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 6.3.4.0)

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.3.4.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.3.4.0 is a Maintenance release of the JTLS-GO 6.3 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as a minor model functionality improvement and three system security improvements mandated by various customer Cybersecurity organizations. 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
1200 Piedmont Ave, Suite A
Pacific Grove, California 93950-5157
United States
jtlsgo@valkyrie.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.3.4.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.3.4.0 is a Maintenance delivery for the JTLS-GO 6.3 series of releases.

JTLS-GO 6.3.4.0 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 "wespac63". There were no database format modifications between this Maintenance release and the original JTLS-GO 6.3.0.0 version. Appendix B of the *JTLS-GO 6.3.0.0 Version Description Document* summarized the database format changes made between the JTLS-GO 6.2 series and this JTLS-GO 6.3 series of the software system.

Detailed descriptions of a minor model Engineering Change Proposal (ECP), as well as three security improvements mandated by various customer Cybersecurity organizations, are provided in Chapter 2.0. Chapter 3.0 summarizes the Software Trouble Reports (STR) that have been corrected and are delivered with this version of JTLS-GO 6.3.

JTLS-GO 6.3.4.0 executes on the Red Hat Enterprise Linux Version 9.4 and Oracle Linux 9.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any 64-bit operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

This section lists documents and software that are relevant to JTLS-GO. All JTLS-GO documents included in this delivery are provided in PDF format within a documents subdirectory.

1.2.1 Obsolete/Outdated Documents

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

1.2.2 Unchanged Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.3.3.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.3.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.3.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.3.0.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.3.0.0)

1.2.3 Updated Documents

- *JTLS-GO Air Services User Guide (*JTLS-GO Document 02, Version 6.3.4.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.3.4.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.3.4.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.3.4.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.3.4.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.3.4.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.3.4.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.3.4.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.3.4.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.3.4.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.3.4.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.3.4.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.3.4.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.3.4.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.3.4.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)
- Geo-Spatial Service (GSS)
- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)

- Reformat Spreadsheet Program (RSP)
- JTLS Symbols Application (JSYMS)
- Database Development System (DDS)

  Database Configuration Program (DCP)
  DDS Client User Interface (DDSC)

- ATO Translator Service (ATOT)
- ATO Generator Service (ATOG)
- ATO Retrieval Program (ATORET)
- JTLS Convert Location Program (JCONVERT)
- Count Critical Order Program (CCO)
- JTLS HLA Interface Program (JHIP)
- After Action Review Client (AARC)
- Scenario Data Client (SDC)
- Order Entry Client (OEC)
- Order Verification Tool (OVT)
- JTLS Object Distribution Authority (JODA)

  The current JODA build number is 214.

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

  Apache Server (APACHE) version 2.4.62
  JTLS XML Serial Repository (JXSR)
  Order Management Authority (OMA)
  Synchronized Authentication and Preferences Service (SYNAPSE)
  XML Message Service (XMS)
  Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service
- Keyhole Markup Language (KML) Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)

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

  JTOI_ICC320 - Used to feed NATO ICC Version 3.2.0 system.

  JTOI_ICC340 - Used to feed NATO ICC Version 3.2.0 system.

  JTOI_ICC350 - Used to feed NATO ICC Version 3.2.0 system.

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

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

- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool
- JTLS Version Conversion Program (VCP)

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

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

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

  VCP63 - Converts a JTLS-GO 6.2 database to a JTLS-GO 6.3 formatted database.

Instructions for installing JTLS-GO 6.3.4.0 are provided in the *JTLS-GO Installation Manual*. Compared to the JTLS-GO 6.2 series, the JTLS-GO 6.3 series uses a significantly different version of PostgreSQL and the Linux operating system. If an organization has not already upgraded to the JTLS-GO 6.3 version, ensure special attention is given to following the documented operating system and PostgreSQL installation procedures. No other upgrade beyond installation of the compressed TAR files or CD is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called “repository63”. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space.

- The scenario “wespac63”, which is suitable for training and demonstrations. The scenario has been updated to use the newest version of engineering level data.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.3.4.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

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

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

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

  Oracle Linux 9.4 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by Defense Information System Agency (DISA) for use by U.S. Government Agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must be a 64-bit architecture with a Java-enabled web browser. JTLS-GO 6.3.4.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 9.4

  Oracle Linux 9.4

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

- JTLS-GO 6.3.4.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 442 package. Both the ICP and DCP have the option for an organization to increase the maximum memory heap for the WHIP and DDSC. For large scenarios and databases, an organization should consider increasing the maximum heap size.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.4.

- JTLS-GO 6.3.4.0 is being delivered PostgreSQL 15.10 that has been compiled under Linux 9.4 and is bundled with the tar files for this release. This version of PostgreSQL is the latest security patch release of PostgreSQL and is being delivered in accordance with US Department of Defense Cybersecurity requirements. It is not necessary to use the delivered solution, but it is the easiest method to meet the requirements of JTLS-GO 6.3.4.0. There are several alternative methods available for obtaining the PostgreSQL 15.10 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.

