JTLS-GO Document 17

JTLS-GO

Version Description Document

February 2025

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.2.9.0)

[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 combined joint and coalition environment.

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.2.9.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.2.9.0 is a Maintenance release of the JTLS-GO 6.2 series that includes a repository of standard data, a demonstration scenario based in the western Pacific. A minor user interface functionality improvement is being delivered with this maintenance release and is described in Chapter 2, Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

JTLS-GO Director of Development
Valkyrie Enterprises LLC
1200 Piedmont Avenue, Suite A
Pacific Grove, California 93950
United States
jtlsgo@valkyrie.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.2.9.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.2.9.0 is a Maintenance delivery for the JTLS-GO 6.2 series of releases.

JTLS-GO 6.2.9.0 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 “wespac62”. One minor User Interlace Engineering Change Proposal (ECP) was implemented for this release and is described in Chapter 2.0. Chapter 3.0 describes the software errors that have been fixed since the last release of the JTLS-GO 6.2 series.

JTLS-GO 6.2.9.0 executes on the Red Hat Enterprise Linux Server Version 8.7 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 a documents subdirectory.

1.2.1 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

1.2.2 Unchanged Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.2.8.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 02, Version 6.2.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.2.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.2.7.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.2.5.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.2.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.2.5.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.2.2.0)
- *JTLS-GO JOBE Quick Start Guide* (JTLS-GO Document 11, Version 6.2.2.0)

- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.2.3.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.2.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.2.3.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.2.5.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.2.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.2.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.2.5.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.2.0.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.2.3.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 24, Version 6.2.0.0)

1.2.3 Updated Documents

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.2.9.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.2.9.0)

1.2.4 New Documents

No new documents are delivered with this version of the software.

1.2.5 Delivered Software Components

JTLS-GO 6.2.9.0 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)

- 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 188 and has not been changed as a result of this release.

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

  Apache Server (APACHE)

  JTLS XML Serial Repository (JXSR)

  Order Management Authority (OMA)

  Synchronized Authentication and Preferences Service (SYNAPSE)

  XML Message Service (XMS)

  Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service

- Keyhole Markup Language (KML) Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool
- JTLS Version Conversion Program (VCP)

  VCP60 - Converts a JTLS-GO 5.1 database to a JTLS-GO 6.0 formatted database.

  VCP61 - Converts a JTLS-GO 6.0 database to a JTLS-GO 6.1 formatted database.

  VCP62 - Converts a JTLS-GO 6.1 database to a JTLS-GO 6.2 formatted database.

Instructions for installing JTLS-GO 6.2.9.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.2.9.0 is not necessary. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

The basics of installation have not changed significantly, but due to many Cyber-Security improvements, a new Linux RPM package named “xerces-c”, is required to run JTLS-GO 6.2.2.0 and all later versions. The Synapse will not function without this package. Prior to installing JTLS-GO 6.2.9.0, please run the RPM checking script delivered with JTLS-GO to ensure that this package and all other packages are installed as part of your Linux operating system.

Due to repeated problems at exercises, JTLS-GO Version 6.2.7.0 changed the method the ICP uses to save WHIP passwords. This improvement resulted in a version change to a scenario’s ICP database files. If you did not upgrade to JTLS-GO Version 6.2.7.0, please refer to the complete explanation and instructions on how to alter existing scenarios to use the new ICP database structure and format in Chapter 2 of the *JTLS-GO Version Description Document* for Version 6.2.7.0. This document is delivered with JTLS-GO 6.2.9.0 in the $JTLSHOME/documents directory.

During installation of JTLS-GO 6.2.9.0, users must perform the following steps:

1. **Unload** their scenarios in their current version of JTLS-GO 6.2 (prior to installing JTLS-GO 6.2.9.0).

2. Install JTLS-GO 6.2.9.0, following the instructions in the *JTLS-GO Installation Manual*.

3. Run the **mend-db-rns** script, located in the $JTLSHOME/script directory, for each of their scenarios.

4. **Load** their scenarios, and then immediately **unload** them.

