JTLS-GO Document 17

JTLS-GO

Version Description Document

March 2022

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.1.2.0)

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.1.2.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.1.2.0 is a Maintenance release of the JTLS-GO 6.1 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as major 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 Development Team Leader
ROLANDS & ASSOCIATES Corporation
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940 USA
jtlsgo@rolands.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.1.2.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.1.2.0 is a Maintenance delivery for the JTLS-GO 6.1 series of releases.

JTLS-GO 6.1.2.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations called “wespac61”. There were no database format changes made to support this maintenance release. Database modifications that were accomplished to upgrade the previous JTLS-GO database format to the JTLS-GO 6.1 series format were summarized in the VDD for Version 6.1.0.0 delivered as part release in the documents directory.

Detailed descriptions of the Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0, and a summary of all Software Trouble Reports (STRs) corrected in this Maintenance delivery are summarized in Chapter 3.0.

JTLS-GO 6.1.2.0 executes on the Red Hat Enterprise Linux Version 8.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

This section lists documents and software that are relevant to JTLS-GO. All JTLS-GO documents included in this delivery are provided in PDF format within 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 6.1.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.1.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.1.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.1.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.1.0.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.1.0.0)

- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.1.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.1.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.1.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.1.0.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.1.0.0)
- *JTLS-GO ATO Services User Guide* (JTLS-GO Document 24, Version 6.1.0.0)

1.2.3 Updated Documents

- *JTLS-GO Executive Overview* (JTLS-GO Document 02, Version 6.1.2.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.1.2.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.1.2.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.1.2.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.1.2.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.1.2.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.1.2.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.1.2.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.1.2.0 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 172.

- 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.

Instructions for installing JTLS-GO 6.1.2.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.1.2.0 is not necessary. No other upgrade beyond installation of the compressed TAR files or CD is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called “repository61”. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space.
- The scenario “wespac61”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.1.2.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Server (ES) Edition Version 8.4, 64-bit architecture.

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

RedHat Linux 8.4 - this operating system license must be purchased.

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

CentOS 8.4 - a free version of Linux 8 that has not been approved by DISA for use by U.S. Government agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.1.2.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.9 and 8.4

  CentOS Linux Version 7.9 and 8.4

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

- JTLS-GO 6.1.2.0 is delivered with the AdoptOpenJDK 1.8 Update 322 package, which is equivalent to the current version of OpenJDK. Using AdoptOpenJDK provides two benefits:

  a. Only the JTLS-GO account on the system servers access this version of Java. An installation site can use the JTLS-GO servers for programs other than JTLS-GO without impacting the version of Java used by other programs.

  b. Security releases of AdoptOpenJDK software are produced on the same schedule as the Oracle OpenJDK security release procedure. An organization can expect to receive a bug release version of JTLS-GO within two-weeks of a new Java 1.8 security release. As long as a user organization installs all of the JTLS-GO bug releases, JTLS-GO can guarantee that the latest Java security release is being used on the servers. JTLS-GO no longer depends on system administration for implementing proper Java security update procedures.

- 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 database tools require a certified PostgreSQL 11.13 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.1.2.0. There are several alternative methods available for obtaining the PostgreSQL 11.13 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.

- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of the supported versions of Red Hat Linux ES.

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

- The Perl script language is used by the JTLS-GO system and game setup scripts. The version of Perl included with the supported versions of Red Hat Linux ES is sufficient. The Perl program is typically located in the /usr/bin directory. If Perl is installed in a another location, a link should be created from the /usr/bin directory to this program.

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

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

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

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

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

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

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

- KML Operational Interface (KOI)

  The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth^{TM}. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using an operational Command, Control, Communication, Computer Information (C4I) systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

Chapter 3 of the *JTLS-GO C4I Interface Manual* describes requirements and procedures for using the KOI capabilities.

1.3.2 JTLS-GO Cybersecurity Compliance

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

One of the DoD requirements is that the software must implement a methodology that ensures that the end user 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. After a certain date, all Java related programs stopped working, whether connected to an open network or not. All JTLS-GO releases were closely linked to the Java expiration date.

OpenJDK^{TM} has not implement an expiration date. In order to fulfill this DoD Cybersecurity requirement, JTLS-GO has moved to AdoptOpenJDK, 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, 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 6.1.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the bug released versions when distributed.

Contact the U.S. Government Program Manager, Mr. James Blank (james.e.blank.civ@mail.mil) to obtain the completed Cybersecurity paperwork and a current Gate completion certificate.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.1.2.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

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

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the $JGAME/data/hla directory, Federation testing of JTLS-GO with CAE's GESI model have been accomplished. Future plans include expanding the capabilities of the GlobalSim federation.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.1.2.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation (www.pitch.se). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

Significant database structure differences exist between the JTLS-GO 6.1 series of the software and the previous JTLS-GO 6.0 series database structure. There are no database format changes between JTLS-GO 6.1.2.0 and the initial JTLS-GO 6.1.0.0 version of the software.

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

STR JTLS-2022-15607 Grid Lower-Left Location Not Exact On Restart on Page 3-21 required a change to the ASCII checkpoint format. Thus any saved checkpoints from previous versions of the JTLS-GO 6.1 series will not work with JTLS-GO 6.1.2.0.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 6.1.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.1.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario_name>.gs and databases symbol.scf file.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.1.2.0, 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

R&A has continued to improve and expand the unclassified data repository, which has been renamed to “repository61.” The DDS comparison and synchronization function can be used to determine if any of the changes delivered are of use to a JTLS-GO user organization.

1.5 INSTALLATION

The *JTLS-GO Installation Manual,* a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery, It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 11.11 required to operate JTLS-GO 6.1.2.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes new model capabilities added to JTLS-GO 6.1.2.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 TLS-2021-15155 JTLS-GO Link To VBS4

Summary of Model Change Request

JTLS-GO should link to Bohemia Interactive Simulation (BISim) Virtual Battle Space 4 (VBS4), much in the same way that it can link to other simulations. The capability includes shadowing objects in VBS4 and non-Lanchestrian kinetic interactions between the two models. Refer to the JTLS-GO Federation Users Guide for complete information on the capabilities of this coordinated execution. Everything needed from the JTLS-GO perspective is delivered with this release. Contact BISim for information on what is needed from the VBS4 perspective.

2.2 JTLS-2021-15408 Create Initialization File For Link To VBS4

Summary of Model Change Request

A development test occurred for a direct link between the simulations JTLS-GO and VBS4. In order to process weapon fire interactions, the VBS4 simulation needed access to the names of the targetable weapons in the JTLS-GO scenario.

Design Summary

A new initialization file was created to provide the required data for VBS4. This file was created using the same method as several other initialization files. This mapping file has a file extension of ".vbs" and it includes the names and DIS-Codes for all weapons in the scenario. It is created when the SIP performs the setup a scenario, and also when the CEP starts execution.

2.3 JTLS-2021-15459 Display OTH-Gold Track Time-Late Values

Summary of Model Change Request

One of the most important pieces of data in Over-the-Horizon (OTH) Gold that may be updated is a track’s time of last detection. If a track has not been detected within a certain time margin, it is considered to be “time-late” and may be dropped from the Common Operational Picture (COP). The ability to monitor and perform maintenance on time-late tracks is important to exercise staff. The purpose of this ECP is to provide new ways for operators to monitor track time-late data.

