JTLS-GO Document 17

JTLS-GO

Version Description Document

February 2023

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.1.9.0)

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

JTLS-GO 6.1.9.0 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 Development Team Leader
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 6.1.9.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.1.9.0 is a Maintenance delivery for the JTLS-GO 6.1 series of releases.

JTLS-GO 6.1.9.0 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. In addition, the NATO Communications and Information (NCI) Agency requested system corrections to potential cyber-security issues. The corrections to these problems are discussed in Chapter 3.0, along with other Software Trouble Reports (STRs) identified during testing and operational exercises. A minor improvement to the model is summarized in Chapter 2.0.

JTLS-GO 6.1.9.0 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 Director Guide* (JTLS-GO Document 07, Version 6.1.0.0)
- *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)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.1.9.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.1.9.0)
- *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 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 Player Guide* (JTLS-GO Document 12, Version 6.1.9.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.1.9.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.1.9.0)
- *JTLS-GO ATO Services User Guide (*JTLS-GO Document 24, 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 6.1.9.0 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 6.1.9.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.1.9.0 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 6.1.9.0 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 6.1.9.0 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 6.1.9.0 is delivered with the Adoptium™ project Temurin Java Development Kit (JDK) 1.8 Update 362 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 6.1.9.0. 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.

Please note that the instructions for installing PostgreSQL, detailed in Chapter 6 of the JTLS-GO Installation Manual, have been extensively reorganized in this version of JTLS-GO.

- 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 6.1.9.0 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 6.1.9.0 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 6.1.9.0 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 has continued to improve and expand the unclassified data repository, which has been renamed to “repository61.” 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.

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 6.1.9.0.

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-2023-16039 Convert Naval Move Data To JTLS-GO Orders

Summary of Model Change Request

A specific user need to be able to convert naval movement data from a spreadsheet file to standard JTLS-GO Naval Move orders.

Design Summary

The user provided a sample spreadsheet containing the naval movement data to be converted. The new conversion capability was based on the provided sample input data. A new option was added to the User Provided Functions under the Alter Database menu in the SIP, named "Translate Naval Movement Data".

The input file (named NavalMoveData.csv) is placed in the <scenario_name>/alterdata directory. The input file is a space-delimited CSV file that defines movement for ships and formations. Each ship or formation has a Sea Route defined with one or more route points with speeds in knots. The CSV file columns follow the specific sequence depicted in the sample spreadsheet. Coordinates are in decimal latitude and longitude. Times are in standard DTG format.

The output file (named NAVALMOVEORDERS) is generated from the input CSV file in the <scenario_name>/alterdata directory. The user must move the output file to the $JGAME/<scenario_name>/orders directory, where it is accessible to an executing CEP. To submit the orders to the CEP, run the Order Verification Program to verify the orders in the file are error free, and then select the desired orders to send.

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-2022-15947 Minefield Depth Restriction Not Enforced

When a unit or HRU is told to emplace mines at a water location, the depth restrictions of the minefield are not enforced.

When an air mission lays mines, the model properly enforces the depth restriction of the minefield type and the water location. These restrictions were not enforced for minefields laid by units. The routine that checks whether the location is legal for the type of minefield being emplaced is now used under all circumstances.

3.2 JTLS-2022-15949 Alter Orbit Order Crash

An air mission was enroute to an orbit location. An Alter Orbit quick order was sent with a different orbit location specified. The model immediately crashed.

A logic error caused both the Move task (to the old location) and the new Orbit task to be deleted from the mission's task list. Only the Move task and the associated route should have been deleted. The model crashed trying to reference the next task, which was the now deleted Orbit task.

The logic flaw was traced to an incorrect AAR reason code. The reason code was changed, which prevented the Orbit task from being deleted.

3.3 JTLS-2022-15952 Synapse Disallows TRIPP Preferences Files

The Synapse does not allow the TRIPP to save or retrieve preference files in the Synapse root directory if the TRIPP is running without the JODA. The Synapse also denies a valid TRIPP's request to accomplish the necessary file path operations without having a running WHIP by the same name.

Due to increased security improvements, the Synapse employs stricter file path management for requests involving directory paths. Any starting WHIP will notify the Synapse during initialization, via the JODA service, so the Synapse can track the path request coming from the WHIP. A TRIPP does not require the JODA to be running, and does not notify the Synapse on startup.

The Synapse was modified to register a validated starting TRIPP during initialization. This allows the Synapse to track all path requests coming from a starting TRIPP, without requiring a running WHIP of the same name.

