

JTLS-GO

Version Description Document

November 2023



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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.2.3.0)**

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.2.3.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.2.3.0 is a Maintenance release of the JTLS-GO 6.2 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as a few minor model functionality improvements implemented as Engineering Change Proposals (ECPs). These ECPs are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

JTLS-GO Director of Development
ROLANDS & ASSOCIATES LLC
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940 USA
jtlsgo@rolands.com

Copyright 2023 - Valkyrie Enterprises LLC - All Rights Reserved

[Blank Page]

TABLE of CONTENTS

ABSTRACT	iii
1.0 INTRODUCTION	1-1
1.1 SCOPE	1-1
1.2 INVENTORY OF MATERIALS	1-1
1.2.1 Obsolete/Outdated Documents.....	1-1
1.2.2 Unchanged Documents	1-1
1.2.3 Updated Documents.....	1-2
1.2.4 New Documents.....	1-2
1.2.5 Delivered Software Components	1-2
1.2.6 Released Databases.....	1-4
1.3 INTERFACE COMPATIBILITY	1-5
1.3.1 Support Software	1-5
1.3.2 JTLS-GO Cybersecurity Compliance	1-7
1.3.3 JTLS-GO High Level Architecture Compliance.....	1-8
1.4 DATABASE MODIFICATIONS	1-8
1.4.1 JTLS-GO Using Legacy Default Symbol Set	1-9
1.4.2 JTLS-GO Using New Default Symbol Set.....	1-9
1.4.3 Standard Repository Changes	1-10
1.5 INSTALLATION	1-10
2.0 ENGINEERING CHANGE PROPOSALS.....	2-1
2.1 JTLS-2023-16321 ADJUST SSM SETUP TIME WHEN CONTROLLER CHANGES DATA	2-1
2.2 JTLS-2023-16381 QUICK REFUEL MISSION ORDER	2-1
2.3 JTLS-2023-16423 ADD C3 QUALITY INFORMATION TO NAVAL UNIT IMT SCREEN ...	2-2
2.4 JTLS-2023-16433 ALLOW ICAO CODES TO BE REMOVED	2-2
3.0 SOFTWARE TROUBLE REPORTS.....	3-1
3.1 JTLS-2023-16322 DAMAGE SUPPLIES ORDER MAX TOO SMALL	3-1
3.2 JTLS-2023-16326 DIS CODE REPORT INCOMPATIBLE FORMAT	3-1
3.3 JTLS-2023-16328 ALLOW FULL TWENTY GAME SIDES	3-1
3.4 JTLS-2023-16330 AIR MISSIONS NOT REMOVED FROM JODA	3-2
3.5 JTLS-2023-16332 JODA DOES NOT DISPLAY BOOLEAN FLAG ATTRIBUTES	3-2
3.6 JTLS-2023-16334 AIR MISSION SUPPLY SOURCE PRIORITY	3-2
3.7 JTLS-2023-16337 DDS REPORT TABLE RENDERED INCORRECTLY	3-3
3.8 JTLS-2023-16338 JODA CRASH USING CONSOLE FIND BY NAME COMMAND	3-3
3.9 JTLS-2023-16340 JODA AAR DEBUG FLAG INCORRECT KEYWORD	3-3
3.10 JTLS-2023-16342 INTERCEPTED MISSION PERCEIVED NAME CHANGED	3-3
3.11 JTLS-2023-16344 BASIC JEDI OBJECT QUICK RETRIEVAL BY ID	3-4
3.12 JTLS-2023-16346 UPDATE MISSILE NAME AFTER LINK 16 DETECTION	3-4
3.13 JTLS-2023-16348 NO ROUTE TO SUPPORT UNIT MISSED VERIFICATION	3-4
3.14 JTLS-2023-16350 ELS DESTROYED EXTRA AIRCRAFT ENTITIES	3-4