Design Summary

The OTH-Gold Message Service (OTHGMS) and OTH-Gold WHIP Module have been updated to include track time-late information in the object summary table. Time late values are calculated based on the system time of the host running the OTHGMS.

2.4 JTLS-2021-15483 Change Airborne Emitter Status

Summary of Model Change Request

The CHANGE.MISSION.PARAMETER order should have the capability to turn sensors and jammers on or off for an air mission. This is needed to make it easier for the user to accomplish the task. The current Emission Control (EMCON) Order also has this capability, but is more difficult to fill out. The capability to turn on and off sensors and jammer has not been removed from the EMCON order.

Design Summary

The capability to turn mission sensors and jammers on or off was added to the CHANGE.MISSION.PARAMETER order.

2.5 JTLS-2021-15513 IIP Unit And Target Detection Multipliers

Summary of Model Change Request

The user had difficulty trying to detect objects that are in the BE Facility List.

Design Summary

If a user adds objects to a BE Facility List, then it can be assumed that the user’s side already detected the object. If the IIP detection values for the unit or target are zero, a SVP Warning should be created with the exception of target objects that are considered 'UNDERGROUND'

The new Warnings 1614, 1623, and 1627 were added.

2.6 JTLS-2022-15553 New Reports For Air Response Cell Reporting

Summary of Model Change Request

Three reports were developed to support the Air Response Cell requirements to track own-side air missions.

Design Summary

These three reports (the Mission History Report, the Air Surface Damage Report, and the Air Loss History Report) have been integrated into the code.

2.7 JTLS-2022-15578 Imaging Of Chemical Or Nuclear Impacts

Summary of Model Change Request

The WHIP should automatically display Contamination areas on the Map when chemical and nuclear events occur in the game. This can be a setting in the Contamination filters to automatically show the areas of impact as they happen. These are important events and the areas should "pop-up" when created,

Design Summary

On the Contamination Area filer panel, there is a new button to indicated whether areas should automatically be displayed when they are created. If set, the contamination areas will be automatically displayed. If not set, the WHIP operates as it has in previous versions. The user must explicit turn on and turn off the contamination areas. For consistency the same capability was applied to the Operations Areas, Boundaries and DSAs filters panels. A detailed design is available on www.rolands.com.

2.8 JTLS-2022-15623 DDS Table Data Filter By Geographic Region

Summary of Model Change Request

It would be valuable to display relevant Database Development System (DDS) table data for objects in a specific geographic region.

Design Summary

The DDS implementation include changes made in both DDS Client (DDSC) and DDS Server.

New capabilities were added to the DDS Client. Users can now:

- Define polygon areas on the map as customized regions.
- Decide whether to show or hide customized regions on the map.
- Filter data from certain tables by customized region, by selecting existing customized regions or OPAREAs from the "Regions" tab on the table's configuration panel.

Customized regions are now saved whenever a user defines them, and can be recalled and used as data filters after the user exits and restarts the DDS Client. The DDS Server was modified to support these changes, by retrieving and filtering data from tables whenever the user requests data from specific geographic regions.

As a side note, the same capability is planned for the Web Hosted Interface Program (WHIP) and will be delivered in a future version of JTLS-GO.

2.9 JTLS-2022-15624 Add Screen Definition File Attributes

Summary of Model Change Request

The XML configuration files needed to have a geographic region filter to support the changes implemented in JTLS-2022-15623 DDS Table Data Filter By Geographic Region.

Design Summary

The DDS checker had a check added for the individual DDS screen definition files. If the table contains a location-related column, geographic region filter attributes are added to the DDS screen definition file.

The Verify IMT stylesheet had a similar check added, where the IMT screen's object type must be a unit, target, air mission, convoy, formation, or satellite and must contain a location-related column.

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-2021-15405 Errors Restoring Entities In ELS

The ELS was restored from a saved checkpoint, but some of the entities were not restored in the same state as they were before the checkpoint: crew entities which were mounted on vehicles were not shown as mounted after restoration, and some command objects had valid symbol names before the checkpoint but had blank symbols after restoration.

A flag was not set when the ELS was restored from a checkpoint to restore the mounting of crew on vehicles. A code change was made to correct this error. Similarly, the symbol names were not assigned to command objects as part of the restoration. Code changes were made to correct this error.

3.2 JTLS-2021-15410 TEMS Shared File Issue

The Tactical Electronic Intelligence Message Service (TEMS) uses a temporary file to generate headers and footers for its ELINT messages. This file had a generic name that, when running multiple TEMS instances, would attempt to be shared across those instances. This caused unexpected behavior and crashes.

The temporary header/footer working file used by the TEMS has been modified so that each instance of the TEMS has its own unique file.

3.3 JTLS-2021-15411 DDS Report Wrong Total Personnel Number

The DDS Combat System Rollup Report did not show the correct Total Personnel number.

Only the CS SPECIAL CAPABILITY of PERSONNEL was considered as Personnel Count. The problem was corrected by including the CS SPECIAL CAPABILITY of both PERSONNEL and CREW as Personnel Count.

3.4 JTLS-2021-15413 Checkpoint Available Diskspace Delay

At the start of a checkpoint, the amount of time before starting to write the checkpoint files kept getting longer.

The *spacecheck* script determines if there is enough diskspace available to write the checkpoint files. To do this, the script creates a list of all of the checkpoint directories, then determines which one is the largest and applies a multiplicative factor. This was logical when disk space was limited, but given the size and cost of modern disc sizes, we simplified the available space determination by simply ensuring that at least 5% of the disk was available.

3.5 JTLS-2021-15415 Database Connection Failure

The code used to check that the Database Development System (DDS) Graphical User Interface (GUI) table definition files are accurately defined ran into a problem when run at a user site. The program failed while trying to connect to the database. The code was hard coded to access the R&A Postgres server.

The hard coded server name was replaced with the real password info that the parsing .pgpass file class provides.

3.6 JTLS-2021-15437 MIDB Tool Empty Values

The Modernized Integrated Database (MIDB) Tool generates SQL scripts to insert records into a JTLS-GO scenario database. The syntax for Oracle and PostgreSQL database systems have subtle differences; PostgreSQL does not accept quoted empty values in INSERT strings, while Oracle does.

The MIDB Tool has been updated to write NULL values instead of empty, quoted values.

3.7 JTLS-2021-15439 OPM Regeneration From Checkpoint

JTLS-GO 6.1.0.0 allows units to start with less than their full load of supplies, records supplies processed through MHE, and to allow air tracks to be reacquired with the same track name. The attributes required for these capabilities were added to the CEP checkpoint, but they were not added to the Scenario Initialization Program (SIP) code. The SIP also reads these attributes when re-generating OPMs from a checkpoint.

The SIP code was updated to handle the new attributes.

The CEP Preamble was written so that the attribute to track supplies processed through MHE was both a uniquely defined attribute and also used as a define to mean sharing memory with supplies due in. This would cause the uniquely defined attribute to never be used and everywhere in the code that the attribute was set it would over-write the supplies due in. The define to mean was removed from the CEP Preamble.

3.8 JTLS-2021-15440 COP Detections With ATO Lite Flag On

The ATO Lite Flag is designed to quickly run an ATO in the CEP while attached to a Common Operational Picture (COP) to view flight paths and behaviors. The flag skips many events that occur in the CEP so that CPU is not wasted on items that are of no interest for this function. One of the processes that is currently skipped is the processing of detections. Own-Side detections should still be processed in order to properly populate the COP.

