

# JTLS-GO

## Version Description Document

April 2020



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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS  
(JTLS-GO 5.1.7.0)

[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 combined joint and coalition resource air, land, naval, and Non-Governmental Organization (NGO) environments.

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 5.1.7.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 5.1.7.0 is a Maintenance release of the JTLS-GO 5.1 series that includes an updated wespac51 demonstration database as well as updated repository data held in the repository51 database. There are no major Engineering Change Proposals (ECPs) included with this release, but there are a few minor ECPs that required no new data or data format changes, which are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

JTLS-GO Development Team Leader  
ROLANDS & ASSOCIATES Corporation  
120 Del Rey Gardens Drive  
Del Rey Oaks, California 93940 USA  
jtlsdev@rolands.com

WARNING – This document contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S. C., Sec 2751, et seq.) or the Export Administration Act of 1979, as amended, Title 50, U.S.C., App. 2401 et seq. Violations of these export laws are subject to severe criminal penalties. Disseminate in accordance with provisions of DoD Directive 5230.25.

Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoD Directive 5230.25 (date of determination). Controlling DoD office is Joint Staff, J7 - Joint Force Development.

Copyright 2020 - ROLANDS & ASSOCIATES Corporation - 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 Information Assurance Compliance .....	1-7
1.3.3 JTLS-GO High Level Architecture Compliance .....	1-8
1.4 DATABASE MODIFICATIONS .....	1-9
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 .....	1-9
1.5 INSTALLATION .....	1-9
1.5.1 Installation Instructions .....	1-9
1.5.2 Oracle Compatibility and Installation .....	1-9
1.5.3 Special Installation Instructions .....	1-10
1.5.3.1 OTH-Gold Ship Types .....	1-10
1.5.3.2 Generation Of Combat System Summary Files .....	1-11
1.5.3.3 Aircraft Type Foreign Key Definitions .....	1-12
2.0 ENGINEERING CHANGE PROPOSALS.....	2-1
2.1 JTLS-2020-14709 Create Controller Kill Aircraft Spreadsheet .....	2-1
2.2 JTLS-2020-14717 WHIP Layout - Save IMT State .....	2-1
3.0 SOFTWARE TROUBLE REPORTS .....	3-1
3.1 JTLS-2020-14702 Crash Accessing Destroyed Update Entity .....	3-1
3.2 JTLS-2020-14703 Crash Splitting Mission In Delay State .....	3-1
3.3 JTLS-2020-14704 Crash Canceling HRU Overwatch Task .....	3-1
3.4 JTLS-2020-14705 Crash Magic Moving Formation With Waiting Ship ..	3-2
3.5 JTLS-2020-14706 Default LOGFAS Profile If No Profile In Database ...	3-2
3.6 JTLS-2020-14707 Transferring UAV Mission From Naval Unit Crash ...	3-2
3.7 JTLS-2020-14708 Crossing Line Segments In Weather Front .....	3-3
3.8 JTLS-2020-14710 Crash Running Away Over Pole .....	3-3
3.9 JTLS-2020-14711 Restart Computing Estimated Return Time Crash ...	3-3
3.10 JTLS-2020-14712 Further Improve ATO Lite Capability .....	3-3
3.11 JTLS-2020-14713 Expand Information In Performance Report .....	3-4

3.12	JTLS-2020-14714 DDSC Map Layer Not Clearing Units/Targets .....	3-4
3.13	JTLS-2020-14715 Make Mission Order Group Viewer Consistent .....	3-4
3.14	JTLS-2020-14716 Add COMAO ID To WHIP SITREP .....	3-5
3.15	JTLS-2020-14718 Checkpoint With ATO.LITE.FLAG Restart Crash .....	3-5
3.16	JTLS-2020-14719 Bad Utility Field Not Caught On Game Startup .....	3-5
3.17	JTLS-2020-14720 Building Network Arc Mismatched Argument List ..	3-5
3.18	JTLS-2020-14721 Tech Tool Fixes .....	3-6
3.19	JTLS-2020-14722 Improve ICP Initialization Performance .....	3-6
3.20	JTLS-2020-14723 OTHGold Validate Track Block Start And End .....	3-6
3.21	JTLS-2020-14724 DDS Table Quick Link Button Labels .....	3-6
3.22	JTLS-2020-14725 Attach Order Rejected .....	3-7
3.23	JTLS-2020-14726 Emplaced Bridge Not Useful Over Barrier .....	3-7
3.24	JTLS-2020-14727 SVP Warning 1449 Auto-Corrections .....	3-8
3.25	JTLS-2020-14728 Air Mission Returned To Attached Unit .....	3-8
3.26	JTLS-2020-14729 Tech Tool Translate UOM Data In Messages .....	3-8
3.27	JTLS-2020-14730 Active Sonars With Limited Or No Power .....	3-8
3.28	JTLS-2020-14731 PSYOP/Unit Effectiveness Corrections .....	3-9
3.29	JTLS-2020-14732 Unused Subroutines Removed .....	3-9
3.30	JTLS-2020-14733 ATOT Airbase Editing .....	3-9
3.31	JTLS-2020-14735 Count Critical Orders Player Breakdown Info .....	3-10
3.32	JTLS-2020-14736 Trusted Air Operator Menu Alter Orbit Missing ....	3-10
3.33	JTLS-2020-14737 Submerged Submarine Tactical Intel Detection ..	3-10
3.34	JTLS-2020-14738 Drawing Tool Color Change Polyline To Polygon ..	3-10
3.35	JTLS-2020-14739 More Than 1000 Combat Systems BDA Report ..	3-11
3.36	JTLS-2020-14740 ATO Parser Ignores Airdrop Records .....	3-11
3.37	JTLS-2020-14741 Squadron Without Aviation Fuel Crash .....	3-11
3.38	JTLS-2020-14742 MDP Default Message Body Does Not Exist .....	3-12
3.39	JTLS-2020-14743 ATOT Airborne Point Taskings .....	3-12
3.40	JTLS-2020-14744 Improve Transportation Class SVPR Checks .....	3-12
3.41	JTLS-2020-14745 View End Time For Ambush Setup .....	3-12
3.42	JTLS-2020-14746 SVPR Warnings Not Generated .....	3-13
3.43	JTLS-2020-14747 C2PC WHIP Slide Overlay Radius Error .....	3-13
3.44	JTLS-2020-14748 MDP Logging Undeliverable Problem Messages ..	3-13
3.45	JTLS-2020-14749 Acquiring New Quick Altitude/Speed Orders .....	3-14
3.46	JTLS-2020-14750 Change ROE Against Unknown Side Message .....	3-14
3.47	JTLS-2020-14751 Squadron Names MDP Mission Report Error .....	3-14
3.48	JTLS-2020-14752 Counter Battery Response Weapon List Length ..	3-15
3.49	JTLS-2020-14753 Checkorder Case Mix Error .....	3-15
3.50	JTLS-2020-14754 Routine Check ADA Entity Exits Early .....	3-15
3.51	JTLS-2020-14755 ATO Parser Allows Zero Altitudes .....	3-15
3.52	JTLS-2020-14756 Set Aircraft Load Assignment Requirements .....	3-16
3.53	JTLS-2020-14757 Link 16 Block Filters And Reset Network .....	3-16
3.54	JTLS-2020-14758 WHIP Polygon Areas Duplicate Points .....	3-17

3.55 JTLS-2020-14759 ATOT Non-ATO Mission Refuel Chit Orders .....3-17

3.56 JTLS-2020-14760 Re-selection Order Graphics Display .....3-17

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 WSM - Many Messages Cause Lockup ..... 4-1

APPENDIX A. ABBREVIATIONS AND ACRONYMS ..... A-1

APPENDIX B. Version 5.1.7.0 DATABASE CHANGES ..... B-1

APPENDIX C. Version 5.1.7.0 REPOSITORY CHANGES ..... C-1

## 1.0 INTRODUCTION

### 1.1 SCOPE

This *JTLS-GO Version Description Document (VDD)* describes Version 5.1.7.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO<sup>®</sup>) software suite. JTLS-GO 5.1.7.0 is a Maintenance release for the JTLS-GO 5.1 series.

