JTLS-GO Document 17

# JTLS-GO

# Version Description Document

May 2024

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

## JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
## (JTLS-GO 6.2.5.0)

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

JTLS-GO 6.2.5.0 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, as well as a few minor model functionality improvements implemented as Engineering Change Proposals (ECPs), These ECPs are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

JTLS-GO Director of Development
Valkyrie Enterprises LLC
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940 USA
jtlsgo@valkyrie.com

[Blank Page]

# TABLE of CONTENTS

# 1.0 INTRODUCTION

## 1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.2.5.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.2.5.0 is a Maintenance delivery for the JTLS-GO 6.2 series of releases.

JTLS-GO 6.2.5.0 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”. Detailed descriptions of some minor Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0, and Chapter 3.0 describes the software errors that have been fixed since the last release of the JTLS-GO 6.2 series.

JTLS-GO 6.2.5.0 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 6.2.0.0)
- *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 6.2.3.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.2.0.0)
- *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 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 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 6.2.5.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.2.5.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.2.5.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.2.5.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.2.5.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.2.5.0)

### 1.2.4 New Documents

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

### 1.2.5 Delivered Software Components

JTLS-GO 6.2.5.0 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.

- 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 6.2.5.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.2.5.0 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 6.2.5.0, 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, we have changed the method the ICP uses to save WHIP passwords. This improvement resulted in a version change to a scenario’s ICP database files. Please refer to the complete explanation and instructions on how to alter existing scenarios to use the new ICP database structure and format in JTLS-2024-16644 ICP Manages WHIP Passwords Independently on Page 2-1.

### 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 6.2.5.0 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 6.2.5.0 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 6.2.5.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 412 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 6.2.5.0. 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.

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

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

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

- The JTLS-GO DDS application uses these open source libraries:

  JFreeChart, licensed under a GNU Lesser General Public License (LGPL) by Object Refinery Limited, http://www.object-refinery.com

  JCommon, licensed under LGPL2.1 (GNU Lesser General Public License version 2.1 or later) by Object Refinery Limited, http://www.object-refinery.com

  Commons-math3-3.0.jar, licensed under Apache Software Foundation (Apache License, Version 2.0) http://www.apache.org/licenses/LICENSE-2.0HLA Compliance

- KML Operational Interface (KOI)

  The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth$^{TM}$. 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.

JTLS-GO has moved to Adoptium, a full OpenJDK Java environment with licensing alternations 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 release, JTLS-GO provides a bug release version (JTLS-GO 6.2.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.

Contact the U.S. Government Program Manager, Ms. Jessica Camacho by email at jessica.l.camacho.civ@mail.mil to obtain the completed Cybersecurity paperwork and a current Gate completion certificate.

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 JTLS-GO 6.2.n.0 maintenance release.

### 1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.2.5.0 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 6.2.5.0 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 this release, STR JTLS-2024-16648 Bad Symbol ID Code From JSYMS on Page 3-6 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.

#### 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 6.2.5.0.

# 2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.2.5.0 as a result of implementing authorized minor Engineering Change Proposals (ECPs).

## 2.1 JTLS-2024-16644 ICP Manages WHIP Passwords Independently

### Summary of Model Change Request

Security requirements in a previous version required WHIP users to be able to set their own login password, from an initial default password set by Tech Control, when they first log in. When users change their password, the new password is written directly to the ICP database files.

Tech Control can modify user passwords through the ICP. The ICP is also used to change configurations for Web Services, WHIP menus, and many other settings. The ICP would save all configuration data simultaneously.

An issue arises when the following events occur:

1. Tech Control opens an ICP, and the ICP reads the current WHIP password files.
2. A WHIP user logs in for the first time and updates their password, while the ICP is still open at Tech Control.
3. Tech Control saves the ICP and overwrites the password that was set in Step 2, with the old password that was read in Step 1.

This change aims to fix the issue of Tech Control overwriting passwords set by WHIP users.

### Design Summary

The ICP now manages WHIP password files in unique files for each WHIP. When a WHIP user updates their password, that password is written to a file unique to that WHIP.

This change also allows the ICP to only modify WHIP password files that have been edited during that ICP session. The ICP now tracks which WHIP passwords it has edited. When the ICP saves, it only writes those edited password files.

A scenario’s ICP Database files have been consolidated and moved to the scenario’s: **$JGAME/<game>/config/icp**.

An automatic ICP database conversion capability was not created to support this ECP, because the procedure to support this change is easy to implement manually.

Before using an existing JTLS-GO 6.2 scenario with the JTLS-GO 6.2.5.0 maintenance release, Technical Control should execute one of the following procedures:

### Procedure Option 1: Recreate ICP Database

If possible, the easiest method is to delete the current ICP database and allow the ICP to create the newly-formatted database. This procedure will require that Technical Control recreate all WHIPs associated with a scenario. To implement this procedure, execute the following steps:

1. Stop all WHIPs and processes associated with a scenario. This includes Apache, AAR Glassfish, JODA, JXSRs, OMA, XMS, Dashboard, AARC, JOIs, and Replay JXSRs.

2. Ensure that the ICP is not executing.

3. Go to the $JGAME/<scenario_name>/config directory.

4. Ensure the terminal window is in the proper directory by entering the command: "pwd". The response should be the name of the proper directory.

5. Delete the entire contents of this directory by entering the command: "rm -rf".

6. Start the ICP. This will automatically create the new directory structure under the $JGAME/<scenario_name>/config directory.