The entire Air Mission Detection logic was previously skipped when the ATO Lite Flag was enabled. This has been modified so that air mission detections are now processed for detectors that belong to the same Side as the air mission. This will provide the blue detections for a blue ATO to populate the COP.

It should be noted that when restarting from a checkpoint there might be residual air missions from other sides that are of no interest to the operator but will still be processed for their own Side detections. This additional and unnecessary processing is considered to be minor.

3.9 JTLS-2021-15442 Political Country Priorities

Political Countries (POC) are represented as closed polygonal geographic region. Each region is given a priority with a lower index number representing a higher priority (1 is higher than 2). This is to handle situations where a country is totally surrounding by a larger country. For example, consider the situation in which Vatican City is surrounded by Italy. Within JTLS-GO, when determining the country associated with a given location, the code searches through a set of all represented POCs and finds the first country polygon in which the location exists.

The set of POCs was not ordered properly. So the lower priority numbers were not placed first in the set of POCs. Thus a location that was in Vatican City was coming to the conclusion that it was in Italy. This is exactly the opposite of the desired use.

The set, which is defined in the Preamble, was changed to specify explicitly that ordering is by LOWER priority number, which results in items being filed in the order 1, 2, 3, and so on.

3.10 JTLS-2021-15444 Country Code UIC And LOGFAS Usage

A database is built with a specific Country Code format, such as NATO Tri-Graph. A configuration-managed translation file exists to translate a Country Code from the database format to another format, such as NATO Di-Graph. The user also has the ability to provide an override and addition version of the Country Code translations by adding it to the $JGAME/data_site directory. These Country Codes are used in model Intelligence Reports, LOGFAS Profiles, UIC Generation and other places within the JTLS-GO system.

When generating the LOGFAS Profiles, which expect a NATO Tri-Graph to be used as part of item names, the system Country Code is not translated from the database format to NATO Tri-Graph. The same issue exists when generating UICs for units, which should start with a NATO Di-Graph. The translations exist and are available, and the system should therefore provide the Country Code in the expected output format for these functions.

An examination of the code showed that in the majority of cases the Country Code is translated into the proper format for the usage. The only exceptions found was the generation of LOGFAS names and the generation of UICs. The code was modified in these two functions to provide the desired format. The translation routine, which was returning an empty string when a translation did not exist, was also modified to return the database code when a translation did not exist. Other routines were changed to provide a cleaner interface to the translation capability.

3.11 JTLS-2021-15446 ATOT Linking Data Missing Meta Files

As part of initialization, both the SIP and the CEP write out a default linking file for the ATO Translator. A separate file is written for every Force Side. The file is written so that the Synapse can deliver it to the WHIP ATOT module when it starts up. However, a corresponding Meta file, which the Synapse uses to identify the file and recipients, is not being written. This causes the Synapse to reject the linking files and not send them to the ATOT module.

A 3-line Meta file was created to match each of the Force Side linking data files. The Meta file identifies the link data file and its recipients.

3.12 JTLS-2021-15448 Track Reacquisition By Ground Based Radar

Air tracks that should be reacquired by ground-based Link 16 radars are not being reacquired and reported to the Link 16 JOI. Upon reacquisition, the model is passing the previous track name and network, but it should also be passing the current track name. By only passing the previous track name, the acquisition routine assumed the track was already acquired. By passing the current track name, when it is the null string the acquisition routine attempted to assign a new track name, with preference given to the previous track name, which will trigger the reacquisition.

AWACS missions that possessed a Link 16-capable ground search sensor were picking up detections as they flew their ingress routes, but were never releasing those detections. Furthermore, once detected by a Link 16 radar, ships were reevaluating their detection probability with each successive detection. This did not match with the air mission logic that says once detected by a Link 16 asset, the object remains detected as long as you are covered by a Link 16 asset.

The code was modified to pass both the current track name and the previous track name to the acquisition routine.

The issue of failure to release naval tracks was traced down to the code that processes the periodic C4I updates. If there were no sensors covering the naval units, then the routine to process detections and identify old Link 16 tracks was never called. This was corrected.

Code was also added to the detection routine to check if the detector is a Link 16 asset and if the ship has a current Link 16 detection. If so, the new detection is automatic.

3.13 JTLS-2021-15450 LC2IS Unit Surname

A new field has been requested for unit names in the SIF file generated by the LC2IS Message Service (LC2MS). The field is the surname of the unit, which will be filled with the unit short name. This is a duplication of the name field, but is still a requirement.

The additional tag and entry was added to the generated SIF files.

3.14 JTLS-2021-15453 Error Pushing Orders

Checkpoint files were moved from the exercise game to a secondary system for testing. Due to a mismatch in systems, a keyword error occurred when pushing orders. The CEP informed the user that an error occurred with the current file and provided a line number. The name of the current file had scrolled off the saved lines of the terminal and the line number was wrong, leaving it difficult to determine the problem.

Before providing the feedback mentioned above, the CEP was attempting to write an error message to the Controller detailing the issue. Because the message directory had not been created, this caused a crash. This issue had to be fixed before the user was provided the details about the bad keyword.

The error message was expanded to provide the name of the current file and to provide the proper line number of the problem.

In order to ensure error messages are delivered, rather than causing a crash, the code was modified so that the necessary directory and files for messages is created early in the start/restart procedure.

3.15 JTLS-2021-15454 MIDB Tool Unique Target Names

The MIDB Tool inserts Targets from MIDB into a JTLS-GO scenario. These Targets must have unique names in order to be processed. The MIDB Tool was failing to give these Targets unique names in some situations.

Additionally, Targets would not be inserted because their associated category code did not yet exist in the JTLS-GO database.

The MIDB Tool has been fixed to give inserted Targets unique names. The tool also ensures that a Target's category code exists in the JTLS-GO database before inserting the Target.

3.16 JTLS-2021-15456 Minor Order Changes

A small labeling error existed in the Manage Movement Network order for the Modify Road Arc and Delete Road Arc groups. The field for the arc name was labeled Rail Arc instead of Road Arc.

The label was corrected in the Manage Movement Network order.

NATO/JWC requires a method to quickly set a large number of tracks to either have their Special Designator Flag set or to have the flag cleared.

Spreadsheets were added to the Track C4I Special Flag order to allow the clearing of the Special Designator Flag or the setting of the Special Designator Flag in Link 16 J2.2 and J3.2 messages.

When adding Combat Systems to a unit via the Set Individual Unit Parameter order the CEP will also add supplies for the Combat Systems if needed. This is not discussed in the order help.

The Set Individual Unit Parameter order had help added to the Combat System List field that explains the adding of supplies if needed.

3.17 JTLS-2021-15462 AAR Tables Missing Detection Events

An intelligence update shared by Side A with Side B is not being put into the AAR tables as a detection event with related detection items.

The code was modified to create a detection event with the AAR and to gather the shared objects and report them as detection items to the AAR.

Formation ships use some of the same code when a minefield is encountered. Since the minefield was already being reported to the AAR by other means, it was decided to rework the minefield reporting code to not cross over into the sharing intelligence between Sides code.

3.18 JTLS-2021-15464 HRU Patrol Repeatable Route Ends

An HRU was on a repeatable route patrol task with the “new tasking” end time selected. The HRU stopped after one circuit and entered a defend posture with a defend mission instead of repeating the patrol route.