JTLS-GO 5.1.7.0 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 “wespac51”. No database format modifications have been made for this release, but a static data error for allowable OTH-Gold Ship Types was corrected in Version 5.1.1.0. This does require a user to execute a correction procedure for all of their Version 5.1.0.0 scenarios loaded in Oracle. Information on this procedure can be found on [Page 1-11](#).

Descriptions of minor Engineering Change Proposals (ECPs) implemented for this release are provided in [Chapter 2.0](#). Explanations of all Software Trouble Reports (STRs) corrected in this release are provided in [Chapter 3.0](#). Outstanding STRs are provided in [Chapter 4.0](#). Changes made to the JTLS-GO 5.1 engineering data repository are provided in [APPENDIX C](#).

JTLS-GO 5.1.7.0 executes on the Red Hat Enterprise Linux Version 7.6 64-bit operating system. 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 Portable Document Format (PDF) 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 5.1.3.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 5.1.2.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.1.3.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 5.1.2.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 5.1.3.0)



- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 5.1.6.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 5.1.5.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 5.1.3.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 5.1.2.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 5.1.2.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 5.1.5.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 5.1.2.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.1.2.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.1.5.0)
- *JTLS-GO Air Services User Guide* (JTLS-GO Document 24, Version 5.1.2.0)

### 1.2.3 Updated Documents

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.1.7.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.1.7.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.1.7.0)

### 1.2.4 New Documents

No new documents are delivered with JTLS-GO 5.1.7.0.

### 1.2.5 Delivered Software Components

JTLS-GO 5.1.7.0 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)
- 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)

When operating the TRIPP capability in current JTLS-GO releases, users are not prevented from logging into an actively running TRIPP resulting in a duplicate connection to the same Replay JXSR. A TRIPP, as documented, requires its own Replay JXSR to control and perform the replay of the recorded simulation events.

As a consequence of this situation, more than one user may concurrently control the playback of the game. The shared Replay JXSR will honor each of the user's playback requests and will then change what each connected user sees on their TRIPP instance. This issue will be addressed in a future JTLS-GO release by only permitting one login per TRIPP instance.

Organizations should develop their own procedures to ensure that only one user logs into a given TRIPP at a time.

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link 16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service
- 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

Instructions for installing JTLS-GO 5.1.7.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 5.1.7.0 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 “repository51” serves as a repository of engineering level data. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your Oracle scenario table-space. With JTLS-GO 5.1.7.0, it is possible to access and copy records from the repository51 database into your own developed scenarios.
- The scenario “wespac51”, which is based on the Western Pacific theater of operations, is suitable for training and demonstrations.

### 1.3 INTERFACE COMPATIBILITY

#### 1.3.1 Support Software

JTLS-GO 5.1.7.0 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 Edition Version 7.6 (ES), 64-bit architecture.

Security Enabled (SE) Linux must not be enabled on systems used to execute JTLS-GO or its components. Tests reveal that SE Linux-enabled systems cause frequent and random JXSR crashes and errors. Use of SE Linux to support JTLS-GO is currently not authorized.

- JTLS-GO 5.1 has been tested with the following versions of Linux 7:

Red Hat Linux 7.6 - This operating system license must be purchased, but it has been approved by the Defense Information Systems Agency (DISA) for use by U.S. Government Agencies.

Oracle Linux 7.6 - 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 Linux 7.6 - A free version of Linux 7 that has **not** been approved by DISA for use by U.S. Government Agencies.

- There are no restrictions on the operating system used for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 5.1.7.0 has been tested on the following operating systems:

Red Hat Linux Enterprise Edition Version 7.6.

CentOS Linux Version 7.6.

Windows 7 and Windows 10, which can be used only if the workstation is an external HTTP client of the simulation network.

- The JTLS-GO 5.1 series no longer uses Oracle Java, and has moved to the latest version of OpenJDK 8, which is OpenJDK 8 Version 242. We no longer deliver the Java Runtime Environment (JRE) within the JTLS-GO delivered software TAR files. Each user organization must obtain the latest version of the OpenJDK Red Hat Package Manager (RPM) and install the RPM on the servers used by JTLS-GO.
- JTLS-GO uses IcedTea to provide the OpenJDK web start capability that implements the web-enabled JTLS-GO functionality. The current version of JTLS-GO supports IcedTea version 1.8.3.

IcedTea version 1.7.1 does not support use of unsecured HTTP to access web-enabled JTLS-GO components for the DDS Client application. Users must configure their Glassfish instance to use HTTPS, by enabling the "Use SSL" column within the DCP when running with IcedTea version 1.7.1. The use of SSL is the current default setting in the DCP.

Red Hat Linux version 7.7 continues to distribute with IcedTea version 1.7.1. There are available RPM packages for a later version of IcedTea, so users who wish to use unsecure HTTP with JTLS-GO must explicitly install IcedTea 1.8.3 on the JTLS-GO servers and client workstations.

- JTLS-GO database tools require use of a certified Oracle database server and the full Oracle Client installation for runtime requirements. Additional installation details can be found in [Section 1.5.2](#) of this chapter.
- 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 service 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 II.5 (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 by a U.S. Government Agency that can obtain source code and plans to re-compile JTLS-GO SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc.

- ANSI C Compiler: It is not necessary to use a C compiler to execute JTLS-GO. This compiler is needed only by a U.S. Government Agency that can obtain source code and plans 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 needed only by U.S. Government Agency that can obtain source code and plans 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 (Database Development System) application uses these open source libraries:

JFreeChart, licensed under LGPL (GNU LESSER GENERAL PUBLIC LICENSE) 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 the COP, C2PC, or other C4I systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

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

- JTLS-GO 5.1 implements SSH Tunneling between Apache and the services, and among the services. Rigorous testing should be done prior to use in any exercise, and particular attention should be paid to network performance under load.

### 1.3.2 JTLS-GO Information Assurance Compliance

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

One of the DoD IA requirements is that the software must implement a methodology that ensures that the end user keeps the software up-to-date and properly installs all security patches. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. After a certain date, all Java-related programs stopped working, whether connected to an open network or not. All JTLS-GO releases were closely linked to the Java expiration date.

Information available at the time of this release indicates that OpenJDK will not implement an expiration date. In order to fulfill the DoD IA requirement, Maintenance releases of JTLS-GO will be released as new versions of OpenJDK are released. In a future version of JTLS-GO, all delivered Java programs will check that the latest version of Java is being used. Each local installation will be able to override the warning generated when the program detects that the latest version is not properly installed.

JTLS-GO has completed the IA program mandates and the JTLS-GO 5.1 series of releases has been granted an Authority To Operate (ATO) on DoD systems. Contact the U.S. Government Program Manager, Mr. Don Weter ([donald.e.weter.civ@mail.mil](mailto:donald.e.weter.civ@mail.mil)), for additional information.

### 1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 5.1.7.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO uses the Federation Object Model (FOM), located in the \$JGAME/data/hla directory. Federation testing of JTLS-GO with CAE's GESI wargaming system has passed all internal test procedures. Future plans include expanding the capabilities of the JTLS-GO-GESI federation, called "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 as a federation. 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.

All JTLS-GO software needed to run GlobalSim is included in this delivery.

The HLA RTI (Run Time Infrastructure) executive program recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 5.1.7.0 delivery. Users may obtain a full installation package of the RTI executive program from Pitch Corporation ([www.pitchtechnologies.com](http://www.pitchtechnologies.com)). 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 5.1 series and the previous JTLS-GO 5.0 series database structure.

To upgrade your JTLS-GO 5.0 scenario to JTLS-GO 5.1 compatibility, see instructions listed in Chapter 3.1 of the *JTLS-GO DDS User Guide*.

### 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 5.1.0.0 formatted data from your Oracle database server into the JTLS-GO 5.1.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in Appendix B.11 of the *JTLS-GO DDS User Guide*. This procedure will reorganize the structure of the .gs and .scf symbols-related files.

### 1.4.2 JTLS-GO Using New Default Symbol Set

R&A considers the Default Symbol Set delivered with JTLS-GO 5.1.7.0 as a configuration managed item. End user organizations should refrain from altering the Default Symbol Set, but are free to use these symbols in their scenarios and alter the scenario symbol set to meet specific organizational needs.

