JTLS-GO Document 17

JTLS-GO

Version Description Document

May 2023

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.1.10.0)

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

JTLS-GO 6.1.10.0 is a Maintenance release of the JTLS-GO 6.1 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as a major model functionality improvement implemented as Engineering Change Proposals (ECPs), This ECP is summarized 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
ROLANDS & ASSOCIATES LLC
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940 USA
jtlsgo@rolands.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.1.10.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.1.10.0 is a Maintenance delivery for the JTLS-GO 6.1 series of releases.

JTLS-GO 6.1.10.0 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 “wespac61”. There were no database format changes made to support this maintenance release. Database modifications that were accomplished to upgrade the previous JTLS-GO database format to the JTLS-GO 6.1 series format are summarized in the VDD for Version 6.1.0.0 delivered in the documents directory.

This release is being provided as part of the normal JTLS-GO maintenance program, in which new maintenance release versions are closely coordinated with Java security patch releases. Two minor improvement to the model is summarized in Chapter 2.0. Chapter 3.0 describes the Software Trouble Reports (STRs) identified during testing and operational exercises.

JTLS-GO 6.1.10.0 executes on the Red Hat Enterprise Linux Version 8.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any 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 the 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 Executive Overview* (JTLS-GO Document 02, Version 6.1.7.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.1.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.1.9.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.1.9.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.1.0.0)

- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.1.9.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.1.9.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.1.9.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.1.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.1.7.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.1.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.1.2.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.1.7.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.1.0.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.1.7.0)
- *JTLS-GO ATO Services User Guide (*JTLS-GO Document 24, Version 6.1.9.0)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.1.10.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.1.10.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.1.10.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.1.9.0)

1.2.4 New Documents

No new documents were produced for this version of the software.

1.2.5 Delivered Software Components

JTLS-GO 6.1.10.0 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)
  Glassfish Version 5.1.0

- 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 173.

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

  Apache Server (APACHE) Version 2.4.54
  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)

- Database Development Server and its component programs

  Glassfish 5.1.0

  Database Development System Client (DDSC)

- 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.

Instructions for installing JTLS-GO 6.1.10.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.1.10.0 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 “repository61”. 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 “wespac61”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.1.10.0 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 8.4 or higher, 64-bit architecture.

  JTLS-GO 6.1 has been tested with the following versions of Linux 8:

  RedHat Linux 8.6 - this operating system license must be purchased.

  Oracle Linux 8.4 and 8.6 - 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 DISA for use by U.S. Government Agencies.

  CentOS 8.4 - a free version of Linux 8 that has not been approved by 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 have a Java-enabled web browser. JTLS-GO 6.1.10.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.9, 8.4, and 8.6

  Oracle Linux 8.4 and 8.6

  CentOS Linux Version 7.9 and 8.4

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

- JTLS-GO 6.1.10.0 is delivered with the Adoptium™ project Temurin Java Development Kit (JDK) 1.8 Update 372 package, which is equivalent to the current version of OpenJDK. Since the release of JTLS-GO 6.1.8.0, JTLS-GO has been using the Temurin JDK, which is optimized for speed, Prior to the release of JTLS-GO 6.1.8.0, JTLS-GO used AdoptOpenJDK, which was optimized for memory. All tests indicate that there is an increase in the amount of memory needed by our Java applications, but the increase is not significant. 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 1.8.8.

- JTLS-GO database tools require a certified PostgreSQL 11.13 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.1.10.0. There are several alternative methods available for obtaining the PostgreSQL 11.13 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.

- TCP/IP is required for inter-process communication between the JODA data server and all user interface programs. The version of TCP/IP included with the supported versions of Red Hat Linux ES is sufficient.

- 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. The code was compiled using Simscript III Version 5.3.

- 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^{TM}. 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 keeps the software up-to-date and all security patches are properly installed. In order to fulfill this DoD Cybersecurity requirement, JTLS-GO has moved to Adoptium^{TM} JDK, a full OpenJDK Java environment with a licensing paradigm allowing an application to deliver the software. The following procedure has been established and approved by the JS/J7 Cybersecurity branch:

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

- Within two-weeks of the Adoptium JDK release, JTLS-GO provides a bug release version (JTLS-GO 6.1.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 completed Cybersecurity paperwork and a current Gate completion certificate can be obtained by contacting Ms. Jessica Camacho, the U.S. Government Program Manager, at jessica.l.camacho.civ@mail.mil.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.1.10.0 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 have been accomplished. Future plans include expanding the capabilities of the GlobalSim federation.

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 6.1.10.0 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 6.1 series of the software and the previous JTLS-GO 6.0 series database structure. There are no database format changes between JTLS-GO 6.1.10.0 and the initial JTLS-GO 6.1.0.0 version of the software.

To upgrade your JTLS-GO 6.0 scenario to JTLS-GO 6.1 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

STR JTLS-2022-15607 Grid Lower-Left Location Not Exact On Restart was included in JTLS-GO 6.1.2.0, released in March 2022. This STR required a change to the ASCII checkpoint format. Any saved checkpoints from JTLS-GO 6.1.0.0 and JTLS-GO 6.1.1.0 will not work with JTLS-GO 6.1.3.0 and above.

STRs JTLS-2022-15810 and JTLS-2022-15839, delivered with JTLS-GO 6.1.7.0, required changes to the checkpoint format. Any saved checkpoints from any previous JTLS-GO 6.1.0.0 versions will not work with JTLS-GO 6.1.7.0 and above.

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 6.1.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.1.0.0 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

No new symbols have been added to the JTLS-GO Default Symbol Set. The JTLS-GO Development Team suggests not making any modifications to the delivered Default Symbol Set, 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. If in the future, the JTLS-GO Default Symbol Set is changed, a procedure to make these symbols available to all your scenarios will be delivered as part of the release. Such a procedure will not work if the end-user organization alters the official JTLS-GO Default Symbol Set.

1.4.3 Standard Repository Changes

R&A is no longer improving and expanding the unclassified data repository “repository61.” All depository work has been moved to the JTLS-GO 6.2 series.

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 11.13 required to operate JTLS-GO 6.1.10.0.