3.4 JTLS-2022-15953 Packager Script Needed Update

The Java packager threw an error after a JOBE package was generated. The packager script was not pointing to the correct version of the .jar file in $JTLSHOME/lib/java/lib.

The Java packager script was updated to point to the correct .jar file.

3.5 JTLS-2022-15954 Manage OPAREA Issue

The JTLS-GO 6.1.8.0 Preamble contained an inconsistency that allowed OPAREAs to be deleted using the Manage Operations Area order, while masking flawed code. This caused the Delete option on the Manage Operations Area order in JTLS-GO 6.2.0.0 to be nonfunctional.

The logic that checks the legality of a Modify or Delete request also checks the validity of a specified color change. If no color was specified, the logic concluded (incorrectly) that the specified color was invalid, rather than that no color had been specified in the order. This has been corrected.

The aforementioned error caused the logic to assign an order status that was incorrectly defined in the Preamble as an even value, rather than an odd value; all bad order statuses are assigned odd values in the Preamble. The CEP simply checks for an order status with an odd value to determine whether the order should be rejected or not. In this case, the order was considered valid because the order status was an even number, and the OPAREA was consequently deleted. The Preamble code was corrected.

3.6 JTLS-2022-15964 Order Checker Logic Error

When the order checker checks certain group fields in an order panel, it checks whether a referenced field must have a side_selector attribute. The logic error was that the order checker checked for a side_selector attribute in all referenced fields in the group. However, a side_selector attribute requires only one referenced field in each option. Consequently, the checker generated an error, even though the Manage DSA order was correct.

The order checker now ignores referenced fields for every option that has one side_selector attribute.

3.7 JTLS-2022-15966 ICP SDR Database Configuration Confusing Warning

When a user set up a game scenario, a warning message said that the SDR Database was not properly configured. The warning further stated that the Glassfish was inactive, and suggested the user save the updates and contact the system admin. The message was confusing, especially when the user had not yet set up the SDR database to collect the AAR data.

The ICP only gives warnings about the SDR database not being properly configured on start up when either Glassfish or the SDR Clients (AARC, OEC, SDC) are active. If they are all inactive, the check is not accomplished.

However, the ICP will check for the proper SDR Database configuration if, and only if, Glassfish or one of the SDR Clients are active.

3.8 JTLS-2022-15969 Incorrect Ship Real-World Identifier Checks

The Scenario Verification Program (SVP) failed to catch an error involving a sub-surface vessel given a Real-World Identifier as a Surface Ship

The SVP uses the ship Real-World Identifier to obtain the Platform Type. It then compares the text values of the Platform Type and the Basic Ship Type to determine if there is a mismatch. If the Platform Type returned based on the Real-World Identifier was empty, it returned a zero. This should be a mismatch, but the code was not set up to check for this case.

The code was modified to check for returns of zero and the error is now properly generated.

3.9 JTLS-2022-15970 HUP Target Supply Error

Error 236 needs options to review SSM targetable weapons, and to display all supply categories.

Error 236 was triggered when a SAM/AAA or SSM Target does not have enough supplies to fire at least one round of their targetable weapons. Error 236 was split into Error 236 for SSM and Error 238 for SAM/AAA:

- Error 236 opens the Targetable Weapon table for review and modification, displays the complete Supply Category table for the HUP for review, and displays the HUP Target table to allow Target data modification.
- Error 238 opens the complete Supply Category table for the HUP for review, and displays the HUP Target table to allow Target data modification.

3.10 JTLS-2022-15971 Insecure Transport: Weak SSL Protocol

A Fortify scan of 6.1.8.0 found two instances of a Critical issue called Insecure Transport: Weak SSL Protocol.

Fortify recommended that the generic "TLS" and "SSL" strings used to specify the communication protocol be replaced with the string "TLSv1.2". This has been done.

3.11 JTLS-2022-15972 Synapse WHIP Request Paths After Restart

The Synapse recently added a security feature to disallow requests that contain a directory path that is not approved for typical WHIP requests. After the Synapse is started, WHIPs are normally started, and any request the WHIP makes for a specific directory path will be allowed by the Synapse. However, if the WHIPs are left running while the Synapse is stopped and restarted, the Synapse denies the requests for the typical directory paths until the WHIPs are restarted.

The Synapse gets WHIP registry information from the JODA. When the WHIPs remain running, but the Synapse is restarted, no new registration information comes from the JODA for an already running WHIP. In this case, the Synapse cannot establish a directory path for these WHIPs. The Synapse must also use the WHIP login information it receives when a WHIP routinely and periodically requests data.

The Synapse was modified to use the WHIP login information, in addition to JODA registration data, to establish directory paths specific to each WHIP.