3.15 JTLS-2023-16352 SYNAPSE BUILDING VALIDATION PATHS IMPROPERLY 3-4

3.16 JTLS-2023-16353 SEND A REQUISITION MEMORY LEAK 3-5

3.17 JTLS-2023-16355 OPM REGENERATION FROM CHECKPOINT CRASHES 3-5

3.18 JTLS-2023-16356 AAR FRACTIONAL OTHER AIRCRAFT KILLS 3-5

3.19 JTLS-2023-16358 LC2MS URL ALWAYS USES DEFAULT SERVICE NAME 3-5

3.20 JTLS-2023-16359 SYNAPSE NOT REPORTING USER AND FILENAME 3-6

3.21 JTLS-2023-16360 RESTORING ELS WHEN JODA CONNECTION LOST 3-6

3.22 JTLS-2023-16363 ELS UNABLE TO REVIVE COMBAT SYSTEM ENTITIES 3-6

3.23 JTLS-2023-16364 ELS MISSING CREW ENTITIES INAPPROPRIATE ERRORS 3-6

3.24 JTLS-2023-16375 AAR AIRCRAFT KILL SUMMARY GROUND LOSSES 3-7

3.25 JTLS-2023-16376 INCORRECT SUNK SHIP MESSAGES 3-7

3.26 JTLS-2023-16378 JOIN FORMATION CRASH 3-8

3.27 JTLS-2023-16380 SHOW DSA ORDER CRASH 3-8

3.28 JTLS-2023-16382 IDENTIFY NUMBER OF OBJECTS IN TERRAIN GRID 3-8

3.29 JTLS-2023-16383 ERRORS IN AAR SCRIPTS TO MINE SDR DATA 3-8

3.30 JTLS-2023-16384 AIR MISSION RETURNING TO MOVED HOME CRASH 3-9

3.31 JTLS-2023-16386 MDP DELIVERY FORMAT DOES NOT SHOW WHIP NAMES 3-9

3.32 JTLS-2023-16387 ERRORS PROCESSING EXTERNAL INTELLIGENCE COLLECTION 3-9

3.33 JTLS-2023-16388 EXTERNAL MODEL DOES NOT GET AIR-TO-AIR IMPACTS 3-9

3.34 JTLS-2023-16389 NAVAL SHIP CLASS NOT SHARED ON RESTART 3-10

3.35 JTLS-2023-16391 BAD CHECKPOINT WITH SAVED MISSION ENGAGEMENTS 3-10

3.36 JTLS-2023-16393 GRAPHICS SYMBOL UPDATE WHEN UNIT ECHELON CHANGES 3-10

3.37 JTLS-2023-16394 AAR OBJECT EVENT REPORT DETAILED VIEW DOES NOT RENDER 3-

11

3.38 JTLS-2023-16395 ADDING MISSION TASKS AFTER PACKAGE OPERATIONS 3-11

3.39 JTLS-2023-16397 KOI NOT READING FILTER FILE 3-11

3.40 JTLS-2023-16399 ELS CRASH WHILE DISAGGREGATING SHIP 3-11

3.41 JTLS-2023-16404 MEAN FOR MESSAGE DELAY CAN BE SET TO ZERO 3-12

3.42 JTLS-2023-16406 TEMS CRASHES ON BAD PULSE RATE 3-12

3.43 JTLS-2023-16407 MDP ERRORS IN INTEL REPORT MESSAGES 3-12

3.44 JTLS-2023-16409 MANAGE ATTACK PACKAGE ORDER HELP TEXT 3-12

3.45 JTLS-2023-16411 SUPER WHIP AUTO-SELECT ORDER FIELDS 3-13

3.46 JTLS-2023-16412 SENDMAIL NOT SETTING HEADER 'TO' FIELD 3-13

3.47 JTLS-2023-16415 SUPER WHIP MAP FINDS OBJECTS FOR ANY PERCEPTION ... 3-13

3.48 JTLS-2023-16416 DASHBOARD CRASH WHEN WRITING DATA 3-13

3.49 JTLS-2023-16418 HOME SQUADRON IN AIR LOSS HISTORY REPORT 3-13

3.50 JTLS-2023-16419 MISSING DATA FILE LINK 3-13

3.51 JTLS-2023-16421 ATO PARSER DOES NOT REPORT CRASH REASON 3-14

3.52 JTLS-2023-16422 ATO PARSER REPORTS TWO ERRORS AS ONE 3-14

3.53 JTLS-2023-16425 UNIT HIERARCHY REPORT LOCATION NOT CORRECT 3-14

3.54 JTLS-2023-16428 LINK 16 SENSOR NOT ON NETWORK DETECTS SHIP ERROR 3-14

3.55 JTLS-2023-16429 ORBITING MISSION ORDERS INCONSISTENT CONSTRAINTS . 3-15

3.56 JTLS-2023-16432 VERY LARGE ATO MESSAGES CRASH ATOT 3-15

3.57 JTLS-2023-16434 ATOT MOBILITY MISSION ORDERS MISSING LOCATION 3-15

3.58 JTLS-2023-16435 ATOT SIDES MISSING FROM UNTRANSLATED ACO 3-15

4.0 REMAINING ERRORS..... 4-1

 4.1 DDSC/WHIP/JOBE - CADRG MAP ZOOM 4-1

 4.2 MHE TARGETS LOADING AIR MISSION CAN CAUSE A CRASH 4-1

 4.3 THE JTLS-GO STRATEGIC LIFT MISSIONS ARE NOT WORKING PROPERLY 4-1

 4.4 TACTICAL GROUND FORMATION ATTACKS DO NOT WORK 4-1

 4.5 ATOT SPREADSHEET LACKS DETAILED FIELD CHECKING 4-1

APPENDIX A. ABBREVIATIONS AND ACRONYMSA-1

APPENDIX B. VERSION 6.2.0.0 DATABASE CHANGES B-1

APPENDIX C. VERSION 6.2.0.0 REPOSITORY CHANGES C-1

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document (VDD)* describes Version 6.2.3.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) software suite. JTLS-GO 6.2.3.0 is a Maintenance delivery for the JTLS-GO 6.2 series of releases.

JTLS-GO 6.2.3.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations called “wespac62”. Detailed descriptions of some minor Engineering Change Proposals (ECPs) implemented for this release are provided in [Chapter 2.0](#), and [Chapter 3.0](#) describes the software errors that have been fixed since the last release of the JTLS-GO 6.2 series.

JTLS-GO 6.2.3.0 executes on the Red Hat Enterprise Linux Server Version 8.7 64-bit operating systems. The Web-Hosted Interface Program (WHIP[®]) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Obsolete/Outdated Documents

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

1.2.2 Unchanged Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.2.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 02, Version 6.2.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.2.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.2.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.2.2.0)
- *JTLS-GO JOBE Quick Start Guide* (JTLS-GO Document 11, Version 6.2.2.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.2.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.2.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.2.0.0)

- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.2.0.0)
- *JTLS-GO Air Services User Guide* (JTLS-GO Document 24, Version 6.2.0.0)

1.2.3 Updated Documents

- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.2.3.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.2.3.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.2.3.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.2.3.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.2.3.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.2.3.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.2.3.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.2.3.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.2.3.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.2.3.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.2.3.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)
- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)
- JTLS Symbols Application (JSYMS)
- Database Development System (DDS)

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

The current JODA build number is 188.

- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:
 - Apache Server (APACHE)
 - JTLS XML Serial Repository (JXSR)
 - Order Management Authority (OMA)
 - Synchronized Authentication and Preferences Service (SYNAPSE)
 - XML Message Service (XMS)
 - Total Recall Interactive Playback Program (TRIPP)
- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service

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

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

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

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

Instructions for installing JTLS-GO 6.2.3.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.2.3.0 is not necessary. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

The basics of installation have not changed significantly, but due to many Cyber-Security improvements, a new Linux RPM package named "xerces-c", is required to run JTLS-GO 6.2.2.0 and all later versions. The Synapse will not function without this package. Prior to installing JTLS-GO 6.2.3.0, please run the RPM checking script delivered with JTLS-GO to ensure that this package and all other packages are installed as part of your Linux operating system.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called "repository62". Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space.
- The scenario "wespac62", which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.2.3.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Server (ES) Edition Version 8.7, 64-bit architecture.

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

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

Oracle Linux 8.7 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.2.3.0 has been tested on the following operating systems:

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

Oracle Linux 8.4 and 8.7

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

- JTLS-GO 6.2.3.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 392 package, which is equivalent to the current version of OpenJDK.
- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.8.
- JTLS-GO database tools require a certified PostgreSQL 11.19 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.2.3.0. There are several alternative methods available for obtaining the PostgreSQL 11.19 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.
- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of the supported versions of Red Hat Linux ES.