JTLS-GO 6.3.2.0 was released with PostgreSQL 15.8. There is no currently known available method to upgrade PostgreSQL 15.8 to PostgreSQL 15.10.

- If your organization requires the use of the most current security release of PostgreSQL, download each of your scenarios held by PostgreSQL. Install PostgreSQL 15.10 by following the instructions in Chapter 6 of the *JTLS-GO Installation Manual* and reload your scenarios.

- If your organization is willing to skip this PostgreSQL maintenance release, JTLS-GO will operate without error using the previously delivered PostgreSQL 15.7 or PostgreSQL 15.8 server. U.S. Government organizations should note that doing so may be contrary to your authority to operate JTLS-GO on Government computer systems.

- 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.

- JTLS-GO 6.3.4.0, using the JODA service, allows connections and data exchange with customer client programs. The customer client programs are linked with a set of JTLS-GO-provided API libraries that permit a TCP/IP connection between the JODA and the client program. These API libraries, called JDSP libraries, are built for Linux and Windows and allow customers to built client applications on either of these operating systems. Below are the development environments under which each of the JDSP libraries are built:

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

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

1.3.2 JTLS-GO Cybersecurity Compliance

Because of recent incidents of intrusions into software systems, the United States Department of Defense (DoD) has implemented a strong and strictly enforced Cybersecurity program. JTLS-GO, as software that executes on DoD systems, must comply to the mandates of the program, along with all of the third party software used by JTLS-GO, such as PostgreSQL and Java.

One of the DoD requirements is that the software must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. The concept of an expiration date has been removed from the DoD requirement, but the concept of always using the latest version of third-party software remains a strong component of DoD Cybersecurity requirements.

The following procedure has been established and approved by the JS/J7 Cybersecurity branch to meet the software update requirement:

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

- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a bug release version (JTLS-GO 6.3.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the bug released versions when distributed.

The JTLS-GO 6.3 series has been issued an Exit Gate letter and certification from the JS/J7 Cybersecurity branch. Please contact the U.S. Government Program Manager, Ms. Tara Kea-Edwards (latara.k.keaedwards.civ@mail.mil) to obtain the completed Cybersecurity paperwork.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.3.4.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

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

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the $JGAME/data/hla directory. Federation testing of JTLS-GO with CAE's GESI model has been accomplished. The reader should note that the JTLS-GO Development Team, to date, has not been able to test this federation. If there is interest in running this federation, please contact the JTLS-GO Help desk at jtlsgo@valkyrie.com.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.3.4.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation (www.pitch.se). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

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

Due to changes made in support of STR JTLS-2025-17024 SVP Corrections Inserting Records Nonfunctional, the JTLS-GO 6.3.4.0 database format has been changed to add a new ROWID column to the text_symbol_lu table, related to unit symbols.

As a result, users should **unload** their JTLS-GO 6.3 series scenarios prior to installing JTLS-GO 6.3.4.0, and then **load** them following installation, in order to preserve their changes.

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

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

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

Due to the mandated Cyber-Security improvements made to the Database Configuration Program (DCP) and the Database Development System Client (DDSC), each PostgreSQL database scenario needs to be reconfigured.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 6.3.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.3.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario_name>.gs and databases symbol.scf file.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.3.4.0, but end-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

This means that the user can easily move in this new symbol set. Please follow the steps outlined in the *JTLS-GO DDS Users Guide*, Section B.13, Updating Scenario Symbol Set.

1.4.3 Standard Repository Changes

The JTLS-GO Database Team has continued to improve and expand the unclassified data repository, which has been renamed to “repository63”. The DDS comparison and synchronization function can be used to determine if any of the changes delivered are of use to a JTLS-GO user organization. Specifically, significant effort has been started to represent additional Combat Systems to more closely match the Combat Systems recognized by the Joint Live Virtual

Constructive (JLVC) federation of models. This effort is expected to be an ongoing effort for the next three to five months.

1.5 INSTALLATION

The *JTLS-GO Installation Manual,* a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery, It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 15.10.

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

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes the model and system capabilities added to JTLS-GO 6.3.4.0 as a result of implementing authorized Engineering Change Proposals (ECPs). Three important Cyber-Security ECPs are delivered to solve the outstanding Plan Of Action and Milestone (POA&M) security waivers that were granted to the JTLS-GO system to operate on secure defense computing networks.

In addition, two exercise construct ECPs were delivered to fulfill important JTLS-GO use case issues.

2.1 JTLS-2013-11738 Deterministic Damage Combat System

Summary of Model Change Request

The purpose of this change request is to give the Controller more deterministic control over magically created damage to a unit when using the Damage Combat System and Controller Change Unit orders.

Design Summary

During the course of an exercise, an event may specify that some very specific changes be made to a unit. The Controller has three orders available to magically implement these scripted unit alterations:

- Damage Supplies - No change were made to this order.
- Damage Combat Systems - A change to this order was implemented. Although the Controller could specify the type and number of Combat Systems that should be damaged, the Controller could not specify what should happen to the crew once the systems are damaged. The Controller must accept the random crew damage that the model computes. This limitation was corrected. Two new fields were added to the order.