There was existing logic written to allow the HRU to patrol a single route, instead of a repeatable route, that was not fully implemented. The continue until “New Tasking” option sets the Pending Execution Task's End Time to zero, and the model interpreted the patrol to be complete because the HRU had no route points remaining in the patrol route and the PET.END.TIME was zero. This logic has been removed and the HRU continues the repeatable route when given the continue until receiving “New Tasking” option. The HRU single route option was also tested and works as desired.

3.19 JTLS-2021-15466 Techtool Copy Order Groups To Remote Host

The Techtool was not able to display the order tree or copy orders over to a remote system.

This issue occurred when the directory structure that held the order groups and shared orders changed. The Techtool was updated to use the whip-list.xml file to build the order group copy trees. The Techtool was searching for the remote scenarios whip-list.xml file on the local file system and was not able to build the remote order group tree or perform copies. The Techtool can now access the remote file and perform order group copies to and from the remote host.

It was not possible to copy orders from a single-sided WHIP to a multi-sided WHIP (for example, the US WHIP could not copy over orders to a multi-sided WHIP that accessed the US side). This has also been corrected.

3.20 JTLS-2021-15469 UOM Lifting Remaining Unit Supplies

When moving a unit from one location to another, either as part of an Amphibious Operation or an Airlift, the combat systems owned by the unit along with their needed supplies are moved first. Once that is complete the model moves all remaining supplies. When determining the remaining supplies that can be placed on the movement asset, the model did not properly consider the unit of measure for WET supplies, such as fuel. The result was that the model was attempting to move 10,000 gallons of fuel and considered it as 10,000 tons of fuel.

Wet supplies are now converted to an appropriate weight using the parameter TONS.PER.GALLON. This problem only existed when moving left over supplies. Wet supplies were properly converted when moving Combat Systems and the needed supporting supplies.

3.21 JTLS-2021-15470 Terminating Push On Game Restart

A capability to terminate a push on game restart has existed for many years. This was accomplished by inserting an extra hash character, #, after an order in any of the checkpoint order files that were being pushed. When the hash character was encountered the pushing of orders for that file, and all other files in the push, would terminate. This ability has been lost, and the current ability terminates the push of the current file, but continues pushing successive files in the run sequence. This makes it difficult to terminate a push at a specific time.

The code that continued pushing successive order files was a mistake that was introduced when the format of the runs file was updated. The old capability has been restored so that an extra hash after an order will now serve as an end of the entire push.

3.22 JTLS-2021-15472 Destroyed Link 16 Dropping Detected Objects

When a Link 16 network is destroyed, the code looks for all units, targets, and air missions that are part of the network and removes them. The code does not look for unit, air mission, and missile tracks that have been detected by a network asset and break the detection link. This leaves the model in a high risk situation for a crash.

The code was modified to go through all unit, air mission, and missile tracks, and the track is dropped if the detection is on the network being deleted. The code also checks the previous track detection, which is used to reassign the same track number for a reacquisition, and if it is on the network then the values are cleared.

3.23 JTLS-2021-15474 IMT Columns Too Short

Several IMT screens displayed data showing abbreviated string values for some screen columns. The result was an incomplete data value for items like combat systems and supplies and would confuse users when different data values had similar or identical beginnings of names.

The IMT screen definition files were modified to correct character display limits for data values that were cut short.

3.24 JTLS-2021-15476 Techtool All Order/Utilities Do Not Import

If a WHIP does not import all orders/utilities in a group because it does not have those orders/ utilities in its orders menu, the user is not notified.

The user is now notified in a dialog window when copying order groups between WHIPs that do not have the same order menu, that it may result in an incomplete copy of the orders.

3.25 JTLS-2021-15477 IMT Combat System Resupply Category

When viewing the Combat Systems for a unit, the resupply category column is always blank.

There was an error when setting the resupply category in the code. The JODA was erroneously informed that the object was an Inventory Item when it should have been marked as a Supply Category. The problem has been fixed.

3.26 JTLS-2021-15478 Repeated Clear Naval Mines Message

A ship was ordered to clear naval mines in a minefield target. The model created a Clear Mine task for the ship, but the task was not executed. A non-compliance Player message was generated multiple times stating that the ship had no mine clearing capability for the minefield type. The non-compliance messages continued to generate over & over again in the Message Browser.

The ship's SUP Mine Clearing Capability (MCC) for the minefield type (in this case, Drifting Contact) held a per-day clear rate of zero for all depth zones. Consequently, the Clear Mine task could not be executed. Instead of canceling the task, the model repeatedly tried to execute the same task and generate the same non-compliance Player message.

The logic was changed to cancel the Clear Mine task when the per-day clear rate is zero at all depth zones for the minefield type or when there is no MCC specified for the ship SUP (i.e. the ship is not capable of clearing any minefield type). Two new task cancellation messages were added to notify the Player.

3.27 JTLS-2021-15480 Javamenu Inconsistent Issues

The Javamenu scenario selection window stayed open after the user clicked the "OK" button. In addition, the terminal window closes after the user executes a script, without asking for a carriage return from the user.

The code was modified to close the Javamenu scenario selection window and to keep the terminal window open after executing a script.

3.28 JTLS-2021-15482 WHIP Foreign Object Invalid Menu Option

A detected foreign air mission displayed the option to start an IMT screen for its Weapon Load List from the object's context-sensitive menu. Although the IMT did not display any information, this option should only be available on a WHIP of the same Force Side, or from a Controller WHIP.

The context-sensitive menu configuration file was corrected to use the proper Force Side permission attribute for the defined Weapon Load List option. A similar correction was applied to the Show ATO Message menu option.

3.29 JTLS-2021-15484 WHIP IMT ICAO Search

On the Airbase IMT screen, users were not able to perform a search on the ICAO column. Additionally, existing ICAO field search functionality did not work.

The behavior of IMT search attributes was corrected.

3.30 JTLS-2021-15487 Add TRIPP Layout Option

The Layout Save/Restore option was not presented in the TRIPP Tools Menu.

The Layout Save/Restore option was added to the TRIPP Tools Menu. Users can now save the layout in the WHIP and restore it in the TRIPP, and vice versa.

3.31 JTLS-2021-15488 Cancel Mobility HRU List Causes Crash

If a downed pilot is placed in the water and the user sends a Mobility Mission to pick up the pilot, rather than extract the HRU from the water, the model crashes.

The air mission could not land on the water to pick up the HRU. This causes the task to cancel. The logic was set up for canceling an Aggregate Resolution Unit (ARU) Load Unit Task, but the logic is different for canceling a Load Unit Task for a HRU. The model was looking for the Offload Unit task, but there is no requirement for a Mobility Mission to have such a task for an HRU. The logic was corrected to account for the possibility that there is no Offload Task for an HRU.

3.32 JTLS-2021-15491 L16MS Crashes On Network Deletion

The Link 16 Message Service (L16MS) processes creations, updates, and deletions of Link 16 Network objects. When a Link 16 Network was deleted, the L16MS failed to process the action and crashed.

The L16MS has been updated to properly process a L16 Network deletion from the JODA.

3.33 JTLS-2021-15493 L16MS JREAP Message Headers

A Link 16 message packet consists of a header and the accompanying message body containing track data. When the L16MS is configured to use the JREAP header, and the Link 16 message feed is toggled off, the L16MS continued to send JREAP packet headers, but without the message body content. This can confuse receiving C4I systems.