- TCP/IP is required for inter-process communication between the JODA data server and all user interface programs. The version of TCP/IP included with the supported versions of Red Hat Linux ES is sufficient.
- The Perl script language is used by the JTLS-GO system and game setup scripts. The version of Perl included with the supported versions of Red Hat Linux ES is sufficient. The Perl program is typically located in the /usr/bin directory. If Perl is installed in a another location, a link should be created from the /usr/bin directory to this program.
- SIMSCRIPT III (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code.
- ANSI C Compiler: It is not necessary to use a C compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO component programs. The C Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.
- C++ Compiler: It is not necessary to use a C++ compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO HLA component programs. The C++ Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.
- The JTLS-GO DDS application uses these open source libraries:

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

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

Commons-math3-3.0.jar, licensed under Apache Software Foundation (Apache License, Version 2.0) <http://www.apache.org/licenses/LICENSE-2.0>HLA 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™. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using an operational Command, Control, Communication, Computer Information (C4I) systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

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

1.3.2 JTLS-GO Cybersecurity Compliance

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

JTLS-GO has moved to Adoptium, a full OpenJDK Java environment with licensing alternations allowing an application to deliver the software. The following procedure has been established and approved by the JS/J7 Cybersecurity branch:

- Within days of an Oracle Java security release, Adoptium produces an equivalent version using infrastructure, build and test scripts to produce pre-built binaries of the OpenJDK class libraries. All Adoptium binaries and scripts are open source licensed and available for free.
- Within two-weeks of the Adoptium release, JTLS-GO provides a bug release version (JTLS-GO 6.2.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the bug released versions when distributed.

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

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

1. Convert your scenarios to JTLS-GO 6.2.0.0 using the Version Conversion Program (see Chapter 13 of the *JTLS-GO DDS User Guide* for instructions).
2. Perform Option 3, “Setup System For A Specific Scenario”, for each scenario.
3. Perform Option 5, “Run Interface Configuration Program”, for each scenario.

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

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.2.3.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

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

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the \$JGAME/data/hla directory. Federation testing of JTLS-GO 6.2.0.0 with CAE's GESI model has not been accomplished. CAE should be contacted concerning the continued support of GlobalSim.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.2.3.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation (www.pitch.se). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

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

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

Users should download and re-load their scenarios into PostgreSQL, due to changes made in STR JTLS-2023-16290 PSQL Statement Fails For SVP Warning Correction, included in JTLS-GO 6.2.2.0, which was released in July 2023. This will recreate the related check constraints in the database schema. This procedure only needs to be execute once for every scenario. If the procedure was followed after the release of JTLS-GO6.2.2.0, it does not need to be followed again,

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 6.2.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.2.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario_name>.gs and databases symbol.scf file.

1.4.2 JTLS-GO Using New Default Symbol Set

The symbol 2525 ID Codes, which are used by C4I systems to identify the type of object, were expanded and refined as part of the JTLS-GO 6.2.2.0 release. If you have scenarios based on the JTLS-GO repository symbols and have not done so since the release of JTLS-GO 6.2.2.0, you are encouraged to update the symbol sets for those scenarios. This can be accomplished by executing the following steps:

1. Go to the \$JDATA/scenario/<scenario_name>/symbols directory. Enter the following command:

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

2. Compare your symbol.scf file to the JTLS 6.0 version under the repository62 scenario. Enter the following command:

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

3. If the files are identical, or if your symbols are a subset of the JTLS 6.0 version, you should update your symbol file (otherwise, perform Steps 4 through 6). Enter the following command:

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

Your symbol set will now be updated.

4. If your symbol file has additional symbols not found in the JTLS 6.0 version, you can still update your symbol file, but will need to perform a manual integration of the symbol.scf file.

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

5. Perform a save in the jsyms program. Selecting "Yes" will update the <scenario_name>.gs file, and update the graphic symbols in the DDS if the scenario is loaded; selecting "No" will only update the <scenario_name>.gs file.

If your scenario is not loaded into the DDS, the next time you perform a load the updated symbols will be loaded from the <scenario_name>.gs file.

6. Exit the jsyms program.

1.4.3 Standard Repository Changes

R&A has continued to improve and expand the unclassified data repository, which has been renamed to “repository62”. The DDS comparison and synchronization function can be used to determine if any of the changes delivered are of use to a JTLS-GO user organization.

1.5 INSTALLATION

The *JTLS-GO Installation Manual*, a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery. It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 11.19 required to operate JTLS-GO 6.2.3.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.2.3.0 as a result of implementing authorized minor Engineering Change Proposals (ECPs).

2.1 JTLS-2023-16321 Adjust SSM Setup Time When Controller Changes Data

Summary of Model Change Request

Changing the setup time for an SSM using the Controller Set SSM Data order does not affect the Ready to Fire events already scheduled in the model. This can cause fire events to occur before the altered SSM is actually set up. The Set SSM Data order should reschedule these fire events when the setup time has been changed, in order to prevent these premature fire events from occurring.

Design Summary

The model now checks to see if there are any Ready to Fire events scheduled when the Set SSM Data order changes the setup time for an SSM. Ready to Fire events that reference the altered SSM are then rescheduled, by adding the new setup time to the time the Player sent a Change Launcher Status order to set up the SSM. In the event that the rescheduled time is calculated to be before the current game time, the Ready to Fire event is scheduled for the current game time, and the SSM status changes to "Prepared" immediately.

2.2 JTLS-2023-16381 Quick Refuel Mission Order

Summary of Model Change Request

There needs to be a quick way to request air-to-air refueling. The existing Manage Air Refueling order, although very useful for scheduling air refueling operations, requires multiple inputs which can hamper a Player during a high-tempo air battle. Often, a mission simply needs an immediate top-off between intercepts.

Design Summary

A new Refuel Mission quick order was implemented in the WHIP. This new quick order allows the Player to request an immediate refuel with only two inputs: a receiving mission and a refueling mission. The order automatically calculates the amount of fuel necessary to top-off the receiving mission and creates a refuel chit with the current game time. The mission immediately heads toward the tanker to refuel.

By design, only missions from the Player's Force Side can be selected in this order. The verbose Manage Air Refueling order must be used if the missions are from different Sides.

2.3 JTLS-2023-16423 Add C3 Quality Information To Naval Unit IMT Screen

Summary of Model Change Request

The C3 Quality information was only on the Unit Info IMT screen. It should also be placed on the Naval Unit Information IMT screen.

