

JTLS-GO

Version Description Document

November 2020



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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.1.10.0)

[Blank Page]

ABSTRACT

The Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) 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 5.1.10.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 5.1.10.0 is a Maintenance release of the JTLS-GO 5.1 series that includes an updated wespac51 demonstration database as well as repository data held in the repository51 database. There are no Engineering Change Proposals (ECPs) included with this release. 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1.2.1 Obsolete/Outdated Documents	1-1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 Cyber Security Information Assurance 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	1-9
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 Generation Of Combat System Summary Files	1-10
1.5.3.2 OTH-Gold Ship Types	1-11
1.5.3.3 Aircraft Type Foreign Key Definitions	1-11
2.0 ENGINEERING CHANGE PROPOSALS.....	2-1
3.0 SOFTWARE TROUBLE REPORTS.....	3-1
3.1 JTLS-2020-14863 Helicopter Landing To Pickup Unit Error	3-1
3.2 JTLS-2020-14864 Crash When Unit Starts Move At Node	3-1
3.3 JTLS-2020-14865 Duplicate Political Boundary Points	3-1
3.4 JTLS-2020-14868 Undefined Force Side For Contamination Areas	3-1
3.5 JTLS-2020-14869 Cancellation Of Delayed Air Mission Crash	3-2
3.6 JTLS-2020-14871 Aircraft End Maintenance Leaving Squadron Error ..	3-2
3.7 JTLS-2020-14873 Missing CSP CAT Record	3-2
3.8 JTLS-2020-14876 Crash Deleting Unit Destroy Target Task	3-2
3.9 JTLS-2020-14878 Zero HRU Assessment Duration Error	3-3
3.10 JTLS-2020-14882 Alert Status Improperly Conveyed To C4I Systems	3-3
3.11 JTLS-2020-14884 Variation In Computations Of Unit Radii	3-3
3.12 JTLS-2020-14886 Moving Unit Not Stopping When Task Canceled	3-4
3.13 JTLS-2020-14888 Stop Glassfish Server If Times Out	3-4

3.14	JTLS-2020-14890 UNIT WORKING SET Not Cleared After Event	3-4
3.15	JTLS-2020-14892 SET.SC.PARAMETER Bad Message Trace	3-5
3.16	JTLS-2020-14894 No Unit Authority Report When Using CONT WHIP .	3-5
3.17	JTLS-2020-14896 Ship Joining Formation Logic Error	3-5
3.18	JTLS-2020-14897 Replay JXSR Freezes When Playing Backward	3-6
3.19	JTLS-2020-14900 Missing Intelligence Report	3-6
3.20	JTLS-2020-14901 NEC CCIS JTOI CWIX Issues	3-6
3.21	JTLS-2020-14904 Shadowing Ship Did Not Alter Speed	3-7
3.22	JTLS-2020-14907 JNLP File Not Removed From Downloads Folder ...	3-7
3.23	JTLS-2020-14909 Unknown Object Type In Mission Report Error	3-8
3.24	JTLS-2020-14915 SVP Option For Warning 1442	3-8
3.25	JTLS-2020-14917 Magic Move Submarine With Future Task Crash ...	3-8
3.26	JTLS-2020-14919 Checkorder Not Checking AAR/DDS Reports	3-8
3.27	JTLS-2020-14921 SCP Correction Different Comparison Operators ...	3-9
3.28	JTLS-2020-14924 Setting Link16 Network Throws Error	3-9
3.29	JTLS-2020-14926 Building Ship Movement Plan Crash	3-9
3.30	JTLS-2020-14927 Non-POT Invalid Target Ranges Not Identified	3-9
3.31	JTLS-2020-14930 Large WSM Log/Error Log File WSM Crashes	3-10
3.32	JTLS-2020-14936 OAS Crash At End Of Single Path Orbit	3-10
3.33	JTLS-2020-14937 Change Country Code Values	3-10
3.34	JTLS-2020-14938 Setting Shelter Priority Mislabeled Order Field ...	3-10
3.35	JTLS-2020-14940 Error 317 Not Functional	3-11
3.36	JTLS-2020-14941 SET.SC.PARAMETER Message Change	3-11
3.37	JTLS-2020-14945 DDSC/SVP Unnecessary Verify Link	3-11
3.38	JTLS-2020-14954 JSAT Help Text Outdated	3-11
3.39	JTLS-2020-14957 Embarking Units Formation Unable To Split	3-11
3.40	JTLS-2020-14959 Mismatched Code For Offloading Convoy	3-12
3.41	JTLS-2020-14962 Naval Formation Magic Move With Ship Routes ..	3-12
3.42	JTLS-2020-14968 Negative Fuel Percent In IMT	3-12
3.43	JTLS-2020-14970 Error 425 Will Not Generate	3-12
3.44	JTLS-2020-14972 AAR Object Attack Report Exception	3-13
3.45	JTLS-2020-14975 Unit Relative Position Not Maintained	3-13
3.46	JTLS-2020-14977 Error 428 Will Not Generate	3-15
3.47	JTLS-2020-14982 Formation Icons Displayed at Equator	3-15
3.48	JTLS-2020-14988 JOI OTHGold Unreported Errors	3-16
3.49	JTLS-2020-14994 ATOG Not Considering Shared Foreign Runways .	3-16
3.50	JTLS-2020-15002 Jtlsmenu Improperly Launches OVT GUI	3-16
3.51	JTLS-2020-15004 Air Mission/Convoy Ordered To Move Same Unit	3-16
3.52	JTLS-2020-15009 Order Checker Did Not Catch Error	3-17
3.53	JTLS-2020-15011 Unable To Display Range Rings For Satellites	3-17
3.54	JTLS-2020-15023 Changes To Air Mission DSA Processing	3-17
3.55	JTLS-2020-15028 Changed Aircraft Type Not Updating Ranges	3-18
3.56	JTLS-2020-15034 OVP Added JODA Retrieval Value Check	3-18

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

APPENDIX A. ABBREVIATIONS AND ACRONYMS A-1

APPENDIX B. Version 5.1.10.0 DATABASE CHANGES B-1

APPENDIX C. Version 5.1.10.0 REPOSITORY CHANGES C-1

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 5.1.10.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO[®]) software suite. JTLS-GO 5.1.10.0 is a Maintenance release for the JTLS-GO 5.1 series.