7. Redefine all services and WHIPs and save the ICP.

8. Your scenario is ready to restart. This procedure does not require that you do a game start. You can restart the game from any saved checkpoint.

### Procedure Option 2: Manually Edit ICP Database To New Format

If an existing scenario has a large and complicated configuration, with numerous defined WHIPs, it may be easier to manually change the pre-JTLS-GO 6.2.5.0-formatted ICP database files to the new version. To implement this procedure, execute the following steps:

1. Stop all WHIPs and processes associated with a scenario. This includes Apache, AAR Glassfish, JODA, JXSRs, OMA, XMS, Dashboard, AARC, JOIs, and Replay JXSRs.

2. Ensure that the ICP is not executing.

3. Go to the $JGAME/<scenario_name>/config directory.

4. Ensure the terminal window is in the proper directory by entering the command: "pwd". The response should be the name of the proper directory.

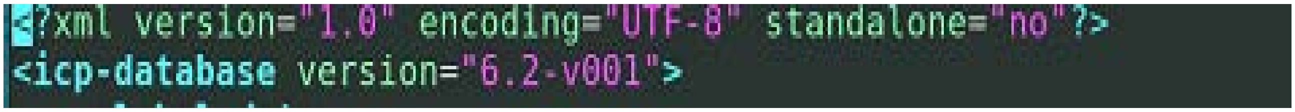
5. Create the new ICP database subdirectory. Enter the command "mkdir icp".

6. Copy the existing two ICP database files into this newly created subdirectory by entering the command: "cp *.db.xml icp". The copy command was selected to ensure the original versions of these files are available as a backup.

7. Move into the newly-created ICP subdirectory by entering the command: "cd icp".

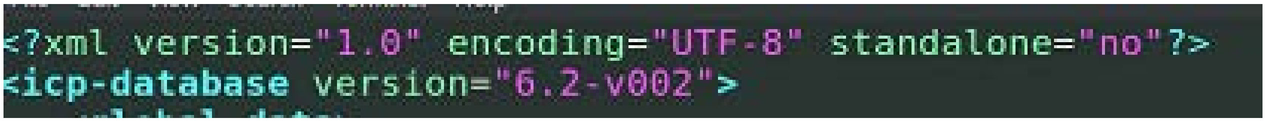
8. The two ICP database files named: <scenario_name>,icp_db.xml and <scenario_name>.whip_icl.xml should exist. This can be confirmed by entering the command: "ls"

9. Edit the <scenario_name>.icp_db.xml file, using any desired text editor. The second line of the file should indicate that the ICP database is written in Version="6.2-v001", as indicated by the following picture:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<icp-database version="6.2-v001">
```

It should be changed to indicate that the ICP database is written in Version="6.2-v002", as indicated by the following picture:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<icp-database version="6.2-v002">
```

10. Make the same change to the <scenario_name>.whip_icl.xml file.

11. Bring up the ICP for the scenario.

12. Using the "Set Default Password Option", create a new Default WHIP password. Do not use a Default password of "NONE". If no WHIP passwords are desired, that option can be selected in the final step of this procedure.

13. Save and stop the ICP.

14. From a terminal window, go to the $JGAME/<scenario_name>/config/icp/whip_pw directory. There should be a separate file for each of the WHIPs defined in the scenario.

15. If no WHIP passwords are desired, reopen the ICP and using the "Set Default WHIP Password" option, set the default password to "NONE". Save the ICP again, and the model is ready to execute with the new ICP database format.

## 2.2 JTLS-2024-16646 Airbase Consumption Report

### Summary of Model Change Request

The user needs information to determine the daily consumption of supplies at airbases, including use of weapons and fuel by air missions whose squadron was using the airbase as a home base.

### Design Summary

A new AAR report was written to collect and provide quantities of supplies used at either a specific set of airbases - which could include FARPs or aircraft-capable Naval Units - or all airbases on a Force Side. The report generates a consolidated report of supply consumption at the selected airbases, and a second version provides a breakdown by airbase and squadron.

# 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-2024-16595 Creating New TW Ruins JODA Weapon Count

**Restarting from a checkpoint in the Shadow game generated a JODA Logic Error.**

The problem was tracked down to the creation of a new SSM Weapon Type by the Controller, which ruined the weapon count in the JTLS-GO Object Distribution Authority (JODA). The problem was fixed.

The Logic Error code was also improved to provide better information concerning the object that had a problem with its weapon count data, if for any reason a similar problem exists in the future.

## 3.2 JTLS-2024-16596 Target Arriving On Formation Status Incorrect

**Targets owned by a unit, which arrived in the game while embarked on a Formation, were still listed as OUT OF GAME.**

**In addition, when the embarked unit is magic moved off of the formation, a Logic Error was generated because the model was attempting to retrieve the target from a grid. Embarked targets are not in a grid.**

Both situations were solved by altering the logic used when a unit arrives on a formation and is magic moved off of a formation.

When a unit was magic moved onto a formation, the unit’s owned target icons were not moved on the WHIP. The targets were moved, but the WHIP was showing them still on land. This has also been corrected.

## 3.3 JTLS-2024-16599 Order Report Allowed Intercepts Incorrect

**When a user requests a report of allowed intercepts, the resulting report is inaccurate.**

The issue was caused by changes made to the order in support of the Super WHIP. The order panel had not been properly changed. This has been corrected.