  The Controller can now select the type of damage that should be applied to the Combat Systems. The options are summarized in Table 1:

Table 1. Combat System Damage Options

OPTION	EXPLANATION
Use Probabilities	This is the algorithm that is in effect in all previous versions of JTLS-GO. When the Combat System is damaged, the model accesses the following data to determine whether the system should go into maintenance or whether it is considered a catastrophic kill.

Table 1. Combat System Damage Options

OPTION	EXPLANATION
Use Probabilities (Con't)	CS COMBAT ARMS EQUIVALENT - This database parameter, which holds a Combat Arms Type (CAT), is accessed for the one Combat System that represents Crew. The data parameter is needed to determine how long it will take to repair each of the wounded crewman.
	CSP CAT PROB SYSTEM RECOVERED - This database parameter is access by accessing the damaged unit's Combat System Prototype CSP and the CAT used by Combat Systems. For each damaged system, the model draws a uniformly distributed random variate and compares it to this probability. If the determination is that the system can be recovered, then the Combat System is placed in maintenance. If the Combat System cannot be recovered then a Catastrophic Kill object is placed on the ground and is visible to collection sensors.
Always Catastrophic	None of the described above data is accessed. All damaged Combat Systems are assumed to be killed and will become catastrophic kill objects. The unit's Killed In Action (KIA) column will be updated with these systems.
Always Repairable	None of the described above data is accessed. All damaged Combat Systems are assumed to be damaged, but repairable. They will show up in the unit's maintenance column and will come out of maintenance based on the database specified repair times.

The crew logic for each damaged Combat System was also changed as a result of this ECP. There are five different options available to specify what should happen to the crew of damaged systems. These options are summarized in Table 2:

Table 2. Crew Options

OPTION	EXPLANATION
Use Probabilities	This is the algorithm that is in effect in all previous versions of JTLS-GO. When the Combat System is damaged, the model accesses the following data to determine what to do with the crew:
	CS CREW COUNT - A Combat System that is considered "Manned" always has its full complement of its crew count.
	CS PROB KILL - This is the Probability that each represented crew is killed when the Combat System is killed. If the CS CREW COUNT is 5, then 5 Uniformly distributed random variates are drawn and compared to this probability of kill. If the crewman is considered killed, it is moved to the Killed In Action (KIA) column of the unit's Combat System data.

Table 2. Crew Options

OPTION	EXPLANATION
Use Probabilities (Con't)	CS PROB WIA - After the kills are determined, the model looks at all remaining crew to determine if they are wounded. Assume in the example, that one crewman was killed, leaving four crewman that still need to b handled. In this situation, the model will draw four uniformly distributed random variates between 0.0 and 1.0 and compare it to the Probability that the crewman is wounded. Any crewman that is not wounded, will automatically be placed back in the available column and can be used by other Combat Systems as needed.
	CS COMBAT ARMS EQUIVALENT - This database parameter, which holds a Combat Arms Type (CAT), is accessed for the one Combat System that represents Crew. The data parameter is needed to determine how long it will take to repair each of the wounded crewman.
	CAT REPAIR TIME - This database parameter is access for the CAT used for the Crew Combat Systems. For each wounded crewman, an exponentially distributed random variable with this mean is drawn. This determines the amount of time needed to repair the wounded crewman.
	FC WIA MAX TREAT UNIT TIME - This database parameter is used to determine which of the wounded crewman are so badly wounded that they need to be evacuated. Each of the repair times computed above are compared against this database parameter for the damaged unit's Faction. If the repair time is greater than this parameter, then the crewman is moved to the Wounded In Action (WIA) column and must be evacuated. If the random repair time is less than or equal to this database parameter, the crewman enters unit maintenance and eventually will be returned to action
Always Kill	None of the described above data is accessed. All crewman are assumed to be killed and will show up in the unit's KIA column for Combat System Crew.
Wounded All Evacuated	None of the described above data is accessed. All crewman are assumed to be wounded so severely that they need to be evacuated. They will show up in the unit's WIA column for Combat System Crew.
Wounded None Evacuated	None of the described above data is accessed as part of the damage algorithm. All crewman are assumed to be wounded and will go into maintenance at the unit. They will show up in the maintenance column and eventually be returned to action after a random repair time based on the database parameter CAT REPAIR TIME.
Not Affected	None of the described above data is accessed. All crewman will be placed back in the unit's Available column where they could immediately be assigned to an available Combat Systems waiting for crew; otherwise they will stay in the Available column of the Unit's Combat System array.

- Controller Change Unit - A change to this order was implemented. The Controller can specify the new desired unit strength that should be achieved, but there was no control over how to prioritize the Combat Systems that should be damaged to achieve the strength goal. This limitation was also corrected.

The Controller Change Unit order was modified to include a list of combat systems that must be damaged and the same two fields added to the DCS Order to control how the damaged systems are represented. The Combat System damage list is applied first to the unit and if more systems must be removed to achieve the desired new strength, the old algorithm to equally distribute the removed systems is applied. The Controller does not need to specify a list of Combat System to destroy, it is an optional field. If no list is specified, the algorithm works as it always has.