Design Summary

The Current C3 Effectiveness value and the current C3 Quality columns were added to the Naval Unit IMT screen.

2.4 JTLS-2023-16433 Allow ICAO Codes To Be Removed

Summary of Model Change Request

The Controller Set Unit Data order allowed an ICAO code to be added or changed, but not removed. The capability was missing to remove an existing ICAO from an airbase, squadron, or ship to correct a scenario database error.

Design Summary

The Set Unit Data order was enhanced to remove an assigned ICAO code when NONE is entered in the order. The ICAO code attribute associated with the Unit is blanked-out in the model and the airbase IMT displays a blank when there is no longer an assigned ICAO code.

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-2023-16322 Damage Supplies Order Max Too Small

The maximum number of supplies that could be entered into the Controller Damage Supplies order was too small. Users had to enter multiple Damage Supplies orders to deplete the total quantity of supplies held by units in the game.

The maximum number of supplies that could be entered was increased by two orders of magnitude.

3.2 JTLS-2023-16326 DIS Code Report Incompatible Format

The format of the DIS Code report was changed. This XML file was used by the MUSE and NWARS external models to identify the DIS Codes used by JTLS-GO for combat systems, aircraft classes, weapons, and targets. The external programs were unable to parse the DIS Code data because of the change in format.

The Scenario Initialization Program (SIP) was modified to include a routine which writes the DIS Code report in the accepted format. It was modified so that this XML file was automatically generated when the SIP was run to set up a scenario.

The routine was also added to the CEP as part of the initialization procedures.

The DIS Code report was also made available as a generated report when using the order to write initialization files for the ELS, NWARS, or MUSE options.

3.3 JTLS-2023-16328 Allow Full Twenty Game Sides

Although JTLS-GO was expanded to support scenarios with 20 Sides, the method used by the CEP and JXSR to identify Common Objects, which are special objects available to every Side, confused those Common Objects with objects owned by the 20th Side. This limited the practicable number of Sides to 19.

The problem was that the CEP and JXSR code used to identify Common Objects did not allow for 20 Sides to be identified - only 19 Sides could be identified in that code, with the space that should have been used for the 20th Side used for the Common Objects.

The CEP and JXSR code has been changed to allow for all 20 Sides, as well as the Controller side, to be identified and not conflict with the identification of Common Objects.

3.4 JTLS-2023-16330 Air Missions Not Removed From JODA

Air missions are persisting in the JODA after removal by the CEP.

All objects in the JTLS-GO game have perception values that determine whether they can be perceived by a specific Force Side. These values indicate which Side does or does not have perception of that object. Some objects have both Common Knowledge and Perception Knowledge attributes, and air missions had their Common Knowledge attribute values marked for 20 Sides.

However, when these air missions were being deleted by the JODA, only the Sides defined in the game were being removed. This left perception values marked for Sides that did not exist in the game, and the JODA would retain a record for the object.

The CEP was changed to only mark as set the perception values for those Sides defined in the scenario. When the CEP removes an object having Common Knowledge attributes, it deletes the perceptions for the Sides defined in the scenario, removing all perception for the object. This allows the JODA to remove transient objects like air missions.

3.5 JTLS-2023-16332 JODA Does Not Display BOOLEAN Flag Attributes

When a Satellite is displayed in the JODA console, all attributes of this object are displayed in text-style format. However, the display_on_map attribute of this object shows up blank, as if this value was never set.

The display_on_map attribute of the Satellite object is a Boolean flag attribute that has possible values of "TRUE" and "FALSE". The correct value is supposed to be displayed in text format by the JODA. However, an earlier correction to the JDSP library broke the function to copy the characters that make up the "TRUE" or "FALSE" value for display, so the value is no longer displayed.

The JDSP library was changed to fix the function displaying the "TRUE" or "FALSE" values for display in the JODA console.

3.6 JTLS-2023-16334 Air Mission Supply Source Priority

Normally, an air mission obtains its supplies from its home base or carrier. However, if the squadron does not have an assigned home base (such as an independent rotary squadron), or the home base has insufficient supplies, the model was not giving priority to the mission's own squadron as the unit from which to draw supplies before checking other nearby squadrons, support units, or targets.

The logic was corrected to consider the mission's squadron as the unit from which to draw supplies first if the squadron does not have a home base.

3.7 JTLS-2023-16337 DDS Report Table Rendered Incorrectly

The Combat System Roll Up Report generated tables rendered incorrectly. Additionally, the Specific Capability table displays a list that included the number zero, which was not necessary in the list.

The stylesheet for the Combat System Roll Up Report tables was corrected. The Specific Capability table HTML file was corrected to omit the unnecessary zero.

3.8 JTLS-2023-16338 JODA Crash Using Console Find By Name Command

If a FIND command is entered in the JODA console with two arguments, when the first argument is an object type that does not have a name attribute and the second argument is a string name, the JODA will crash.

The JODA must first ensure the entered object type has a name attribute. An object such as a COMMAND_AUTHORITY_TAG does not have a name attribute, and the JODA FIND routine was attempting to reference a null, causing a crash.

The JODA code for a two argument FIND was changed to ensure any type of object included in the find will have a name attribute prior to dereferencing the name.

3.9 JTLS-2023-16340 JODA AAR Debug Flag Incorrect Keyword

The JODA console can directly send special orders to the CEP that are intended to set or unset debug flags in the game. The command to send the AAR collection order does not work.

One of the special orders available from the JODA, the SET.AAR.COLLECTION.FLAGS order, has recently changed. It now uses a different keyword to change all the AAR collection capabilities. This order is implemented in the JODA and needed the corresponding AAR keyword changed to the same keyword used in the order.

3.10 JTLS-2023-16342 Intercepted Mission Perceived Name Changed

Side A was intercepting a Side B mission. The mission moved out of a Link 16 coverage area and Side A's perceived name of the intercepted mission changed. The task that told the intercepting mission which mission was being intercepted was not changed.

This issue only caused a problem if a checkpoint was taken while the model was in this situation. When trying to come up from the checkpoint, the model was attempting to figure out which Side B mission was being intercepted. Since no existing missions had the old perceived name, the link could not be made.

Whenever the perceived name of an air mission is changed, the model now determines if anyone is intercepting the mission, and alters the INTERCEPT task to use the new perceived name.