### 1.4.3 Standard Repository Changes

The JTLS-GO 5.1 series of JTLS-GO is the first series in which R&A is delivering an unclassified data repository called “repository51”. [APPENDIX B](#) provides a summary of the data structure changes made to the data repository. Since no data structure changes have been made in this Maintenance release, [APPENDIX B](#) is empty. Refer to Appendix B in the *JTLS-GO 5.1.0.0 Version Description Document*, included with this release, for data structure changes made for the JTLS-GO 5.1 series.

## 1.5 INSTALLATION

### 1.5.1 Installation Instructions

The *JTLS-GO Installation Manual*, a PDF file available for direct download, is part of this JTLS-GO delivery. It provides detailed instructions for installing a new version of JTLS-GO.

### 1.5.2 Oracle Compatibility and Installation

A full Oracle Client (not Instant Client) installation that matches your database server version is currently a requirement for running some JTLS-GO applications. The Oracle Instant Client is not sufficient for JTLS-GO applications because certain Oracle utilities, such as sqlldr, imp, exp, and tnsping, are missing. If you have applied a patchset to your database server, the same patchset



should be applied to the Oracle Client installation. A 64-bit Oracle Client installation must be used.

The JTLS-GO scenario database modification process also expects Oracle 11.2.0.1 or higher full Oracle Client installation. Some sites NFS mount their database server as Oracle Client; other sites prefer a full installation of the Oracle Client in a different directory that mounts to JTLS-GO (a simple NFS mount will suffice). Your system administrator can choose the appropriate installation.

Assigning the full Oracle Client installation location (or mount point) as the ORACLE\_HOME in the JTLS-GO .cshrc file allows connecting to an Oracle database server (11.2.0.1 or higher - including 11gR2 XE) running on any Oracle-certified database server platform.

Oracle offers free Express Editions (XE) of the Oracle relational database management system. Compared to the 11gR2 XE version, the newer 18c XE has a larger footprint and a much more complex database architecture. For test environments and scenario building purposes, or for collecting AAR data for a short period of time, the installation and setup of the 11gR2 XE version is much simpler.

The DDS application utilizes the Oracle GlassFish J2EE server, which, like the JTLS-GO WHIP Apache server, is delivered with JTLS-GO and requires no separate installation.

Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional details pertaining to the Oracle installation.

### 1.5.3 Special Installation Instructions

This section describes special instructions that should be followed because of errors corrected in this version and previous bug releases for the JTLS-GO 5.1 series.

#### 1.5.3.1 OTH-Gold Ship Types

JTLS-GO 5.1.0.0 was delivered with an old list of OTH-Gold ship types. This problem was corrected in JTLS-GO 5.1.1.0. To properly implement this solution, users must execute the following additional procedures for each of their JTLS-GO Version 5.1.0.0 scenarios loaded in Oracle.

Users who have already executed this procedure for their scenarios after installing JTLS-GO Version 5.1.1.0, 5.1.2.0, 5.1.3.0, 5.1.4.0, 5.1.5.0, or 5.1.6.0 do not need to execute the provided procedure again.

New OTH-Gold ship types were added to JTLS-GO, due to STR JTLS-2019-14238.

1. Execute the following command:

```
cd $JTLSHOME/script/dds/version5.1/scripts/
```

2. Execute the following command:

```
sqlplus yourScenario/OraclePassword @update_oth_gold_types.sql
```

3. Verify the related execution listing file under the \$JDATA/scenario/ directory for errors.
4. Unload your scenario using the JTLS-GO Menu, Options 1 -> 1 -> 4
5. Verify the .srw ascii file for your scenario.

To fix the problem renaming Force Sides, due to STR JTLS-2019-14267:

6. Reload the database.

### 1.5.3.2 Generation Of Combat System Summary Files

STR JTLS-2019-14518 Move Combat System Summary Files, delivered as part of JTLS-GO 5.1.4.0, solved the problem of overwriting the summary Combat System Character Separated Value (.csv) files during Batch Runs. This STR was solved by moving the location of the .csv files from the game/<scenario\_name>/location directory to a sub-directory under each checkpoint.

For every active game scenario, one of the following procedure options must be executed to establish the directory structure needed by the STR solution. Users who have already executed this procedure for their scenarios after installing JTLS-GO 5.1.4.0, do not need to execute one of the selected options again:

- Option 1: Rerun the Setup Procedure for each active scenario.
- Option 2: Hand-create a cbtsys\_summary sub-directory in the game/<scenario\_name> directory. This can be accomplished using the following steps from a command terminal for each existing game that has already been set up and prepared for execution:
  - a. Enter the command: **game** - this puts the terminal in the \$JTLSHOME/game directory.
  - b. Enter the command: **cd <scenario\_name>** (for example, **cd wespac51**) - this puts the terminal in the game/<scenario\_name> directory.

- c. Enter the command: `mkdir cbtsys_summary`. This creates the necessary new directory.

### 1.5.3.3 Aircraft Type Foreign Key Definitions

STR JTLS-2019-14541 Referenced Aircraft Class Can Be Deleted, delivered as part of JTLS-GO 5.1.5.0, solved the problem that allowed the deletion of aircraft types that were referenced by units. The STR was solved by removing a rule in the database that allowed the user to set the aircraft type-related foreign keys of the unit tables to NULL.

After loading JTLS-GO 5.1.5.0, the user must unload and then reload their JTLS-GO 5.1 scenarios to have the new foreign key definitions in their database schemas.

Users that have accomplished this procedure after installing JTLS-GO 5.1.5.0, do not need to re-execute this procedure after installing this version of JTLS-GO.

## 2.0 ENGINEERING CHANGE PROPOSALS

The following model capabilities were added to JTLS-GO 5.1.7.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

These ECPs were built on an emergency basis to support a large upcoming NATO exercise. The documentation for these ECPs will be completed as part of the JTLS-GO 6.0 series, due to be released in the Summer of 2020. These ECP are technically not closed until all documentation requirements have been fulfilled. These ECPs will also be included in the JTLS-GO 6.0 Version Description Document.

### 2.1 JTLS-2020-14709 Create Controller Kill Aircraft Spreadsheet

#### Summary of Model Change Request

A spreadsheet order should be added to the Air Mission IMT screen to kill highlighted air missions.

#### Design Summary

This capability was added.

To access this new spreadsheet, open a Controller WHIP. Bring up an Air Mission IMT and highlight some air missions that are flying. Right-click and select the "Kill Mission" spreadsheet. One order will be created for each Air Mission and the spreadsheet can be submitted.

### 2.2 JTLS-2020-14717 WHIP Layout - Save IMT State

#### Summary of Model Change Request

IMT States, such as the ordering, sorting, and widths of columns, were not saved to the WHIP Layout data.

#### Design Summary

The following IMT state information is now saved to WHIP Layouts:

- Column ordering
- Hidden columns
- Sorted columns
- Column sizes



## 3.0 SOFTWARE TROUBLE REPORTS

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

[Chapter 4.0](#) describes STRs that remain outstanding from previous versions; however, because of the model enhancements included with JTLS-GO 5.1.0.0, no STRs identified in prior JTLS-GO series are considered valid. Uncorrected errors identified for JTLS-GO 5.1.7.0 are documented in this chapter and should be corrected in future maintenance releases.

### 3.1 JTLS-2020-14702 Crash Accessing Destroyed Update Entity

*The model was accomplishing an instantaneous update of an enemy object just before firing on the object. The update was rejected because the air mission held too many updates in its information set. This eventually caused a crash, because the firing logic assumed that the update was still active.*

STR JTLS-2020-14657 corrected a problem that too many “ghost” detections were held in an air mission’s information set. That STR reduces the probability that this problem can ever occur. The user that reported this crash was not using the version of the code in which STR JTLS-2020-14657 was delivered, but the error occurred and needed to be corrected.

If the update is rejected because of too many updates held in the mission information set, the calling routine was notified of the rejection and now properly handles the situation.

### 3.2 JTLS-2020-14703 Crash Splitting Mission In Delay State

*The model crashed when a user attempted to split an air mission that had no assigned aircraft while it was in an aircraft delay state.*

The model did not check whether the specified parent mission had any assigned aircraft. It only checked that the mission was not in a “Scheduled” posture state.