2.2 JTLS-2022-15976 Encrypt All Passwords Used To Start WHIP Or DDSC

Summary of Model Change Request

This is the first highly important Cyber-Security issue that needed to be solved. Encrypted passwords for the WHIP were delivered in previous versions of JTLS-GO 6.3. The same capability is needed for the Database Development System Client (DDSC).

Design Summary

The DDSC now uses the same password construct as the WHIP. The Database Configuration Program (DCP) is used to set the default password for each DDSC. The first time users sign into their assigned DDSC, the system requires a new unique password that only the DDSC user will know. The passwords are stored in an encrypted manner on the server.

2.3 JTLS-2023-16449 Run JTLS-GO From User Account

Summary of Model Change Request

This is the second highly important Cyber-Security issue that needed to be solved. When JTLS-GO executes on a highly secure computer system, Cyber-Security regulation mandate that multiple users cannot have the password to a single account. This mandate is universal across of US Department of Defense systems and NATO systems.

The JTLS-GO structure paradigm assumes that all Technical Control personnel can log on to the JTLS-GO account from any NFS mounted JTLS-GO server to conduct Technical Control functions, such as starting, stopping, and altering each of the JTLS-GO system programs.

The US provided JTLS-GO a waiver for this mandate, but NATO refused to do so. Thus JTLS-GO needed to adjust its paradigm to allow each person associated with Technical Control to have their own account and still manage all needed Technical Control functions. This ECP implements the needed paradigm change.

Design Summary

An organization does not need to use this new multi-user paradigm. It is a choice that can be made by each organization.

If an organization chooses to use the multi-user Technical Control environment, a user account needs to be created for each member of the Technical Control team, in the same group as the single JTLS-GO installation account. After installation, a script has been created to alter the permissions of all needed files and executable code from User-only privileges to Group privileges. Once Group privileges are in place, the processes associated with JTLS-GO can be executed by an user account in the JTLS-GO group.

The issue of allowing multiple users to interact, as needed, with the Combat Events Program terminal has been solved by running the CEP in a Multiplexer Terminal (TMUX). The user starts the CEP in a TMUX terminal, and must provide permission access to other Technical Control accounts to the TMUX terminal. Once accomplished, the permitted accounts are able to monitor and interact with the CEP.

The numerous details associated with running JTLS-GO in a multi-user environment is discussed in the *JTLS-GO Installation Manual* and the *JTLS-GO Technical Controller Manual*.

The Development Team wants to emphasize that JTLS-GO does not need to use the Multi-User Technical Control environment. Installing this version of JTLS-GO does not need to affect the manner in which Technical Control runs and uses the JTLS-GO system. Each organization's Cyber-Security regulations should be consulted when making the decision to use or not use the Multi-User Technical Control environment.

2.4 JTLS-2024-16879 DDSC Privileges

Summary of Model Change Request

This is the third highly important Cyber-Security issue that needed to be solved. Every DDSC had the ability to execute all DDSC capabilities. In the past this has led to numerous issues such as:

- Two DDSC users attempting to perform a database download at the same time.
- A DDSC user using the migration and synchronization tool, or the Automatic Supply Calculation (ASC) tool, without fully being trained it how to use them.
- A DDSC user altering prototype data, even though they were given access to the DDS to alter Order of Battle (ORBAT) information only.

To solve these problems, the Database Team needs to ability to limit access of certain functions to each DDSC,

Design Summary

The DCP was changed. In previous versions, the DCP controlled which DDSC users could perform migration and synchronization functions. This privilege configuration has been expanded to include permission to:

- Download the data.
- Run the Scenario Verification Program (SVP).
- Run the ASC.
- Run JINN-Capable options.

In addition, each DDSC can be assigned a Edit Role file that manages which database tables are allowed to be edited, The concept of this file is similar to that of the WHIP Menu Definition files, In the DCP, Database Control can select which role to assign to each DDSC, thus configuring the table editing privilege for that client. For example, it is possible to select the ORBAT role. In this Edit Role file, only the ORBAT-related tables can be altered. Each DDSC client can see all table data, but are only allowed to change the data in the tables for which permission has been granted in the DDSC's Edit Role file.

JTLS-GO is delivered with some configuration-managed Edit Role files, but each user organization can define their own Edit Role files to meet their specific operational needs.

2.5 JTLS-2025-17035 Pass Object Ownership Using ELS

Summary of Model Change Request

The capability to support ownership transfer of objects was added to the ELS. This was part of continuing efforts to improve the connection between JTLS-GO and the external model VBS4.

Design Summary

Code changes were made to the ELS and the CEP to allow the control of objects to be transferred from one model to another. The implementation is complete, but the full capability will not be ready until further testing and integration with VBS4 has been done.

3.0 SOFTWARE TROUBLE REPORTS

Software Trouble Reports (STRs) describe software code errors that have been discovered by JTLS-GO users or developers and have been corrected.