The L16MS was modified to not sent JREAP message headers if the feed has been toggled off.

3.34 JTLS-2021-15496 GeoTiff Layer Stops After Layout Restoration

After a user restored their saved WHIP Layout, the GeoTiff and CADRG Layer did not match the restored Map Projection.

The behavior of the restored map layout layers panel was corrected.

3.35 JTLS-2021-15498 Match Units On Map With Selectable Names

A relatively new JTLS-GO capability is to select the WHIP object name for some objects that have multiple names. For example, an air mission has its assigned name and its "Call Sign" name. When a user selected a new display name from the WHIP Symbol panel, the names associated with the object on the map changed as requested by the user. However, when a user clicked on top of a group of units, the names shown did not match the selected display name choice.

When a user selects a group of units on a map, the display names shown are of the type selected by the user. In other words, if the user indicated "Call Sign" names should be displayed, then the WHIP displays the call sign name when clicking on a single object or a group of objects.

3.36 JTLS-2021-15499 ATOT Does Not Use Mobility Stop Locations

Whenever an ATO Mobility mission uses a location or airspace location as a stop for tasking, the ATOT reports the location as an invalid ICAO and fills the associated order with an empty field for the stop.

The ATOT only checked the stop location as an ICAO while looking for a valid ICAO name. However, for rotary wing missions the stop may be a latitude/longitude location or be a reference to an airspace name from the ACO message.

The ATOT code was changed to first compare the stop name against linked ICAO names, then any airbase names, and finally any names of airspaces. In the case of a specified airspace name, the ATOT will give a warning report, "using a location" rather than a unit name for the stop.

3.37 JTLS-2021-15501 ATO Parser Tasks From Refuel Locations

The ATO Parser is not able to identify routing locations as part of a mission's tasks when the name of the location is the same as any of the names used for refuel locations within the *mission's records. The resulting mission data excludes the tasks by the same name as the refuel locations.*

It is unusual to have a task and a refuel location both using the same airspace name within a single mission. It may be that a CAS mission is tasked with protecting the refueler from which it is also taking its fuel. In order for the Parser to differentiate the refuel locations from the tasks, it was modified to use the refuel time along with the refuel location name provided in the ARINFO records.

3.38 JTLS-2021-15504 WHIP Vocabulary Fields Without Entries

The user was unable to bring up the Create Unit order when running the Civilian game. The Aircraft Class vocabulary field had no entries.

The vocabulary order field was changed to allow an order panel to proceed to open with only the empty option for any vocabulary data field with no entries.

3.39 JTLS-2021-15505 JDS Packet Data Mixing

A major difference between JTLS-GO 6.0 and JTLS-GO 6.1 concerns the sending of data to the JODA. Within the JODA, there is the concept of perceived and common attributes. A common attribute has a single value that is shared to all sides that want the info. A perceived attribute has separately tracked values for each side and the Controller and will only send out the value for the side(s) requested.

In JTLS-GO 6.1, the model handles the issue of mixing common and perceived attributes by separating the two types of updates and producing logic errors if common and perceived data are being mixed in a single update.

There were several places in the CEP code where these attributes were being sent in a single update and logic errors were occurring.

Routines which updated several attributes in a single update were reviewed and the attributes were set in separate updates.

3.40 JTLS-2021-15506 ATO Parser Withholding Call Sign In Reports

The ATO Parser lists the mission name whenever Warning 115 is invoked by a mission marked as residual in the ATO. The mission is better identified in some cases from its call sign. Having the call sign aids troubleshooting.

The call sign for any mission having residual status in the ATO will be included in the Warning 115 report.

3.41 JTLS-2021-15509 JOI Module Exit Menus

JTLS-GO users are trained to use the explicit Exit options available from the file menu of programs, such as the Interface Configuration Program and Web Services Manager. This reduces the chances of closing a critical service or program such as the CEP. The JOI modules (Link 16 and OTH-Gold) are missing these exit options from their main menu.

The Link 16 and OTH-Gold WHIP modules have been updated to include Exit options in their main menus.

3.42 JTLS-2021-15511 ATOT Improper Order Option Location

The ATOT processes ATO package missions, but any of these missions that are carrying out the attack on an airspace area without a specific target object are built incorrectly. The ATOT creates the associated Attack Package Mission order with the Location option selected, but without a location.

The ATOT was modified to compute the center location of any airspace provided by the ATO mission and enter this location latitude/longitude as the Location option of the Attack Package Mission order. Also, when a GTGTLOC record is not provided for the mission the ATOT uses a default target Type list as provided by the user in the targets linking in the ATOT.

3.43 JTLS-2021-15515 Reformat Spreadsheet Program BE Facilities

A user attempted to use the Reformat Spreadsheet Program to produce a Manage DSA Order using the "Create New BE Facility DSA" option. The RSP produced an error and failed to create the JTLS-GO Order because BE Facilities were not recognized by the RSP.

The RSP has been updated to recognize known DSAs. When the RSP connects to the JODA, BE Facility objects are downloaded as well.

3.44 JTLS-2022-15517 WHIP Consistent Wild Card Searches

A wildcard can be the first character when using the Map search field, but cannot be when using the Command Hierarchy or a Logistics Hierarchy search field.

This inconsistency was corrected by allowing wildcards to be used as the first character when using Command Hierarchy and Logistics Hierarchy search fields.

3.45 JTLS-2022-15518 Synapse Log Entry Missing Date/Time Prefix

The Synapse makes error and informational log entries having a date and time prefix. This is true for all services,. However, for errors with emailing, the Synapse does not include the prefix on the related error log entries.

The JTLS-GO library of functions includes a macro definition for streaming errors and informational data to the log. This macro inserts the time and date automatically when the caller uses the macro for 'logerr' or 'loginfo'. Most services are using this library, but the library for handling the email feature employed by the Synapse does not. This library uses the c++ standard cerr and cout, which does not have any log entry formatting.

The emailing library has been changed to include the library that defines the macro for inserting the time and date in the log entries.

3.46 JTLS-2022-15519 Synapse Nonstandard Log Entry Format

All services use a time-date prefix for any of their error or informational log entries. The formatting for these log entries is defined in the common jtls library. The Synapse does not use this library but mimics the formatting under its own macro definitions for its log entries. In order to coordinate a consistent format for logging in all services, the library that defines the format for logging should be used.

The Synapse's own definition of the log entry format was removed, and the special JTLS-GO library for formatting any log entries was included in the Synapse for consistency.

3.47 JTLS-2022-15538 Warning 1254 Corrections Nonfunctional

Some of the automatic corrections for Warning 1254 do not work.

Two routines use Warning 1254 to report issues with Bring To Theater values for air fuel supplies. Two of the parameters in one of the routines were reversed, causing the problem. The code was corrected.

3.48 JTLS-2022-15539 Missing WHIP Panel During Recall

If a user saved a WHIP Order with an Alert option and recalled it, the Leave Option fields were not visible. The user had to click on the Alert radio button to make the fields reappear.

The WHIP code was corrected to solve the invisible field problem.

3.49 JTLS-2022-15541 ELINT Code Missing Cases

A sensor using the type .GUIDANCE.GPS.JAMMER did not have a translated ELINT Code. The CEP would simply produce a logic error stating “Inconsistent entity to assign ELINT Code”.

