

# 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 6.1.12.0)**

[Blank Page]

## ABSTRACT

The Joint Theater Level Simulation - Global Operations (JTLS-GO<sup>®</sup>) 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 6.1.12.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.1.12.0 is a Maintenance release of the JTLS-GO 6.1 series that includes a repository of standard data, a demonstration scenario based in the western Pacific as well as a few minor model functionality improvements implemented as Engineering Change Proposals (ECPs). These ECPs are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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 - a Valkyrie Enterprises Operation  
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1.2.1 Obsolete/Outdated Documents.....	1-1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-2023-16424 NAVAL UNIT IMT C3 QUALITY INFORMATION .....	2-1
3.0 SOFTWARE TROUBLE REPORTS.....	3-1
3.1 JTLS-2023-16323 DAMAGE SUPPLIES ORDER MAX TOO SMALL .....	3-1
3.2 JTLS-2023-16324 SUPER WHIP FORMAT PROPERTIES SAVED UNDER CONT .....	3-1
3.3 JTLS-2023-16327 DIS CODE REPORT INCOMPATIBLE FORMAT .....	3-1
3.4 JTLS-2023-16329 ALLOW FULL TWENTY GAME SIDES .....	3-1
3.5 JTLS-2023-16331 AIR MISSIONS NOT REMOVED FROM JODA .....	3-2
3.6 JTLS-2023-16333 AIR MISSION SUPPLY SOURCE PRIORITY .....	3-2
3.7 JTLS-2023-16343 INTERCEPTED MISSION PERCEIVED NAME CHANGED .....	3-2
3.8 JTLS-2023-16345 BASIC JEDI OBJECT QUICK RETRIEVAL BY ID .....	3-3
3.9 JTLS-2023-16347 UPDATE MISSILE NAME AFTER LINK 16 DETECTION .....	3-3
3.10 JTLS-2023-16349 NO ROUTE TO SUPPORT UNIT MISSED VERIFICATION .....	3-3
3.11 JTLS-2023-16351 ELS DESTROYED EXTRA AIRCRAFT ENTITIES .....	3-3
3.12 JTLS-2023-16354 SEND A REQUISITION MEMORY LEAK .....	3-4
3.13 JTLS-2023-16357 AAR FRACTIONAL OTHER AIRCRAFT KILLS .....	3-4
3.14 JTLS-2023-16361 RESTORING ELS WHEN JODA CONNECTION LOST .....	3-4
3.15 JTLS-2023-16362 ELS UNABLE TO REVIVE COMBAT SYSTEM ENTITIES .....	3-4
3.16 JTLS-2023-16365 ELS MISSING CREW ENTITIES INAPPROPRIATE ERRORS .....	3-4
3.17 JTLS-2023-16377 INCORRECT SUNK SHIP MESSAGES .....	3-5

- 3.18 JTLS-2023-16379 JOIN FORMATION CRASH ..... 3-5
- 3.19 JTLS-2023-16385 AIR MISSION RETURNING TO MOVED HOME CRASH ..... 3-6
- 3.20 JTLS-2023-16390 NAVAL SHIP CLASS NOT SHARED ON RESTART ..... 3-6
- 3.21 JTLS-2023-16392 BAD CHECKPOINT WITH SAVED MISSION ENGAGEMENTS ..... 3-6
- 3.22 JTLS-2023-16396 ADDING MISSION TASKS AFTER PACKAGE OPERATIONS ..... 3-6
- 3.23 JTLS-2023-16398 KOI NOT READING FILTER FILE ..... 3-7
- 3.24 JTLS-2023-16403 ELS CRASH WHILE DISAGGREGATING SHIP ..... 3-7
- 3.25 JTLS-2023-16405 MEAN FOR MESSAGE DELAY CAN BE SET TO ZERO ..... 3-7
- 3.26 JTLS-2023-16408 MDP ERRORS IN INTEL REPORT MESSAGES ..... 3-7
- 3.27 JTLS-2023-16410 SUPER WHIP AUTO-SELECT ORDER FIELDS ..... 3-8
- 3.28 JTLS-2023-16413 SENDMAIL NOT SETTING HEADER 'TO' FIELD ..... 3-8
- 3.29 JTLS-2023-16414 SUPER WHIP MAP FINDS OBJECTS FOR ANY PERCEPTION ..... 3-8
- 3.30 JTLS-2023-16420 MISSING DATA FILE LINK ..... 3-8
- 3.31 JTLS-2023-16426 UNIT HIERARCHY REPORT LOCATION NOT CORRECT ..... 3-8
- 3.32 JTLS-2023-16427 LINK 16 SENSOR NOT ON NETWORK DETECTS SHIP ERROR .. 3-9
- 3.33 JTLS-2023-16430 ORBITING MISSION ORDERS INCONSISTENT CONSTRAINTS ... 3-9
- 3.34 JTLS-2023-16431 VERY LARGE ATO MESSAGES CRASH ATOT ..... 3-9
  