3.1 JTLS-2025-17024 SVP Corrections Inserting Records Nonfunctional

When using the Scenario Verification Program (SVP) automatic correction capability, all corrections that required a new record to be inserted into the data failed to execute.

The new version of PostgreSQL used in the JTLS-GO 6.3 series, requires that each database record include a ROWID attribute. The SVP Insert record corrections did not include this new ROWID attribute. The attribute was added to each of the automatic insert record corrections, which solved the problem.

3.2 JTLS-2025-17025 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.

3.3 JTLS-2025-17027 WHIP ATOT Did Not Take Off Unit From Task List

The WHIP's ATO Translator (ATOT) interface did not allow the user to remove a unit from a task list in the selection dialog. When the tasked unit exists in the ATOT linking data, but not the scenario game (such as detached units), the selection dialog failed to open to remove the tasked units.

The problem was solved by notifying the user that the tasked units do not exist in the scenario game, and the selection dialog will open showing only the existing tasked units.

3.4 JTLS-2025-17028 SVP Crash For Scenario Converted From 6.2 To 6.3

The Version Conversion Program was used to convert a JTLS-GO 6.2 scenario to 6.3. The 6.3 SVP crashed when it was used to verify the 6.3 scenario.

The JTLS-GO 6.2 SVP Preferences file has three default groups of preferences: Terrain, Federation, and TBMCS Data. After the SVP reads in these three check groups, it reads the warnings that should be ignored. Each warning is listed line-by-line. After the final warning, an asterisk ("*") is entered on the next, and last, line of the preference file. While converting the scenario from 6.2 to 6.3, the user ran the copyscenario script, which copied the JTLS-GO 6.2 version of the preferences file into the directory structure for the JTLS-GO 6.3 version of the scenario.

The JTLS-GO 6.3 SVP Preferences file has four (not three) check groups: Terrain, Federation, TBMCS, and LOGFAS Data. The SVP crashed because it read in four check groups, but the file only had three. The fourth group read in an asterisk, and when the code attempted to read the first warning it came to an End Of File, causing the crash.

The preferences file is now deleted during the conversion process. The user will need to reset their desired preferences. This is reasonable because the SVP warning numbers may change from version to version, and the user may not want to skip the warnings that were set up for 6.2.

Note - if the 6.2 preferences file listed some warnings, the first warning would not have been recognized because it was read in and placed in the variable CHECK.LOGFAS. The remainder of the warnings would have been read without incident and the SVP would have run as expected, except the first warning would not have been skipped.

3.5 JTLS-2025-17029 SVP Error 503 Additional Correction Needed

A new Small Boat (SB) was created, with a new Combat System (CS).

When an HRU was created in the database, Error 503 was generated, because the HUP did not have the necessary type of SB. In fact, the HUP had the correct boat type (by name) but the equivalent CS for this boat did not have a Special Capability of AMPHIB, or a Special Subtype equal to the SB class.

Error 503 needs a correction to bring up the relevant CS and special capability and special subtype for SB column in the CS table.

The original Error 503 was designed for Naval units, but was being applied to all other types of units. The process of assigning Small Boats to Naval units, rather than to other types of units, is different. The Warning was split into Warning 503 for Naval units, and 504 for all other types of units.

The corrections for each Warning were expanded to provide the user the options necessary to look at all the tables and data need to correct the problem based on unit type.

3.6 JTLS-2025-17030 Allowed Intercept Report Order Help

The order help and field help text for the Allowed Intercept Report order was confusing.

The order help and field help text for the order was updated for clarity.

3.7 JTLS-2025-17031 Warning 1255 Values Nonfunctional

Warning 1255 identifies when a Unit has insufficient fuel supplies to support its Combat Systems. However, selecting one of the corrective options does not correct the problem, and the SVP continues to report the Warning.

The corrective option displayed an incorrect value. The routine was corrected to pass the correct value.

Additionally, the corrective option's wording was confusing. Users believed that the value presented by the correction option represented the correct total amount of fuel, but the value actually represented the amount that should be added to the existing amount of fuel. The option's wording was changed to make this more clear.

3.8 JTLS-2025-17032 Errors When Changing Load Assignment Array

The Set Load Assignment order had errors when removing a load from the assigned loads for aircraft.

The code did not properly handle the cancellation of missions when the load assignment was changed. The code to process this order was not well organized, and there was a great deal of duplicate code used in several places.

The code was rewritten to determine whether or not a mission should be canceled when the load assignment was modified. Significant changes were made to the main routine to clean up redundant code and facilitate future debugging if necessary.

3.9 JTLS-2025-17033 AAR Damage Report Missing Runway Damage

A 10,000 foot runway was reduced to about 3000 feet by an air strike, and confirmed by the Damage Report and Situation Report. However, the AAR Air Surface Damage Report showed a value of zero in the Number Killed column.

The user noted that the Object Damage Report showed the actual fraction destroyed by each weapon under the Number Killed column. The user also noted that the Engagement Surface Table in the AAR database holds the amount of damaged inflicted on the runway for each weapon. Therefore, the runway damage data exists in the database.