3.12 JTLS-2022-15973 WHIP Order Panels Unnecessary Beeping

Each time the user opened an Order panel that contained one or more of the unit-of-measure (UOM) drop-down lists, these fields would invoke an unnecessary beep. Because the UOM lists are searchable, and to prevent the beeps, searchable lists require that its current list is updated, instead of replacing the list model altogether whenever adding the UOM items.

The drop-down list on the UOM field was changed to update the list of items for the UOM types instead of replacing the list model. The beeps are now avoided.

3.13 JTLS-2022-15974 WHIP/TRIPP Had No Indication Of Full Start

When starting the WHIP, the user had no way of knowing that the application is ready for use and has fully started.

The mouse cursor is now set to the busy cursor to indicate that the WHIP or TRIPP has not completed its full start, and then returns to the standard cursor once it is ready to be used. The busy cursor is set as the WHIP/TRIPP begins its construction and is returned after the Map completes loading.

3.14 JTLS-2022-15977 Email Not Displaying As Attachment

The Message Delivery Program (MDP) allows the Technical Controller to specify messages to be delivered by email to specific clients, and whether the contents of those messages should be the body of the message or be included as an attachment. When “include as an attachment” is selected, the received messages are still showing up as the body rather than as an attachment in Microsoft Outlook.

There was no empty line between the message header information and the body of the message, causing the body of the message to be run straight against the attachment separator. This empty line is a requirement. Once the empty line was added, the messages appeared properly as attachments in Microsoft Outlook, and were no longer prevented from being sent through a mailguard.

3.15 JTLS-2022-15978 New Target Cannot Be Controlled

A Mobile Sensor Site was created and assigned to a moving Unit. The Target showed up on the Command Hierarchy under the Unit, but was grayed out with a status of "Out of Theater". When the Unit stopped moving, the Target status updated to "Turned On", but the Target remained grayed out, preventing any orders from being sent to it.

This would also affect Jammers, Comm Sites, SAM/AAA, SSM, and Bridge Targets, in addition to Sensor Site Targets. Within the model, the Targets are considered active when the Unit stops, but because the WHIP believes they are inactive, no orders can be sent to them.

The code for new Targets was modified so that if the Target is owned, and the owning Unit is in theater, the Target will be assigned a status of "Turned Off" or "Not Prepared" for SSM and Bridges. If the Unit is not moving, this is automatically changed to "Turned On" for emitter Targets later in the creation sequence. If the Unit is moving, the emitter Targets will be "Turned On" when the Unit stops. The only difference is that the Target should be considered active in the JODA, and will not be grayed out.

3.16 JTLS-2022-15979 ROE Message Crash For Unknown Side

The model crashed when the new expanded ROE implementation message was being generated for an ROE order against an unknown Force Side.

The problem existed for both the Unit version and Air Mission version of the new ROE message and has been corrected.

3.17 JTLS-2022-15980 Squadron Decisively Engaged Infinite Loop

When a squadron is decisively engaged with an enemy unit, it is not allowed to launch aircraft. When the situation is encountered, the model can end up in an infinite loop.

When the situation is encountered, the model sets up another check at the next Assess Combat Period. It is possible that this check will occur at exactly the same time as the next Assess Combat Period. The model computes that the next period will happen in 0 seconds and the event to check is scheduled over and over again with no time advancement.

The solution was to ensure the check happened a few seconds after the next Assess Combat Period, not at the same time as the Assess Combat Period.

3.18 JTLS-2022-15982 Target Array Detach Task Saved Improperly

If a unit holds a Detach Task, with a list of specified Targets to detach, when a checkpoint is taken, the checkpoint incorrectly saves the task. Restarting from the checkpoint and eventually executing the task causes a crash.

The tasks target array was being incorrectly saved as a polygon array. The code to save and read the target array of a Detach Task was corrected.

3.19 JTLS-2022-15984 Zero True Distance Network Arc Logic Error

A logic error was generated when the code encountered a road arc where the attribute NA TRUE DISTANCE had a value of zero.

The arc was a road arc that was automatically created by the model when the user emplaced a bridge. The internally generated arc had its attribute NA DISTANCE properly set, but it did not set the attribute NA TRUE DISTANCE. The problem was fixed.

3.20 JTLS-2022-15986 HRU Start Move Does Not Cancel C4I Update Process

There is a background event that will automatically send periodic updates to attached C4I systems for all objects that are not moving. When an object starts to move, the process is supposed to be stopped, and when the object stops, the process is supposed to be restarted. When an HRU stopped, the model attempted to restart this C4I update process but found that the process was already running.