2.0 ENGINEERING CHANGE PROPOSALS

This section describes the new capabilities, known as an Engineering Change Proposal (ECP), that are delivered with this release.

2.1 JTLS-2023-16043 AAR Weapon Damage Report

Summary of Model Change Request

In order to demonstrate the process, steps, and code required to generate an AAR Report to customers, a new Weapon Damage Report was created. Part of the demonstration was to illustrate the steps to develop a report to the coding standards of JTLS-GO. Because this new report was developed to those standards, and exists it was decided to include it in the Configuration Managed version of JTLS-GO.

Design Summary

The Weapon Damage Report provides all ground damage done by a unit, target, or specific missile. If a unit is selected, the user may choose to include or exclude damage done by owned HRUs and owned Targets. The user also has the ability to restrict the report to damage by a specific Targetable Weapon if desired.

2.2 JTLS-2023-16126 Add Location To Unit Hierarchy Report

Summary of Model Change Request

The Unit Hierarchy Report provides an indented list of units within a Command Hierarchy, but no additional unit information is provided on the report. During an exercise, a Response Cell indicated that including the location of each unit within the report would greatly assist them in supporting the exercise.

Design Summary

Unit location was added to the Unit Hierarchy Report to meet this exercise requirement.

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-2023-16041 Satellites Inconsistently Displayed On WHIP

When a new satellite was created in the future, the satellite icon appeared on the WHIP before it was supposed to exist. Similarly, when a satellite reached the last point on its route, it continued to appear at that location indefinitely.

Satellites are meant to only appear on the WHIP when they are actively moving along a route. The code was modified to change the Display on Map, attribute depending on whether or not the satellite was moving. The satellite was made to be displayed when game time reached the intended starting time. Likewise, the satellite display was stopped when it reached the end of its route.

3.2 JTLS-2023-16044 LC2IS Message Service Reporting Attached Units

Inactive units, including units that were attached, were showing up in the SIF file generated by the LC2IS Message Service (LC2MS). Additionally, unique IDs, which should always consist of six digits where the first digit is "1", were being assigned with a first digit of "0" (zero).

The code was modified to exclude inactive units when generating a SIF file. The code was also modified to ensure six digit IDs start with the numeral "1".

3.3 JTLS-2023-16045 Checkpoint Crash Saving Supply Statistics Data

The model crashed attempting to save Supply Usage Data as part of the checkpoint procedure. The category of the used data was NULL.

The issue was linked to the fuel that a MUSE mission used when it was created by MUSE. The code to create the shadowed air mission in JTLS-GO did not assign a Fuel Type, so when it launched it registered the use of fuel without a fuel Supply Category. This has been corrected.

3.4 JTLS-2023-16046 MUSE Air Mission Reports Cannot Be Parsed

When a MUSE air mission completed, JTLS-GO generates a Mission Report (MISREP) for the mission. The MISREP that was generated could not be parsed.

The MUSE mission MISREP was corrected, and some missing information needed for the report was provided.

3.5 JTLS-2023-16047 ELS Crash Determining Grid Location

The ELS model crashed when it computed the grid location for an object outside the bounds of the grid layer.

The true cause of the issue was not found, but the code was patched to ensure the crash situation would not occur. If the situation does occur, more information is now provided to help determine the root cause of the problem,

3.6 JTLS-2023-16048 Mission Refuels Before Tanker Is Close Enough

A mission headed to the tanker's expected refueling orbit area. Once the mission arrived at the orbit area, the mission received fuel, even though the tanker was still miles away moving toward its assigned orbit area.

The problem was fixed. The receiving mission continues to orbit at the tanker's orbit area, waiting for the tanker to arrive.

3.7 JTLS-2023-16049 Supply Shipment Error Does Not Apply To One Side

The Scenario Verification Program (SVP) was generating errors that a Side with only one unit did not own any assets to ship certain supplies. Given there is only one unit on the Side, that unit has no one to which supplies could be shipped. It is a false error and should not be reported.

The code was changed. Sides that have no units or only one unit are now skipped in the transportation availability check.

3.8 JTLS-2023-16050 SVP Find Node For Zero Distance Arc Error

The SVP checks all network arcs to determine if any of them have an extremely small length. Any such arcs should be deleted as unnecessary and computationally inefficient. The SVP Viewer allows the database builder to find the nodes of the small arc on the map. This capability was not working.

The location of the arc's nodes was not being output by the SVP. The code was changed to provide the needed information. The capability now works.

3.9 JTLS-2023-16052 Not Checking All RPMs For Execution Environment

A script exists to check if all required software packages are installed for a JTLS-GO development site or a simpler JTLS-GO exercise site. The packages that are checked are listed in Appendix C of the JTLS-GO Installation Manual. Within that manual, the required packages are broken into three groups representing Executable RPMs, Support RPMs, and Development RPMs, and the first two groups are always required.

However, when the script was run to check for an exercise site, it would only check the first group of RPMs, instead of the first two.

The script was modified to always continue to check the Support RPMs and to exit after the Support RPMs if only the RPMs for an exercise site are required.

3.10 JTLS-2023-16054 GIS Tool Exception Importing Jungle Terrain Grids

The GIS Tool attempted to import the .tl_gd file of a scenario that had terrain grids classified as Jungle. The tool threw an exception, indicating that it did not recognize the terrain type Jungle.

The implementation for importing terrain grids has a check for the terrain type of each grid. It was missing the check for the Jungle terrain type, which caused the exception. This omission has been fixed.

Further examination of the code indicated that the Jungle grids were not being rendered in their proper colors (grids are color-coded based on their terrain type). This error has also been fixed.

Note, the current GIS tool does not produce jungle grids. The default shape files delivered with JTLS-GO do not distinguish between Forest shapes and Jungle shapes. The GIS tool only produced Forest terrain grids, but the Terrain Modification Utility in the DDS allows the user to manually change Forest grids to Jungle grids as desired.

3.11 JTLS-2023-16055 Computing Trucks For Transporting Unit Crash

The CEP crashed while attempting to compute the number of trucks needed to transport a unit by convoy from one location to another.