This is a problem of formatting the runway damage output. As mentioned above, runway damage is stored in the AAR database as the fraction of the runway destroyed by each weapon, so the values are between 0.0 and 1.0. The Air Surface Damage Report was formatting the damage values as an integer, and this caused the report to output a zero in the Number Killed column. The formatting has been corrected to output the values as real numbers.

3.10 JTLS-2025-17034 Controller Change Target Order AAR Result

The Controller Change Target order allows the Controller to change the capability of a Target between 0.0 and 1.0 (for a Runway Target the specified capability is restricted to the Minimum Runway Percentage and 1.0). Unless the specified capability is the same as the Target's current capability, this should, in general, result in either a CONTROLLER FAILURE or CONTROLLER REPAIR maintenance event being recorded in the AAR database. However, this has not been happening.

Many calls to send the results to the AAR were either missing, or misplaced, in the CEP. The Controller Change Target order results are now recorded as either a CONTROLLER FAILURE or CONTROLLER REPAIR maintenance action. Each damage/repair is also recorded in the AAR TARGET EVENT table. When damaging targets such as SUPPLY STORAGE types, a percentage of their supplies are killed. These losses are now recorded in the AAR OTHER LOSSES table.

3.11 JTLS-2025-17036 Assess Weapon Damage Against Satellites

Satellites could not be used as the specified target in the Assess Weapon Damage order.

The capability was never added when satellites were included in JTLS-GO. Structures were added to the existing code to support the damaging of satellites using this order.

3.12 JTLS-2025-17037 Convoys Left Over When ELF Processing Canceled

When unit-related events were canceled, any processing by elementary loading facilities (ELF) was canceled. This occurred for various reasons such as when the unit was magic moved, or when it was removed from the game via TPFDD order. However, when the ELF processing was canceled, any associated convoy continued to exist with the same incomplete loading or unloading task.

The code to cancel the events for the unit was modified to cancel the current task for the convoy and then send the convoy back to its home unit. While canceling the current task, the ELF processing event is destroyed. The convoy returned to the original unit as expected.

3.13 JTLS-2025-17038 OPM Errors in Hyperlinks

When a player opened the Online Player Manual (OPM), some of the links did not point to the correct file location. In some cases, the links were associated with Controller OPM pages. In other cases, the links pointed to the incorrect file name.

Changes were to the Scenario Initialization Program code were made to fix the links in the OPM pages. The corrections were mainly associated with links to specific unit and target data, such as for higher headquarters, a Target’s owning unit, or the operating unit for pipelines.

3.14 JTLS-2025-17049 ATOT Generic Named Side List Utility

The ATOT generates an order utility containing all Force Side names for Sides sharing the ATO. This utility is given a name, which does not identify it as belonging to a specific ATO period. The ATOT saves this Side list utility in the Basic order group, where it may be deleted when the group is deleted. This leaves the orders in the ACM order group, which also reference this utility, without a Side list utility.

The routine that names the Side list utility was modified to include the ATO ID and change number from the period for which the ATO was translated. But, when the ATOT is run in the absence of an ATO (only translating an ACO), the ATOT uses the same utility naming criteria except using zero ("0") for the ID and zero ("0") for the change number.

Additionally, the ATOT does not include the Side list utility in the Basic order group for the ATO period. This way it is available for reference by orders in both the Basic and the ACM order groups. Deleting either group will allow the utility to remain available for orders in the other referring group.

3.15 JTLS-2025-17050 ACP Parameter Improved Documentation

The documentation for the ACP.NUMBER.AC.IN.DETECTION.GROUP was inadequate, and did not properly explain what the parameter is meant to do.

The entry for the parameter in the *JTLS-GO Data Requirements Manual* was updated to be more thorough, and the *JTLS-GO Analyst Guide* was expanded to elaborate more on how this parameter works.

3.16 JTLS-2025-17052 WHIP Error Message Connecting To JXSR

The WHIP Debug Console sometimes displays numerous error messages when the WHIP first establishes a connection to the JXSR. These were reported as "Error, service_id 1", which gives the user no indication of what is causing the problem.

The problem is caused when the Apache JXSR module has not yet discovered all of the running JXSR services. When the WHIP starts making requests to the JXSR, the requests pass through the JXSR module, and the module can only direct the request to a running JXSR service if it has a reference to the specified JXSR service from the request.

The text of the error message was modified to reflect this. The problem self-corrects.

3.17 JTLS-2025-17053 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.

3.18 JTLS-2025-17055 DDS 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.

3.19 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.

3.20 JTLS-2025-17059 DDS Repository Cannot Synchronize Updates

When synchronizing a Tactical Unit Prototype's graphic symbol data, although the status said synchronization was complete, the graphic symbols of the destination database were not updated.

The text_symbol_lu table was missing the rowid column. However, the content of the text_symbol_lu table may not be hidentical between different scenarios, so a migration rule checked if the referenced graphic symbol existed in the destination database's text_symbol_lu table, and if not, nullified the field. The rowid is necessary for this process.