The logic was changed. If the aircraft has no aircraft assigned, then the split is based on the requested number of aircraft. If the aircraft has any aircraft assigned, then the split is based in the current number of aircraft assigned to the parent mission.

### 3.3 JTLS-2020-14704 Crash Canceling HRU Overwatch Task

*Units that are in an Attack posture are not allowed to have HRUs providing help in an Overwatch mode. This is also true for targets owned by units in the Attack posture. If an HRU was overwatching a target and the target's owning unit was given an Attack order, the model canceled the HRU's Overwatch Task. When writing out the Task cancellation message, the model crashed.*

The code used to generate the task cancellation message did not consider the possibility that the HRU was overwatching a target owned by the unit that just started to fulfill its Attack Task. The code was corrected to consider this possibility.

The cancellation message was also improved to make sense when the HRU was overwatching a target and not the unit itself.

### 3.4 JTLS-2020-14705 Crash Magic Moving Formation With Waiting Ship

*The magic move formation order crashed when it had a ship that was waiting, because it could not find a good station location. The ship was expected to have a waiting task, but it had none. This caused the model to crash.*

Three changes were made to the model to fully correct the problem.

1. A situation was found in which the logic did not properly create the ground wait task.
2. Through testing, it was discovered that the user was allowed to cancel the waiting task, altering the expected structure of the waiting ship.
3. For safety, when processing the magic move order, the model now checks if the waiting ship in fact does have the waiting task. If not, a logic error message is generated allowing the magic move to be properly completed.

### 3.5 JTLS-2020-14706 Default LOGFAS Profile If No Profile In Database

*JTLS-GO is capable of creating LOGFAS data files for the main Force Side in a game. This main Force Side is known as the Common Operational Picture (COP) side. The entire force structure on the COP side needs to be in a LOGFAS Profile. The SIP helps the database builder, by creating the LOGFAS profiles specified in the data initialization files. If not all COP units are included in a database-specified LOGFAS profile, the SIP automatically defines a Default Profile to hold the units, not in any of the database profiles.*

*The SIP was creating a default profile that contained all COP units when the database did not include any LOGFAS profiles. In other words, it was impossible to “turn off” LOGFAS setup, even when it was unneeded and even undesired for a given scenario.*

The SIP was changed. It only creates a default profile when the database contains at least one LOGFAS profile.

### 3.6 JTLS-2020-14707 Transferring UAV Mission From Naval Unit Crash

*The model crashed when the user transferred a UAV mission that launched from a naval unit to another naval unit.*

The logic error associated with transferring non-squadron air missions has been fixed.

During testing, it was found that if the UAV returned to a non-squadron unit that previously did not have any UAVs, the newly arrived UAV could not take off again. This problem was also solved.

### 3.7 JTLS-2020-14708 Crossing Line Segments In Weather Front

*A Weather Front polygon had crossing line segments.*

SVP Error 166 is now generated if a Weather Front polygon has crossing line segments.

### 3.8 JTLS-2020-14710 Crash Running Away Over Pole

*The model crashed trying to compute the next best runaway location when an air mission was running away from an interceptor and was directly over the North Pole.*

The logic was corrected to handle this issue, but a new issue has been opened. In this specific case, the interceptor could not move forward because it was told to avoid air defense. The interceptor was stationary but the mission being intercepted continue to run away. This issue will be solved in future versions of JTLS-GO.

### 3.9 JTLS-2020-14711 Restart Computing Estimated Return Time Crash

*The model crashed restarting from a checkpoint. There was an active mission that had no tasks and the “estimate return time” algorithm crashed.*

The cause of the problem was not discovered, but it was determined that the mission had no tasks and no events scheduled. The “estimate return time” code was changed to check for this situation, and print out a logic error when the exception is encountered. The engineering staff is still attempting to find the sequence of events that resulted in an Escort Mission having no events and no tasks scheduled.

### 3.10 JTLS-2020-14712 Further Improve ATO Lite Capability

*In JTLS-GO 5.1.6.0, an Engineering Change Proposal (ECP) was delivered to create an Air Tasking Order (ATO) light capability that would allow a user to run the model quickly in order to watch the air movement plan. While testing this capability, the ECP sponsor wanted the game to run even faster and decided to stop air missions from detecting things and things detecting air missions.*

The following routines are now also skipped if in ATO Lite Mode:

- Recce missions looking for targets.
- AWACS missions looking for IADS communication sites to which they can link.
- Placing the mission's sensor coverage tags on the game board.
- Air Mission Detections are disabled.



### 3.11 JTLS-2020-14713 Expand Information In Performance Report

*When the built-in performance flag function is turned on, the model keeps track of game speed since the last time game speed was set and the average amount of time spent in each event. This report lists the time of the report and the internally collected statistics.*

*A user requested that the time of last game change also be printed in the report. This enabled the performance evaluation team to decide the length of time over which the statistics were collected.*

The requested data was added to the report.

### 3.12 JTLS-2020-14714 DDSC Map Layer Not Clearing Units/Targets

*When leaving the IADS edit button on, the DDSC map layer did not always turn off the displayed IADS targets.*

When the IADS edit mode is selected on the DDSC, the IADS edit layer automatically displays SAM\_AAA, SENSOR\_SITE, and COMM\_SITE targets. These targets are needed to properly create IADS links. There are two ways to exit the IADS editing mode.

- The user can unset the IADS edit mode by clicking a second time on the IADS edits button. Using this method, all of the displayed IADS targets are automatically cleared on the map.
- The user can select a different edit mode by clicking on the alternate edit button. When using this method, the displayed IADS target were not automatically cleared. This problem was corrected.

### 3.13 JTLS-2020-14715 Make Mission Order Group Viewer Consistent

*The WHIP Order Group Viewer was not consistent across all air missions that could be generated by the Air Tasking Order Translator (ATOT) and passed to the WHIP for viewing and submission.*

Each Air Mission order was reviewed and the Mission Name field and the Launching Squadron field are now consistently displayed in the Order Group Editor. These two fields along with the following fields are viewable in the Order Group Viewer:

- Call Sign
- NATO COMAO ID
- Air Mission Package ID

### 3.14 JTLS-2020-14716 Add COMAO ID To WHIP SITREP

*As an extension of the NATO Combined Air Operations ID (COMAO), a mission's COMAO ID should be added to the WHIP SITREP for the mission.*

The Air Mission SITREP now includes the COMAO ID. The field ordering and organization on the SITREP were reviewed and some minor data ordering changes were also made.

### 3.15 JTLS-2020-14718 Checkpoint With ATO.LITE.FLAG Restart Crash

*A user turned on the ATO.LITE.FLAG, proceeded to run the game, and then tried to take a checkpoint. The model refused - just as the design indicated. The user then went to the JODA console and executed 'order dloff ATO\_LITE\_FLAG' to turn the flag off and then took a checkpoint. The model crashed coming up from the checkpoint.*

If the user tries to turn off the ATO.LITE.FLAG after it has been turned on, the model will refuse the order. The controller will receive a message explaining once in ATO Lite Mode, there is no ability to go back.

### 3.16 JTLS-2020-14719 Bad Utility Field Not Caught On Game Startup

*While developing JTLS-GO 6.0, the Implementation Team encountered a problem when loading the definition of feasible model orders. A utility referenced another utility that had not yet been defined*

*In this situation, the code called the SNAP.R error handler, which normally would put the user in simdebug and ask if they wanted a checkpoint. However, because the reading of the order definition data happens during initialization, a checkpoint is not possible. The only thing that happened was that the scenario lock file was removed and the model continued, with the user being unaware of the issue.*

The Implementation Team decided that although this issue did not exist in JTLS-GO 5.1 series, a check should be added for safety reasons. The call to the SNAP.R error handler was switched to a call to SIMDEBUG.R, which will always put the model into the SimDebug mode. In this way, the Test Team can ensure any future problems are found before the delivery of a release.

### 3.17 JTLS-2020-14720 Building Network Arc Mismatched Argument List

*A mismatch was uncovered in the arguments passed to a routine that adds a road node during the creation of a new arc. An extra parameter specifying the distance was being passed. This error would likely result in a crash by corrupting the value returned from the routine, the pointer of the new node.*

The calling routine was modified to omit the extra parameter.