These instructions are due to changes made in JTLS-GO 6.2.5.0, related to random number seeds, as documented in STRs JTLS-2024-16756, JTLS-2024-16758, and JTLS-2025-17069.

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 “repository62”. 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 “wespac62”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.2.9.0 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 8.7, 64-bit architecture.

  JTLS-GO 6.2 has been tested with the following versions of Linux 8:

  > RedHat Linux 8.7 - this operating system license must be purchased.
  >
  > Oracle Linux 8.7 - 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.

- 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 6.2.9.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Server Edition Version 7.9, 8.4, and 8.7

  Oracle Linux 8.4 and 8.7

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

- JTLS-GO 6.2.9.0 is delivered with Apache 2.4.62, the latest available Apache security patch release.

- JTLS-GO 6.2.9.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 442 package, which is equivalent to the current version of OpenJDK.

- 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 11.19 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 6.2.9.0. There are several alternative methods available for obtaining the PostgreSQL 11.19 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.

- 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.

- SIMSCRIPT III (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code.

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

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

- JTLS-GO 6.2.9.0, using the JODA service, allows connections and data exchange with customer client programs. The customer client programs are linked with a set of JTLS-GO-provided API libraries that permit a TCP/IP connection between the JODA and the client program. These API libraries, called JDSP libraries, are built for Linux and Windows and allow customers to built client applications on either of these operating systems. Below are the development environments under which each of the JDSP libraries are built:

  RedHat Linux 8.7 using gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-15.0.2).

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

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 Apache, PostgreSQL IceTea, and Java.

JS/J7 Cybersecurity branch has approved the following update procedure. On a quarterly basis, a maintenance release of JTLS-GO is produced which includes the latest security patches for Java 1.8, PostgreSQL 11, IceTea 1.8, and Apache 2.4. To remain compliant with these approved procedures, users should upgrade and use the latest maintenance release of JTLS-GO.

Contact the U.S. Government Program Manager, Ms. Latara K. Kea-Edwards by email at latara.k.keaedwards.civ@mail.mil to obtain the completed Cybersecurity paperwork and a current Gate completion certificate.

Note: PostgreSQL 11 is no longer supported by the PostgreSQL organization. JTLS-GO 6.2.9.0 is being delivered with the last PostgreSQL 11 security release from November 2023. The JTLS-GO 6.3 series has moved to the newest supported version of PostgreSQL, but due to some major changes in PostgreSQL it is impossible to use this newer version of PostgreSQL with the JTLS-GO 6.2 series. If this situation causes issues gaining local cyber approval to run JTLS-GO 6.2.9.0, an organization has no choice but to move to the JTLS-GO 6.3 series.

As a result of new security requirements built into JTLS-GO 6.2.0.0 as part of ECP JTLS-2022-15976 “Encrypt Passwords To Start WHIP/DDSC”, users must delete **all** pre-JTLS-GO 6.2 scenarios from their $JGAME directory. This will require you to set up your scenarios from a fresh state:

1. Convert your scenarios to JTLS-GO 6.2.0.0 using the Version Conversion Program (see Chapter 13 of the *JTLS-GO DDS User Guide* for instructions).

2. Perform Option 3, “Setup System For A Specific Scenario”, for each scenario.

3. Perform Option 5, “Run Interface Configuration Program”, for each scenario.

This is **not** a requirement if you have already installed JTLS-GO 6.2.0.0 and are upgrading to a JTLS-GO 6.2.n.0 maintenance release.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.2.9.0 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 6.2.0.0 with CAE's GESI model has not been accomplished. CAE should be contacted concerning the continued support of GlobalSim.

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 6.2.9.0 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

Several database structure differences exist between JTLS-GO 6.2 series and the previous JTLS-GO 6.1 series database structure.