The model crashed while attempting to divide by zero when determining the number of trucks required to move a unit. In this case, the denominator was the carrying capacity for a single transportation class, which had no capacity for the needed supply category. The code was modified to expect such a situation. In addition, the operation cancellation message no longer reports the number of trucks needed; instead, the message indicates the total amount of excess supplies that could not be lifted.

3.12 JTLS-2023-16056 Crash Destroying Stand Alone Pumping Station

The model crashes if a pumping station that has no association with a pipeline Network is destroyed.

When the model attempts to determine if the pumping station is associated with a Pipeline node, an ill-formed check caused the model to access the target's associated node. There was no associated network or node and the model crashed. Applying proper parenthetical delimiters to the check solved the problem.

3.13 JTLS-2023-16058 Continue Air Intercept Parameter Ignored

A DCA mission was manually paired against an enemy DCA mission. The interceptor's weapons were out of range, but the interceptor was within range of the enemy mission's weapons. The interceptor immediately broke off, citing "UNEQUAL-ODDS" as the reason. The Continue Engage Parameter for the interceptor aircraft type was then reduced to a very small number using the Controller Set Aircraft Data order (by design, a small value is intended to make the interceptor "braver" and less likely to break off). The manual pair order was resent. Again, the interceptor broke off. The reduced Continue Engage Parameter appeared to be ignored by the interceptor.

The problem was in the continue intercept decision logic in the above situation. Because the interceptor's weapons were out of range, the Ph*Pk was zero, and the computed Measure of Effectiveness (MOE) value was also zero. The MOE was always less than the Continue Engage Parameter, causing the interceptor to break off every time. The Continue Engage Parameter therefore had no effect on the decision to continue the intercept.

The JTLS-GO Design Team believes the algorithm is working as intended. One issue was discovered by conducting this investigation. The intercept algorithm now looks at the ROE range of the intercepted mission, and not simply whether the intercepted mission has a positive ROE. This fix allows the intercepting mission to continue intercepting until the next move will result in a distance less than or equal to the intercepted mission’s ROE distance.

A perfect solution to this issue has been implemented in the JTLS-GO 6.2 series. Within that series, the database parameter AC CONTINUE ENGAGE MULTIPLIER can be set to zero, resulting in the intercepting mission being completely fearless and it will continue the intercept under all circumstances.

3.14 JTLS-2023-16060 Warning 1214 Corrections Nonfunctional

SVP Corrections for Warning 1214 for Ship Unit Prototypes do not work.

The SVP Template was using the wrong data items names for Warning 1214. This has been corrected.

3.15 JTLS-2023-16061 ATO Translator Writes Bad Fire Mission Order

The ATO Translator (ATOT) generates several types of orders from a user-provided Air Tasking Order (ATO). When the ATO contains Fire Missions the ATOT encodes the parameters from the ATO, such as firing unit and weapon type, into the Fire Mission Order. However, even though the named weapon is properly mapped in the ATOT to the JTLS-GO weapon name, the weapon type field is empty in all Fire Mission orders generated. This causes the order to fail checking, and the user must manually enter the weapon name in the weapon field.

The weapon field identifier of the Fire Mission order had recently changed, from TARGETABLE_WEAPON to just WEAPON. The ATOT was not updated to properly write the WEAPON field identifier. This has been changed so the Fire Mission order is properly filled using this weapon field.

3.16 JTLS-2023-16062 ATOT Module Error Listing Load Inconsistent

The ATOT WHIP module loads several data files written by the ATOT service. The module did not always reload the mission summary or error listing data files correctly after translating the ATO.

The module's data file handling code has been rewritten to consistently load all data files.

3.17 JTLS-2023-16063 OTHGold Configuration Fields Refuse Spaces

The OTHGold module allows users to configure the message traffic with several text fields, such as Report Command. However, these fields were not consistent in accepting a space character. Some fields would create errors when submitting the request to the server.

The OTHGold module's text-based configuration fields have been made consistent in their handling of spaces.

3.18 JTLS-2023-16065 AAR Database Rejects Surface-Air Engagements

Surface-Air Engagements were not getting filed into the AAR database because the type of the object, AAR_ENGAGEMENT_SURFACE_AIR, was longer than the allowed length of 25 for the type field in the AAR Object Lookup table.

Whenever an object or event record is inserted into the AAR database, a trigger also inserts a record into the AAR Object Lookup table, which serves as the master location to resolve all object JODA Entity Data Identifiers (JEDI) into names. The AAR Object Lookup table contains columns for the checkpoint the object was created, the JEDI of the object, the integral type of the object, the named type of the object, and the object name.

In previous versions of JTLS-GO, the named type and the object name used a shared macro, defined as 30, for the length of the two columns. This length was not long enough for some IMT spreadsheet order names, which would cause an error on the insertion of the order record. The macro was switched from being 30 characters to 60 characters long, to accommodate the large name lengths. In an effort to save space, the macro for the named type was switched to an existing macro that was only defined for 25 characters.

A new macro was introduced for the maximum table name length of 30 characters, and this was assigned to the named type column in the AAR Object Lookup table. Because the named type is actually the name of the table, and because Postgres has a 30 character limit on table names, this should accommodate all objects within the AAR.

3.19 JTLS-2023-16068 ATOT Uses Old Parameter In Multi-Target Utility

The ATOT is writing attack package missions using the multi-target option, but the multi-target utility order is incorrect. The order will not verify.

The definition for the Multi-target utility order was recently updated, to change the name of the utility and one field type. The ATOT was using the older name and field type. The ATOT code was modified to write the proper name and field type for the utility.

3.20 JTLS-2023-16070 RPM Test Script Missing Package Reference

The test program checkJtlsRequiredRPMs, does not report the openssl-devel package when testing for required RPMs for a JTLS-GO development server.

The missing RPM package was not properly included in the required RPM list when testing a development server. This package name has been added to this list.

3.21 JTLS-2023-16071 ATOT Log Error Message

While processing a spreadsheet ATO, the ATOT error log in the Web Services Manager indicated that a breadth was missing for the ACO circle shape. As a result, the ATOT used a default radius of 1 km for the circle ACO.