.GUIDANCE.GPS.JAMMER was an existing type defined in the preamble, but there was no defined case for this type in the routine where the logic error was being produced. .GUIDANCE.GPS.JAMMER and several other types were added as case statements to the routine ELINT.FUNCTION.CODE.

3.50 JTLS-2022-15542 WHIP Force Flow Graphic Drawings Collapse

When formatting the force flow arrow graphics, the arrow would collapse in size after confirming a color change. This occurred after the initial drawing was edited. The format editor window was not always loading the correct dimensions for the current force flow arrow.

The force flow graphic editor panel was corrected to consistently open with the correct values for the current force flow shape. A flag was also added to set new values to the shape only when any of the dimensions are changed by the user, and not otherwise.

3.51 JTLS-2022-15544 Add Airbase Long Name To IMT

The airbase long name should be one of the available columns on the Airbase IMT, to make identifying the airbase easier for operators.

The airbase long name is now shown on the Airbase IMT.

3.52 JTLS-2022-15546 Techtool Order Group Crash

The Techtool crashed on initialization when it encountered an order group for a deleted WHIP.

The Techtool will no longer include order groups for a WHIP that has been deleted in the Order Group Editor Tree. Order groups from deleted WHIPs will now be skipped over.

3.53 JTLS-2022-15547 Command Authority Non-Arrived HRU

The model crashed when processing a Change Command Authority Order for a future arriving unit that owned an HRU.

The model crashed when it was accessing the HRU's perception information. HRUs are not given perception information until their arrival event occurs.

A check was added to make sure that the HRU parent was in the game. If not, the HRU command authority data is not created. When the unit does arrive in the game and its child HRUs are created, they will receive the command authority of the parent unit.

3.54 JTLS-2022-15548 TPFDD Out Submarine Being Detected

A stationary submarine was TPFDDed out of the theater and it continued to be detected by sonars within the region.

The issue was caused by the new C4I PROCESSING EVENT in which ships that are stationary can be periodically detected by other stationary ships. When the submarine was TPFDDed out of the theater, the event which processes continued detections was not stopped. The event is now canceled when a ship leaves the theater.

3.55 JTLS-2022-15549 ELS Crash Change In Mission Return Location

A UAV air mission was flown from a ground unit. While in flight, the return location was changed to a different ground unit. The ELS crashed as a result of this parameter change.

The ELS crashed while attempting to change the parent squadron for the air mission. The code was expecting the new squadron to already have an assigned aircraft type. This failed because there were no aircraft at the new return location. Code was changed was made to use the aircraft type from the mission instead of using that from the unit.

3.56 JTLS-2022-15551 Crash Airlifting Ground Unit With UAV

The model crashed attempting to determine how many UAVs could be lifted by a mobility mission. UAVs are represented as aircraft and have no weight, causing a divide by zero error.

The code was changed to assume that all UAVs can be placed on the mobility mission because, they have no weight. This is not a perfect solution, but until new data can be added to the database, this is the only workable solution.

3.57 JTLS-2022-15552 Crash Depleting Weapons From Air Mission

A user submitted an Magic Air Ops Order to deplete all weapons from an air mission, and the model crashed.

The code was incorrectly accessing an incorrect variable when attempting to access the mission’s weapon set. The proper variable was used.

3.58 JTLS-2022-15554 Crash Updating Late CAS Requests

The model crashed when a CAS Request was being updated with the JODA as a late request. The crash was caused by passing a "Y" character string to the CEP C code functions that handle JODA updates and attempting to cast it as an integer "1".

The code was changed to send the number "1" instead of a character to the JODA interface, removing the attempt to convert a character into an integer.

3.59 JTLS-2022-15556 AAR Not Connected Messages During Start

When the CEP is finished reading in a database, it will schedule initial events. This is done before the CEP connects to the JODA, and the act of scheduling the initial events generates a lot of AAR traffic.

Because the AARC, which connects to the CEP via the JODA, is not connected at this time, a number of messages are created that warn that the AAR Collection Flag is on but that there is no connected AAR Client. This leads many Tech Controllers to assume that the model is waiting for the AAR Client to connect or that there might be an error, when neither is true.

If the JODA is not connected, the AAR warning messages being queued are now suppressed. Once the JODA connects, any new messages will generate the standard warning message.

3.60 JTLS-2022-15558 JXSR Crash Processing Order Group Filter

The JXSR uses order group filtering to decide which group of fields to provide to the WHIP while it is applying known fields in an order. When the order panel allows the user to have a deleted object in one of the fields used in a group filter, the JXSR will crash. This can happen when an order panel is filed with a air mission name and the mission has completed, but the order panel remains open with the name of the old mission.

The JXSR uses the name of an object provided in an order field from which a group filter is composed to locate an object having the specified name. If an object is not found, the JXSR has nothing to pass as the argument to the filter function. The argument is left empty and the filter function crashes while using the empty argument.

This code was modified to return an invalid index rather than call the filter function. The WHIP's order panel handles the invalid index properly and the JXSR does not crash.

3.61 JTLS-2022-15559 JXSR Crash Updating Super WHIPs

The JXSR sometimes crashes when an update publication for a Super WHIP contains a deleted object.

Whenever the JODA sends an object delete to the JXSR, the JXSR looks in its pool of pending creates and updates. The JXSR attempts to remove any creates or updates, because the object will be added to the pool of deletes. Once the creates, updates, and deletes are adjusted, the JXSR is free to send all of these to the associated WHIP as a publication. The pool of creates, updates and deletes are stored by Side perception, so any additions or removals are applied for each Side to which the WHIP is subscribing.

Super WHIPs that include the Ground Truth side were not getting the JXSR to apply the delete properly for the Ground Truth side. Any Ground Truth object deletes were not removing the deleted objects from the create and update pools. When the publication is sent, the JXSR would dereference a pointer to the deleted object and crash.

This has been modified so the JXSR includes the Ground Truth perception whenever a deleted object is included in a Super WHIP publication.

3.62 JTLS-2022-15560 JXSR UI Objects Constraint Filter

When the user selects an Unidentified (UI) object as a field of an order that includes a constraint with another field, the other field is returned as "invalid object". The other field should be returned as a list of valid objects.

The JXSR stores objects in a hash using a key, which typically is the name of an objects in the game. When the user selects an object in one field that is constrained by another field, the JXSR searches the hash for the object using its name. The JXSR was storing the objects in the hash by its true name, so searching for a perception name would fail and return "invalid object" for the other object. The JXSR now adds objects to the hash using the perception names for an object.

3.63 JTLS-2022-15564 Completed Mission Still Being Shadowed

Air Mission A was instructed to shadow Air Mission B, which puts Air Mission A in the Guiding Mission Set of Air Mission B for the duration of the shadow task. Air Mission B landed and completed its mission and scheduled its removal from the game. Air Mission A was still shadowing Air Mission B at this point. When Air Mission B was removed from the game and destroyed it caused a crash because Air Mission A was still in the Guiding Mission Set of Air Mission A.

When a mission is completed and is being destroyed, the code checks for other missions that are shadowing it. Any such missions that are found now have their shadow task canceled and will execute their next task.