To upgrade your JTLS 6.1 scenario to JTLS-GO 6.2 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1, followed by the instructions in Chapter 13 of the *JTLS-GO DDS User Guide.*

> Users should download and re-load their scenarios into PostgreSQL, due to changes made in STR JTLS-2023-16290 PSQL Statement Fails For SVP Warning Correction, included in JTLS-GO 6.2.2.0, which was released in July 2023. This will recreate the related check constraints in the database schema. This procedure only needs to be execute once for every scenario. If the procedure was followed after the release of JTLS-GO 6.2.2.0, it does not need to be followed again,

1.4.1 JTLS-GO Symbol Set

Over the past several years, the Database Team has added and updated the Default Symbol Set used for the delivered scenarios. An organization is not required to use the JTLS-GO Default

Symbol Set, but If there is a desire to do so, the following information is provided to help determine how to use the JTLS-GO Default Symbol Set improvements and changes that have been made.

With the release of JTLS-GO 6.2.5.0, STR JTLS-2024-16648 Bad Symbol ID Code From JSYMS resulted in several changes to the Default Symbol Set. There are two possible situations in which a user organization can currently stand with respect to the use of the JTLS-GO Default Symbol Set. The method that should be used to update a scenario's Symbol Set to the latest Default Symbol Set is included for each situation. Please see the *JTLS-GO 6.2.5.0 Version Description Document* for further information.

1.4.1.1 User Organization Using Pre-JTLS-GO 5.0.0.0 Legacy Default Symbol Set

To change a scenario's Symbol Set under this situation, prior to unloading your JTLS-GO 6.2.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.2.0.0 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.

1.4.1.2 User Organization Using Post-JTLS-GO 6.0 Symbol Set

The symbol 2525 ID Codes, which are used by C4I systems to identify the type of object, were expanded and refined as part of this JTLS-GO 6.2.5.0 release. If you have scenarios based on the JTLS-GO repository symbols, you are encouraged to update the symbol sets for your organization's scenarios. This can be accomplished by executing the following steps after the installation of JTLS-GO 6.2.5.0:

1. Conduct a download of the scenario.

2. Go to the $JDATA/scenario/<scenario_name>/symbols directory by entering the following command:

   ```
   cd $JDATA/scenario/<scenario_name>/symbols
   ```

3. Compare your symbol.scf file to the JTLS 6.2.5.0 version under the repository62 scenario by enter the following command:

   ```
   diff symbol.scf $JDATA/scenario/repository62/symbols/symbol.scf_jtls60
   ```

4. If the names of symbols in the scenario symbol file are identical to the repository symbol file or if your scenario symbols are a subset of the repository62 version, update your symbol file by entering the following command:

   ```
   cp $JDATA/scenario/repository62/symbols/symbol.scf ./symbol.scf
   ```

Your symbol set will now be updated. For safety and to ensure Step 4 is done correctly, do a load of the scenario. Check the log files to ensure there was a clean load

5. If your symbol file has additional symbols not found in the current repository62 version, you can still update your symbol file, but will need to perform a manual integration of the symbol.scf file.

   Bring up the jsyms program for your scenario, by selecting Option 1 “Prepare or Alter a Scenario Database”, and then Option 5 “Configure Symbols”, from the JTLS Menu.

6. Perform a save in the jsyms program.

- If the scenario is loaded on the DDS, select “Yes” to fully update the symbols in the scenario.
- If your scenario is not loaded into the DDS, select “No” to simply update the ASCII version of the scenario’s usable graphics symbol located in the <scenario_Name>.gs file. The next time you perform a load, the most current updated symbols will be loaded from the <scenario_name>.gs file.

7. Exit the jsyms program.

1.4.2 Standard Repository Changes

The JTLS-GO Database Team has continued to improve and expand the unclassified JTLS-GO data repository, named to “repository62”. 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 11.19 required to operate JTLS-GO 6.2.9.0.

2.0 ENGINEERING CHANGE PROPOSALS

The following new model capability was added to JTLS-GO 6.2.9.0 as a result of implementing authorized minor Engineering Change Proposals (ECPs).

2.1 JTLS-2024-16999 Add Long Unit Name To Situation Report

Summary of Model Change Request

The unit short name in the Situation Report (SITREP) is not sufficient to meet exercise requirements. The unit long name needs to be added to the SITREP.