The spreadsheet ATO parser lacked the capability to process a breadth attribute for circle-shaped ACOs. The breadth attribute existed in the ATOT schema definition file but was not used in the ATO parser. The input ACO spreadsheet did not include a column for the breadth attribute.

The missing code was added to the ATO parser to read a breadth value from the ACO spreadsheet and write the value out in the parsed XML files directory. The ATOT now uses the input breadth value instead of defaulting to 1 km.

3.22 JTLS-2023-16075 Update Warning 1412 And Related Warnings

SVP Warnings 1412, 1288, 1289 and 1290 are not working as originally intended. Additional criteria needs to be added.

The warnings indicated were intended to identify when a unit with a SAM target that had an inherent Fire Control Radar also had the Fire Control Radar built as a separate target, which was unnecessary. The user may have built what appeared to be the extra Fire Control Radar for use as a general-purpose Air Search Radar.

The new criteria look for the same SAM target and to see if it has a matching Sensor built of the same subcategory as the Fire Control radar for the SAM site. If so, then the SVP logic looks to see if there is yet another Air Search radar. If there is, then the JTLS-GO Database Team recommends deleting the possible duplicate Fire Control Radar.

Warning 1412 was removed. Warnings 1488-1490 now become Warnings 1488-1495, based on whether the SAM, Fire Control and Air Search radars are built as POTs or real targets.

3.23 JTLS-2023-16077 Improper Column Heading For POTs

The column header on the TUP and SUP detail page for prototype targets uses the term CCF Number. It should say POT Base Name.

A POT Target's CCF Number is a concatenation of the unit name and the POT Base Name. The change was made for both TUPs and SUPs to distinguish between the two concepts.

The second column header was also changed from "Long Name" to "POT Long Name" for consistency. This was also done for TUPs and SUPs.

3.24 JTLS-2023-16081 GIS Tool Allows Grid Elevation Less Than 5000 Feet

The DDS constrains non-Ocean terrain grid elevations (average and maximum) to be between a negative 5000 feet (-5000) and 32767 feet. A problem occurred when the terrain from a GIS Tool project was exported to JTLS-GO scenario text format and then loaded by the DDS. The DDS found an Open type terrain grid with an average elevation of -6227 feet. The load failed as a result. The tool needs an elevation check like the one implemented for the DDS.

The GIS tool now checks the average elevation and maximum elevation of non-Ocean grids against the allowable elevation range. If the elevation is less than -5000 feet, it is set to -5000. If the elevation is greater than 32767 feet, it is set to 32767.

The model was modified to limit depths to be no deeper than the maximum depth of the deepest depth zone when changed using the order to alter terrain.

3.25 JTLS-2023-16083 Undamaged Aircraft Always Avoid Maintenance

Undamaged aircraft were observed not entering maintenance after completing their missions. This behavior was true for missions that launched only once, and for missions that launched multiple times from alert status or air transport. Alert missions that never launched did not enter maintenance, as expected.

The air mission attribute AM.NUMBER.MISSIONS.FLOWN tracks the number of times a mission takes-off. After the mission is completely finished, the number of take-offs influences the likelihood an aircraft requires maintenance.

Investigation revealed that the AM.NUMBER.MISSIONS.FLOWN was always zero. Therefore, aircraft never entered maintenance upon mission completion, unless damaged. The AM.NUMBER.MISSIONS.FLOWN was not being incremented each time a mission took-off. The missing code was added to correct the problem.

As designed, missions that take off at least once have a probability of requiring “malfunction” maintenance upon completion. The greater the AM.NUMBER.MISSIONS.FLOWN, the greater the likelihood each aircraft will enter malfunction maintenance. Aircraft that avoid malfunction maintenance (due to the randomness of the process) will enter routine maintenance for a much shorter length of time. Again, alert missions that never launch will not enter any type of maintenance.

3.26 JTLS-2023-16085 National Boundary Arcs May Be Too Short

National boundary arcs that are too short may result in computational roundoffs, causing the incorrect conclusion that an object will not cross the boundary.

Changes were made to the code to generate SVP Error 171 if the distance between two national boundary points is too small.

In the CEP, checks were added to ensure that no altered arc segments had zero length in national boundaries and terrain barriers.

3.27 JTLS-2023-16086 Assess Weapon Damage Refused Direct HRU Action

The assess weapon damage order did not allow for direct action by HRU. A direct attack by HRU could not be represented by this order.

An option was added to the assess weapon damage order to support direct action. The radius of effects was set to be that for the selected weapon.

3.28 JTLS-2023-16089 Controller Can Not Set POC Priority

There is no capability to set the Priority for a Political Country in the running simulation.

The option to modify the POC Priority was added to the SET.POLITICAL.COUNTRY.DATA order.

3.29 JTLS-2023-16091 Attack Order Constraint Failure Message Revised

The failure message for the Attack order needed to be revised when the constraint "Capability_Disabled" was invoked.

The failure message was revised to indicate that the issue with the order was the selection of the Group Attack option. This option is still disabled and resources are being allocated to correct the code to re-implement the Group Attack option.

3.30 JTLS-2023-16093 Air Transport Mission Launches Before Loading

A Mobility order was submitted to direct an Air Transport mission to load supplies at its home base. The mission took off and immediately landed before loading the specified supplies. The mission accomplished the loading and continued normally.

This behavior was observed consistently each time an Air Transport mission was ordered to load supplies at its current airbase. A logic flaw caused the mission to take off even though it was already at the correct location to load supplies. Specifically, a Take Off task was always added when the mission was not airborne and in the correct location. The error was corrected by skipping the Take Off task if the currently processing task is a .LAND.TASK, indicating that the mission had just landed.

3.31 JTLS-2023-16096 Air Transport Mission Repeatedly Launches

A Mobility order was submitted to direct an Air Transport mission to unload supplies at an airbase with a Leave Option Duration of three minutes. The Order generated a Ground Wait task after the Offload Supplies task in the Air Mission Tasks IMT as expected. During execution, the mission completed offloading the specified supplies. The mission then took off and landed three consecutive times before starting the Ground Wait task. After the Ground Wait, the mission continued normally.