### 3.18 JTLS-2020-14721 Tech Tool Fixes

*Several issues were found and fixed in the Tech Tool.*

The following issues were found and fixed in the Tech Tool:

- When printing out classification, the Tech Tool would print out the delimiter “&” along with the actual classification string. The “&” delimiters are no longer included in the classification that appears at the top of a message.
- The Tech Tool would also crash when opening messages that contained country code data. The Tech Tool was not loading Country Code data, so whenever the Tech Tool attempted to display a message that used Country Code data, it would throw a null pointer exception. Country Code data is now loaded to the Tech Tool upon initialization.
- In addition, the Tech Tool is no longer case sensitive when implementing searches.

### 3.19 JTLS-2020-14722 Improve ICP Initialization Performance

*The Interface Configuration Program (ICP) save process took a long time at initialization when trying to stop a yet-configured Glassfish domain.*

The issue was caused by the ICP attempting to stop a empty domain. Logic was added to check the domain before issue a stop command.

### 3.20 JTLS-2020-14723 OTHGold Validate Track Block Start And End

*The OTHGold Message Service was being started with an old ICP configuration. This contained Track Block Start and End values of -1. This caused the OTHGold Message Service issues when generating unique track IDs.*

The OTHGold Message Service now validates the Track Block values on startup. If bad values are encountered, the service will exit.

### 3.21 JTLS-2020-14724 DDS Table Quick Link Button Labels

*Some DDS tables have quick-link buttons that open their child tables when pressed. The buttons are labeled by the names of the child tables, and if a parent table can be deep-copied, its child tables whose data will be deep-copied have a “Double-Dagger” symbol (‡) appearing next to the table name.*

*However, a few of the tables such as “Supply Type” and “BE Facility” have their child tables mistakenly labeled with the double-dagger symbol, when in fact the data from these child tables was not deep-copied.*

The DDS screen definition files were corrected to match the values of the "deep-copy" attribute for a "quick" element that defines a table's quick-link button with what data is being actually deep-copied.

### 3.22 JTLS-2020-14725 Attach Order Rejected

*An Attach order was submitted to a ground unit that owned aircraft combat systems, but the receiving unit did not. The order was not accepted by the WHIP order panel. The error message indicated that the aircraft classes did not match between the two units.*

The Attach order prevents a player from attaching a unit with aircraft (e.g. RQ7B.SHADOW) to another unit that also has aircraft, but of a different type (e.g. MQ9.REAPER). This feature is by design because mixed aircraft types are not permitted in the same unit.

In the reported case, the receiving unit had no aircraft, which should not have prevented the unit with aircraft from attaching. If the attaching unit had no aircraft and the receiving unit did, the order should also be permitted. The constraint in the Attach order panel was modified to accept the order for those two cases.

As a result of this correction, new code was added to the transfer combat systems logic to assign the aircraft type to the receiving unit if it didn't already have aircraft assigned. This change avoids a subscript out of range error when the model tries to initialize the aircraft range rings (based on aircraft type) for the receiving unit.

### 3.23 JTLS-2020-14726 Emplaced Bridge Not Useful Over Barrier

*A Bridge Operation order was submitted to an engineering unit to emplace a bridge across a tank ditch (barrier) without specifying an associated road or rail arc. After the bridge target was emplaced, ground units were unable to cross the tank ditch.*

The problem was found in the logic that searches for any barrier whenever a river arc is not near the bridging location. The logic failed to initialize a variable used to compare the distance between a candidate barrier and the bridge to find the closest barrier segment. As a result, all candidate barriers were ignored. The bridge was emplaced, but not usable because the model didn't find any barriers that needed to be crossed. The error was corrected by properly initializing the comparison variable. Note this problem did not exist for a bridge emplaced near a river arc.

While investigating this problem, an incorrect reference to the barrier's width was discovered and corrected to access the proper data element.

Finally, the Bridge Operations Report was improved to notify the player when a bridge is emplaced too far from the barrier or river it is intended to facilitate crossing. Also, the name of the associated arc was added to the message.

### 3.24 JTLS-2020-14727 SVP Warning 1449 Auto-Corrections

*When SVP Warning 1449 is generated, Auto-Correction Options 1 and 2 bring up the supply table for the unit's HUP/SUP, so the user can set the Bring to Theater Value or Basic Load value, for the specified supply category. In the message, the user is informed of the exact value that they should use to alter the table.*

Options 1 and 2 now automatically execute the update statement using the supply values calculated by the routine CHECK.SUPPLY.LEVELS.

Two new options have also been added. Option 6 brings up the unit's Prototype Table Editor so the Automatic Supply Calculation Tool can be executed for the TUP/SUP. Option 7 brings up the Unit's Table Editor so the Automatic Supply Calculation Tool can be executed.

### 3.25 JTLS-2020-14728 Air Mission Returned To Attached Unit

*A user sent an order for an air mission to launch from one squadron and return to a second squadron. Between the launch and the execution of the transfer task, the "return" squadron had been attached to a different squadron. The mission should not return to an attached squadron since it is considered inactive.*

Before an air mission executes a TASK CHANGE HQ, the model checks to make sure the return unit is still good. If the unit has a posture of WIPED OUT or OUT OF GAME, then the task is canceled and the squadron will stay with its launching unit. If the return unit has a posture of ATTACHED, the mission will return to the unit that the return unit is attached to if it is still a good unit. If there are layers of attachments, the mission will return to the final unit that is not attached to any other unit.

### 3.26 JTLS-2020-14729 Tech Tool Translate UOM Data In Messages

*When outputting messages that contained UOM data, such as time or weight, the Tech Tool would print out "error" in place of the UOM.*

The Tech Tool now properly converts UOM data when transforming messages. The problem was that the tables that were supposed to hold UOM data were empty. The internal tables are now being populated with proper UOM data.

When converting an XML time string to a "gametime" decimal value, the calendar object being passed was set using PST. The calendar object should have been using "GMT" since this is the timezone used in the game.

### 3.27 JTLS-2020-14730 Active Sonars With Limited Or No Power

*An active sonar was not able to make a detection. The user had to increase the power of the sensor in order to make detections.*

A new SVP Warning 1276 has been added to make sure the sensor has enough power (ST.POWER) to achieve the full range (ST.RANGE) of the sensor. It was also discovered that active sonars in the wespac51 scenario have a power of zero. Hence, a new SVP Error 254 has been added to alert users when an active sonar has a zero power.

### 3.28 JTLS-2020-14731 PSYOP/Unit Effectiveness Corrections

*Several issues were uncovered concerning the Combined Effectiveness column displayed on the Unit Info IMT.*

Each Unit has a Unit Effectiveness and a PSYOP Results Multiplier attribute. The Combined Effectiveness is the product of the current unit effectiveness and the PSYOP Results Multiplier. As the Unit experiences enemy induced PSYOP (leaflets or broadcast), its PSYOP Results Multiplier is reduced causing the Combined Effectiveness value to decrease.

Although the model was correctly assessing PSYOP effects on the Unit, an error in the code prevented the Combined Effectiveness value from being recalculated at the Unit Hourly Processing event. The code also did not properly pass the Combined Effectiveness and PSYOP Results Multiplier to the JODA for display on the IMT. These errors were corrected.

Because the PSYOP Results Multiplier is a perceived attribute, it appears on the Foreign Unit Info IMT under a column labeled Unit Effectiveness. This caused confusion. The column was renamed Perceived Effectiveness to better define the value. Also, the Help texts were revised in both IMTs and the JDSP protocol.

### 3.29 JTLS-2020-14732 Unused Subroutines Removed

*Several model subroutines were identified as obsolete and no longer used. The routines were formerly used to calculate off road travel time by ground units. The routines were replaced and superseded by new routines during the transition to terrain grid movement methodology.*

The obsolete subroutines were removed from the CEP directories. They were not called by any other routines.

### 3.30 JTLS-2020-14733 ATOT Airbase Editing

*While using the Airbase linking screen in the ATOT WHIP module, a user can only enter a single character when entering the airbase name in the drop down list. For example, if looking for "Roosevelt", only "R" can be searched for.*

The ATOT WHIP module has been modified to use a new table cell editor that allows for more robust searching. This new editor is now used by all drop-down lists across the ATOT.