3.64 JTLS-2022-15565 Checkpoint Convoy Update Crash

A mismatch existed between writing and reading an intelligence update on convoys. The update contains information on which Transportation Classes were seen by the detecting object. These were written out as Transportation Class names, but were being read in as Combat System names. This would cause a model crash when restarting from a checkpoint when the update was processed.

The code was modified so that the read and write match, using the Transportation Class names. Other code that accessed the intelligence update were also modified to ensure they were also accessing a Transportation Class, rather than a Combat System.

3.65 JTLS-2022-15566 Divide By Zero Checking Boundary Restrictions

A crash occurred when a naval unit was looking for any possible boundaries when performing a move of less than one meter. The crash happened when attempting to create a Great Circle Arc for the two points which were almost, but not exactly, the same.

The code was already protected from two points being exactly the same, but not from two points being almost the same. The protection check was modified to allow a 1-meter resolution to consider the points the same.

3.66 JTLS-2022-15569 Crash Adjusting Barrier Points

A crash occurred in the JODA interface code when the Controller sent an order to adjust the points of a barrier. Adjusting the barrier points requires two pieces of information on a single

JODA update packet: the new number of points and the array of points. These two pieces of information were being sent separately, losing the link between the pieces.

The code was modified so that the two pieces of information are sent as a single update packet to the JODA.

3.67 JTLS-2022-15572 Target Subcategory Update Failure

The DDS Target Table acts as a dynamic lookup table and is managed by database triggers. When DDS users update the subcategory data fields of individual Targets, the subcategory data field of the Target Table fails to be updated accordingly.

The PostgreSQL database triggers and related functions were modified so that when the subcategory data field of a Target is updated, the subcategory data field of the Target Table is now updated accordingly.

3.68 JTLS-2022-15574 WHIP Help Posture Symbols Missing

The Unit Posture symbol images were missing from the WHIP help.

The links to the images were corrected.

3.69 JTLS-2022-15576 ATO Parser Adjust Start Time Checkbox Issue

Mousing over the Adjust Start Time checkbox or label of the Java-based ATO Parser would reset the associated adjust start time data.

The Adjust Start Time checkbox was modified to only register specific mouse events (like clicks).

3.70 JTLS-2022-15579 Crash Adding Weapons To Air Mission

The model crashed when a Magic Air Ops order was sent by the Controller to add weapons to an air mission. The code expected a fuel supply to be included on the order, but because only weapons were being added, and not fuel, this did not exist.

The code was modified to properly handle changes to weapons alone, changes to fuel alone, and changes to both.

During the fix it was noted that, in the code, an attempt was made to provide fuel from the Player's designated specification first, and if that was not available to use other aircraft-capable fuel if available. This code was not working as intended and was redesigned to properly prioritize the user-specified fuel, and then check all other fuel categories.

3.71 JTLS-2022-15580 Replay JXSR Crash Sim State Current Time

The Replay JXSR crashed after it obtained an invalid real value, used to represent the simulation state current game time, from a Delta log file, created by the Logging JODA. This value is converted into simulation game days by calling a system routine localtime(), which crashes if the provided real value is invalid.

The simulation time obtained from the Delta file was an invalid "nan" (not a number) value. The library routines for reading the Delta file were changed to report the invalid value in the error log and not use the current time value that was read. The recording library used by the Logging JODA was changed to report the invalid value and not write the invalid time value.

3.72 JTLS-2022-15581 JOIs Do Not Display Side Name

The JOI web services state the perception for which they have been configured in the Web Services Manager Status tab. This perception value used the JDSP-enumerated value, but not the scenario-specific Side name. This required the user to then convert that value to the scenario Side manually.

The JOI web services now list the scenario Side name and JDSP-enumerated value under their perception value.

3.73 JTLS-2022-15582 TEMS PRI Field Format

The TACELINT Message Service (TEMS) was sending TACELINT messages with invalid PRI data. This specifically applied to the level of decimal precision for pulse rates and sending empty pulse rate values.

The Tactical Electronic Intelligence (TACELINT) Message Service (TEMS) was updated to set pulse rate decimal precision to 4 decimal places, and empty values are now handled by sending a '-' character in the PRI field.

3.74 JTLS-2022-15583 Order Group Verification Dialog Fails To Save

If an order group fails verification in the Order Group Editor, a popup is displayed containing the orders that need attention in some way. Users are able to remove orders from the order group using this popup. However, removing orders did not trigger the associated order group data to be saved to the server as well as locally to the WHIP.

Order groups are now saved to the server when removing orders from an order group using the verification popup dialog.

3.75 JTLS-2022-15584 AAR Visual Report Lists The Month Incorrectly

The AAR Visual Report List was displaying dates off by 1 month. This is due to the fact that in Javascript, years and days are 1-based, while months are 0-based.

The AAR Visual Report List was updated to account for Javascript's date representation quirk.

3.76 JTLS-2022-15585 WNT Message Very Large Dialog

The WHIP Notification Tool (WNT) displays a confirmation dialog after sending a notification. The dialog contains the name of all WHIPs to which the message was sent. The dialog did account for displaying a large number of WHIPs, so the dialog could span beyond the screen's visible area.

The WNT's message notification dialog was modified to have a static size and use a scrollable view of the notified WHIPs.

3.77 JTLS-2022-15586 LOGFAS Service Crash On Object Deletion

The LOGFAS/JLOI service would crash when handling an Inventory Item or Equipment Item deletion from the JODA.

The LOGFAS/JLOI service was updated to handling Inventory Item and Equipment Item deletions.

3.78 JTLS-2022-15587 AAR Weapon Usage Report Fails

The AAR's Weapon Usage Report would fail to generate due to a invalid SQL query submitted to the SDR Postgresql database.

The Weapon Usage Report was modified to correctly format the multiple JOIN statements of its SQL query.

3.79 JTLS-2022-15588 DDSC Table Not Removing Rows

When an adhoc filter on a column is applied on a DDSC table, the table is filtered only to show rows that satisfy the adhoc filter condition. When the user changes a column value to something that would violate the adhoc filter condition, the table throws a null error and the row remains empty.

The issue was due to the new PostgreSQL change-data-capture function, which tracks only data that have been changed, instead of the whole row that Oracle’s flashback-query function tracks. Glassfish now directly returns a "delete" command to the DDSC.

3.80 JTLS-2022-15590 Modify JTOI Build Process

The JTOI programs link to third-party libraries that are built locally. The build processes for the JTOIs required the libraries to be built manually.

The JTOI build processes have been updated so their required third-party libraries are built automatically.

3.81 JTLS-2022-15591 Crash Determining If HRU Parent Has Resources

The model crashed when the user attempted to create an HRU that needed Combat Systems that were never available at the parent unit.

The code error was fixed, and an HRU creation request where the parent unit does not have the necessary Combat Systems is now properly rejected.

3.82 JTLS-2022-15606 Crash Destroying Satellite With Information Set

The model crashed when the Controller destroyed a satellite that had a non-empty information set.

The code was modified to empty and destroy the information updates from the satellite before destroying the satellite itself.

3.83 JTLS-2022-15607 Grid Lower-Left Location Not Exact On Restart

The model crashed attempting to remove a unit from the game. The unit was located directly on top of a longitudinal grid line. The unit thought it was in grid 6,5, but the model computed that it was in grid 5,5. The model attempted to take the unit out of grid 5,5 when it was really in grid 6,5.