The database schema was modified so that the rowid column was added to the text_symbol_lu table. To solve this problem, each JTLS-GO 6.3 scenario should be reloaded in PostgreSQL.

3.21 JTLS-2025-17060 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.

3.22 JTLS-2025-17062 Error 247 Corrections Do Not Always Work

When executing the correction for Error 247 to add a CREW combat system record, the correction failed.

The code is checking for crew and personnel requirements. The code was not properly set up to distinguish between a CREW combat system record that did not exist in the database and one that existed but had a TOE of zero. Therefore, when an attempt to insert a new CREW combat system record when a record with a TOE of zero existed, the correction would not work.

The code now makes the distinction between a nonexistent record and a record with a TOE of zero, for both SUPs and TUPs. It will appropriately decide whether to insert a new record with the proper TOE or update the current record.

3.23 JTLS-2025-17063 Illegal Characters In WHIP Password

When assigning a password for a WHIP, the Password Prompt panel checked off the password requirements as they were fulfilled. A password must have at least one uppercase letter, one lowercase letter, one digit (0-9), and one valid symbol from a list of special characters.

There was nothing to stop the user from entering an illegal character outside of the list of special characters. There was no indication of what the error was, but the "Confirm" button on the Password Prompt panel remained disabled. The user was left confused why the new password cannot be changed, because all of the password requirements were fulfilled and shown in green.

The Password Prompt panel's code was changed to forbid users from entering illegal characters; the character will be rejected and the user informed that an illegal character was entered. When all of the password requirements are properly fulfilled, the Confirm button on the password prompt panel will be enabled.

3.24 JTLS-2025-17064 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.

3.25 JTLS-2025-17066 Incorrect VDD For 6.3.3.0

The JTLS-GO 6.3.3.0 release included an incorrect Version Description Document (VDD).

The correct 6.3.3.0 VDD has been included in the Documents directory of this release.

3.26 JTLS-2025-17067 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.

3.27 JTLS-2025-17070 Display OPM Custom Query Active Count

Custom Queries are available in the OPM tables. These queries allow you to filter tables based on custom data constraints. The number of active queries was not displayed to the user when the query builder is hidden.

The Custom Query builder has been updated to display the number of active queries on the Custom Query builder button.

3.28 JTLS-2025-17071 SVPR Error 451 Correction Does Not Work

In the SVP, Error 451 has two corrections. The correction offering to “blank out the track blocks” for a squadron has no effect on the existing track blocks.

The error was checking data in two different tables to see if a squadron had JU numbers or source numbers and, if it did, whether it was assigned to a Link 16 network. If there was an error, the correction assumed the error was generated solely based on the data in the squadron table. In fact the error was also generated if the squadron had source JU numbers in the Squadron JU table and no Link 16 network was assigned.

Instead of confusing the user by combining corrections to two different tables into Error 451, the errors associated with the Squadron JU number table were moved to new Error 422.

3.29 JTLS-2025-17073 Leaflet OAS Mission Does Not Fly

An Orbiting Air Support (OAS) order was submitted to direct an air mission to drop leaflets (1000KEAFLET.BOX) on a ground Unit. The mission entered Launch Capable posture, but immediately completed with a message saying “the mission is not carrying the right weapon to use. None of the weapons are effective against the specified target.”

To use leaflets, the supply category for the weapon must have a special capability of LEAFLET. In the database, the 1000LEAFLET.BOX targetable weapon uses the supply category LEAFLETS-1000LEAF.BOX which correctly has a special capability of LEAFLET. Also, the 1000LEAFLET.BOX weapon has a surface kill lethality of NON.LETHAL_SKL, with zero Probability of Hit (PH) and zero Probability of Kill (PK) tables.

The problem was in the logic that determines which weapons can be fired. Any weapons with a PH and PK of zero were deemed “ineffective”. If no weapons were effective, the mission was immediately terminated before launch.

To fix this problem, the code was changed to allow weapons with a special supply capability of LEAFLET to be considered effective, regardless of PH and PK values.

3.30 JTLS-2025-17075 Air Mission IMT Fuel Remaining Format Error

When an air mission is in a scheduled mode, the IMT Fuel Remaining column has a formatting error, caused by the air mission not having been assigned a fuel type. Fuel type cannot be assigned prior to going through the Launch Capable posture.

When the CEP is incapable of assigning a fuel type to a scheduled air mission, the IMT will use the designated use-category specification in the IMT's configuration file to prevent the Fuel Remaining row from displaying the “Format Error”, and will instead display the Fuel Remaining in the specified basic unit-of-measure.

To maintain consistency with the SITREP, the same was done for the SITREP tool.

3.31 JTLS-2025-17076 Air Mission Duration Field Does Not Allow Seconds

The Duration field of an air mission order allows the user to input times using hours, minutes and seconds. However, if only seconds are entered, the value is ignored.

The Duration field in an order specifies the minimum and maximum permitted values. The WHIP will now set the minimum permitted duration to a field whenever the user assigns a value less than that minimum, and will allow a value of zero when the minimum permitted is zero. The maximum permitted value already exhibited this behavior.