In the Mobility order, the Transport Instruction specified “Location Option: Land Specific Unit”, which designated the airbase where the mission went to execute the Offload Supplies task. The model created the Offload Supplies task with the location and name of the airbase. The same Transport Instruction also specified “Leave Option: Task Duration”, which required the mission to remain on the ground after completing the Offload Supplies task. The Ground Wait task included the coordinates of the airbase center, but not the name of the airbase itself.

When the mission landed, the model randomly placed it within the airbase radius, which in this specific case was 12 KM. After the offload completed, the model checked the location of the mission to determine if the mission was in the correct location to execute the Ground Wait. Because there was no unit to reference in the Ground Wait task, the model substituted the global MAX OFFSET DISTANCE of 1 km as the radius.

The logic properly determined that the mission was too far away and dutifully created a Takeoff and Move task. When the mission landed, the logic randomly placed the mission near the airbase center, but more than 1 km (plus the mission radius) away from the coordinates in the Ground Wait task. The mission repeated this behavior twice more before the random location selected by the model placed the mission less than 1 km from the airbase center.

To avoid this behavior, the initial processing of the Transport Instruction was changed to include the unit name in the Ground Wait task when the “Location Option: Land Specified Unit” is selected in the Mobility order. This change now provides an actual unit radius to assess the location of the mission relative to the unit's center.

3.32 JTLS-2023-16099 Insert Extract Mission Crash

The CEP crashed while attempting to execute an air wait task during a mobility mission.

The crash was caused when the model tried to take the square root of a negative number. This calculation error occurred because the air mission had an altitude of negative one (-1). This value was incorrectly assigned to the mission when no altitude was specified on the Mobility Mission Order. The CEP logic was corrected to not use the -1 value from the order; instead, the mission will now fly at its database cruise altitude when no altitude is specified on the order.

3.33 JTLS-2023-16102 Mobility Self-Lift Crash

A Mobility order was submitted to a fixed-wing airlift squadron that specified to conduct a self-lift with all of its aircraft. The order specified a destination that did not have an airbase or runway. When the self-lift mission arrived at the destination, the “No Suitable Landing Place” tactical alert was generated and the model crashed.

The model crashed while attempting to find a suitable divert location. The logic assumed the squadron still had a home base assigned, but it did not because the entire squadron had been self-lifted. The squadron's home base was needed to determine if there was an alternate base assigned. The code attempted to reference the null home base and crashed.

The problem was corrected by adding a check for a non-null home base before determining if an alternate base is assigned. If the alternate base is assigned, it is selected as the divert location and the mission flies to that location and lands. If the home base is null, the logic looks for a suitable divert location.

- If found, the mission flies to that location and lands. It may be that the “best” divert location has a runway that is shorter than the aircraft's required landing length in which case a number of aircraft may be destroyed due to crash landing.
- If no suitable landing location is found, then all the aircraft in the mission are lost.

Note that none of this logic, other than the check for a non-null home base, was altered.

3.34 JTLS-2023-16105 SIP Crash For Color CLEAR

The SIP crashed trying to set a Force Side color to CLEAR.

JTLS-GO may have up to 20 sides and each side must have a unique color. There are only 20 significantly different colors defined within JTLS-GO, and the DDS ensures that no color is used twice. Once all 20 Sides are created within a scenario, and all 20 colors used, there was no way to change colors for a Side without duplicating a color already used and the DDS does not allow the database builder to do this. To solve this dilemma, the color CLEAR was added to be used only as a temporary filler when changing Force Side colors.

New Error 179 was added to the SIP to prevent the crash and to inform the user the color CLEAR needs to be changed.

3.35 JTLS-2023-16108 Warning 1422 Not Ignored

Warning 1422 is still created, even when the user has set their preferences to not display that Warning.

SVP warnings are created to inform the user that there is some inconsistency in their data that will not cause the model to crash, but may cause unpredictable results. Because there are many warnings that may not be of significance to the user, the user has the capability to suppress the generation of any or all Warnings.

The code for Warning 1422 did not check to see whether the user had set their preferences to suppress generating that warning. The check was added to the code, and several other warnings that had similar missing checks were corrected as well.

3.36 JTLS-2023-16112 AAR Weapon Damage Report Mismatched Data

The stylesheet for the AAR Weapons Damage Report was listing killed systems using the data for damaged systems and vice versa.

The stylesheets were corrected to print out the proper data parameter

3.37 JTLS-2023-16113 Controller Repair Sinking Ship In Formation Crash

A ship that belonged to a non-moving formation was damaged and started to sink. The Controller submitted a Controller Change Unit order to restore the strength of the sinking ship, which stopped the ship from sinking. When the code attempted to update the ship’s speed, the model crashed.

The problem occurred because the ship's formation had an initial speed of zero, and the model was attempting to restart the formation because the sinking ship had been repaired. The model incorrectly assumed that the formation speed was zero because it had stopped to wait for the sinking ship. In this case, this was a poor assumption - the formation speed was zero, because the formation was not supposed to be moving, and had no assigned tasks.

The logic was improved to determine the reason the formation speed was zero. It now only attempts to update the formation speed and restart its movement task if the formation was stopped because of the sinking ship.

3.38 JTLS-2023-16116 WHIP AAR Orders Incorrect Game Utilities

The WHIP AAR report orders that reference a utility were failing to open the correct utility panel, or in some cases no utility panel at all. The problem occurred because the path to the AAR utilities was not set.

The event that opens an AAR utility from a report order now sets the AAR flag properly, which sets the correct path to the AAR utilities. The checkorder script also now validates duplicated AAR order file names.

3.39 JTLS-2023-16119 Command Authority Report

It was not possible to get a report of the current Command Authority for each unit.

The user may use the new Query Command Authority order to generate a command structure report, showing the primary WHIP for each unit and the type of authority.

3.40 JTLS-2023-16121 ATO Spreadsheet Parser Missing Side Attribute

The ATO Service crashed trying to parse the XML file.

