logic.

The code was reorganized for efficiency.

3.42 JTLS-2025-17233 Missing DRM Entries

The Miscellaneous Modeling Parameter OPM page has no DRM entries for data items.

The page was updated to include the DRM entries.

3.43 JTLS-2025-17234 SVP Error 672 Problems

The automatic correction for Error 672 to move the Target does not work.

The automatic correction option is not intended to be an automatic correction. The appropriate Target table is opened for the user to manually change the Target location. The wording of the correction was modified to ensure the user understands it is a manual correction, not an automatic correction.

3.44 JTLS-2025-17235 Orphaned Lookup Table In DDS Schema

Each sensor name ends in a three letter code following an underscore character. This code is intended to aid those who may not recognize the name of the sensor and indicates the sensor's proper use within the game (such as air search, sonar, etc). The first letter indicates where the sensor can be found. The second letter indicates the type of search that the sensor performs in the game. The third letter indicates the actual function of the sensor in the real world.

The Sensor Type collection mode links to the sensor_mode_lu lookup table, but the collection mode is now built into the Sensor Type name. The orphaned sensor_type_lu table should be deleted.

The sensor_type_lu table was dropped from the DDS Schema.

3.45 JTLS-2025-17236 Number Of Open Files Exceeded In Replay JXSR

The Replay JXSR is exceeding the number of open file descriptors permitted whenever a TRIPP is started and attaches to it. The permitted maximum number is 4096, which represents the number of open files and sockets that the process should have. This causes the Replay JXSR to get stuck in a state where the TRIPP keeps asking for information and the Replay JXSR keeps writing an error message, because the number of file descriptors has been exceeded.

The problem was traced to code that was introduced to support the TRIPP over multiple runs. Whenever a TRIPP request is received, the Replay JXSR checks the scenario runs file to determine if it has changed. This check opened the runs file and, if the file had not changed, was failing to close the file. Because this check was being done for every JXSR request, the number of open files quickly exceeded the limit. The code was modified to properly close the file.

3.46 JTLS-2025-17237 Aircraft Class DIS Code Attribute Change

When using the Set Aircraft Data order to change the Aircraft Class DIS Code, the change was being saved internally by the model but not passed correctly to the JODA.

The object being updated was the "equipment_item_object", not the "aircraft_class_object". The code has been changed to update the correct object to the JODA.

3.47 JTLS-2025-17238 Satellite Names And Altitudes

When using the JTLS Satellite Service (JSAT), three issues were found:

- The JSAT failed to correct problems with the names of satellites. Some of the names included a slash, which was an illegal character.
- When calculating the route for one of the satellites, the route points had altitudes with negative values. These are not physically possible and are not allowed in either the database or the model.
- The minimum value for the satellite altitude was incorrectly set to 1,000,000 feet, instead of 100,000 feet.

The JSAT GUI and the JSAT service were modified such that any instance of a slash was replaced with an underbar ("_") in the satellite name.

The negative altitudes were generated by the JSAT for a satellite which had a highly decayed orbit. In the real world, that satellite would have already burned up in the atmosphere. The JSAT service now checks if the altitude is negative. If that occurs, then the last positive value for the altitude is used for those route points.

The database schema and the related orders were modified to use a minimum altitude of 100,000 feet for all satellite route points.

3.48 JTLS-2025-17239 CSP Faction Attribute Change Message

When trying to view the Combat System Prototype (CSP) attribute for a given Faction, there is no such value listed for the Faction in the message returned by the Faction Attribute Change order.

The Faction CSP has been added to the message received after the Controller order is submitted.

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.

The following list of issues is known and have not been fixed in time to make it into this release of JTLS-GO 6.3.6.0.

4.1 DDSC/WHIP/JOBE - CADRG Map Zoom

When using the CADRG map projection, if the width of the map is less than the height, the zoom tool does not work correctly.

4.2 MHE Targets Loading Air Mission Can Cause a Crash

MHE targets should be avoided for loading and unloading air missions. It is suggested that the database be set to “Do Not Use” for Air Missions.

4.3 JTLS-GO Strategic Lift Missions Are Not Working Properly

Strategic Lift Missions, used to move TPFDD assets into the Theater and report the results to a real-world TPFDD processing system, has not been updated to work within JTLS-GO 6.3.

4.4 Tactical Ground Formation Attacks Do Not Work

The ability to send a Tactical Ground Formation on an Attack mission has been temporarily disabled due to reliability issues.

4.5 ATOT Spreadsheet Lacks Detailed Field Checking

The ATOT Spreadsheet Parser has been found to have numerous issues within the Spreadsheet format that are not caught and cause the spreadsheet parser to crash. Fixing the uncovered issues are being worked and should be fixed prior to the next maintenance release of the JTLS-GO 6.3 series.

4.6 Moving Combat System Supplies Can Reduce Unit Strength To Zero

If a user does a mandatory transfer of Combat System supplies from one unit to another, the providing unit can be emptied out and exists without any Combat Systems or personnel. This situation needs to be thoroughly and properly handled.

4.7 Upgrade Procedures For Maintenance Release of PostgreSQL

The Development Team continues to look for viable options to upgrade from one maintenance release of PostgreSQL to a newer maintenance release.

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
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
SDR	Scenario Data Repository
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 6.3.6.0 DATABASE CHANGES

Due to changes made in support of STR JTLS-2025-17024, the JTLS-GO 6.3.4.0 database format was changed to add a new column to the text_symbol_lu table, related to unit symbols.

This change requires that users **unload** their scenarios prior to installation of JTLS-GO 6.3.4.0, and then **load** them following installation, if users are upgrading to JTLS-GO 6.3.6.0 from JTLS-GO 6.3.3.0 or earlier.

APPENDIX C. VERSION Version 6.3.6.0 REPOSITORY CHANGES

No significant changes have been made to the structure of the JTLS-GO 6.3.6.0 repository.