### 3.31 JTLS-2020-14735 Count Critical Orders Player Breakdown Info

*The Count Critical Order (CCO) program provides a method for the Controller to determine the number of orders that have been sent or the number of messages that have been generated. When dealing with orders, it provides a breakdown of which types of orders were sent between every checkpoint and a breakdown of how many orders were sent by each player. The section that listed the breakdown of orders sent by each player was actually listing the breakdown of orders sent by each side.*

The problem was being caused by the CCO reading an old format for the orders. The format for the orders had been changed, removing two unused integers but adding an XML time and the sending force side, but the code had not been updated. An update of the read code fixed the issue.

### 3.32 JTLS-2020-14736 Trusted Air Operator Menu Alter Orbit Missing

*JTLS-GO is delivered with a number of configuration-managed Player menus which control which orders, quick buttons, and IMT screens, and modules are available to a WHIP. The one for the Trusted Air Player was missing the ability to send an Alter Orbit order, either via the menus or via a quick button. This is an essential task for any Air Player.*

The data file for the trusted air operator menu was modified to include the Alter Orbit order within both the pull-down menus and as a quick button.

### 3.33 JTLS-2020-14737 Submerged Submarine Tactical Intel Detection

*A submerged submarine with passive and active sonars turned off detected a foreign submerged submarine using only its organic tactical intelligence range capability. A submerged submarine should be excluded from the organic reporting capability.*

While logic exists that takes into account a submarine's depth when assessing detectability by active and passive sonar sensors, there was no such logic when assessing detectability by the organic intelligence capability. Submarines were automatically detected if within range of the organic tactical intel capability, regardless of depth.

New code was added to exclude submerged submarines from the organic tactical intelligence range detection by all units, not just submarines. As before, submarines that are at snorkel/periscope depth or surfaced are not excluded and will be detected if within tactical range.

In addition, new code was added to exclude covert HRUs and Unmanned Underwater Vehicles (UUV) HRUs from detection by the organic tactical intelligence capability.

### 3.34 JTLS-2020-14738 Drawing Tool Color Change Polyline To Polygon

*If a user would change the color of a Polyline, the Polyline would turn into a polygon.*

The color change problem on a Polyline was corrected by internally not treating the Polyline as polygon. While testing this fix, it was discovered that when previewing a drawing change and exiting the preview mode with the “X” button, the change was committed. This was deemed inappropriate. The preview is only committed, when the user selects the preview “OK” button, which returns the previewed drawing into its original form.

### 3.35 JTLS-2020-14739 More Than 1000 Combat Systems BDA Report

*The BDA Report was being generated using a three-digit integer for a combat system index. However, within some scenarios, combat systems now number over 1,000. This results in the message being printed with three asterisks (\*\*), instead of a valid integer. The end result is that the message does not provide the information to the user that is required.*

This is not a problem solely limited to combat systems. With modern scenarios and the growing size of the repository, we expect to see many entities with counts over 1,000.

To accommodate these growing database sizes, we have adopted the coding standard that we should provide five integers for all database entity types when printing out an index. Rather than just fixing this single issue, a full code review of the CEP was conducted to ensure that database entity index values use five integers when being output.

### 3.36 JTLS-2020-14740 ATO Parser Ignores Airdrop Records

*The ATO Parser does not consider locations specified in an AIRDROP record.*

The NATO Interactive Command and Control (ICC) Parser reads ATO records associated with mobility missions, as it does for the records of all ATO missions. However, a record that is used for specifying air drop locations, the AIRDROP record, was not being considered by the Parser. This record gives locations in latitude/longitude format that the Parser can process. When an AIRDROP record is found, the Parser obtains the location and reverts the location from the previous AIRMOVE record to this AIRDROP location, if it is specified. The AIRMOVE gives the tasking but the AIRDROP refines this tasking into a specific location rather than a landing site.

The Parser has been modified now to process the locations found in the AIRDROP record.

### 3.37 JTLS-2020-14741 Squadron Without Aviation Fuel Crash

*A naval helicopter squadron was magic moved to a shore location and then mission orders were sent to it. The squadron did not have any aviation fuel, previously relying on the ship it was on board to provide such assets. The mission was therefore rejected, but the model crashed when attempting to format a rejection message, assuming the squadron would have an aviation fuel record. Since it had no fuel, and never had any fuel, it did not have such a record.*

The code was modified to check for the existence of the aviation fuel record before attempting to access it. If the record does not exist then it is assumed that the mission could not be created due to a lack of fuel.



### 3.38 JTLS-2020-14742 MDP Default Message Body Does Not Exist

*When setting up an email recipient with an attachment in the Message Delivery Program (MDP) configuration GUI, a file with the name default\_body.txt is used to provide the contents for a message body. This file has traditionally resided in the \$JGAME/data\_site/mdp directory, but is no longer there.*

Traditionally the \$JGAME/data\_site directory is where user configured files should reside, and it should never have included the Configuration Managed default\_body.txt file. The decision was made to restore this file but place it in the \$JGAME/data/mdp directory with other non-user configured files. Individual users may still create and add their own message body files to the \$JGAME/data\_site/mdp directory and the MDP configuration GUI will permit a file from either location to be chosen as the message body file.

### 3.39 JTLS-2020-14743 ATOT Airborne Point Taskings

*The ATOT does not consider airborne points for mobility mission types. The tasking points are all made into landings at a specific unit where the unit name is always "NONE".*

The ATOT code was modified to write mobility mission orders with airborne points whenever the ATO contains a mobility mission with an airdrop versus a landing unit.

### 3.40 JTLS-2020-14744 Improve Transportation Class SVPR Checks

*The code that performs checks on the transportation classes is inconsistent and incomplete.*

The code in Routine CHECK.TRANSPORTATION.CLASS was restructured and made more efficient to check the following functions of the transportation classes:

1. Does a TANKER Truck carry only WET supplies, A CARGO Truck Dry supplies and a HET carries Self-Propelled Supply categories.
2. Can every supply category be carried by at least one of the transportation classes.
3. If a combat system is designated as a special capability of CARGO, TANKER or HET, is the Combat System Special Capability Subcategory the correct type for it's purpose.

The corresponding SVPR Warnings are 1143 through 1149.

### 3.41 JTLS-2020-14745 View End Time For Ambush Setup

*When an HRU is going to perform the AMBUSH task, the HRU must first perform the AMBUSH.SETUP task to prepare for the ambush. When the AMBUSH.SETUP tasking appears on the IMT, the user cannot see when the setup will be complete.*

The routine HRU.PREPARE.FOR.AMBUSH always calculated a "SETUP.DONE.TIME." However, it was not displayed in the IMT. This PET.END.TIME is now initialized to the SETUP.DONE.TIME and can be viewed in the IMT HRU Tasking window.

### 3.42 JTLS-2020-14746 SVPR Warnings Not Generated

*SVPR Warnings 1232 and 1234, which perform checks on the number of standard responses, are not being generated.*

There was a simple error in the top of the routine which caused the routine to exit without performing the checks for Warnings 1232 and 1234. Once the correction was made, additional changes were needed to improve the reporting of the Warnings 1232 and 1234.

Warning 1151 was also added to report when combat systems of conflicting attrition types were using the same combat arms types, and Warning 1157 was added to report any combat arms types that are not used. The code was corrected and tested.

### 3.43 JTLS-2020-14747 C2PC WHIP Slide Overlay Radius Error

*A slide, which included a circle, was created on a WHIP and then converted to a C2PC overlay. The circle had a radius of approximately 100 KM on the WHIP slide, but this showed up as an enormous circle covering half the world on the C2PC overlay.*

*This was tracked down to the radius Unit of Measure (UOM) used in the WHIP slide, which is normally decimal degrees but was in this case set to kilometers. The conversion stylesheet assumed 100 KM was 100 degrees and converted this value to nautical miles for the C2PC overlay.*

The WHIP uses the openmap libraries to construct graphics. The stylesheet that converts a WHIP slide to a C2PC overlay was expanded to handle the various UOMs that might be used to generate a circle radius.