The model determined that the unit was in grid 6,5 at game start, but after an ASCII checkpoint restart the model determined the unit to be in grid 5,5. This was caused by a slight difference in the value of the lower left-hand corner of the grid, as read in by the model when an ASCII checkpoint is used to restart the game, versus the originally-computed location of the corner when the game is started. This slight difference caused the unit to be computed to be located in a different grid after a restart of the game.

To correct this problem, the lower-left hand corner information is no longer saved as part of the ASCII checkpoint. Instead it is always computed whether the game is starting or restarting. This change results in a change to the ASCII checkpoint file. This means checkpoint files made with previous versions of JTLS-GO 6.0 will not work with this version of JTLS 6.0.

In addition, the model does not need to compute the grid location of the unit when the unit is to be removed from the game. The unit correctly holds this data, and the model should have simply accessed the unit's saved grid data, rather than compute it. The computation was removed from the Remove From Game routine and all other routines in which a unit is removed from a grid set, and the routines now access the saved grid data from the unit.

3.84 JTLS-2022-15609 Fire Artillery Order Has No Weapons

An Amphibious Assault order was submitted to offload an artillery unit. After the unit was completely ashore, including all supplies and weapons, the Fire Artillery order did not have any *weapons to select from the Projectile Type drop-down list. Consequently, the unit could not fire its artillery, even though the Weapon Count IMT showed that artillery rounds were available.*

In the model, each unit's Weapon Owner array holds the artillery, mines, and SSM targetable weapon types available to the unit. This array is used to populate the weapon drop-down lists in several different orders, including the Fire Artillery order.

During the amphibious operation, the Weapon Owner array was not updated as each portion of the unit was offloaded, nor was it updated after all the unit's assets were ashore. As a result, the array did not hold any available weapon types for the unit. To correct this problem, two subroutine calls that update the array were added to the routine that delivers each portion of the unit.

3.85 JTLS-2022-15610 ATOT Fails To Clear Assigned Squadrons

The ATOT interface module allows squadron assets to be assigned to a tasking unit. If the user tried to clear all assigned squadrons from a tasking unit, the process failed and any existing selections were preserved. Additionally, the list of selectable squadrons may fail to display all eligible squadrons when comparing against their aircraft classes.

The ATOT interface module was updated to fix an issue when setting a tasking unit's assigned assets to an empty value. The module was also updated to better represent eligible squadrons when comparing aircraft classes.

3.86 JTLS-2022-15611 AARC System Function On Checkpoint

Exercise personnel wanted to maintain a running backup of the SDR database. The best way to achieve this would be for the AARC to invoke a system script whenever it took a checkpoint. This is actually the same behavior that existed prior to JTLS-GO 6.0, when the AAR database would only contain a single run. This same behavior was desired so that the site could setup site-specific actions to be performed on the SDR database at the time of a checkpoint.

The AARC was modified to make a system call to execute the sdr_checkpoint script when a checkpoint start notification is received. The sdr_checkpoint script, which contained obsolete instructions to perform a database dump, was modified to leave an open section for site-specific instructions.

Note that the exact mechanism to back up the SDR database is a site-specific choice. This change does not provide that mechanism, but only provides the method to invoke the mechanism.

3.87 JTLS-2022-15613 OMA Unique Record Check For Utilities

A Move order was sent with a lead unit specified and a Ground Unit List of other units that should move with it. The same order appeared twice in the Ground Unit List, and this caused a crash when the move was canceled. The order XML for the utility clearly specifies that the unit name should be a unique field, but the OMA did not catch the error and reject the order.

The Order Verification Tool (OVT) library, which the OMA uses to verify orders, was using an obsolete method to check for uniqueness within utility records. The obsolete method would consider two records the same if every field in the two records was the same. The proper and newer method relies on a unique-key attribute of the order fields to compose a unique field or combination of fields. The unique record checking has been updated to use the unique key combination of fields.

3.88 JTLS-2022-15614 Satellite Sensors Not Turning On/Off on IMT

Satellite sensors can be turned on or off individually, but when this is done the results are only visible on the IMT screen for sensors that were part of the original satellite load. Sensors that were added to the satellite do not properly update their on/off status.

Sensors that were part of the original satellite load were given a receiver number equal to the satellite's receiver number plus the sensor index. Sensors added to a satellite were given the next receiver number available, but the JODA initialization code would improperly create the object using the satellite receiver number plus sensor index algorithm. Thus, when the sensor was changed it was attempting to update a receiver number object that did not exist on the JODA. The code was changed to use, assign, and expect the next receiver number algorithm when assigning sensors to a satellite, whether as part of the initial load or as part of an addition.

3.89 JTLS-2022-15617 Mapping JTLS-GO Data To GESI

As part of the GlobalSim federation, data in a JTLS-GO scenario must be mapped into the required structures in a database used by the GESI simulation. These data included the unit names, prototypes, munition names, combat system names, highest-scoring combat systems in each unit, the DIS code assignments, and whether weapons were direct or indirect in nature.

A new subroutine in the alter-data program was added to write the required mapping data into comma separated variable (CSV) text files. A total of four files were generated, for ground units, naval units, squadrons, and munitions. The database builder must run the alter-data option for their scenario, and then manually pass these CSV files to the GESI scenario builder.

3.90 JTLS-2022-15618 Lift Unit By Air Never Finishes

An airlift mission designed to move a unit would never finish if it has extra supplies which are not necessary to support its combat systems.

As the unit was lifting the leftover supplies, the supplies that were being loaded on the aircraft were not properly being subtracted from the unit. This has been corrected.

3.91 JTLS-2022-15621 Transportation Class No Capacity To Carry

The Scenario Initialization Program (SIP) crashed generating Online Player Manuals (OPMs) for Transportation Class data.

The OPM crashed because the Transportation Class being printed did not have any supply capacity to carry records. Code had been added to check for the situation and report it as Error 137.

3.92 JTLS-2022-15628 Air Missions Deleted Immediately After Completion

Air Missions are being deleted as soon as they complete. The current design is that the missions should remain visible in the system until their Mission Report (MISREP) is generated or the ATO Period is complete, whichever is longer.

When fixing issues with passing Common Knowledge and Perceived Knowledge to the JODA, this error was introduced. Only foreign sides lose access to the air mission when the mission completes.

3.93 JTLS-2022-15629 JTLS-GO Documentation Updated

The JTLS-GO documentation suite needed to be updated, to account for the changes that have been made to the system since the previous maintenance release (JTLS-GO 6.0.5.0).

The JTLS-GO documentation has been updated.

4.0 REMAINING ERRORS

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

As JTLS-GO 6.1.0.0 represents a major release of new functionality, all outstanding errors have been reviewed. If the error could not be reproduced, it was considered obsolete and no longer relevant to JTLS-GO. These errors have been removed from consideration for correction at this time. This chapter contains the remaining errors that could be reproduced. As development team assets become available, these remaining known issues will be corrected.

In future maintenance releases, newly uncovered outstanding errors related to JTLS-GO will be listed in this chapter, along with information regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

4.1 DDSC – TMU Line Mode Changes Multiple Grids

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

4.2 DDSC – Multiple Types In DDS History Table

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

4.3 WHIP - Pipeline Not Shown On IMT

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

4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

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

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
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
SDB	Standard Database
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)

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

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

APPENDIX B. VERSION 6.1.0.0 DATABASE CHANGES

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

APPENDIX C. VERSION 6.1.0.0 REPOSITORY CHANGES

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