3.11 JTLS-2023-16344 Basic JEDI Object Quick Retrieval By ID

If the user clicks on the Quick Retrieval Button labeled "Prototype" on the Naval Unit IMT screen, nothing is retrieved.

The underlying JXSR quick specification on a basic JEDI object, such as the name of a SHIP UNIT PROTOTYPE, was not supported. The IMT was changed to handle retrievals by ID for any basic JEDI object.

3.12 JTLS-2023-16346 Update Missile Name After Link 16 Detection

The code to update the new perceived name of a missile after it is detected by Link- 6 was being rejected by the JODA.

The update was using the Receiver Number of the Fire Mission, not the Receiver Number of the missile. The problem was corrected.

3.13 JTLS-2023-16348 No Route To Support Unit Missed Verification

The Scenario Verification Program (SVP) did not identify all units that were on islands, and could not receive truck convoys to deliver supplies.

The issue was that the islands were close to the mainland. When the SVP gathered feasible nodes on which to drop off supplies, it found a node within the required distance of the receiving unit which was on the mainland. When convoys attempted to deliver supplies to the units within the model, the model correctly determined the convoy could not get from the delivery node to the receiving unit on the island. This caused convoys to stack up within the model. This problem was corrected.

3.14 JTLS-2023-16350 ELS Destroyed Extra Aircraft Entities

During an event with the ELS running, a single-aircraft mission was killed by Controller order. The mission was properly destroyed, but the ELS killed two aircraft entities.

The ELS did not properly reconcile the destroyed entities when the mission was destroyed. One aircraft was destroyed when the mission was killed, and then a second aircraft was killed when the squadron listed the combat system as KIA. There was an error in the code that has been corrected.

3.15 JTLS-2023-16352 Synapse Building Validation Paths Improperly

The Synapse does not allow persistent WHIPs to request a LIST of files in the imt_orderspreadsheet and layout directories. WHIPs are persistent when the CEP has been restarted, but the WHIPs have remained running during the restart.

For security, the Synapse is able to check all files the WHIP may attempting to add, delete or modify on the hard drive for the game. This includes a variety of configuration, preference and order files. The Synapse checks the path, the name and the content of the files as well as the activity the WHIP is attempting to preform, such as changing the mode or deleting files.

The Synapse builds an internal list of allowed directories for each WHIP, but when the Synapse has not received a download from the CEP, the Synapse has not yet had the opportunity to create the list of directories. Therefore, any persistent WHIPs (and TRIPPs) will continue to request file actions, such as a LIST from the directories the Synapse has not yet created. This requested WHIP action is denied until the CEP downloads to the JODA and the Synapse receives the download.

The Synapse was changed to create the list of allowed directories following the first read of the CIF file, which is generated by the ICP. The Synapse checks this file for modification continuously, before the CEP has completely started.

3.16 JTLS-2023-16353 Send A Requisition Memory Leak

In the Send A Requisition routine, there was a local array reserved that is never released.

The array is now properly released.

3.17 JTLS-2023-16355 OPM Regeneration From Checkpoint Crashes

The Scenario Initialization Program (SIP) crashed reading in an checkpoint as part of the procedure to generate Online Players Manuals (OPM).

The Target Read was not updated to use the new JTLS-GO 6.2 Target ASCII Read format, nor the new Inventory Item Owner format. This has been corrected.

3.18 JTLS-2023-16356 AAR Fractional Other Aircraft Kills

The After Action Report (AAR) reports fractional aircraft kills for "Other Kills", which is normally reserved for Controller kill actions.

There was a mismatch between the data that the model was sending and what was expected to be received, which corrupted the information reported to the AAR. This error was fixed.

Additionally, the Development Team uses a tool that is supposed to identify these problems, but in this case it failed to do so. This tool was also fixed and several other similar errors were discovered and corrected.

3.19 JTLS-2023-16358 LC2MS URL Always Uses Default Service Name

While writing documentation on how to interface with the LC2IS Message Service (LC2MS), it was discovered that the browser interface always wrote links using the default service name of

joi-lc2is. There is always a default service created using this name, but the Tech Controller can add other LC2MS services and specify any name if they want to distribute processing. However, the interface was always forcing the requests to the default service, which may or may not be running.

The code was modified to get the service name from the ICP-generated configuration file, convert it to lowercase, and use it in the URL. This matches what the Apache server expects.

3.20 JTLS-2023-16359 Synapse Not Reporting User And Filename

Whenever the Synapse is validating files that are PUT by a user, a file may fail XML validation. In this case the Synapse reports the line number and column number of the format problem within the XML file, but not the file name or which user performed the PUT action.

The Synapse routine for processing PUT requests was modified to report the name of the WHIP and the name of the file.

3.21 JTLS-2023-16360 Restoring ELS When JODA Connection Lost

The ELS was running normally when the connection to the Primary JODA was lost. This happened because the JODA was restarted by Controller action. The ELS should be allowed to continue when the JODA has resumed execution. However, the ELS crashed while attempting to continue.

The ELS crashed while attempting to clear out the existing sets of objects in memory. There was an error in the code which related to the clearing of units which are subordinate to other units. The error was corrected, and the ELS now continues after losing connection to the JODA.

3.22 JTLS-2023-16363 ELS Unable To Revive Combat System Entities

During an event with the ELS running, the Controller revived some aircraft combat systems in a squadron, because these aircraft were inadvertently killed and were needed later in the event. The aircraft were revived in the aggregate game, but the equivalent combat system entities remained dead in the ELS.

The ELS was not properly handling the resurrection of entities when they were revived in the CEP. ELS code changes were made to correctly reflect the combat system counts from the aggregate model.

3.23 JTLS-2023-16364 ELS Missing Crew Entities Inappropriate Errors

The ELS was running with a database which was configured to not publish entities for all personnel-type combat systems. Some equipment combat systems had required crew counts, but the ELS was not creating or publishing the associated crew entities. When an HRU was created, the ELS was attempting to assign crew entities from the parent unit. Because there were no available crew entities, the ELS reported this as an error for inconsistent crew counts.

Because there were no crew entities in the ELS in the database by design, these reported errors were not valid. The code used to create HRUs was corrected to handle the case where crew combat system entities do not exist in the ELS. This change eliminated the false errors from appearing on the ELS console and in the optional error log.