The code was fixed so the background reporting process is now properly canceled each time an HRU starts a move.

3.21 JTLS-2022-15987 Improvements To LC2IS Message Service Layers

A number of minor changes were discovered while importing layers into the Land Command & Control Information Service (LC2IS). The layers are generated by the JTLS LC2IS Message Service (LC2MS):

1. ***A deconstruction of the JTLS-GO SOF symbols into LC2IS SIF tags did not exist, so SOF units did not show the proper symbology on LC2IS.***

2. ***Strength percentages for personnel and major equipment were being output to two significant digits (94.25), and LC2IS expects integer values only (94).***

3. ***The unit radius was being written with a trailing blank in the tag.***

4. ***The JTLS-GO Target Category was being used as the Vehicle Type for JTLS-GO Targets being reported as equipment. Some of these Vehicle Types were not recognized by LC2IS, and other Vehicle Types expect additional tags.***

5. ***A minor error was discovered in the code that would allow name matches for matching substrings, rather than the entire name.***

1. LC2IS documentation was found that specified the proper tags to be used for 2525 SOF symbology and this was employed to expand the symbol deconstruction routine to handle SOF symbols.

2. The output of the unit radius was converted from being a float output without significant digits to an integer output, which removed the trailing blank.

3. Personnel and major equipment strengths are now output as integers, removing the significant digit issue.

4. JTLS-GO Targets are now output with a Vehicle Type of OTHER, which is recognized by LC2IS with the current tags.

5. The substring match issue was solved by ensuring we check for the string termination character within a match.

3.22 JTLS-2022-15988 AAR Orders Names Too Long From IMT

Some orders being generated by the IMT Order Spreadsheet capability end up with reference names that are longer than the expected limit of 25 characters for AARC object names. This prevents the orders from being inserted into the AAR database.

The allowed length of order names was increased to 60 characters, and at the same time the allowed maximum name length for objects was also increased to 60 characters. The second variable is used by the AAR_OBJECT_LOOKUP table, which is automatically filled by triggers when any object is created.

During changing this it was also noted that the checkpoint length for all the tables was set to 25, when our checkpoint names are only ever four characters long. These fields were reduced to four characters to reduce wasted space.

3.23 JTLS-2022-15989 Synapse Email Address Issue Error

The Synapse service was logging an error with the “From” email address as “jtlsdev”. The Message Browser was defaulting the sender to “jtlsdev” whenever the “From” address was not provided by the user. This same issue was seen in the Report Browser.

The hard-coded “From” address was removed, and a dialog message now informs the user that a sender must be first provided before continuing to email Message Browser messages and Report Browser reports.

3.24 JTLS-2022-15990 Picking Up Airlift Load Logic Error

The model generated a logic error, indicating that the same set was being used to compute what combat system should be picked up as part of an airlift load, and once the load was computed

the set was used to reassign crew personnel to the most beneficial systems that were remaining at the unit.

The logic error was self-correcting, but a review of the code was conducted to make sure that the overlap was not causing a problem. Once the load was computed, the set was cleared and then the algorithm to reassign crew to the most beneficial remaining systems was executed.

3.25 JTLS-2022-15991 Crash After Postponing Single-Pass HRU Patrol Task

The shadow game crashed when a Single Route Path Patrol HRU ran into a river that it could not be cross. The task was postponed, but once the crossing issue was solved, the model crashed attempting to restart the Patrol Task.

The crash occurred after a barrier problem was solved and the unit tried to execute the Patrol Task again. Because it was a Single Path Patrol Route, the location of the Patrol Task was back at its starting point. The HRU moved back to the starting point and then tried to execute the Single Path Patrol again. This is where the model crashed, because the Patrol Task no longer had access to the desired Single Path Patrol Route. The HRU held it in its route and when the HRU ran into the river, the HRU's route was cleared.

The solution was two-fold when postponing an HRU's Single Path Patrol:

- Take the HRU's current route, which holds the remainder of the single path and put it back onto the Patrol Task.

- Change the starting location of the Patrol Task to the HRU's current location.

3.26 JTLS-2022-15993 Synthetic Aperture Radar Collection Mode

When a Synthetic Aperture Radar (SAR) detects an object, it must assign a detection source that will be displayed on the WHIP SITREP. This source was not being set properly for a SAR sensor, causing a crash.

The Collection Source for a SAR sensor was set to RADAR, the same as a RADAR sensor.

3.27 JTLS-2022-15996 ICC JTOI Crashes In Initialization Mode

In a previous NATO Integrated Command and Control (ICC) system JTOI update, the JTOI was modified to provide timestamps with all object updates/creates. This works in Running/Update mode of the JTOI, but causes a crash in Initialization mode.