The parsed XML code created by the ATO Spreadsheet Parser did not create the "SIDE" attribute correctly when multiple Sides were contained in a spreadsheet ATO. This problem was fixed.

3.41 JTLS-2023-16124 Missile Object Has Incorrect Weapon Identifier

The JODA CRUISE_MISSILE object was not properly setting the identifier for the type of weapon that was fired.

The model incorrectly formed the JODA Update with the pointer to the weapon, and not the receiver number of the weapon. In order to ensure that the model was properly setting this JODA parameter, a Missile IMT screen was created to look at the data.

3.42 JTLS-2023-16128 WHIP Illegal Chars Attribute

The illegal characters attribute read by the WHIP for the orders did not match the order schema definition. The problem was that the WHIP incorrectly looking for the XML attribute 'illegal-chars', (plural) instead of 'illegal-char' (singular) as is defined in the schema.

The attribute name read in the WHIP as 'illegal-chars' was changed to 'illegal-char', matching the orders schema as it is defined. The capability to specify illegal characters within a WHIP Order text field now works. This corrected capability was used to ensure that Air Mission Call Signs did not include illegal characters such as “-” and “_”.

3.43 JTLS-2023-16132 Bad Mission Report Canceled By Formation Move

A mission report indicated that a mission was canceled due to Reason Code 28, and the message did not know how to interpret this reason code.

The mission was in aircraft delay from a squadron that was on board a ship that belonged to a moving formation. The formation moved, causing the ship to move, causing the squadron to move, causing the model to re-look at whether the delayed missions were still viable. The conclusion was that the aircraft delayed mission was not viable, and it was canceled due to the unrecognized reason code.

The code was changed to recognized the reason, and the format of the Mission Report message definition file was changed to list an appropriate written reason.

3.44 JTLS-2023-16133 AAR DSA Collection Report Not Generated

The DSA Collection Report failed to be generated.

The Glassfish log showed a null pointer exception. The problem only occurs when the Detecting Force Side is “All”. This causes an object called "DcrObjectDynamics" to never to be set. This is fine when the side is “All” but must be instantiated when the side is a Single side or Multiple Sides are specified. The code executed a line referencing DcrObjectDynamics that expected a non-null object, but without checking whether it was null.

A check is now in place before that line is executed. If the object is null, the code now returns immediately to the calling method.

3.45 JTLS-2023-16135 DSA Spreadsheet File Creates Hidden Files

The output files from the DSA Spreadsheet Program (DSP) were created as hidden .dot files in the orders directory.

The DSP program uses the “REFERENCE ID” which is supposed to be the second variable on the second line of the input data. It uses the ID to form a valid and unique file name for this set of DSA orders. If the field is blank, then the file name is created with a leading dot (.) or period character, which in a Linux system gives it the appearance of being a hidden file. It was confirmed with the user that the data had been left blank.

To fix the problem, the program checks for the blank field and, if so, generates a warning message and replaces the blank field with the text “NONAME”.

There is a third field on the same line, which the user had also left blank. This field contains the simulation month and year (APR23) that is needed to correctly format the Date/Time Group field for DSAs that have a cancellation time.

A new warning was added to inform the user of this problem. Because most simulations are run using the current date and time, the program will also substitute the current system month and year for the blank field entry.

3.46 JTLS-2023-16138 DSA And Contamination Area Default Color

The map colors file had no default color defined for Directed Search Areas (DSA) and Contamination Areas. When these areas were created with no assigned color, the areas were being displayed as black.

Default colors for tactical and national DSAs were added to the map colors file for the displayed DSAs outline and the area fill colors. Similarly, default colors were also added to represent nuclear and chemical Contamination Areas.

3.47 JTLS-2023-16141 Errors Generating MUSE Snap Reports

The ELS crashed when the MUSE external model sent an order to the ELS to generate a Snap report. This capability allowed a synthetic UAV, as controlled by the MUSE simulation, to send Fixed Frame Imaging information to the JTLS-GO simulation. The end result of this interaction is the production of an IIR message in JTLS-GO which was initiated by MUSE.

The ELS crashed because it was attempting to insert information into the IIR report for air missions on the ground. Specifically, there were aircraft on the ground, which had been catastrophically killed while they were On-Alert. The ELS code was not properly processing the physical size that the aircraft occupied on the ground.

3.48 JTLS-2023-16144 Warning 1132 SVP Correction Unnecessary

The correction for Warning 1132 to set the DIS Code to 0.0.0.0.0.0.0 is unnecessary.

The code for checking DIS Codes generates Warning 1132 when certain conditions are met and the DIS.CODE is "0.0.0.0.0.0.0". One of the corrections for fixing warning 1132 is to set the DIS.CODE to "0.0.0.0.0.0.0". This is not necessary because that was DIS.CODE was already determined to be that value. The correction to set the DIS.CODE to "0.0.0.0.0.0.0" was removed.

The same problem was also noted for Warning 1159 and corrected.

3.49 JTLS-2023-16147 Error 5000 Correction Nonfunctional

The correction to open the unit table for Error 5000 does not work.

Error 5000 checks the federation compatibility for a unit and subordinate unit. If the Parent unit is anything except JTLS_ONLY, then the subordinate unit cannot be JTLS_ONLY. The correction for Error 5000 should open the unit table for the subordinate unit to allow correction of the Owning Federate type. There was a mismatch in the parameters passed to the SVPR, causing the table not to open.

The SVPR correction was corrected to properly display the table for the subordinate unit. A new correction was added to open the table for the parent unit to allow correction of its Owning Federate as well.

3.50 JTLS-2023-16150 CEP Crash When Ground Mission Damaged

Weapon fire caused some but not all of the aircraft in a ground mission to be killed. The mission was ordered to be on ground alert for an indefinite amount of time. The damage was caused by Controller order using Assess Weapon Damage.

As part of the process to damage the aircraft, the model attempted to take the mission off of alert. This step implied that the event to complete the mission was rescheduled to occur immediately. However, because this mission was on an indefinite alert with no end time, no such mission complete event existed. Logic in the model did not account for this circumstance.

Code modifications were made to generate an event which immediately completed the mission. This halted the alert status and allowed the aircraft on the ground to be properly damaged.