3.24 JTLS-2023-16375 AAR Aircraft Kill Summary Ground Losses

The user launched an air-to-ground-mission targeting a squadron. The squadron lost combat systems, including aircraft. The AAR Aircraft Kill Summary Report should have tallied the aircraft losses as Ground losses, but the total losses in the Ground column of the report were zero.

Aircraft lost by air missions that were on alert on the ground were properly reported by the Aircraft Kill Summary Report in the Ground column of the report. However, if a squadron was attacked and lost some aircraft, the losses were not tallied and never appeared in the Ground column.

The entry in the Scenario Data Repository (SDR) database in the `damaged_system` field for any aircraft combat system was "AIRCRAFT", instead of the name of the aircraft class (AC NAME). A check was added to the model to see whether the damaged system is a Combat System with the Special Capability of AIRCRAFT. When reporting damage to such a Combat System, the AC NAME is entered in the `damaged_system` field; otherwise, the CS NAME is entered as before.

3.25 JTLS-2023-16376 Incorrect Sunk Ship Messages

A naval unit sank after incurring multiple hull hits. Two Player messages were immediately generated.

- **The “Unit Destroyed/Redeployed Report” showed that the ship's targets were transferred to the ship's support unit, even though those targets were destroyed when the ship sank.**
- **The “Unit Destroyed Future Arrivals Affected” message was generated even though there were no squadrons scheduled to TPFDD to the destroyed ship. These messages were confusing to the players.**

A flaw in the logic that transfers targets from a destroyed unit to the destroyed unit's support unit (or suitable alternative unit) failed to exclude ship-board targets from this process. Because all targets were destroyed on the sunken ship, it made no sense to transfer the targets. The logic was changed to skip the transfer of targets owned by the destroyed ship, preventing the message from indicating the targets had been transferred. A new sub message was added to indicate that all targets were destroyed when the ship sank.

The Future Arrival message was superfluous and should not be generated if no squadrons were arriving. Logic was added to skip the message, unless there is at least one squadron scheduled to be TPFDD to the destroyed unit (which could be a ship, FARP, or airbase).

3.26 JTLS-2023-16378 Join Formation Crash

A ship was ordered to Change Station. While the ship was enroute to its newly assigned station location, a Join Formation order was sent with an execution time of Now and specified a different formation to join. The model crashed.

When the Join Formation order was sent, the currently executing Change Station task was canceled as intended. The code did not properly clear the pointer to this canceled task when the task was destroyed. When the CEP attempted to schedule the new Join Formation task, the model entered the wrong logic branch because the original first task was not NONE, and crashed when attempting to reference the original first task pointer.

The problem was fixed by adding code to set the original first task to NONE when the Change Station task is canceled. This fix only applies to the Change Station task. All other executing naval tasks are postponed and not canceled when a new “Now” task is sent to a Naval Unit.

3.27 JTLS-2023-16380 Show DSA Order Crash

Sending the Show Directed Search Area order from the Controller WHIP cause a model crash.

The Controller Show DSA order and Player Manage DSA order shared code, which was not expecting the Controller Show DSA order to use its code. The shared code was improved to properly process the Controller Show DSA order.

3.28 JTLS-2023-16382 Identify Number Of Objects In Terrain Grid

The SVP should be able to identify how many Units and Targets are in a Terrain Grid.

SVP Warning 1900 has been added, which looks for Terrain Grids that contain a combined total of more than 100 Unit and Target objects. The number 100 is an arbitrary figure - database developers can change this number. Warning 1900 can safely be ignored or simply used to identify grids with an unusually large number of units which can slow down model execution throughput.

3.29 JTLS-2023-16383 Errors In AAR Scripts To Mine SDR Data

Several scripts were delivered with JTLS-GO which can be used to mine AAR data from the SDR database. Four of these scripts attempted to access data for an obsolete attribute. As a result, the script failed to run.

When viewing AAR data for engagements, the attribute called NUMBER_SYSTEMS was used, but this attribute is no longer valid in the database. The associated value was unused in the script. This attribute was removed and the scripts functioned normally.

3.30 JTLS-2023-16384 Air Mission Returning To Moved Home Crash

The last task in every air mission's task is the "Mission Complete" Task. When an air mission was on the last leg of its route heading home and its home moved, the mission knew that it had to recompute a new route to its home unit. When executing this logic, the model cleared not only the mission's Move Task, but also its "Mission Complete" task. This caused the model to crash.

The mission's Move task was automatically created to move the mission to its next Player-assigned task. Under most circumstances, when an automatically-assigned move task is canceled, the associated Player-assigned task is also canceled. There are a few circumstances when the Player-assigned task should not be canceled. The situation in which a mission's home moves while the mission is heading home is one of the exceptions. The code now recognizes this situation as one of the exceptions and the "Mission Complete" task is not deleted.

3.31 JTLS-2023-16386 MDP Delivery Format Does Not Show WHIP Names

As part of the MDP configuration, the Tech Controller can select which directed messages to which WHIPs should be managed. However, the list for WHIP names is empty.

The WHIP names were previously retrieved from the <scenario>.icp_db.xml file. They have been removed from that file to allow individual password settings and now reside in the new <scenario>.whip_db.xml file. The code was updated to retrieve them from this new location.

3.32 JTLS-2023-16387 Errors Processing External Intelligence Collection

The link between JTLS-GO and MUSE was used to test the external collection of intelligence. A UAV was flown in the MUSE simulation and reflected inside JTLS-GO. This external mission sent an Ad Hoc request for intelligence collection in a specified polygonal area. The orders sent from MUSE failed to be processed by JTLS-GO.

There were minor processing errors in both the ELS and the CEP. The ELS crashed while processing the external order because a routine was using the wrong object class when accessing the owning unit for a combat system entity. After this error was corrected, the order was processed by the ELS and then an order was submitted to the CEP to report the intelligence collection.

Subsequently, the CEP crashed when attempting to create an associated DSA. The routine was using the wrong order field to create a DSA. The processing routine was changed to use the correct field and the order was then processed correctly. With these minor changes, the external intelligence updates were generated properly.

3.33 JTLS-2023-16388 External Model Does Not Get Air-to-Air Impacts

When linked to an external model, air-to-air weapon shots are not being sent out, so the external model cannot assess the damage to its air mission.

The code that is sent when an Air Defense site fires on an external air mission was working as desired. The impact message was put into a commonly accessed sub-routine and now both the Air Defense Site assessment and the Air-to-Air assessment logic calls the single air impact notification message.