In addition, it was also noted that the color used in the WHIP slide were color names, when traditionally we have only seen encoded RGB test (such as #FA238D). C2PC expects decimal red-green-blue (RGB) colors in the overlay. Therefore, the stylesheet was also expanded to recognize a color name versus an RGB-encoded color, and if a color name is used attempt to lookup the proper decimal RGB values.

### 3.44 JTLS-2020-14748 MDP Logging Undeliverable Problem Messages

*The MDP occasionally cannot process a message from the CEP and does not transmit the message, (such as an email to a remote client). No report of the message's non-delivery is given to the user.*

The MDP had code for error reporting of some message processing problems, but the format of the report was generated improperly. This formatting error caused them to be ignored by the log display in the MDP\_GUI.

The code for the MDP\_GUI has an extensive Application Interface (API) for logging various reports, but was lacking a report for undelivered messages. The following changes were made:

- The MDP\_GUI was modified to allow reports of undelivered messages.
- The JTLS-GO message library was modified to detect and reveal various stylesheet errors during the parse process (message body and subject).
- The MDP was modified to detect the errors, adjust the existing incorrect message delivery problems to the new report API, and report them properly through the new undelivered message report in the MDP\_GUI. This new report gives the message number, checkpoint and description.

#### 3.45 JTLS-2020-14749 Acquiring New Quick Altitude/Speed Orders

*To be consistent with the previously existing Change Mission Parameter Order, the quick versions of the Change Mission Speed and Change Mission Altitude orders were labeled as acquirable. The routine that generates the acquired information was not changed to expect these two orders.*

The routine that generates the message to acquire this information was changed to handle the randomly acquired Change Mission Speed and Change Mission Altitude Orders.

#### 3.46 JTLS-2020-14750 Change ROE Against Unknown Side Message

*If a foreign side acquires information concerning a Change ROE order against Unknown objects, the generated acquisition message does not properly display.*

The message was not expecting to display the "UNKNOWN" Side. This problem was solved.

#### 3.47 JTLS-2020-14751 Squadron Names MDP Mission Report Error

*The MDP uses the system sendmail command to deliver messages to specific email addresses. When attachments are used, a complex series of eight command line statements are used to compose the message which is sent to sendmail.*

*One of these commands echoes a portion of the subject line to use as a file name. For mission reports, this includes the name of the squadron. One of the squadron names contained parenthesis in the name, which caused the system command to fail, and ultimately the mission reports from this squadron were never delivered.*

*In addition, the results of this command line series are not saved anywhere, making it difficult to perform debugging when email is not being delivered. An interface to "sendmail" allowing easier debugging should be used.*

The required code already existed for the sending of files via sendmail. However code expected a completed file for delivery, and it is the composition of that file in the MDP which is complex. Therefore, the code was extended to allow the composition of an email component by component with the pieces assembled into a single file and delivered to sendmail. This allows the composition of the message to be done via file writes negating the need for the series of system terminal commands.

In addition, the assembled file is not removed, but is overwritten on subsequent uses of the class, enabling easier debugging.

#### 3.48 JTLS-2020-14752 Counter Battery Response Weapon List Length

*When a user creates a Counter Battery Response, the list of available Targetable Weapons is not limited to weapons that can only be fired by artillery. This change was made to the Fire Artillery Order, but the same change was never applied to the prearranged Counter Battery Responses.*

The Counter Battery Response utility now limits the list of Targetable Weapons using the same methodology as the Fire Artillery order.

#### 3.49 JTLS-2020-14753 Checkorder Case Mix Error

*The checkorder script did not catch a field in an order XML file where "Surface\_Subsurface" was entered instead of "Surface\_SubSurface".*

The checkorder script was modified to catch this case-sensitive issue.

#### 3.50 JTLS-2020-14754 Routine Check ADA Entity Exits Early

*The SVP CHECK.ADA.ENTITY routine does not perform all error checks.*

While reviewing the code in CHECK.ADA.ENTITY, a mistake in the code was discovered which caused the program to exit prematurely when it encountered a particular error instead of performing the remaining checks for ADA target prototypes.

The coding error was corrected and small corrections were made to make the routine more efficient.

#### 3.51 JTLS-2020-14755 ATO Parser Allows Zero Altitudes

*An OAS mission in a USMTF ATO was converted by the Parser and Translator to a CEP order having a ten foot altitude.*

The Altitude field of the AMSNLOC record in the USMTF ATO format allows the value zero. The altitude was supplied as exactly "0" so the Parser converted this value specifically to zero. The Translator identified this as a valid value but also identified it to be too small for the mission order. The Translator therefore increased the altitude to the lowest allowed value for the order, 10.0 feet.

The Parser was changed to default the altitude for supplied values which are below the minimum of 10.0 feet. In this way the Translator will use either the altitude specified for the ACO airspace, if identified or the default for the aircraft type.

### 3.52 JTLS-2020-14756 Set Aircraft Load Assignment Requirements

*While testing the SET.LOAD.ASSIGNMENT order for aircraft with a control mechanism of LOITERING\_MUNITION, it was discovered that Aircraft loads could not be removed for Aircraft of this type. However, LOITERING\_MUNITIONS should be able to remove Aircraft Loads.*

Aircraft Classes labeled as LOITERING\_MUNITION can now remove load assignments.

There is a new requirement when it comes to removing load assignments. If the load the user is trying to delete is the last load in the load assignment array, it cannot be deleted. This requirement goes for AC Types of MANNED, UNMANNED, and LOITERING\_MUNITION. The Controller will receive a "CANNOT COMPLY" message explaining that the load cannot be deleted/removed.

### 3.53 JTLS-2020-14757 Link 16 Block Filters And Reset Network

*Two issues were encountered in the Link 16 message service.*

*A Link 16 object had a JU number, a Source JU number, and a Track JU number. This occurred because the object was initially encountered as a track, then was given a self-reporting JU number. At this point it is considered a PPLI, and Track JU number filters should not be considered against it.*

*A Link 16 object was registered with a Network and was reporting on that Network. Then the CEP removed that object from the Network. The Link 16 message service was failing to recognize this removal and continued to hold the object on the Network.*

If a Link 16 object has a self-reporting JU number, which makes it a PPLI, Track JU filters are no longer considered when calculating if the object should be filtered on or off.

The Link 16 message service now recognizes when the CEP removes an object from a Link 16 Network. The service handles the removal and the object no longer reports on that Network.

### 3.54 JTLS-2020-14758 WHIP Polygon Areas Duplicate Points

*Creating a polygon area on the map using the WHIP allowed duplicate points to be entered. This often occurred when the user double-clicked to complete the drawn polygon and the consecutive clicks were too slow, causing a duplicate point to be produced at the final location. These duplicate points were then part of the order submitted to the model.*

Any duplicate point added to a polygon area is now removed as the points are set by the user, preventing duplicates from being submitted with the order.

### 3.55 JTLS-2020-14759 ATOT Non-ATO Mission Refuel Chit Orders

*Several missions contained in a USMTF ATO included references to air refueling by tanker missions that were not in the ATO or belonged to a foreign side. The ATOT did not give a warning report to the user for the missing-mission condition. The ATOT also assumed all refuel missions in the ATO belonged to the translation side. In each case the ATOT simply generated the refuel chit orders. This caused an error while checking group orders prior to sending to the CEP.*

The ATOT did not check the existence of named tanker mission at the time the ATO is translated. If the tanker mission name was not blank, the ATOT simply created the chit using the names provided.

The ATOT has been changed to verify that the tanker mission exists in the current ATO and that the tanker mission belongs to the side translating the ATO. An appropriate warning is reported to the user for the missing-mission case.

### 3.56 JTLS-2020-14760 Re-selection Order Graphics Display

*The connected graphics display for an order with connected unit and location fields was not redrawing the graphics correctly when a connected field was given a new entry. After selecting a different value such as a unit name or location on the map, the prior location corresponding to the previous selection was not being properly cleared, and continued to incorrectly persist on map.*

This was fixed by clearing the associated graphic location for the updated field in the order panel.