## 3.4 JTLS-2024-16600 Canceling Amphib Operation With Boat Waves Crash

**Postponing an ongoing Amphibious Assault that still had some Small boat waves going ashore caused a crash.**

The Amphibious Assault was correctly postponed, but the code did not set the SUCCESS.FLAG to YES. The end result was that two move events were scheduled for the formation and eventually caused the model to crash. This has been corrected.

### 3.5 JTLS-2024-16603 Aiming At Coordinates Logic Error

**A Logic Error was generated when processing an attack coordinate task.**

The code used to find the task's associated object failed, as it should have because there is no object associated with the attack coordinate task. Checking for the situation was moved to the proper place in the code to avoid the generation of the Logic Error.

### 3.6 JTLS-2024-16604 Package Mission Had No Rendezvous Point Crash

**The model crashed when an Air Mission thought it was in a package and it had no rendezvous point.**

The issue was very complicated, but it had to do with adding missions to existing Air Mission Packages. Several issues were uncovered and corrected:

- When an orbiting mission is added to a package, it postpones the orbit task and properly places in a rendezvous task, but does not schedule the event to execute the task. The execute task is now scheduled.
- When all of these altering tasks occur, there is a check that the last mission task is a mission complete task. This logic worked but it did not properly delete the auto move task for the mission complete task. This caused the mission to have an incorrect task list, which the user was trying to correct when the error occurred. The logic now properly deletes the auto move task.

### 3.7 JTLS-2024-16606 Formation Name Inconsistency

**A Formation name can be 20 characters long in the database, but the Information Management Tool (IMT) only shows 12 characters, and the order used to create new Formations only allows 9 characters.**

The order and IMT screen were fixed to support the 20 character name limit, established in the Database Development System (DDS).

### 3.8 JTLS-2024-16608 Manage OPAREA Not Suitable For Controller

**Because an Operations Area (OPAREA) holds numerous Rules of Engagement specifically for the owner of the OPAREA, an OPAREA cannot be owned by the Controller side. This means it cannot be entered by a Controller WHIP. It is currently included on several Controller menus.**

The order was changed to a Player-only order and removed from all of the Controller menus.

### 3.9 JTLS-2024-16611 Improve DSA Spreadsheet Location Capability

The Directed Search Area (DSA) Spreadsheet Program (DSP) needs to be able to convert spreadsheet location information into the decimal latitude and longitude format required by JTLS-GO orders. During an exercise, DSA spreadsheets were delivered with several different location formats within the same spreadsheet. The DSP programs needs to be able to handle this type of spreadsheet.

The Reformat Spreadsheet Program (RSP) already had a robust location conversion routine, which allowed for a variety of location formats within the same spreadsheet. The location conversion routine from the RSP was integrated into the DSP.

### 3.10 JTLS-2024-16612 Trusted Air Player Cannot Create ACMs

The Manage Air Control Mean Area order is not available to the trusted air player.

The order was added to the trusted air player's menu. Adding the new order made some of the menus too large. A menu rearrangement was also made to limit each menu to no more than 10 options for ease of use.

### 3.11 JTLS-2024-16613 Logic Error During HRU Combat

A Logic Error is generated when attempting to retrieve the firing HRU from an HRU engagement-generated fire mission.

The routine did not consider an HRU engagement when retrieving the object performing a fire mission. This option is now included in the routine.

### 3.12 JTLS-2024-16615 Logic Error Determining External Model

A Logic Error was generated by the model while it was attempting to determine the owning federate for an air mission. The mission was on the ground in either an alert or delay posture.

The function used to determine the owning federate for an object did not consider the case when it was a ground air mission. This code worked properly for missions in the air, but not for those on the ground. A small change was made to handle these ground missions.

### 3.13 JTLS-2024-16616 Crash While Assessing External Unit Damage

The model crashed when weapons fire caused damage to a unit being controlled by an external simulation. The model was attempting to assess damage to that unit.

The crash occurred because the model incorrectly accessed a variable that had no assigned value. This error in the code was modified to use the right variable name.

### 3.14 JTLS-2024-16617 ELS Crash When New Faction Changed Sides

The Entity Level Server (ELS) model crashed when a new faction was created, and that faction changed Sides in the game.

A unit was deleted in the aggregate model when the faction changed from one Side to another. The ELS code was not properly handling the clearing of object sets which are owned by the unit. The ELS code was corrected to handle the changing of faction Sides.

### 3.15 JTLS-2024-16618 SVP Not Generating Critical DIS Warning

The Scenario Verification Program (SVP) Warning 1132 reports on the absence of DIS code data for objects. Some targets did not have DIS codes, but Warning 1132 was not generated for them in the SVP.

The SVP code was not properly recording the errors associated with targets which had no DIS codes. There was a logic error when keeping track of which objects were utilized by external federates. Warning 1132 was only relevant to those object which were to be used in federations. The SVP now catches and reports all DIS Code issues when accomplishing federation checks.

### 3.16 JTLS-2024-16619 SIP Crash Checking Unit Counts For Federation

When running in a federated mode, the SIP verification process checked the number of units in the hierarchy for each Force Side and crashed. The GESI federate had a strict limit on the number of units allowed per Side.

The SIP crashed while attempting to access an array using a zero array index. This is not allowed. The code was accessing the command level for the higher headquarters of an Air Wing unit. In this case, the HQ was a ship. In this scenario database, ships were assigned zero ("0") as their command level.

The SIP code was modified to skip instances where the HQ is a ship. Naval unit counts were not limited by GESI in the same manner as ground units. This code change resolved the crash.