The JTOI's object update/create code was modified to only provide timestamps during Running/Update mode.

3.28 JTLS-2022-15997 Utility Directive Name Pasting Illegal Characters

The Order Utility directives disallow illegal characters from being entered in the Name field. However, copying and then pasting illegal characters was allowed in the directive Name field. Illegal characters in the Name field prevent the save of the Utility directive and causes errors in the Synapse service.

The Name field in the Order Utilities now use the same filter check as the Order fields to prevent illegal characters from being pasted into the Name field.

3.29 JTLS-2022-15998 Covert HRU Inaccurate Detection

The computation used to determine when a covert HRU would be detected used the incorrect location of the HRU. This resulted in covert HRUs not being detected as they should have been.

The location error in the time to detection algorithm was fixed, and several code inefficiencies were corrected at the same time.

3.30 JTLS-2022-16000 Crash Writing Out Assess Damage BDA Report

The model crashed attempting to write out the BDA summary report for an assess weapon damage that caused damage to a squadron and at least one other unit.

For every unit damaged, the BDA report determines what the post-strike Reconnaissance mission can see in the way of visible Combat Systems. If, through the random draws, the squadron's Aircraft systems were not visible, the code did not remove those systems from the list of systems killed in the unit. After the kills for the squadron are reported, the model selects the next unit that had damage. The Aircraft kills were left in the list of systems killed at this second unit. This time through the code, the random draws determine that the post-strike RECCE did see the aircraft.

The crash was caused because the message wanted to print out the type of aircraft that were damaged, but the unit being reported did not have an Aircraft Class attribute. The code was fixed so if the Combat Systems are not visible to the post-strike RECCE for a given unit the Combat System is cleared out of the set. Thus, only systems owned by the owning unit will be reported.

Note that in the past there have been complaints that the BDA Summary Report was providing some false information, saying tanks were killed at a unit that did not own tanks, for example. This would also have been caused by this bug.

3.31 JTLS-2022-16001 Debug Flags Inconsistent Assignment

A user submitted a JODA console command to change the current debug flags in the model. These flags were used to determine which types of debugging information were printed on the CEP terminal. When using the default console command, a specific set of debug flags were

activated and all other debug flags were turned off. If the user had previously customized the flags, these settings were inadvertently lost and replaced with the defaults.

Code changes were made to prevent any inadvertent overrides of the current values of the debug flags. A change was made to the JODA console such that all orders to set debug flags must include a parameter which specifies the type of flag. This eliminated the option for default flag settings.

The CEP code was also modified to always turn on the debug flags for: Executing Events, Controller Air to Air Combat, and Controller Damage Reports.

3.32 JTLS-2022-16002 Spreadsheet Generation Nonfunctional

The generation of slides from a spreadsheet ACO does not work.

An error in the code never considered that there may be either an ATO which may not need an ACO, or there may be an ACO only being used to produce slides for display on the WHIP. The error has been corrected.

3.33 JTLS-2022-16003 Context Menu "Alter Orbit" Item Added

The context-sensitive menus needed to add the “Alter Orbit” item for own Side air missions in player and air menus.

The “Alter Orbit” Order was added to the appropriate context-sensitive menus.

3.34 JTLS-2022-16004 Javadoc Generation Errors

Some Javadoc errors were found. These errors were introduced into the system when the new Glassfish 5 J2EE libs were migrated from Oracle to Eclipse Jakata.

The JavaDoc build script was updated to point to the new Eclipse Jakata libs. A few minor Javadoc typos were also fixed.

3.35 JTLS-2022-16005 ICC JTOI Individual Object Update Fails

The ICC JTOI allows users to modify the ICC Role and ICC Key of squadrons. The value used to identify these squadrons by the JTOI was inconsistent. In some places the name_prefix was used, and in other code locations, the reference_name of the object was used. This caused any data changes to the object to fail.

The value used to identify squadrons by the JTOI object modification screen was updated to consistently use the object's reference_name.

3.36 JTLS-2022-16006 GIS Tool Hangs Trimming Nodes

While testing the Trim Nodes feature of the GIS Tool on road networks with the Maximum Deviation Constraint turned off, the tool became unresponsive once the trimming process started. This did not occur when the constraint was enforced.

Additional testing showed that this problem also occurred with the other types of networks (rail, river, etc).

When the Maximum Deviation Constraint was enforced, a necessary check in the trimming process method was performed, but the check was missing when the constraint was turned off. This caused the method to stay in an infinite loop. The check has been added to the processing method when the Maximum Deviation Constraint is turned off.