Design Summary

The routine that generates the SITREP was modified to include the UT LONG NAME (up to 40 characters). The message templates were modified to display the unit long name (and the unit short name) in both the English and MTF formats.

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-2024-16966 Split Mission Not Link 16 Assigned

If a Link 16 mission is split, the split-off mission is not properly assigned a Link 16 network.

The split-off mission is now assigned to the same network as the parent mission.

3.2 JTLS-2024-16975 Crash Send Message To Non-Existent Pending Task

The model crashed attempting to send a message about a Controller Kill Aircraft order.

The user made a squadron change Force Sides. The squadron had several active air missions. One of these air missions was on the ground waiting to take off at a future time, as specified by the original order. The logic incorrectly removed all of the tasks from the air mission, causing the mission to be stuck. It had no pending execution tasks (PETs), so nothing could make it take off and fly back home.

The Player became frustrated and killed the air mission using the Controller Kill Aircraft order. When the logic attempted to tell the Player that they lost some aircraft, the model used a routine called SEND TO PET FUNCTIONS, which accessed the variable representing the PET without checking if it existed, causing a crash.

The model now checks if a PET exists and, if not, sends the message to the WHIP responsible for the Ground function of the mission's home squadron.

Note: The cause of the problem was that the model logic does not properly handle missions on alert when the home squadron changes Sides. This problem has not been solved, but will be in future releases.

3.3 JTLS-2024-16982 Ingress With Auto Reverse Egress Crash

The model crashed when an air mission order used the Ingress Path and the reverse for the Egress Path.

A code change was incorrectly moved from the JTLS-GO 6.3 series into the JTLS-GO 6.2 series. The code was restored to its proper version.

3.4 JTLS-2024-16994 Magic Air Ops Reference Limit

The Change Mission Parameter and Manage Air Mission Tasks orders both accommodate a 25-character Reference field length. The Magic Air Ops order is limited to only a 12-character

Reference field length, which prevents pasting of full mission names. For consistency and ease of use, all three orders' Reference fields should be the same length.

The Reference field was increased from 12 to 25 characters in the Magic Air Ops order. The Help text in all three orders was updated to reflect 25 characters instead of 12.

In future versions of JTLS-GO, the Design Team is evaluating the impact of changing all Reference field names to a consistent length.

3.5 JTLS-2024-16996 OPM CAT Terrain Type Non-Visible Values

The Combat Arms Type/Terrain Type Percent Non-visible value is not in the Online Player Manual (OPM).

The OPM code has been modified to display these data.

3.6 JTLS-2024-16997 Remove Flight Path Parameters From OPM

Flight Path parameters should be removed from the OPM. The time required to produce the Flight Path data for the OPM takes too much time, particularly as the data has almost no use to the user during execution.

The link to the Flight Path Parameter OPM page was left intact, but now only displays a message informing the user that the data should be accessed through the DDS or WHIP map display.

3.7 JTLS-2024-17002 Cannot Save AAR Reports To HTML

The user has a choice to save an After Action Report as either raw XML or HTML. The raw XML option works, but the HTML option does not work, and only contains the table and cell borders of the report, without data.

The method used to generate the HTML file did not retrieve the data values to allow them to be printed in the report. A different method was substituted and the report now generates properly in HTML.

3.8 JTLS-2024-17008 Mobility Mission Actions Missing From MISREP

An Interim Mission Report (MISREP) was requested for a Mobility air mission. The report contained an incomplete reference to a transport action and omitted all tasks completed by the mission.

The Interim MISREP is designed primarily for non-Mobility missions. The Air Movement Report is designed specifically to monitor currently executing Mobility missions and details the mission's lift-related tasks. However, it is not unreasonable to expect the MISREP to contain entries for each completed load/offload task in a Mobility mission.

Investigation revealed several related issues that required correction:

- Completion of Unit pickup and delivery (Airlift/Airdrop) was missing from the final MISREP as well as the Interim MISREP.
- Completion of supply loading & offloading (Air Transport) was missing from the Interim MISREP.
- Completion of HRU movement (Insert/Extract) was missing from the Interim MISREP.