3.34 JTLS-2023-16389 Naval Ship Class Not Shared On Restart

After stopping and starting the game, the ship class for foreign units was no longer known on the JODA. During a normal run of a game, this information becomes available when the detection level is Recognition or higher. The information, once acquired, should persist across a game stop and start.

The JODA download code was providing perception on the Ship Unit Prototype and the Real World OTH-Gold type for Naval units, but not the ship class. The two other attributes are also Recognition detection level information, and the sharing of the ship class was added to the same section of code.

3.35 JTLS-2023-16391 Bad Checkpoint With Saved Mission Engagements

An air mission had a number of either Air-Air or Surface-Air engagements that it saves for end of mission reporting.

During a checkpoint, the specific firing Air Defense Class or Aircraft Class is saved as a three-digit integer, which represents the index of the firing entity. If the database has more than 999 Aircraft Classes or Air Defense Classes, and the firing object is an entity with an index over 999, the checkpoint contains an unreadable entity index causing a bad checkpoint.

The CEP save code was modified to write out the firing object index as five digits instead of three.

3.36 JTLS-2023-16393 Graphics Symbol Update When Unit Echelon Changes

A unit was assigned a Command Level with the wrong echelon. When the Controller changed the unit to use a Command Level with the correct echelon, the WHIP showed the new updated echelon, but the 2525-B Symbol ID Code, which contains the echelon and is used by some external programs to identify the echelon, was not updated.

Besides the echelon, the 2525-B Symbol ID Code also has dynamic information in it to identify the force relationship, the mobility of targets, and the country code of the object. Changes to the relationship within the model already resulted in changes to the 2525-B Symbol ID Code, but the other changes did not update the code with the JODA.

A routine was added to perform an update, and send the update to the JODA, of the code and calls to the routine were added to handle Controller changes to Unit and Target Political Country, Target Mobility, and Unit Command Level.

3.37 JTLS-2023-16394 AAR Object Event Report Detailed View Does Not Render

The Detailed Event view for the AAR Object Event Report failed to render because it was using an old enumeration for Other Losses Aircraft. In earlier versions of JTLS-GO, only aircraft were reported in the AAR Other Losses Air table. In JTLS-GO 6.2, the table has been renamed AAR Other Losses, with a child table of AAR Other Losses Items, to represent damage to all possible items (combat systems, supplies, ship hull, targets, etc) by non-kinetic action such as Controller changes.

The old enumeration was replaced with the newer versions which are Other Losses Inflicted and Other Losses Sustained.

3.38 JTLS-2023-16395 Adding Mission Tasks After Package Operations

A user attempted to add a route to an air mission that was part of a package. The air mission had not yet launched, and the route was to be added after the Escort task, which marks the end of package operations. The model refused the order, insisting the sequence number had to be before the Rendezvous task, which marks the start of package operations, or after the Escort task but before the Mission Complete task. This was an erroneous complaint because the requested task number was after the Escort task and before the Mission Complete task.

A code error was discovered when the logic was determining the task numbers for the existing Rendezvous, Escort, and Mission Complete tasks. Once the Rendezvous task was found, the code stopped searching other tasks, leaving the Mission Complete task without a proper value. This code error was fixed so that the search through the tasks continues and all task numbers are assigned.

3.39 JTLS-2023-16397 KOI Not Reading Filter File

When the KOI is started, it generates many error messages about mismatched tags in its filter file. This file appears to be a good XML file, but because of the errors the data is not loaded, and the KOI does not use any of the defined filters.

Individual data elements within the file were being improperly read by the KOI, prematurely shortening these data elements and preventing the file from being loaded properly. This problem was fixed.

3.40 JTLS-2023-16399 ELS Crash While Disaggregating Ship

The ELS crashed while attempting to disaggregate a naval unit. It was trying to assign a DIS code to the combat system entity for the hull of the ship. The ELS crashed because there was no entity associated with the hull.

The ELS cannot run properly without an entity which represents the ship hull. One is normally created as part of the ELS startup procedure. In this case, the entire template for the specific

ship was missing, due to a procedural error in setting up the game. As a result, the ELS dynamically created a template for the ship unit prototype (SUP) used by the naval unit.

The code to dynamically create the SUP template was incomplete. The code was modified to add a hull position to the SUP template when it was created by the ELS. With this change, the crash was resolved for dynamically-generated templates for ships.

3.41 JTLS-2023-16404 Mean For Message Delay Can Be Set To Zero

The Controller can incorrectly set the Mean Communications Delay Time to zero, causing a crash.

The order panel incorrectly allowed the Controller to change the Mean Message Delay time to zero. The order panel was corrected.

3.42 JTLS-2023-16406 TEMS Crashes on Bad Pulse Rate

The TACELINT Message Service (TEMS) crashed parsing an unexpected value for a pulse rate.

The TEMS has been updated to handle invalid pulse rate values gracefully.

3.43 JTLS-2023-16407 MDP Errors In Intel Report Messages

In the Message Distribution Program (MDP) error log, there were multiple errors associated with the intercepted Intelligence Report USMTF message.

There were two coding errors in the CEP:

1. The Country Code was not assigned a value when the subject was a squadron, which caused the data translation in the XML template to be undecipherable.
2. The Mission Type was omitted when a multi-task mission was involved, which caused the data translation to fail.

Both errors were corrected.

3.44 JTLS-2023-16409 Manage Attack Package Order Help Text

Several help text entries in the Manage Attack Package order were either incorrect or inadequate.

The help text that described ingress and egress routes using Air Control Mean (ACM) regions incorrectly referenced DCA protection zones. The text was replaced with a correct explanation. Also, the help text that described the use of an ACM region as a rendezvous location was expanded to explain which ACM shapes were valid.

3.45 JTLS-2023-16411 Super WHIP Auto-Select Order Fields

Orders that contained auto-select fields did not work in a Super WHIP, because these fields were not enabled for the Super WHIP.

The auto-select feature was enabled in the code to also work in Super WHIPs.

3.46 JTLS-2023-16412 Sendmail Not Setting Header 'To' Field

The automated sendmail library, responsible for sending emails in JTLS-GO, was not setting a “To” field in the email headers it generates. This would cause the emails to fail checks performed by more stringent mail handling services, and ultimately lead the emails to be dropped.