### 3.17 JTLS-2024-16620 Submerged Submarine Visible To External Sim

While connected to an external simulation, an aircraft spotted a submerged submarine. This submarine was detected while generating a snap intelligence report. The external simulation specified an area on the JTLS-GO terrain where perception of objects was shared. The result of this share was an IIR message or Non-Theater Collection Report.

The submarine should not be detected when it had a depth status of submerged. The sensors on the aircraft were not capable of sensing an underwater vessel. The ELS did not take the depth of the ship into account when reporting it as a detected object. The code was changed to skip counting naval units with a depth status of submerged.

### 3.18 JTLS-2024-16621 Generate Traffic HRU Update Location Crash

**The Generate Traffic order crashes the model. The HRU needs its creation location before activating the HRU. The code called a centralized routine to update the HRU's location, but it crashed because it also updated the HRU's perception data.**

The centralized routine is no longer used. Instead, the location of the HRU is updated and the activation logic properly sets all associated attributes when the Side Information array has been created.

### 3.19 JTLS-2024-16645 Numerous Minor AAR Issues

**Numerous minor After Action Report (AAR) issues were encountered during an exercise. Most had no effect on the functionality of the AAR, just the ease of use.**

- The AAR reports were expanded to allow a name field of 25 characters.
- The Weapon Usage Report was modified to allow a user to select a Targetable Weapon from a list, instead of typing it in.
- Two stylesheets has typos in the retrieval of language specific variables that was causing column headers to show default values. These were corrected.
- Five stylesheets were using a default test template for user preference conversions, instead of the template that is tied to the WHIP preferences. These were corrected.
- A new object class was created to determine common object types, such as whether the object is an aggregate unit or not.
- A stack trace was added to a number of reports that were not generating properly.

### 3.20 JTLS-2024-16647 Load LC2IS Message Service Symbology

**The LC2IS Message Service (LC2MS) uses XML files compatible with the Land C2 Information System (LC2IS) to provide own force updates to LC2IS. In the section for force descriptions, the XML files use unique attributes that relate directly to the symbol used for each unit. The translation from the JTLS-provided APP-6A symbol code for a unit to these attributes was hard-coded within the LC2MS, which prevented it from accepting updates.**

Updates to the LC2IS attributes are provided via an XML file that relates the 2525 numeric hierarchy code to the attribute values. A data file already existed within JTLS-GO that listed the 2525 numeric hierarchy and the ID-Code. This data file also now serves as the repository for the LC2MS symbol defining attributes.

A stylesheet was written to accept a new XML file from LC2IS that defines the hierarchy and attribute values and to merge this information directly into the JTLS 2525 symbol data file.

A library module was written that permits the LC2IS to retrieve the attribute values from this data file when writing output for the LC2MS.

### 3.21 JTLS-2024-16648 Bad Symbol ID Code From JSYMS

Some SDC reports included a unit's 2525 APP-6A Symbol ID-Code. One of these codes was illegitimate. The bad code exists within the repository62 symbols.

The symbol was corrected within the data file as a quick solution, but it did indicate that the Symbol ID-Codes needed to be checked within JSYMS. Because JTLS-GO already has a full XML file of symbol codes that should be used with it, a lookup for JSYMS to check codes against this file was developed.

### 3.22 JTLS-2024-16651 IMT Sheltered Combat Systems In Tunnel

A unit that takes shelter in a tunnel will bring its combat systems with it into the tunnel. Consequently, each of those systems displayed on the IMT should indicate that they are sheltered. However, after a unit moves completely within a tunnel, its systems are sheltered, but the IMT does not reflect the systems as sheltered.

The CEP was properly updating the combat system count when the unit is or is not sheltered, but it was not properly updating this value to the JODA because the CEP was not including the proper perception Sides for the unit.

The CEP routine for updating the actual value of the count of sheltered systems was modified to include the proper perception sides. In this way, the value is updated to the JODA, and thus properly displayed on the IMT.

### 3.23 JTLS-2024-16652 JODA Objects With No Perception Received

The CEP code for setting a perception value for an object had an error. Prior to sending the object from the CEP to the JODA, this error allowed the object's perception to be set with blank perception. Internally, the JODA updated all the values for the object, but could not forward any of the values to any of the connected WHIPs because the perception controls which WHIPs should receive the updates, and in this case the perception was blank.

The JODA processing code was changed to detect and inform the operator of this condition.

### 3.24 JTLS-2024-16654 Improve Help For JXSR Lookup Command

The "Lookup" command used in the JTLS XML Serial Repository (JXSR) console does not provide enough details of the arguments for this command to be useful. The base name for the object used in one of the arguments (for example, "OBJ") is not displayed anywhere in the help for this command.

The Lookup command takes three user-provided arguments: Expression, JODA Entity Data Identifier (JEDI), and WHIP name. The help text was changed to properly describe the use of each argument.

### 3.25 JTLS-2024-16656 Dropping AAR Tables When Connection Exists

**If an existing connection to the scenario_sdr database was previously initialized, it could be reinitialized by mistake and the collected AAR data could be lost.**

The related checks were added to the sdr_action script to prevent reinitializing the scenario_sdr database when there are active connections.

### 3.26 JTLS-2024-16658 Barrier Arc Is Not An Arc

**A logic error was generated by the CEP when it had computed two points as being identical or near-identical.**

New SVP Error 180 was added to identify barrier points that are identical or less than 1 meter apart. Error 181 was also needed to identify arcs created by the points that would cross one another.