All missing mission activities were added to the MISREP in chronological order, including the units/HRUs or supplies involved at each load/offload event.

3.9 JTLS-2024-17019 Converted Scenario Does Not Load

A JTLS-GO 6.2 scenario that was missing some Air Control Prototype - Mission Type records was converted to JTLS-GO 6.3. The resulting scenario could not be loaded into PostgreSQL.

After investigating the problem, it was determined that the same problem would occur if a JTLS-GO 6.1 scenario with missing records is converted to JTLS-GO 6.2 format.

The conversion process created the missing records, but the default data for the data field ACP MT ACCEPTABLE LAUNCH FRACTION was given an illegal value of zero.

The Version Conversion Program was changed to properly default the newly-created records. The Combat Events Program and Scenario Initialization Program were also changed to default the value of the database parameter if any records were missing.

3.10 JTLS-2025-17026 Mobility Mission Delivering Supplies To Target Crash

If a Mobility mission is told to drop off supplies at a location without specifying the unit that should receive the supplies, it is possible that the model will crash.

If there are no units at the drop-off location, but there is a Supply Storage Target at the location, the model crashes attempting to write the information to the Mission Report message. The code was attempting to write out the name of the receiving Target, but assumed that the receiving object was a Unit. The code was fixed to use the name of the correct entity (either a Unit or Target).

To make this change, the routine TRANSLATE DATA TYPE needed to be used. The internal routine documentation was confusing and was rewritten to make it easier to understand. No code was changed in this routine. The routine description comments were simply improved.

3.11 JTLS-2025-17054 JXSR Sends IMT Update With Created Objects

When a Controller opened a Unit Combat System Information Management Tool (IMT) window, the unit's Equipment Item objects (for example, trucks or artillery) were properly listed. When the model updated the information being displayed, the JXSR incorrectly sent the Controller WHIP's IMT publication a Create, rather than an object Update. This caused the IMT to never update the information being displayed. The IMT window had to be closed and a new Combat System IMT opened for the unit to get new updated information.

This STR is likely to be the primary, if not the only, problem with several reported issues concerning IMT screens not properly updating. There were two basic independent code issues causing this problem. These are:

- The CEP correctly sent out Creates for Equipment Item objects when the model was started or a unit was given a new Combat System through player action. The CEP then incorrectly sent out updates for Equipment Items to all Sides. This inconsistency has not been solved. It is the "tip of the iceberg" and will be addressed fully and correctly in a future JTLS-GO version.

- When the JXSR received an Update for an Equipment Item, the JXSR noted that there were some additional Sides on the Update. These sides had never received a Create, so the JXSR changed the Update for the new Sides to a Create. The JXSR incorrectly also sent the Controller a Create, even though the Controller side already knew about the Equipment Item. This problem was solved in the JXSR.

The end result is that the Controller's IMT screen will only receive one Create, and all other Equipment Object changes will go out to the WHIP's IMT as an Update. This solved the IMT display updating problem.

3.12 JTLS-2025-17056 DDSC Cannot Add New Transportation Class

When adding a new Transportation Class in the DDS, a "null" Agility Type caused the add to fail.

The Agility Type field was added to the Transportation Class record's "Add" panel, to force the user to specify an Agility Type when adding a new Transportation Class.

3.13 JTLS-2025-17058 JSYMS Failed To Delete Symbol

When deleting a symbol using the Java Symbol Application (JSYMS), an exception was thrown and the symbol was not deleted.

When deleting a symbol, the exception could be thrown while moving the subsequently-indexed symbols' index numbers up and cause the deletion to fail. This is because the symbols are not indexed alphabetically, so subsequently-indexed symbol's may already have had their indices moved. In this case, the exception will be ignored by JSYMS and the symbol will be deleted.