3.37 JTLS-2022-16007 Order Group Editor Improved Verification Dialog

When verified, order groups that contain errors will display a pop-up verification dialog. This dialog listed the orders that contain errors. If the cursor hovered over an order name in the list, the related errors for that order were displayed in a tooltip. This is an ineffective way to display this extra error information because the tooltip automatically disappear after a few seconds. The text cannot be copied, and tooltips are considered a somewhat hidden feature of the GUI.

The error information for each order that failed to verify is now displayed in a separate pane when the order is selected. Multiple orders can be selected to display multiple error listings at once.

3.38 JTLS-2022-16009 SIP Crash Writing Lifeboat Name

The Scenario Initialization Program (SIP) crashed while trying to write the name of the LIFE BOAT type for the DOWNED.PILOT HUP.

After determining that a squadron did not have enough life boats for every aircraft, the SIP tried to write the warning message using the name of the first combat system the squadron had that matched the DOWNED.PILOT HUP lifeboat name. The lifeboat type for the DOWNED.PILOT HUP was NULL, so the array being used to obtain the name of possible matching combat systems was empty. It could not find a combat system name for combat system 0.

If the lifeboat name of the DOWNED.PILOT HUP is blank, Warning 1248, which is used to report a squadron does not have enough lifeboats for all aircraft, was modified to account for this circumstance.

Warnings 1462 and 1464 were also modified to provide a better explanation and detail for cases where there is a lifeboat type, but the squadron does not have enough or any of the needed combat systems.

3.39 JTLS-2022-16010 Duplicate Real And POT Targets

The SIP needed the ability to determine if a unit has both real and prototype-owned (POT) Targets that are of the same Target category and Target subcategory. Duplication of items that cover large areas, such as sensor and jammers, could have a significant impact on simulation speed.

It is not unusual for a unit to have two real Targets or two POT Targets of the same type. It may be unusual that a unit would have the same type (category/subcategory) of Target as both a POT Target and a real Target. Warning 1422 was added to fulfill this requirement.

The new Warning compares each real Target against all of its POT Targets to determine if any are of the same Target category/subcategory. If so, the warning is written. The user will be able to list (display) both the real and POT Targets, as well as use the SVP to delete the real Target or all of the matching POT Targets.

3.40 JTLS-2022-16011 Order Group Editor Sending Multiple Orders

Selecting multiple orders in the Order Group Editor's left-hand pane did not Check or Send all selected orders, only the last one.

A temporary and unsaved order group is created to send the selected group of individual orders. If errors are found in any of the orders, the user is notified and prompted using the same method as when checking and sending Order Groups.

3.41 JTLS-2022-16014 Missing Default Aircraft Load

The SIP crashed writing the Default Aircraft Load Name.

The default load name for the aircraft was NULL. The code assumed every aircraft would have a default load, which is not true. The default load name is optional. When the SIP tried to write the name of the load, there was no pointer and the SIP crashed.

The code has been modified to substitute the word NONE when printing the default load name for aircraft that do not have a default load specified.

3.42 JTLS-2022-16015 Duplicate Sensor Impacting Model Performance

Units had a duplication of the fire control sensor for a SAM site and a separate sensor of the same type as the fire control sensor for the SAM site. Checks are needed to inform the user of the situation.

Although there may be reasons to have a separate sensor with air search capability external to the SAM fire control sensor, duplication could have significant impact on model performance.

The following new Warnings were added to aid the user in identifying possible duplication:

- Warnings 1214 when a POT Target includes overlapping air defense fire control sensors, and a POT sensor Target of the same sensor category as the air defense fire control sensor.

- Warning 1288 when a unit has a real-owned SAM Target and a matching POT sensor Target of the same type as the SAM fire control radar.

- Warning 1289 when a unit has a real-owned SAM Target and a matching real sensor Target of the same type as the SAM fire control radar.

- Warning 1290 when a unit has a POT-owned SAM Target and a matching real sensor Target of the same type as the SAM fire control radar.

3.43 JTLS-2022-16017 Adding Order Utilities To Order Group

If an order containing utilities is placed in a group, those utilities are also inserted into the group. However, if those utilities reference other utilities, then the second-level utilities are not moved into the group.

Each utility referenced by an order is now checked for second-level utilities that are referenced by the utility, and adds them to the group the order was moved into.

3.44 JTLS-2022-16018 New Utility Entry Does Not Add After Record

When adding a new record to a Utility directive's table, the new record was not added after the selected row. Instead, the new record was added before the selected row.