### 3.27 JTLS-2024-16660 ATO Not Separating Load Names By Side

**The ATO Parser discovers all named loads from the records of each mission contained in the ATO message. The Parser does list all these loads at the completion of the parse, but does not indicate from which Nationality (Side) the loads originate.**

**The ATO Translator also discovers the names of each load from the ATO XML provided by the Parser. The Translator simply enters all the load names into the linking of each participating Side in the ATO.**

The Parser was modified to list the names of the loads and the associated Side for each load. This will give the user an idea of the loads by Side as they are defined in the currently parsed ATO.

The Translator was modified to update the linked list for each Side contained in the ATO with the names of the loads used by the Side. When the user has properly mapped the originating squadron and aircraft asset to the corresponding JTLS-GO squadron, the Side for the mission can be resolved and the load is entered in the linking list for that Side. In this way the user will only need to match the load to the weapons for the Side on which the load is used.

### 3.28 JTLS-2024-16662 WHIP Sitrep Default Unit-Of-Measure

**Changing units of measure in the WHIP did not change the affected values when copying and pasting from the Sitrep tool. The values remained in the default units of measure.**

The Sitrep was changed to copy and paste values in the displayed format, as it is performed in the IMT screens.

### 3.29 JTLS-2024-16663 WHIP Drawing Tool Did Not Remove Deleted Regions

**After deleting a region, the Geo Region Drawing Tool retained the deleted region's name in the drop-down list of existing regions. The Drawing Tool's last selected item was being retained the Drawing Tool's list of regions.**

When the last selected region no longer exists, its name is no longer included in the Drawing Tool's list of regions.

### 3.30 JTLS-2024-16664 Deleting Geo Regions Actively Being Edited

**When trying to delete a region while it was actively being edited, the region was not properly deleted, and its graphic would remain displayed. This also caused an exception to occur, and the Map had to be closed and restarted so the region was no longer shown.**

The delete now identifies regions that are being actively edited and deletes them on request.

### 3.31 JTLS-2024-16665 ATO Parser Window Not Fully Displayed

**For scenarios with many Sides, the ATO Parser could not fully display and ensure all its options and buttons are available to the user, especially when using a smaller monitor.**

A scroll feature was added to the window frame of the ATO Parser, so all components are available to user when there are many Sides in a scenario, or the monitor is not large enough to display the full window.

### 3.32 JTLS-2024-16666 Civil Affairs Change HRU Perceived Name

**When a unit collects civil affairs intelligence information, the intelligence collected may change the perceived name of the HRU for the owning side.**

Civil Affairs intelligence information that is available about your own units is now rejected.

### 3.33 JTLS-2024-16668 Crash Extracting HRU In Overwatch

**The model crashed when the user attempted to conduct an Air Mission extraction of an HRU that was currently executing an Overwatch task. The HRU was properly picked up from its ground location. After that was done, the model canceled the HRU's Overwatch task and replaced it with a task to place the HRU back on the ground. This meant that the HRU was on the ground and on board the aircraft at the same time.**

The logic was corrected to first cancel the HRU's Overwatch task, and then pick up the HRU from its ground location.

### 3.34 JTLS-2024-16672 Aircraft Divert To Non-Aircraft Capable Ships

When a ship sinks, its onboard squadrons automatically transfer to nearby friendly ships. These aircraft were picking other naval units that are not aircraft-capable and ships that already were at full capacity.

Both issues were fixed. Inappropriate ships are no longer allowed.

### 3.35 JTLS-2024-16673 Road Node Delay Time Change Not Updated

When the Controller changes the delay time for a Road Network Node, the model did not reflect the change.

The model was updated to reflect the change in delay time.

### 3.36 JTLS-2024-16675 HRU Route Not Updated After Changing Tasks

After altering the list of HRU tasks, the HRU's planned route was not updated.

The HRU’s route is now updated after altering its task list.

### 3.37 JTLS-2024-16677 Two Units Pick Up Same Bridge Crash

A user ordered Unit A to retrieve Bridge X. Moments later the user ordered Unit B to retrieve the same Bridge X. This caused the model to crash.

Unit A looked at the bridge and passed all of the retrieval checks. Unit A started the retrieval process at 0640Z and said it would be done at 0710Z. Unit B looked at the same target at 0641Z and passed all of the retrieval checks, so Unit B started the retrieval process at 0641Z and said it would be done at 0711Z.

At 0710Z Unit A retrieved the bridge and put the bridge in its set of owned targets. At 0711Z Unit B attempted to retrieve the bridge, put the bridge in its set of owned targets, and the model crashed because the bridge can only be owned by one unit at a time.

The existing retrieval checks ensured that the bridge was not owned, but never considered that the bridge was in the middle of being retrieved by a different unit. The retrieval checks were expanded to determine if the bridge was already being retrieved.

### 3.38 JTLS-2024-16679 Unit Symbology Updates

A number of minor issues with unit symbology were identified:

A comparison of our symbol ID Codes with a translation file that NATO had been using in the past identified several iD Codes where the NATO translation was better than the translation used by JTLS-GO.

Installations were using an echelon level identifier, rather than the symbol used to denote an installation. For installations, the CEP overwrote the size/mobility code of "H", which denotes an installation.

The following changes were made to properly use Installation symbology:

- The more robust symbol ID Codes used by NATO were incorporated into the repository database.
- Glyphs to identify echelon levels for installations, which is always the same regardless of echelon level, were created and applied to those symbols that are installations. The JSYMS program was modified to recognize the expanded echelon set.
- The CEP was modified to not overwrite the size/mobility code for installations.