3.14 JTLS-2025-17061 Interceptor Speed Issues

A Player manually paired a Defensive Counter Air mission against a slower enemy air mission. The Player then ordered the intercepting mission to reduce speed. Later, the Player ordered the intercepting mission to break off. The mission did not resume its normal cruise speed after breaking off.

When the mission was ordered to intercept, its speed did not increase to the aircraft's maximum speed. There was no logic in the code that changed the interceptor's current speed to its maximum speed. This error was corrected.

When the mission broke off, there was no logic to change the interceptor's speed to the aircraft's cruise speed. This error was also corrected.

Finally, the logic regarding how speed should be handled during interceptions was reviewed and improved to actively adjust the interceptor's speed based on its air-to-air weapon ranges and the intercepted mission's air-to-air weapon availability. The following rules were implemented:

1. When the interceptor is beyond its maximum weapon's range, maintain maximum aircraft speed.

2. When the interceptor is between its best air-to-air weapon's maximum and optimal firing ranges, there are two different rules based on the weapon status of the intercepted mission:

   a. If the intercepted mission has no air-to-air weapons, the interceptor maintains maximum aircraft speed. The idea behind this rule is that the interceptor is not in harm's way, and quickly closing the gap is advantageous to the interceptor.

   b. If the intercepted mission has air-to-air weapons, the interceptor's speed is changed to match the intercepted mission's speed. The interceptor will fire under this circumstance. The intent of this rule is to not get any closer than necessary to the intercepted mission. This allows a more capable air mission to not get unnecessarily close to the enemy mission.

3. When the interceptor is within its optimum weapons range, change the interceptor's speed to match the intercepted mission's speed.

3.15 JTLS-2025-17065 Unnecessary Code Removed

A section of code in the alert mission logic was not fulfilling any useful purpose.

The section of code that references the local variable LOOK.DISTANCE in routine CHECK.ALERT.MISSIONS served no useful purpose, and the original intent could not be determined. The section was removed.

3.16 JTLS-2025-17068 Weapon Load IMT Default Size Inadequate

The Air Mission Weapon Load IMT contains seven columns. Three of the columns are not visible when the IMT is opened. Players must scroll or drag the IMT window to the right to reveal the three columns. This is annoying and must be accomplished every time the IMT is opened.

The default Weapon Load IMT width was increased to display all seven columns upon opening. Also, the IMT height was decreased to reduce unneeded white space.

Note that customer sites may choose to alter the initial size settings as they see fit.

3.17 JTLS-2025-17069 Unwanted RNS In Database Scripts

For JTLS-GO 6.3, some new random number seeds (RNS) were added to the scenario database. A total of five named seeds were added to support statistical requirements of the Entity Level Server. These seeds were part of a designed improvement in 6.3. These named seeds, specific to the ELS, were never meant to be part of 6.2.

The unwanted RNS names started with the characters "RNS_ELS_". These seeds were automatically added to scenarios in JTLS-GO 6.2 through SQL scripts. The scripts provided starting values for the seeds used by the ELS. The issue was resolved by removing the seeds from the SQL scripts.

We are also delivering a new script, named **mend-db-rns** and located in the **$JTLSHOME/script** directory, to automatically remove the unwanted seeds from the scenario .rns file.

> This STR affects a user’s existing scenarios, and users must follow the instructions on Page 1-5 of this document, instructing them to unload their scenarios, run the new **mend-db-rns** script, load their scenarios, and then unload them.

4.0 REMAINING ERRORS

Every effort has been made to correct known model errors. All reproducible errors that resulted in CEP catastrophic software failures (crashes) have been corrected. Other corrections were prioritized and completed according to their resource cost-to-benefit relationship.

The following list of issues is known and have not been fixed in time to make it into this release of JTLS-GO 6.2.9.0.

4.1 DDSC/WHIP/JOBE - CADRG Map Zoom

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

4.2 MHE Targets Loading Air Mission Can Cause a Crash

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

4.3 The JTLS-GO Strategic Lift Missions Are Not Working Properly

Strategic Lift Missions, used to move TPFDD assets into the Theater and report the results to a real-world TPFDD processing system, has not been updated to work within JTLS-GO 6.2.