- 4.0 REMAINING ERRORS..... 4-1
  - 4.1 DDSC - TMU LINE MODE CHANGES MULTIPLE GRIDS ..... 4-1
  - 4.2 DDSC - MULTIPLE TYPES IN DDS HISTORY TABLE ..... 4-1
  - 4.3 WHIP - PIPELINE NOT SHOWN ON IMT ..... 4-1
  - 4.4 DDSC/WHIP/JOBE - CADRG MAP ZOOM ..... 4-1
  - 4.5 MHE TARGETS LOADING AIR MISSION CAN CAUSE A CRASH ..... 4-1
  - 4.6 THE JTLS-GO STRATEGIC LIFT MISSIONS ARE NOT WORKING PROPERLY ..... 4-1
  - 4.7 TACTICAL GROUND FORMATION ATTACKS DO NOT WORK ..... 4-2
  
- APPENDIX A. ABBREVIATIONS AND ACRONYMS .....A-1
  
- APPENDIX B. VERSION 6.1.0.0 DATABASE CHANGES ..... B-1
  
- C.0 VERSION 6.1.0.0 REPOSITORY CHANGES..... C-1

## 1.0 INTRODUCTION

### 1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.1.12.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO<sup>®</sup>) software suite. JTLS-GO 6.1.12.0 is a Maintenance delivery for the JTLS-GO 6.1 series of releases.

JTLS-GO 6.1.12.0 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 are summarized in the VDD for Version 6.1.0.0 delivered in the documents directory.

This release is being provided as part of the normal JTLS-GO maintenance program, in which new maintenance release versions are closely coordinated with Java security patch releases. Detailed descriptions of some minor Engineering Change Proposals (ECPs) implemented for this release are provided in [Chapter 2.0](#). [Chapter 3.0](#) describes the Software Trouble Reports (STRs) identified during testing and operational exercises.

JTLS-GO 6.1.12.0 executes on the Red Hat Linux Enterprise Server Version 8.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP<sup>®</sup>) 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 the 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 6.1.10.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 02, Version 6.1.7.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.1.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.1.10.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.1.9.0)

- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.1.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.1.9.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.1.9.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.1.11.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.1.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.1.2.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.1.7.0)
- *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 6.1.9.0)

### 1.2.3 Updated Documents

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.1.12.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.1.12.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.1.12.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.1.12.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.1.12.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.1.12.0)

### 1.2.4 New Documents

No new documents were produced for this version of the software.

### 1.2.5 Delivered Software Components

JTLS-GO 6.1.12.0 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)
  - Glassfish Version 5.1.0
- 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 173.
- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:
  - Apache Server (APACHE) Version 2.4.54
  - 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)

- Database Development Server and its component programs
  - Glassfish 5.1.0
  - Database Development System Client (DDSC)
- 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.

Instructions for installing JTLS-GO 6.1.12.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.1.12.0 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 6.1.12.0 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 or higher, 64-bit architecture.

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

RedHat Linux 8.6 - this operating system license must be purchased.

Oracle Linux 8.4 and 8.6 - 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 6.1.12.0 has been tested on the following operating systems:

Red Hat Linux Enterprise Server Edition Version 7.9, 8.4, and 8.6

Oracle Linux 8.4 and 8.6

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 6.1.12.0 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.

Since the release of JTLS-GO 6.1.8.0, JTLS-GO has been using the Temurin JDK, which is optimized for speed. Prior to the release of JTLS-GO 6.1.8.0, JTLS-GO used AdoptOpenJDK, which was optimized for memory. All tests indicate that there is an increase in the amount of memory needed by our Java applications, but the increase is not significant. Both the ICP and DCP have the option for an organization to increase the maximum memory heap for the WHIP and DDSC. For large scenarios and databases, an organization should consider increasing the maximum heap size.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.8.

- 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 6.1.12.0. 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. The code was compiled using Simscript III Version 5.3.
- 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) system 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 keeps the software up-to-date and all security patches are properly installed. In order to fulfill this DoD Cybersecurity requirement, JTLS-GO has moved to Adoptium™ JDK, a full OpenJDK Java environment with a licensing paradigm 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 JDK 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.

The completed Cybersecurity paperwork and a current Gate completion certificate can be obtained by contacting Ms. Jessica Camacho, the U.S. Government Program Manager, at [jessica.l.camacho.civ@mail.mil](mailto:jessica.l.camacho.civ@mail.mil).