### 3.39 JTLS-2024-16680 CEP Crash Treating Egress Route As Unit List

A user entered a ground mining order that provided both a national boundary override list and an egress route for when the task was completed. Within the CEP, egress routes for engineering tasks are stored on the Player order in the same location as unit lists for group moves.

When processing the national boundary list, the CEP incorrectly assumed that it also had a group unit list and loaded an array of unit pointers, which were all null because there were no unit names within the list. These null unit pointers later caused an error when the CEP attempted to access the units.

Only Move and Attack orders perform group or Tactical Ground Formation (TGF) moves, so only these two orders will ever have a group unit list. The code was modified to check if the order being processed is a Move or an Attack before loading an array of unit pointers from the group unit list.

### 3.40 JTLS-2024-16682 Magic Move Unit With Counter Battery Requests

A unit that had fired artillery had been detected and a counter battery barrage was scheduled to fire on it. That unit was then magic moved to a new location. As part of the magic move logic, all counter battery barrages scheduled against the unit are canceled and destroyed. The code to destroy the barrages was not cleaning up the set which holds the responding units, causing a crash.

The code was modified to cycle through all Counter Fire Responders associated with the Request for Fire event, and remove them from the event and destroy them before destroying the event itself.

### 3.41 JTLS-2024-16684 External Air Missions Missing Call Signs

**When running in a federated mode with an external simulation, the externally-controlled air missions did not show the proper Call Signs. Specific Call Signs were given in the ATO for all missions. The external simulation missions did not have any Call Signs assigned to them.**

The External Update order on the WHIP did not allow the external model to change the Call Sign for externally controlled missions. A new order field for the Call Sign was added to this order and the associated CEP code was modified to implement the new value as set by the external model.

### 3.42 JTLS-2024-16685 Naval Formation IMT Missing Foreign Ships

**A formation is allowed to contain ships from multiple Sides. The Naval Formation IMT child table, showing the list of ships in the formation, does not list foreign ships in the formation.**

The problem was corrected by adding a new Quick Retrieval button that allows the user to view foreign ships that belong to the formation.

It was also noted that the context sensitive Formation menu did not work from the Formation IMT screen. This was also fixed.

### 3.43 JTLS-2024-16687 No WHIP Context Menu Without Map Open

**The context-sensitive menu was not being displayed from the IMT and other WHIP components whenever an instance of the WHIP Map was not open. The construction of the menu was attempting to access the active map's filter data. This prevented the menu from being shown.**

A check for an available active map was added, and when there is no map open, the menu's object default name is used instead of attempting to access the name from the map filters. This enables the object's context-sensitive to be displayed when there is no map available to access its filter settings.

### 3.44 JTLS-2024-16688 WHIP Login Error Message Not Informative

**When attempting to log into a WHIP that is already logged in, the error message is generated. This error message, shown in the login window, did not inform users of the cause of the problem.**

The ability to read error messages provided by the Synapse service was added. This provides the user more information about the problem, such as that a certain WHIP is already logged in and is no longer available.

### 3.45 JTLS-2024-16689 DDSC Targetable Weapon Field Label Wrong

**The field label “Communications Link Capable” in the DDS Targetable Weapon table was not matching the field definition in the SIP and CEP.**

The field label was replaced with “Re-targeting Allowed” to match the field definition in SIP and CEP.

### 3.46 JTLS-2024-16690 Logic Error Protection For Empty Formation

**An error, which was properly corrected in JTLS-GO 6.2.4.0, inadvertently emptied out a formation so the Formation was active, but it had no ships. A user did not move in the new JTLS-GO 6.2.4.0 code, and the crash happened again.**

To ensure this does not happen again, a Logic Error was created in the Amphibious Operation order to also catch the problem, even though there is no way known to recreate the situation. The order is rejected if the ordered formation was empty and did not contain any ships.

### 3.47 JTLS-2024-16693 Patch Protection For Zero Idle Launchers

**The model was attempting to set up SSM targets to be ready to launch, and came across an SSM target that needed to be set up, but that had a count of Idle Launchers equal to or less than zero. This will cause the model to crash. There is no known way to recreate the situation that caused the problem.**

Because the problem could not be recreated, a Logic Error was added to the routine to catch the problem and stop the crash from happening in the future. If the Logic Error is encountered again, it may be possible to determine the cause of the illogical situation.

### 3.48 JTLS-2024-16694 Convoy Next Move Combat Crash

**Before a convoy moves into the next grid, the model determines if the next move will encounter enemy forces. If the convoy was heading to emplace a supply cache behind enemy lines, the model would crash. The code was always expecting that the convoy was heading to a unit.**

The code now catches the situation and properly makes the decision based on the situation.

### 3.49 JTLS-2024-16697 Fix Add Mobility Task Crash

**A Player was adding lifting tasks to an air mission. The logic creates a temporary air mission to test if the tasks make sense. In this case, the tasks did not make sense and the model attempted to notify both the Player who sent the order and the Player responsible for the air mission’s home squadron, causing a crash.**

The problem was that the home squadron had not been set in the temporary air mission. The home squadron attribute is now set for the temporary air mission

Because the routine in which the model crashed is only called during the order entry process, only the Player that sent the order should be notified. The Player responsible for the mission’s squadron should not be involved.

A logic error was added to the routine that was attempting to find the pointer to the Home Squadron's air function. This should catch the situation in the future.

### 3.50 JTLS-2024-16698 HRU Overwatch Does Not Help On Raid

A Blue HRU was put into Overwatch for a Blue target, and a Red HRU was told to attack the Blue Target. An HRU-to-HRU fight did not occur. The Red HRU killed the Blue Target immediately.