4.4 Tactical Ground Formation Attacks Do Not Work

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

4.5 ATOT Spreadsheet Lacks Detailed Field Checking

The ATOT Spreadsheet Parser has been found to have numerous issues within the Spreadsheet format that are not caught and cause the spreadsheet parser to crash. Fixing the uncovered issues are being worked and should be fixed prior to the next maintenance release of the JTLS-GO 6.2 series.

4.6 Moving Combat System Supplies Can Reduce Unit Strength To Zero

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

4.7 Squadron Changing Side Does Not Handle Alert Missions Properly

If a squadron is told to change sides, the model can leave alert missions at a foreign base stuck and without tasks. These types of missions must be properly handled.

4.8 CEP Incorrectly Sends Updates For Combat System IMT

The CEP properly sends out Creates for Combat System Equipment Item objects (such as trucks or artillery) when the model is started, or when a Unit was given new Combat Systems through Player action. However. the CEP then incorrectly sends out updates for Equipment Items to all Sides. This is inconsistent and the inconsistency needs to be corrected.

APPENDIX A. ABBREVIATIONS AND ACRONYMS

Terms are included in this Appendix to define their usage in JTLS-GO design, functionality, and documentation.

AAA	Anti-Aircraft Artillery
AADC	Area Air Defense Commander
AAL	Air-to-Air Lethality
A/C	Aircraft
ACP	Air Control Prototype
ADA	Air Defense Artillery
AEW	Airborne Early Warning
AFB	Air Force Base
AG	Air-Ground (Air-to-Ground)
AI	Air Interdiction
AIM	Air Intercept Missile
AIREF	Air Refueling
AKL	Area Kill Lethality
AMMO	Ammunition
AO	Area of Operations
AOC	Air Operations Center
APC	Armored Personnel Carrier
ARECCE	Armed Reconnaissance
ARTE	Air Route
ARTY	Artillery
ASC	Automatic Supply Calculation
ASCII	American Standard Code for Information Interchange
ASW	Anti-Submarine Warfare
ATC	Aircraft Target Category
ATGM	Anti-Tank Guided Missile
ATK	Attack
ATO	Air Tasking Order
ATORET	Air Tasking Order Retrieve Program
ATOT	Air Tasking Order Translator
AWACS	Airborne Warning And Control System
AZ	Altitude Zone

BADGE	Bilateral Air Defense Ground Environment (used by Japan Defense Agency)
BAI	Battlefield Air Interdiction
BDA	Battle Damage Assessment
BDE	Brigade
BN	Battalion
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CA	Civil Affairs
CADRG	Compressed ARC Digitized Raster Graphics
CAP	Combat Air Patrol
CAS	Close Air Support
CAT	Category
CCF	Central Control Facility
CCP	Command Control Prototype
CEP	Combat Events Program
CMDR	Commander
COP	Common Operational Picture
CP	Combat Power
CS	Combat System
CSP	Combat System Prototype
CTAPS	Contingency Tactical Air Planning System
CTG	Commander Task Group
CTRL	Control keyboard command
DCA	Defense Counter Air
DCL	Digital Command Language
DDS	Database Development System
DEMSDB	Demonstration Standard Database
DISA	Defense Information Systems Agency
DIV	Division
DMA	Defense Mapping Agency
DoD	Department of Defense
DOS	Days of Supply
DPICM	Dual Purpose Improved Conventional Munitions