## 4.0 REMAINING ERRORS

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

As JTLS-GO 5.1.0.0 represents a major release of new functionality, remaining outstanding errors from the JTLS-GO 4.1 series and earlier have been considered to be obsolete and no longer relevant to JTLS-GO and have been removed from consideration for correction at this time. In future Maintenance releases, outstanding errors related to JTLS-GO will be listed in this chapter, with information provided 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 WSM - Many Messages Cause Lockup

If a service produces a large number of log or error messages in a short period of time, it can cause the WSM to lockup.





## 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
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
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
DISA	Defense Information Systems Agency
DIV	Division
DMA	Defense Mapping Agency
DoD	Department of Defense
DOS	Days of Supply
DPICM	Dual Purpose Improved Conventional Munitions
DS	Direct Support
DSA	Directed Search Area

---

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

---

IIP	Intelligence Information Prototype
IMT	Information Management Tool
INFO	Information
INTEL	Intelligence
JCATS	Joint Conflict And Tactical Simulation
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	Japan Self-Defense Forces
JTLS	Joint Theater Level Simulation
JTLS-GO	Joint Theater Level Simulation - Global Operations
JTOI	JTLS Transaction Operational Interface
JXSR	JTLS XML Serial Repository
KIA	Killed In Action
KM	Kilometer
KNOTS	Nautical miles per hour
LA	Lethal Area
LAN	Local Area Network
LAT	Latitude
LB	Login Build (JTLS order type)
LDAP	Lightweight Directory Access Protocol
LDT	Lanchester Coefficient Development Tool
LOG	Logistics
LOGIN	Logistics Input
LOGREP	Logistics Report
LONG	Longitude
LOTS	Logistics Over The Shore

---

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

---

OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
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-GO 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
XML	Extensible Markup Language
XMS	XML Message Service

## **APPENDIX B Version 5.1.7.0 DATABASE CHANGES**

No database structure changes were made for JTLS-GO 5.1.7.0.



## APPENDIX C Version 5.1.7.0 REPOSITORY CHANGES

The following changes were made to the JTLS-GO 5.1 repository.

### C.1 Added new UOM:

- CUBIC-METER

### C.2 Updated TERRAIN - BARRIER Table:

- Updated Barrier Type - Wadi - Max Width from 0 to .1
- Updated Barrier Type - TANK\_DITCH - Max Width from 0 to .0033

### C.3 Updated Sensor Table - Power Column:

- AQS10.ASONR\_ASU to 119
- AQS13.ASONR\_ASU to 113
- AQS14.ASONR\_ASU to 119
- AQS18.ASONR\_ASU to 217
- AQS22.ASONR\_ASU to 119
- 2069.ASONAR\_ASU to 119
- 8260.ASONAR\_ASU to 119
- APM60.MAD\_ASU to 115
- APM73.MAD\_ASU to 117
- DHAX1.MAD\_ASU to 115
- DHAX3.MAD\_ASU to 117
- EH101.MAD\_ASU to 119
- ASQ13.MAD\_ASU to 119
- ASQ504.MAD\_ASU to 111
- ASQ508.MAD\_ASU to 115
- ASQ81.MAD\_ASU to 119

- C212.MAD\_ASU to 115
- CAE.MAD\_ASU to 115
- MK44.MAD\_ASU to 119
- MMS106.MAD\_ASU to 115
- OKA2.ASONAR\_ASU to 117
- HISOS.ASONR\_ASU to 117
- HS12.ASONAR\_ASU to 119
- SONR.DOM.MA\_SSS to 440
- SONR.DOM.MA\_USS to 440
- SONR.VDS.HA\_ASS to 240
- SONR.VDS.HA\_SSS to 400
- SONR.VDS.HA\_USS to 400
- SONR.VDS.LA\_ASS to 400
- SONR.VDS.LA\_SSS to 560
- SONR.VDS.LA\_USS to 560
- SONR.VDS.MA\_ASS to 320
- SONR.VDS.MA\_SSS to 480
- SONR.VDS.MA\_USS to 480
- VGS3.ASONAR\_ASU to 115
- SBOYFLD.LGA\_ASS to 400
- SBOYFLD.SMA\_ASS to 330
- SH5.MAD\_ASU to 115
- SONAR.MAD\_ASU to 120
- SONAR.MINE\_SSS to 240

- SONR.DOM.HA\_SSS to 360
- SONR.DOM.HA\_USS to 360
- SONR.DOM.LA\_SSS to 520
- SONR.DOM.LA\_USS to 520

#### C.4 Updated Air to Air Lethality Table:

- Updated 5GEN.FIGHTERS AIR\_TO AIR\_LEATHALITY Table for F22.RAPTOR
- Updated 5GEN.FIGHTERS AIR\_TO AIR\_LEATHALITY Table for F35A.JSF.USAF
- Updated 5GEN.FIGHTERS AIR\_TO AIR\_LEATHALITY Table for F35B.JSF.USMC
- Updated 5GEN.FIGHTERS AIR\_TO AIR\_LEATHALITY Table for F35C.JSF.USN
- Updated 5GEN.FIGHTERS AIR\_TO AIR\_LEATHALITY Table for J20.M.DRAGON
- Updated 5GEN.FIGHTERS AIR\_TO AIR\_LEATHALITY Table for SU57(T50.PAKFA)

#### C.5 Added new Combat Arms Type:

- New COMBAT ARMS TYPE - NAVAL.GUN.HVY
- New COMBAT ARMS TYPE - NAVAL.GUN.LTMED
- New COMBAT ARMS TYPE - NVL.TANK.TURRET

#### C.6 Updated Combat System Table - CS\_COMBAT\_ARMS\_EQUIVALENT Column:

- NVL.GUN-20-27MM\_GO - Changed to NAVAL.GUN.LTMED
- NVL.GUN-20-27MM\_GT - Changed to NAVAL.GUN.LTMED
- NVL.GUN-3.9IN-100-FGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-3.9IN-100-SGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-3.9IN-100-VSG - Changed to NAVAL.GUN.HVY
- NVL.GUN-30-35MM-GO - Changed to NAVAL.GUN.LTMED
- NVL.GUN-30-35MM-GT - Changed to NAVAL.GUN.LTMED
- NVL.GUN-37-40MM-FGO - Changed to NAVAL.GUN.LTMED

- NVL.GUN-37-40MM-FGT - Changed to NAVAL.GUN.LTMED
- NVL.GUN-37-40MM-SGO - Changed to NAVAL.GUN.LTMED
- NVL.GUN-37-40MM-SGT - Changed to NAVAL.GUN.LTMED
- NVL.GUN-3IN-76-FGO - Changed to NAVAL.GUN.HVY
- NVL.GUN-3IN-76-FGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-3IN-76-SGO - Changed to NAVAL.GUN.HVY
- NVL.GUN-3IN-76-SGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-3IN-76-VSG - Changed to NAVAL.GUN.HVY
- NVL.GUN-4.5IN-120-FGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-4.5IN-120-SGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-57MM-FGO - Changed to NAVAL.GUN.LTMED
- NVL.GUN-57MM-FGT - Changed to NAVAL.GUN.LTMED
- NVL.GUN-5IN-130-FGT - Changed to NAVAL.GUN.HVY
- NVL.GUN-5IN-130-SGT - Changed to NAVAL.GUN.HVY
- NVL.MRL-107MM - No Change
- NVL.MRL-BM11 - No Change
- NVL.MRL-BM14 - No Change
- NVL.MRL-BM21 - No Change
- 100MM-TK-TRT - Changed to NVL.TANK.TURRET
- 115MM-TK-TRT - Changed to NVL.TANK.TURRET
- 76MM-TNK-TRT - Changed to NVL.TANK.TURRET
- 85MM-TK-TRT - Changed to NVL.TANK.TURRET

#### C.7 Added to Target Type Groups:

- ART\_NON\_ARM\_TTG - NAVAL.GUN.LTMED

- ARTY\_ARMOR\_TTG - NAVAL.GUN.HVY
- ARTY\_ARMOR\_TTG - NVL.TANK.TURRET

#### C.8 Added New Units and TUPs:

- Added Chinese Units and TUPs
- Added Russian Units and TUPs

#### C.9 Updated CSPs:

- Updated Combat System Prototype for all 10 CSPs, adding the 3 New Combat Arms Types
- Updated Combat System Prototype - FLP\_CSP\_CAT tables for all 10 CSPs, adding the 3 New Combat Arms Types.