The problem is that, during a "Raid/Attack" task, the model enters the attacking HRU at the Firefight stage, and the overwatching HRU is only allowed to engage during the Ambush phase.

The code was changed to begin in the Ambush phase. During this situation, there is now an HRU-to-HRU fight, with damage, and the Red HRU ran away without hitting the Blue Target.

### 3.51 JTLS-2024-16700 Negative Due In After Postponing Replenish Unit

A negative number occurred in the supplies Due In after canceling a Replenish Unit Task.

Unit A was told to Replenish Unit B with 1 Supply Category. Unit A was then given another Replenish Unit B order, but this time with an execution time of "NOW".

This caused the first Replenish Unit task to be postponed halfway through processing. The model gave Unit B half of the ordered supplies, and left the first task with the remaining 50% of the supplies. The second Replenish Unit order was placed on top of the task list, and the postponed Replenish Unit task was now second on the list.

When the postponed Replenish Unit task executes, the Due In value for the receiving unit is correctly set to the remaining supplies, but once the task is complete, the full amount that was originally in the task is subtracted from Due In, thus leading to the negative number.

The logic was corrected to subtract only the remaining supplies when completing a postponed Replenish Unit Order.

### 3.52 JTLS-2024-16709 Non-Support Unit Creates Convoy

It was possible for a squadron to create a convoy when the user submitted a Directed Resupply order with multiple stops. Any unit can carry out a single task Directed Resupply order. Support units will create a convoy, but non-Support units will create an implicit convoy to accomplish the tasks. Only Support Units should be allowed to create a multi-stop Directed Resupply and this should be accomplished as a convoy

The order panel was changed to ensure that only Support Units are given a multi-stop Directed Resupply task.

### 3.53 JTLS-2024-16712 Crash Updating ADA Target Ranges

The CEP crashed while trying to set new ADA altitude zones.

There was an error in the code used to report the change to the JODA. Memory for the altitude zone array in the JODA protocol messages was never reserved, so the crash occurred when accessing it. The code has been corrected.

### 3.54 JTLS-2024-16714 SAM Site Fires When Turned Off

An air mission was shot down by an ADA site which was turned off.

The ADA site was included in the set of SAM/AAA sites that are checked by the model for firing on the air mission because it was part of an IADS network. Even when the site is turned off, it should be allowed to fire, as long as its Fire Control radar is not needed for firing. This is controlled by the database parameter AD UNGUIDED PK MODIFIER. If greater than zero, the site should be able to fire even when turned off if IADS feeds the site with detection information.

The problem was the site that fired had an AD UNGUIDED PK MODIFIER of zero. An Air Defense Site that has its Fire Control sensor alive, but turned off, should not fire, even if it receives detection information from the IADS network.

The logic error was corrected.

### 3.55 JTLS-2024-16717 Logic Error Postponing Naval Shadow Task

A Logic Error was generated when postponing a Naval Shadow Task.

The task was postponed correctly, but the flag indicating that the postponement was successful was not set. This caused the Logic Error to be produced.

The flag is now properly set and the Logic Error is no longer generated.

### 3.56 JTLS-2024-16718 Crash Attempting To Execute Next Naval Task

The CEP crashed attempting to execute the next scheduled naval task in a Naval Move order, when the previous task was destroyed.

In the Naval Move order logic for a “Now” task, there is a call to postpone the current task. In this case, the current task was to Change Station, and this task was canceled and destroyed when postponing the current task. As a result, the local variable FIRST.TASK was a destroyed entity while processing the Naval Move.

The code was fixed by retrieving the first task again after the call to postpone the current task.

### 3.57 JTLS-2024-16721 Pass Intelligence For HRU Strength Fails

**The Manage Pass Intelligence order included an option to pass the strength of a unit, but this order did not appear to work for sending the strength of an HRU.**

The code was modified to add the passing of HRU current strength.

Note that these code changes only work when passing ground truth data. The current strength of HRU is not included in the perceived information for other Side info.

### 3.58 JTLS-2024-16722 Crash Computing HRU Radius

**The model crashed trying to compute the radius of an HRU.**

The HRU had a negative number of available combat systems, which caused the model to attempt to compute the square root of a negative number, causing a crash.

The cause of the negative combat system count could not be identified. Protection code was added to check for the situation and use the HUP Radius instead of computing the value. A Logic Error is now produced if the situation is encountered again. If the Logic Error is triggered again, the cause of the negative Combat System situation can possibly be identified.

### 3.59 JTLS-2024-16727 Convoys Ignoring Mandatory Pickup Rules

**When using the Directed Resupply order with multiple stops, the MANDATORY flag does not work. A multiple point delivery can never move needed supplies from one unit to another.**

The Mandatory flag was not properly saved on the Task. The information was saved, which corrected the problem.

### 3.60 JTLS-2024-16729 Identify Sheltered Supply Crash

**A user was moving Combat System supplies between units, rather than moving Combat Systems themselves. This caused numerous model issues.**

**One such issue was discovered when computing what supplies should be placed in Equipment Shelters along with the Combat Systems placed in the Shelter. The model crashed during this computation.**

There is a deeper problem with this entire situation that will take several weeks of effort to properly fix. These issues will be addressed in future maintenance releases. In the meantime, the crash is averted by adding code to catch the situation and avoid the crash. Users should use the more common method of moving Combat Systems from one unit to another. The preferred method is to detach a portion of one unit and attach the detached unit to the second unit.