New records are now added after the selected row.

3.45 JTLS-2023-16019 PostgreSQL Java Library Vulnerability

The Fortify scan identified a Java library vulnerability with the existing Postgresql .jar file.

The PostgreSQL Java JDBC Library was upgraded from the existing postgresql-42.2.23.jar file to the latest postgresql-42.5.1.jar file.

3.46 JTLS-2023-16020 SVP Error Corrections Nonfunctional

Not all of the corrections for Error 422 worked.

Error 422 reports errors when a ground unit has an SSM that uses water-only weapons. The SSM can be created as an actual real Target or as a POT Target. The corrections' attempt to toggle between real and POT Targets did not always apply, and therefore would not work. The same problem was found for Error 423.

The Errors are now divided into separate errors. The old Error 422 is now Error 461 for POT Targets, and Error 462 for real Targets. The old Error 423 is now Error 463 for POT Targets and 462 for real Targets.

The correction options for all four errors were also enhanced to provide the user more options.

3.47 JTLS-2023-16021 Airlift Estimate Message Missing Unit Name

When a Player requested an airlift estimate, the resulting message did not contain the name of the lifted unit.

The CEP code and associated message were changed to include the name of the lifted unit.

3.48 JTLS-2023-16022 Changing Unit Prototype JODA Error

The controller sent an order to change the prototype for an individual unit. This order produced an error on the JODA. The error reported that the model attempted to modify both perceived and common knowledge attributes in the same packet.

The CEP code was modified to send separate updates for the perceived data and the parameters which contained common knowledge.

3.49 JTLS-2023-16023 Order Definition Files Illegal Characters

A number of order files contained illegal characters. The presence of these characters resulted in Synapse errors when those orders were used.

The parentheses and number sign ("#") characters were used in the labels of some order fields. They were removed and the order definition file was regenerated. This corrected the Synapse errors.

3.50 JTLS-2023-16024 Error 605 Impossible

Error 605 could not be induced to occur.

The JTLS-GO Development Team has an internal database that contains every possible Error the user might receive while building their own database. Our attempts to induce Error 605 failed. A code problem was found and corrected, and the Error can now be induced and will be reported.

3.51 JTLS-2023-16025 Ground Route Ignored Specific Leave Time

A Move order with a specified route was submitted to a ground Unit. The ground route included a route point with a specific leave time. However, the Unit's task list IMT displayed the order submission time instead of the specific leave time in the End Time column. When the Unit arrived at that route point, it immediately continued its next move without regard to the specific leave time submitted in the Move order ground route.

In the routine that processes the Move order, the specific leave time was incorrectly overwritten by the maximum lag distance. The maximum lag distance had a value of -1 because it was not entered in the Move order. In the routine that assigns the end time for the route point, the logic treated the -1 as the current game time.

To correct this problem, the maximum lag distance was assigned to a different, unused field in the Move order routine. This field, in turn, was assigned to the attribute in the routine that creates the ground route task list. The specific leave time was thereby preserved and used in the routine that assigns the end time.

3.52 JTLS-2023-16027 SIP Crash CHECK.BEF.DATA

The SIP crashed in routine chebefdat.sim due to a “Floating Point Error”.

The verify routine was trying to determine the size of a DSA. This particular DSA had only one object, which was to be created in the future. Therefore, the number of objects was zero and, when determining the average latitude of all objects, the SIP tried to divide by zero and crashed.

The code was modified to check for an empty set of objects before continuing with the computations.

3.53 JTLS-2023-16028 Database Error When Table Opened

When the “Target All” table is brought up on the DDS, a database error occurred that said “more than one row returned by a subquery used as an expression”.

The SELECT SQL query statement was refined in the Target screen definition file to return only one result.

3.54 JTLS-2023-16029 Armed Recce Does Not Attack Convoys

An Armed Reconnaissance mission (an OAS mission with the Auto Assign Allowed option set to YES) was launched with a Target Search List of TRUCK_RAIL_TTG. The mission was sent to orbit near an enemy convoy location, as depicted on the Controller WHIP. The mission never fired any weapons at the convoy, despite being within weapon range for fifteen minutes with positive surface ROE. There were no other enemy objects in the vicinity.

This situation generated two logic error messages: “Unrecognized Object Type” and “Attempting to fire on an inappropriate object”.

The mission could not attack a convoy because the convoy object type was omitted from several places in the armed recce search and attack logic. The convoy object type was essentially ignored when detected, which prevented it from being placed in the attack priority list. Since it was not included in the priority list, no weapon could be selected to fire.