3.51 JTLS-2023-16153 Two Mission Unit Lift Different Destination Crash

A user submitted two Mobility Mission Orders to lift the same unit. The pickup location was identical, but the drop off location was not identical. They were close, but the two locations were in different grids. The model crashed when the last load was picked up and the model was attempting to move the unit icon from its pickup location to its drop off location.

When the first load was delivered, the Airlift's partial unit was placed in the grid associated with that delivering mission's assigned destination. It turns out the second mission picked up the last load for the unit. The model attempted to remove the partial unit of the grid associated with the second mission. This was the wrong grid and the model crashed.

The implemented solution only solves the immediate problem. The model now asks whether the Unit is in the grid from which it should be removed. If not, a Logic Error is generated and the unit is removed from the grid in which it is located. A more complete solution, which will require the delivery locations for a multi-mission Airlift to be identical, will be implemented at a later date.

3.52 JTLS-2023-16156 Primary Unit Incorrect Supply Blast Damage

If a weapon was aimed at a unit, that unit is known as the weapon's primary target object. If the weapon also causes secondary blast damage, the computation tended to kill too many supplies.

The issue was based on the Command Control Prototype (CCP) Supply Type (SUT) Base Density (CCP.SUT.BASE.DENSITY) data array. The computation was not properly using this data. In addition, when computing the overlap between the unit radius and the weapon's lethality radius, the overlap computation was using Meters as the size of the radii, but the distance between the objects was using Kilometers. The end results of these two errors was the weapon was assumed to cover too much of the unit, which caused too much damage. This mismatch was corrected.

The computation for blast damage to Combat Systems properly used the Combat System Base Density data but still had the wrong unit of measure for the distance between the unit circle and the weapon's lethal radius circle centered on the impact location. This problem was also solved.

3.53 JTLS-2023-16159 Airbase Fuel Column Shows Ground Fuel

When JTLS-GO moved to multiple fuel types, the ability to list the fuel at different units became more complicated. This complicated situation resulted in an Airbase Unit updating its available aviation fuel whenever it received a shipment of Ground Fuel.

The JTLS-GO Design Team is considering alternative methods to easily provide the user with available fuel summary information. Until a design can be approved, a more well-defined definition of a unit’s fuel IMT parameter has been implemented. On the IMT Unit Logistics Screen, the fuel listed is different for each basic type of unit, using the following rules:

- Airbase - the sum of all aviation fuel held by the airbase.
- Squadron - the sum of all fuel that the squadron's aircraft can use.
- FARP - the sum of all aviation fuel that can be used by helicopters.
- Ground - the sum of all fuel that can be used by the unit's Combat Systems.
- Support - the sum of all fuel - no matter what it is used for.
- Naval - the sum of all fuel that the Ship can use.

3.54 JTLS-2023-16160 Error 5007 Correction Does Not Work

The corrections for SVP Error 5007 do not work.

Error 5007 is generated when a parent TUP has zero combats systems to support a subordinate HUP or TUP. The correction attempted to correct the problem by updating the combat system record for the parent TUP. The correction did not work, because the parent TUP did not have a combat system record to update.

Error 5007 will still be used to update the number of combat systems if a record exists. New Error 5018 was created to insert a new combat system record when the parent TUP does not have a combat system record.

3.55 JTLS-2023-16162 Canceled Mission Has Trouble Completing

If a mission had a Refuel Task scheduled when the user or the model cancels the mission, the mission may have trouble completing.

The issue was caused by the routine to compute the mission's next refuel location. When a mission is canceled, the model purposefully does not clear the mission's task list. This allows users to change their minds and easily get the mission to resume its original flight plan. The problem was that the cancel mission kept using the Refuel Task as its next refuel location, when in fact the mission's return base, as specified in the first “Mission Complete” task, should have been the next refuel location. The mission kept attempting to get to the next refuel location for no reason, which caused it to keep trying indefinitely. The logic was updated to avoid this situation.

3.56 JTLS-2023-16165 Logic Error Checking Destroyed SSM Tear Down

The model generated a Logic Error when it checked whether an SSM site needed to be torn down. The specific SSM Target was destroyed and had a current status of zero. The Logic Error was generated because the code did not recognize a status code of zero.

It is well documented that a status of zero for an SSM target means it is destroyed. The code simply needed to account for the defined STATUS.DESTROYED., which indicates that nothing needs to be done with the target because the target is already assumed to be torn down. The model was functioning properly, and the inefficiency of generating a Logic error has been removed.

3.57 JTLS-2023-16169 Satellite Move Times Not Updated

As a satellite moved in orbit, the times of its last move and its next move were never updated on the JODA. This resulted in incorrect times on the Situation Report and IMT screen for Satellites.

Code was modified to publish the move times on the JODA each time a satellite moved.

3.58 JTLS-2023-16172 Replenishing Unit Can Enter Infinite Move Loop

When a Naval Unit is told to replenish another ship, the replenishing ship follows the intercept route logic so it continues to travel to the current location of the receiving ship. A naval unit conducting a move to get to the ship that was to be replenished entered an infinite movement loop, moving over and over to the same location.

The problem only occurs if the ship executing the task to replenish another ship cannot find a direct route or a sealane route to get to the ship being replenished. If the model needed to use the gridded terrain algorithm to get to the ship being replenished, the situation occurred.

The issue was that the gridded naval terrain algorithm never placed the final destination route point in the replenishing ship’s route. The correction ensures that this last route point is added to the ship's route.

3.59 JTLS-2023-16174 Logic Error Checking Load Validity

If a user specifies a load for a specific air mission, the model ensures that the user-specified load is valid for the type of mission being flown. This routine generated a Logic Error for a MINE.CLEARING mission. This indicated that the model logic had not properly handled the case for MINE.CLEARING missions.

There are no needed load validity checks for a MINE.CLEARING mission. The mission's mine clearing capabilities are organic to the type of aircraft being flown, and not the Aircraft Load being used. The code was changed so that no checks are required for a MINE.CLEARING mission.