### 3.61 JTLS-2024-16733 Crash When Rail Convoy Damaged

**The model crashed when collecting unit statistics concerning damage to Transportation Classes from a damaged rail convoy.**

The model correctly determined that unit statistics cannot be collected on rail convoys because rail assets are not owned by the unit. A misplaced code statement caused the crash. The problem was corrected.

### 3.62 JTLS-2024-16736 Runway Repair Started Without Supplies

**A runway repair started when the owning airbase had no Class IV supplies. This caused the Class IV supplies to go negative.**

A check was added to avoid this problem. A runway repair will not start if the owning unit has no Class IV supplies.

### 3.63 JTLS-2024-16738 Detachment Created Unit Reusing Name

**A crash occurred when attempting to create an intercepted message concerning a change to an HRU's task list.**

The cause of the problem was that there was an Aggregate unit and an HRU with exactly the same name. The HRU was created first, and the Aggregate Unit was created using a Detach Unit order.

The Detach Unit logic now checks to ensure that the provided name is not the same as an HRU.

### 3.64 JTLS-2024-16740 Detailed Lanchester Battle Report Record Size

**The Lanchester Battle detailed report is difficult to read because each line printed out is broken into two lines.**

When the names of units and combat systems were increased, the impact on the format of the Lanchester Battle Damage Report was not considered. The size of each record was increased from 132 characters to 200 characters.

### 3.65 JTLS-2024-16741 WHIP UIC Alias Revealed On Foreign Units

**A unit's display name can be changed to the unit's UIC in the WHIP's symbol filters. This also displayed and revealed the actual UIC for an unidentified foreign ship. The name was revealed on the map, the object's context menu, and the popup menu list displayed for grouped objects.**

The ability to display the UIC and the other symbol aliases was corrected to display these names only if the unit is owned by the WHIP side, or when the WHIP has Controller capability.

### 3.66 JTLS-2024-16742 WHIPs Assigned Unavailable ID

**A few instances occurred where a new WHIP was created during a running game, but the user was unable to start the WHIP. This was traced to a non-unique Peer ID being assigned by the CEP to the WHIP. This can happen because the CEP always assigns a new WHIP a Peer ID of the number of entries in the set of active model interface programs (MIPs). The CEP assigns all MIPs (including WHIPs and AAR Clients) a Peer ID as they are created.**

**However, when a checkpoint is taken, the CEP saves all of the WHIPs, but does not save all of the other MIPs. When you restart a game from a checkpoint that had a running AARC, the CEP will count the number of MIPs that exist and attempt to assign the AARC the total number of MIPs in the set of active MIPs as a Peer ID, which can conflict with the Peer ID of an active WHIP.**

To explain the issue, imagine a game that is started with ten WHIPs and one AARC. The WHIPs will have Peer IDs of 1-10 and the AARC will have Peer ID 11. If two new WHIPs are created, the CEP is immediately notified of them and assigns Peer IDs 12 and 13. Now a checkpoint is taken and the 12 WHIPs are saved along with their PEER IDs (1-10 and 12-13), but the AARC is not.

When the game is restarted from that checkpoint, the existing 12 WHIPs are read in with Peer IDs of 1-10 and 12-13. When the AARC starts, the CEP will be informed of the new connection, file it in the MIP Set, and assign it a Peer ID of the number in the set, which is 13, but that Peer ID is already being used by the last WHIP.

The code was changed so that, instead of assigning a Peer ID of the number in the MIP Set to a new MIP, the CEP will search backwards from the number in the set to 1, looking for an unused Peer ID. There should always be an unused integer, and the CEP will assign this as the Peer ID for the new MIP. In this example, the AARC would be assigned a Peer ID of 11.

### 3.67 JTLS-2024-16743 OTHGold - Add Support For Space Force Codes

**The OTH Gold Message Service (OTHGMS) derives a force code for each track in the game. These force codes describe a relationship and an operational domain for the track. The OTHGMS was not considering SPACE tracks when deriving a force code, so TBMs were always assigned an operational domain of AIR instead of SPACE.**

The OTHGMS now considers TBMs to have an operational domain of SPACE when assigning force codes.

### 3.68 JTLS-2024-16744 SVP Error 217 Description Confusing

**The wording for SVP Error 217 is confusing.**

The wording was modified to make it easily understandable.

# 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 6.2.5.0.

## 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.

# APPENDIX A. ABBREVIATIONS AND ACRONYMS

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

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

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

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

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

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

| | |
|---|---|
| OJCS | Organization of the Joint Chiefs of Staff |
| OMA | Order Management Authority |
| ONC | Operational Navigation Chart |
| OPM | Online Player Manual |
| OPP | Order Preprocessing Program |
| OTH | Over The Horizon |
| OTH Gold | Over The Horizon message specification |
| OTH-T | Over The Horizon-Targeting |
| pD | Probability of Detection |
| pE | Probability of Engage |
| pH | Probability of Hit |
| pK | Probability of Kill |
| 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 |
| 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 continually adding and vetting unclassified data to expand and maintain the JTLS-GO Data Repository. Over the last year as part of the JTLS-GO 6.2 development effort, this entire process has been conducted in both the JTLS-GO 6.1 and JTLS-GO 6.2 versions of the repository. No specific, unique, additions were made to the JTLS-GO 6.2 repository. The repository delivered with this initial version of JTLS-GO 6.2 contains the same data as the JTLS-GO 6.1 version of the repository, except the format has been altered to meet the requirements of JTLS-GO 6.2.