DS	Direct Support
DSA	Directed Search Area
DTG	Date Time Group
EC	Electronic Combat
ECM	Electronic Counter Measure
ECP	Engineering Change Proposal
EEI	Essential Elements of Information
ELINT	Electronic Intelligence
ELS	Entity Level Server
EODA	Entity Level JTLS Object Data Authority
ETA	Estimated Time of Arrival
FARP	Forward Arming and Refueling Point
FLP	Fire Lethality Prototype
FLOT	Forward Location of Troops
FOL	Forward Operating Location
FWL	Frederick W. Lanchester (originated a differential equation model of attrition)
GAL	Gallon
GCCS	Global Command and Control System
GRTE	Ground Route
GS	General Support
GSR	General Support Reinforcing
GUI	Graphical User Interface
HARM	High-speed Anti-radiation Missile
HE	High Explosive
HELO	Helicopter
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HQ	Headquarters
HRU	High Resolution Unit
HTML	Hypertext Markup Language
HTT	High resolution unit Target Type
HUP	High resolution Unit Prototype
ICM	Improved Conventional Munitions
ICP	Interface Configuration Program
ICPLogin	Interface Login Program

ID	Identifier
IFF	Identification Friend or Foe
IIP	Intelligence Information Prototype
IMT	Information Management Tool
INFO	Information
INTEL	Intelligence
JCATS	Joint Conflict And Tactical Simulation
JDA	Japan Defense Agency
JDPI	Joint Desired Point of Impact (formerly DMPI: Desired Mean Point of Impact)
JDS	JTLS Data System
JDSP	JTLS Data System Protocol
JEDI	JODA Entity Data Identifier
JMCIS	Joint Maritime Combat Information System
JMEM	Joint Munitions Effectiveness Manuals
JODA	JTLS Object Distribution Authority
JOI	JTLS Operational Interface
JPL	Jet Propulsion Laboratory
JRSG	Joint Rapid Scenario Generation (formerly JIDPS: Joint Integrated Database Preparation System)
JSDF	Japanese Self-Defense Force
JSYMS	Java Symbols application
JTLS	Joint Theater Level Simulation
JTLS-GO	Joint Theater Level Simulation - Global Operations
JTOI	JTLS Transaction Operational Interface
JXSR	JTLS XML Serial Repository
KIA	Killed In Action
KM	Kilometer
KNOTS	Nautical miles per hour
LA	Lethal Area
LAN	Local Area Network
LAT	Latitude
LB	Login Build (JTLS order type)
LDAP	Lightweight Directory Access Protocol
LDT	Lanchester coefficient Development Tool
LOG	Logistics

LOGIN	Logistics Input
LOGREP	Logistics Report
LONG	Longitude
LOTS	Logistics Over The Shore
LR	Long Range
M&S	Modeling and Simulation
MAPP	Modern Aids to Planning Program
MB	Megabyte
MCP	Mobility Counter-mobility Prototype
MCR	Model Change Request
MG	Machine Gun
MHE	Material Handling Equipment
MIP	Model Interface Program
MOGAS	Motor Gasoline
MOPP	Mission-Oriented Protective Posture
MOSAIC	NCSA user interface software
MOTIF	X Window System graphical interface
MP	Maneuver Prototype
MPP	Message Processor Program
MSC	Major Subordinate Command
MSG	Message
MTF	Message Text Formats
MUREP	Munitions Report
MUSE	Multiple Unified Simulation Environment
NCSA	National Center for Supercomputing Applications (University of Illinois)
NEO	Noncombatant Evacuation Operations
NFS	Network File Server
NGO	Non-Governmental Organization
NIS	Network Information Service or Network Information System
NM	Nautical Mile
NTSC	Naval Telecommunications System Center
OAS	Offensive Air Support
OBS	Order of Battle Service (formerly UGU: Unit Generation Utility)
OCA	Offensive Counter-Air

OJCS	Organization of the Joint Chiefs of Staff
OMA	Order Management Authority
ONC	Operational Navigation Chart
OPM	Online Player Manual
OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKI	Public Key Infrastructure
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
SDC	Scenario Data Client

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 6.2.0.0 DATABASE CHANGES

Refer to the JTLS-GO 6.2.0.0 Version Description Document (VDD) for the list of database changes between the JTLS-GO 6.1 series and the JTLS-GO 6.2 series.

APPENDIX C. VERSION 6.2.0.0 REPOSITORY CHANGES

The JTLS-GO Database Team is no longer updating data in the JTLS-GO 6.2 Data Repository. All improvements are only be made in the JTLS-GO 6.3 repository.