### 1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.1.12.0 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 6.1.12.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation ([www.pitch.se](http://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 6.1.12.0 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 was included in JTLS-GO 6.1.2.0, released in March 2022. This STR required a change to the ASCII checkpoint format. Any saved checkpoints from JTLS-GO 6.1.0.0 and JTLS-GO 6.1.1.0 will not work with JTLS-GO 6.1.2.0 and above.

STRs JTLS-2022-15810 and JTLS-2022-15839, delivered with JTLS-GO 6.1.7.0, required changes to the checkpoint format. Any saved checkpoints from any previous JTLS-GO 6.1.0.0 versions will not work with JTLS-GO 6.1.7.0 and above.

Users should download and re-load their scenarios, due to changes made in STR JTLS-2023-16291 PSQL Statement Fails For SVP Warning Correction, included in this JTLS-GO 6.1.11.0. This will recreate the related check constraints in the database schema. If this has not been done since the release of JTLS-GO 6.1.11.0, it should be accomplished for full SVP functionality.

#### 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 database 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, was expanded and refined as a part of STR JTLS-2023-16310 Incorrect Identification Codes For Symbols released with JTLS-GO 6.1.11.0.

If you have not done so already and your scenarios are based on the JTLS-GO repository, you are encouraged to update the symbol sets for those scenarios using 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 repository61 scenario. Enter the following command:  

```
diff symbol.scf $JDATA/scenario/repository61/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/repository61/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 is no longer improving and expanding the unclassified data repository "repository61." All repository work has been moved to the JTLS-GO 6.2 series.

## 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 11.13 required to operate JTLS-GO 6.1.12.0.



## 2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes the one model capability added to JTLS-GO 6.1.12.0 as a result of implementing an authorized minor Engineering Change Proposals (ECPs).

### 2.1 JTLS-2023-16424 Naval Unit IMT C3 Quality Information

#### 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.



## 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-2023-16323 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-2023-16324 Super WHIP Format Properties Saved Under CONT

**The Information Management Tool (IMT) format properties file for Super WHIPs was being saved under the Controller Side's file path.**

The file path configured for the format properties was corrected to use the WHIP login data's Force Side name.

### 3.3 JTLS-2023-16327 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.

### 3.4 JTLS-2023-16329 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.

### 3.5 JTLS-2023-16331 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.

### 3.6 JTLS-2023-16333 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-2023-16343 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.

### 3.8 JTLS-2023-16345 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 of a basic JEDI object name, 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.

### 3.9 JTLS-2023-16347 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.

### 3.10 JTLS-2023-16349 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.

### 3.11 JTLS-2023-16351 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.

### 3.12 JTLS-2023-16354 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.

### 3.13 JTLS-2023-16357 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.

### 3.14 JTLS-2023-16361 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.

### 3.15 JTLS-2023-16362 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.

### 3.16 JTLS-2023-16365 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.

### 3.17 JTLS-2023-16377 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).

### 3.18 JTLS-2023-16379 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.



### 3.19 JTLS-2023-16385 Air Mission Returning To Moved Home Crash

The last task in every air mission's task list 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.

### 3.20 JTLS-2023-16390 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.

### 3.21 JTLS-2023-16392 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.

### 3.22 JTLS-2023-16396 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.

### 3.23 JTLS-2023-16398 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.

### 3.24 JTLS-2023-16403 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.

### 3.25 JTLS-2023-16405 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.

### 3.26 JTLS-2023-16408 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.

### 3.27 JTLS-2023-16410 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.

### 3.28 JTLS-2023-16413 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.

### 3.29 JTLS-2023-16414 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.

### 3.30 JTLS-2023-16420 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.

### 3.31 JTLS-2023-16426 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.

### 3.32 JTLS-2023-16427 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.

### 3.33 JTLS-2023-16430 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.

### 3.34 JTLS-2023-16431 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.

## 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.

Prior to releasing JTLS-GO 6.1.0.0 as a major release with new functionality, all outstanding errors were reviewed. If the error could not be reproduced, it was considered obsolete and no longer relevant to JTLS-GO. These errors were removed from consideration for correction. 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.

### 4.5 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.

### 4.6 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.

#### 4.7 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.

## 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
EEL	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
HET	Heavy Equipment Transporter
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
OVT	Order Verification Tool
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 6.1.12.0 database. For information concerning the data changes between JTLS-GO 6.0 and the JTLS-GO 6.1 series, please refer to the *JTLS-GO 6.1.0.0 Version Description Document*.





## **C.0 VERSION 6.1.0.0 REPOSITORY CHANGES**

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