Once that error was corrected in the code, it was discovered that the logic that started the firing process failed to use the actual coordinates of the convoy when determining a firing position, which prevented the Fire task from being executed. After that problem was corrected, it was discovered that the subsequent fire missile logic and JDS fire mission initialization code also omitted the convoy object type. Those omissions were also corrected.

Finally, if an ordinary OAS mission (where Auto Assign is not allowed) was tasked to attack coordinates that happened to be near an enemy convoy (within the MAX.OFFSET.DISTANCE), the search mask did not include convoy objects, which prevented detection and attacking of convoys. The omission was corrected.

3.55 JTLS-2023-16031 Add Data Requirements Manual (DRM) Page Link

The title for the Target Type Group DRM entry on the SUP Detail page was not linked.

The title on the SUP detail page is not linked to the DRM entry explaining what the data item is. A code change was made to create the link.

3.56 JTLS-2023-16032 Error 709 Cause Unclear

The cause for Error 709 is not clear.

Error 709 checks to see if the squadron is within the radius of its home base, and then checks further to ensure that units are on the same terrain grid. It attempts to report both of these issues in a single error.

The squadron could be inside the radius of the airbase, but in a different terrain layer grid. The squadron could also be in the same terrain layer grid, but not be within the home base radius. Error 709 has been separated into Errors 705 and 709. Error 705 will report if the squadron is not within the radius of the home base. Error 709 will report if the units are not in the same terrain layer grid.

The user may at first receive Error 705 and, once corrected by moving the squadron, may still receive Error 709 if they are still in different grids.

3.57 JTLS-2023-16033 Land Minefield Clear And Search Value

Land minefields contain values for depth zones other than the surface.

Land minefields can only be placed on the surface which is considered depth zone 1. Values for any other depth zones are not used. New Warnings 1362 for Minefield Clear Capabilities and Warning 1363 for Minefield Search Capabilities have been added.

3.58 JTLS-2023-16034 MSC And MCC Records Out Of Range

The Minefield Search Capability and Minefield Clear Capability tables contain records that are out of range for the Minefield Type.

The MSC and MCC tables contain records for the minefield type and depth zones for a minefield type. The depth zone fields of those records need to be checked against the minefield minimum and maximum depth zones. There should be no MSC or MCC records for a depth zone that is outside the range of the minefield type minimum and maximum depth zones.

New Warnings 1364 (for the MSC) and 1365 (for the MCC) have been added. Each of these new warnings has a correction to delete an individual entry or a group of bad entries.

3.59 JTLS-2023-16035 Crash Generating External Air Mission Remarks

While federated with MUSE, the model crashed when an air mission was completed by the external operator. The MUSE JTLS Interface Module (JIM) was used to control the launching and completion of a UAV mission. The mission was reflected in JTLS-GO as being externally controlled. The crash occurred when JTLS-GO was attempting to compile a final Mission Report.

The CEP was attempting to write a remark which described the type of fuel used by the mission. For external missions owned by MUSE, the fuel type was never specified. The model crashed because the fuel type was undefined. Protection was added to the code to omit this remark for external air missions.

3.60 JTLS-2023-16036 Crash Using Controller Assess Weapon Damage

The model crashed when the Controller attempted to use the Assess Weapon Damage order. This order specified air-to-ground damage from a weapon in a valid aircraft load. The aircraft class was specified, but there were no active missions which used that type of aircraft.

The crash occurred because the model was attempting to set an air mission's attack location for proper reporting in the mission's final Mission Report. The problem was that the assessment was initiated by a Controller Assess Weapon Damage order and there was no mission. The code to set the mission's impact location was skipped under this circumstance.

3.61 JTLS-2023-16037 Magic Move Formation Targets Left Behind

A Magic Move Formation order was submitted to a naval formation consisting of ships that owned various targets (SSMs, sensors, jammers, etc). After the entire formation had relocated, the targets remained at the ships' previous locations. A Naval Move order was then submitted which immediately caused the targets to correctly jump to their owning vessels in their new locations.

This situation did not arise when a formation was moved normally, nor when an individual ship was magic moved or moved normally.

When the Magic Move Formation order was processed, the ships' targets were not removed from the target set held by the terrain grid. After the formation was relocated, the targets could not be placed in the new grid's target set, because they were still in the old target set. Therefore, their coordinates could not be updated. The missing logic was added to remove the targets from the old target set, which allowed the targets to be placed in the new target set and redeployed at new ships' locations.

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 could not be fixed in time to meet the release date requirements. Expect the problem to be solved in the next JTLS-GO 6.1 maintenance release.

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 significant changes were made to the JTLS-GO 6.1 repository.