The sendmail library now sets a “To” field in all email headers it sends.

3.47 JTLS-2023-16415 Super WHIP Map Finds Objects For Any Perception

On a Super WHIP, when the map perception is set to a Side that has no perception of an object on the map, it finds it showing the object's location, although the object is not shown for the selected Side.

Only the selected Side’s perception should find perceived objects’ locations. This has been corrected.

3.48 JTLS-2023-16416 Dashboard Crash When Writing Data

The Dashboard service would crash when serializing non-UTF-8 data into JSON.

The Dashboard service now gracefully catches this serialization issue.

3.49 JTLS-2023-16418 Home Squadron In Air Loss History Report

The home squadron needs to be on the detailed format of the Air Loss History Report.

The home squadron data was added to the report.

3.50 JTLS-2023-16419 Missing Data File Link

The WHIP reads a file to determine the shelf life of specific messages in the Message Browser. During a transition to a new data file linking schema, this shelf life file was missed, so the WHIP could not find it.

The data file linking process was updated to include the message shelf life data file.

3.51 JTLS-2023-16421 ATO Parser Does Not Report Crash Reason

While reading an ATO, the Parser received a error which caused it to terminate. The Parser did not report the fatal error before exiting. Without any indication of where the error lies, the user must use trial and error on the ATO message until the offending error is discovered.

The ATO Parser was modified to write the current stack of errors and warnings prior to exiting with a fatal error. In this way the user has a record of the parsing progress and should be able to conclude the general location of the fatal error.

3.52 JTLS-2023-16422 ATO Parser Reports Two Errors As One

When an ATO contains a mission that is missing a departure and arrival ACO, both errors are reported, but are listed on the same line. This appears as one error in the display window.

The Parser was modified to separate the two errors into two line items in the display.

3.53 JTLS-2023-16425 Unit Hierarchy Report Location Not Correct

The Unit Hierarchy report includes the location of every listed unit, but the location list was incorrect.

The message used a special capability called the Quoted Location function. This function was implemented incorrectly.

When that problem was solved, if the seconds portion of the unit's location was less than 10, the location left a space in the displayed location, making it impossible to quickly find the location on the map. This problem could not be solved. It was determined that the Quoted Location function was unnecessary in this specific message. The default LOCATION function prints out the decimal seconds format correctly; therefore the Quoted Location function was removed from the Unit Hierarchy report.

The same problem also existed in one of the Controller Cannot Reply Messages and this solution was implemented.

3.54 JTLS-2023-16428 Link 16 Sensor Not On Network Detects Ship Error

When a Link 16 Sensor detects a ship, and the sensor is not on a Link 16 network, the model generated a logic error.

The user was warned of the situation in the SVP and decided to run with the situation, so a logic error should not be generated. Code was added to ensure the logic error is not generated.

3.55 JTLS-2023-16429 Orbiting Mission Orders Inconsistent Constraints

All Orbiting Mission orders are supposed to have consistent Main tabs, and the constraints for those basic fields should be identical. An inconsistency existed between the OAS and DCA orders.

Two constraint inconsistencies were noted and fixed:

- If a transit route is entered for a repeatable route, the Altitude field should not be filled.
- If a transit route is entered for a repeatable route, then an End Orbit location must be specified.

3.56 JTLS-2023-16432 Very Large ATO Messages Crash ATOT

While processing a very large ATO message using the ATO Translator, the Translator crashed and left no report of what the problem was. The operator had no useful information in which to easily determine what the problem was.

The ATO Translator reserves 1 MB of memory for an array to hold an entire ATO message, including comments. When the Translator reached this memory limit while reading an ATO of greater than 1 MB, it caused a memory fault and terminated before any report of memory overrun could be recorded.

The Translator was modified to check the status of memory while reading the ATO message and stop reading when the reserved memory becomes full. A proper report of the overrun condition is now available to the user in the error log when this condition occurs. Also, the memory limit was raised in order to accept very large ATO messages.

3.57 JTLS-2023-16434 ATOT Mobility Mission Orders Missing Location

The ATO Translator generates transport instructions with a utility order whenever there are AIRMOVE records for a mission in the ATO message. This utility order contains the incorrect landing location and supply list.

The utility order for mobility transport instructions was recently rewritten. The list of column names were shifted over one field when the ACM_LOCATION field was added. This caused all the following field numbers to shift. The Translator was therefore placing the correct values for the utility in the wrong fields of the utility. The Translator was modified to account for the new field numbering in the new transport instructions utility.

3.58 JTLS-2023-16435 ATOT Sides Missing From Untranslated ACO

Whenever the ATO Translator is required to translate an ATO without an ACO, the Translator reports an error. This error is for the missing sides that are expected and required in the XML tags of the ACO.

Similar to the XML side tags that are required in the ATO, the ACO is also required to provide the side tags. This is necessary for the Translator to inscribe the sides for ACMs whenever the option to generate ACMs is selected and the ATO is not being translated.

The Translator was modified to withhold reporting this error whenever the ACO is specifically not selected during the translation.

4.0 REMAINING ERRORS

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

The following list of issues is known and have not been fixed in time to make it into this release of JTLS-GO 6.2.3.0.

4.1 DDSC/WHIP/JOBE - CADRG Map Zoom

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

4.2 MHE Targets Loading Air Mission Can Cause a Crash

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

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

Strategic Lift Missions, used to move TPFDD assets into the Theater and report the results to a real-world TPFDD processing system, has not been updated to work within JTLS-GO 6.1.

4.4 Tactical Ground Formation Attacks Do Not Work

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

4.5 ATOT Spreadsheet Lacks Detailed Field Checking

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

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

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

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

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

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

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

APPENDIX B. VERSION 6.2.0.0 DATABASE CHANGES

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

APPENDIX C. VERSION 6.2.0.0 REPOSITORY CHANGES

The R&A Database Team is continually adding and vetting unclassified data to expand and maintain the JTLS-GO Data Repository. Over the last year as part of the JTLS-GO 6.2 development effort, this entire process has been conducted in both the JTLS-GO 6.1 and JTLS-GO 6.2 versions of the repository. No specific, unique, additions were made to the JTLS-GO 6.2 repository. The repository delivered with this initial version of JTLS-GO 6.2 contains the same data as the JTLS-GO 6.1 version of the repository, except the format has been altered to meet the requirements of JTLS-GO 6.2.