3.60 JTLS-2023-16178 Missing Information For Satellite Detections

Some of the information was missing or categorized incorrectly when the model made detections using satellites. Specifically, when DSAs were used, the DSA name was not associated with the detection. The source of the detections was labeled as "External" instead of "Satellite" for these detections.

The code was modified to assign the proper DSA name to each of the detection items. For satellites, the detection source was changed from “External” to “Satellite”.

3.61 JTLS-2023-16180 Error 5017 Correction Issue

When corrections are made for Error 5017, the record goes blank.

Error 5017 is only created when the “Check Federation” data flag is set and the JTLS-GO default dry or wet units of measure (UOM) are not compatible with the known federated model list of valid values. The Error 5017 correction asked the DDS to retrieve the record that contained the illegal UOM. The problem record was then displayed. When the user fixed the problem, the filter criteria based on the old UOM for the record no longer applied and the record appears to clear.

Because the global table value is a single record, no special selection criteria is needed. The selection criteria was modified to select the record without qualification.

3.62 JTLS-2023-16189 WHIP Remote Shutdown Did Not Allow Users To Exit

Through the Synapse service, a running WHIP or all running WHIPs can be remotely instructed to shut down. Once a running WHIP receives the shutdown instruction, the WHIP exits after some time, during which the WHIP no longer allowed users to exit the WHIP themselves or save their work during the shutdown countdown. Users were prevented from performing anymore tasks on the WHIP because the exit notifier blocked all input from the user.

The exit notifier was changed to allow user input during the shutdown countdown, so users can save their work and exit the WHIP themselves immediately and gracefully.

3.63 JTLS-2023-16191 Limit Orders In Controller WHIP Menus

There were two orders in the Controller menus for the WHIP that are only meant to be used by technical control. These orders caused problems during an exercise when they were used by someone who did not fully understand their consequences.

The orders to Correct Object Tasks and Enter Debug were removed from the menus used by the WHIP for most Controller menus. Only the primary menu for a Controller WHIP contained these orders. This will allow technical control to limit access to these orders in the future.

3.64 JTLS-2023-16192 JODA Debug On-Off Console Command With No Flags

The JODA allows the user to turn a default set of order debug flags on or off without specifying a specific order. However, this may accidentally turn off a desired debug flag.

The JODA source code which handles terminal commands was modified to simply give a warning whenever the debug flag has not been provided as arguments. If the user gives the "order df" or the "order dfoff" commands, with no following flag, the JODA will report back to the console a warning and take no further action.

3.65 JTLS-2023-16194 WHIP Incorrectly Showing Right-Click Context Menu

Right-clicking on a faction with a short name of “CHI” in the IMT table produced a context sensitive menu for a ship called “CHICAGO_US”. Right-click events on string values had two errors. It performed a regular expression find of the matching object name and compared the value with not only object names but also alias names. If the value did not match a map object name but matched an alias name, that object's context menu would incorrectly show.

The event that shows the context sensitive menu for the corresponding selection was changed to get the exact match of the context menu's object name, and to avoid comparing with alias names when attempting to show a context sensitive menu from the IMT table and the Message Browser.

3.66 JTLS-2023-16199 DDSR Synchronize Tool Cannot Update Location

The DDS Repository's Synchronization tool failed to update the location differences of units.

In the DDS, unit location is a pseudo field, that maps into two actual fields in the database. The server did not take that into consideration. The Glassfish server was changed to handle this situation.

3.67 JTLS-2023-16201 DDSR Synchronize Data

Sometimes no data downloaded when trying to load the data in the DDS Repository's Synchronize Table.

The database account was not set up correctly. The source database account has to have Foreign Data Wrapper for synchronization. But the GUI did not capture the error message to inform the user of such error, so it appeared to hang. The code is changed to capture the error message during the download and inform the user in a popup window.

3.68 JTLS-2023-16205 Improve Task Cancellation Message

A Player was unable to cancel a naval unit’s JOIN.FORMATION task. The message from the CEP only indicated that the task had not been canceled. The message should be more informative.

In this case, Message 2205 was created when a user tried to cancel the Primary Task of JOIN.FORMATION for a naval unit. A simple message was returned, indicating the task was not canceled but did not provide any reasoning.

The message was expanded to inform the user that a Formation primary task for a unit cannot be canceled. The unit needs to be removed from the formation and given a new task.

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 6.1.0.0 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. This chapter contains the remaining errors that could be reproduced. As development team assets become available, these remaining known issues will be corrected.

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.

4.1 DDSC – TMU Line Mode Changes Multiple Grids

When using the line mode in the TMU, more grids than the ones the line passes through are changed. This can also cause a warning about trying to change multiple layers to appear.

4.2 DDSC – Multiple Types In DDS History Table

If records for more than one table type are selected in the DDS History table, “History Details” will display details for only one type.

4.3 WHIP - Pipeline Not Shown On IMT

A pipeline being operated by a non-detected unit is not shown in the pipeline IMT.

4.4 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.

4.5 Manage TPFDD Order Recall May Cause Exception

When recalling an Manage TPFDD Order, there can be as many as 17 requests to the JXSR to fill in the various order fields. Some of these requests may fail and cause a WHIP exception. The order can still be used and corrected by fixing any reported errors. The cause of this problem has been identified, but is more difficult to fix than originally planned. The problem has been placed lower on the bug priority list. If the TPFDD order does not receive the results from the 17 different field related JXSR request, the user can easily click the field refresh button to obtain the missing field information.

4.6 Database Target Owning And Associated Units

It is impossible to create a new target in which the owning unit and the associated unit are the same unit. Similarly, attempting to alter either of these fields to the same unit, will cause a trigger failure. The end result is the change will not be reflected in the database. If the owning unit or associated unit of a target are different, or if one of them is null, the trigger failure issue does not exist and the related record will be successfully updated in the database is successful.

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
HET	Heavy Equipment Transporter
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
OVT	Order Verification Tool
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 6.1.0.0 DATABASE CHANGES

No changes were made to the JTLS-GO 6.1 database.

APPENDIX C. VERSION 6.1.0.0 REPOSITORY CHANGES

No changes were made to the JTLS-GO 6.1 repository.