3.32 JTLS-2025-17077 Total Target Track Numbers Counted Incorrectly

The SVP is miscounting the available Target Track numbers. For example, the SVP says 06001 to 06005 has four available numbers, when the correct number is five.

The logic in the code failed to allow for inclusive numbers. 06001 to 06005 when subtracted is four, then one should be added to allow for inclusive numbers.

While correcting and testing, it was also found that a Link 16-capable Target, where the owning and associated Units are the same, was counted as two Targets, needing two separate Target Track Numbers. This was also corrected.

3.33 JTLS-2025-17079 Incorrect Use Of Sensor Type

In the logic that updates grid effects for a moved sensor target, the sensor type was mapped to a local variable, THE.TYPE, as .ENTITY.ST, .ENTITY.JT, or .ENTITY.AD. THE.TYPE was then passed to routines that created or moved grid effects as appropriate.

Later, the logic incorrectly used THE.TYPE again as the sensor target's use: .PASSIVE.SONAR, .ACTIVE.SONAR, or .SURFACE.SEARCH. The sensor's use was needed to determine the appropriate routines to check detections. Using THE.TYPE in this manner was the wrong context and removed the opportunity to detect surface or subsurface vessels.

The local variable THE.TYPE was replaced in the second context with the actual use of the sensor class (the sensor subcategory) so that the appropriate detection routines are called. In addition, THE.TYPE was misapplied to determine a jammer's use. THE.TYPE was replaced with the actual use of the jammer class (the jammer subcategory).

4.0 REMAINING ERRORS

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

The following list of issues is known and have not been fixed in time to make it into this release of JTLS-GO 6.3.4.0.

4.1 DDSC/WHIP/JOBE - CADRG Map Zoom

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

4.2 MHE Targets Loading Air Mission Can Cause a Crash

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

4.3 JTLS-GO Strategic Lift Missions Are Not Working Properly

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

4.4 Tactical Ground Formation Attacks Do Not Work

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

4.5 ATOT Spreadsheet Lacks Detailed Field Checking

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

4.6 Moving Combat System Supplies Can Reduce Unit Strength To Zero

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

4.7 Upgrade Procedures For Maintenance Release of PostgreSQL

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

APPENDIX A. ABBREVIATIONS AND ACRONYMS

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

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

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

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

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

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

OMA	Order Management Authority
ONC	Operational Navigation Chart
OPM	Online Player Manual
OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
SDB	Standard Database
SDR	Scenario Data Repository
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)

SIP	Scenario Initialization Program
SITREP	Situation Report
SLP	Sustainment Log Prototype
SOF	Special Operations Forces
SP	Survivability Prototype
SQL	Structured Query Language
SR	Short Range
SRP	Start/Restart Program (a JTLS component)
SRTE	Sea Route
SSM	Surface-to-Surface Missile
STR	Software Trouble Report
SUP	Ship Unit Prototype
SVP	Scenario Verification Program
SYNAPSE	Synchronized Authentication and Preferences Service
TADIL	Tactical Digital Interface Link
TCP/IP	Transmission Control Protocol/Internet Protocol
TEL	Transporter Erector Launcher
TG	Target entity attribute prefix
TGS	Terrain Generation Service (formerly TPS:Terrain Preparation System)
TGT	Target
TMU	Terrain Modification Utility
TOE	Table of Organization and Equipment
TOT	Time Over Target
TOW	Tube-launched Optically-tracked Wire-guided missile
TPFDD	Time-Phased Force Deployment Data
TTG	Target Type Group
TTL	Target Types List
TUP	Tactical Unit Prototype
TW	Targetable Weapon
UBL	Unit Basic Load
UIM/X	GUI builder tool
UNIX	POSIX-compliant operating system
UNK	Unknown
UOM	Unit Of Measure

USA	United States Army (U.S. and U.S.A. refer to United States and United States of America)
USAF	United States Air Force
USCG	United States Coast Guard
USMC	United States Marine Corps
USMTF	United States Message Text Format
USN	United States Navy
UT	Unit entity attribute prefix
UTM	Universal Transverse Mercator
VIFRED	Visual Forms Editor
VMS	Virtual Memory System
VTOL	Vertical Take-Off and Landing aircraft
WAN	Wide Area Network
WDRAW	Withdraw
WEJ	Web Enabled JTLS
WHIP	Web Hosted Interface Program
WIA	Wounded In Action
WPC	Warrior Preparation Center
WPN	Weapon
WT	Weight
WW	Wild Weasel
XMS	XML Message Service

APPENDIX B. VERSION 6.3.4.0 DATABASE CHANGES

Due to changes made in support of STR JTLS-2025-17024 SVP Corrections Inserting Records Nonfunctional, the JTLS-GO 6.3.4.0 database format has been changed to add a new column to the text_symbol_lu table, related to unit symbols. This change requires that users **unload** their scenarios prior to installation of JTLS-GO 6.3.4.0, and then **load** them following installation.

APPENDIX C. VERSION Version 6.3.4.0 REPOSITORY CHANGES

No significant changes were made to the JTLS-GO Version 6.3.4.0 repository.