JTLS-GO 5.1.10.0 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 prior to the release of Version 5.1.1.0. Information on this procedure can be found on [Page 1-11](#).

No Engineering Change Proposals (ECPs) were implemented for this release, but for consistency an empty [Chapter 2.0](#) stating this fact is included. 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 usually provided in [APPENDIX C](#), but updates for the JTLS-GO 5.1 engineering data repository are no longer supported.

JTLS-GO 5.1.10.0 executes on the Red Hat Enterprise Linux Version 7.6 64-bit operating system. The Web-Hosted Interface Program (WHIP[®]) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

This section lists documents and software that are relevant to JTLS-GO. All JTLS-GO documents included in this delivery are provided in 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 5.1.9.0)

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.1.7.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.1.7.0)
- *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 5.1.6.0)
- *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 5.1.9.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.1.9.0)
- *JTLS-GO Air Services User Guide* (JTLS-GO Document 24, Version 5.1.2.0)

1.2.3 Updated Documents

- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.1.10.0)

1.2.4 New Documents

No new documents are delivered with JTLS-GO 5.1.10.0.

1.2.5 Delivered Software Components

JTLS-GO 5.1.10.0 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)
- Modernized Integrated Database (MIDBTool)
- JTLS Object Distribution Authority (JODA) Build 137
- 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 making a 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.

This situation as a consequence can have more than one user concurrently control the playback of the game, as 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 has been addressed and fixed within the JTLS-GO 6.0 release by only permitting one login per TRIPP user instance.

Organizations should develop their own procedures to ensure that only one user logins to 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 5.1.10.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 5.1.10.0 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 "wespac51", which is based on the Western Pacific theater of operations and is suitable for training and demonstrations.

- 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 5.1.10.0, it is possible to access and copy records from the repository51 database into your own developed scenarios.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 5.1.10.0 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 5.1.10.0 has been tested on the following operating systems:

Red Hat Linux Enterprise Edition Version 7.6.

CentOS Linux Version 7.6.

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 272. For the JTLS-GO 5.1 series, the Java Runtime Environment (JRE) is not provided 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> OHLA 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 Cyber Security 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 Cyber Security Information Assurance (CSIA) program. JTLS-GO, as software that executes on DoD systems, must comply to the mandates of the CSIA 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 CSIA 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. The primary purpose of this release is to provide a version of JTLS-GO compiled with the latest security release of OpenJDK, OpenJDK 8 Version 272.

To meet these CSIA requirements, each user organization should ensure that this version of OpenJDK is loaded on the JTLS-GO servers and any client machines used to connect to JTLS-GO.

JTLS-GO has completed the CSIA 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), for additional information.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 5.1.10.0 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 been accomplished. 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. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The CAE GESI system is used to represent a virtual battlefield, including weapons, vehicles, aircraft, and ground forces.

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 5.1.10.0 delivery. Users may obtain a full installation package of the RTI executive program from Pitch Corporation (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. 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. 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

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 5.1.10.0, but end user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs.

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”. In future Major releases of JTLS-GO, [APPENDIX C](#) will provide a summary of the data structure changes made to the data repository. No data structure changes have been made in this Maintenance release; therefore, [APPENDIX C](#) is empty.

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 to 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 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 a problem 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.

1.5.3.2 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 5.1.1.0, do not need to execute the 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.

1.5.3.3 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 a 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

No model capabilities were added to JTLS-GO 5.1.10.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

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.

3.1 JTLS-2020-14863 Helicopter Landing To Pickup Unit Error

If a helicopter attempts to land in the middle of nowhere to pick up a unit, the model generates a logic error.

The logic error was self-correcting but needed to be fixed. The issue was that under this circumstance, the type of object at which the unit should try to land was not being properly set. It is now properly set, and the logic error no longer occurs.

3.2 JTLS-2020-14864 Crash When Unit Starts Move At Node

The model crashed when a unit was at the exact starting node location and started a move order to use a road network.

The model checked for this situation, but it used a check for perfect equality. The logic was changed to use the routine called SAME LOCATION which does not check for perfect equality but checks whether the unit is within one meter of the starting node.

3.3 JTLS-2020-14865 Duplicate Political Boundary Points

The model crashed attempting to generate an IIR Report.

The model was trying to determine the political country (POC) for an entity while generating a IIR message. The error was caused by two points in the POC polygon had the exact same points.

New SVP Error 170 was added to identify POC polygons that share identical boundary points.

3.4 JTLS-2020-14868 Undefined Force Side For Contamination Areas

The model generated a logic error when attempting to find the Force Side for a contamination area.

Contamination areas have no Force Side - they represent chemical or nuclear contamination at a specific location. The routine to find the Force Side was modified to return a value which represented any Force Side. The Force Side value was used in the model to specify the effects present in a grid location. The value for any-side was properly assigned during run time and when the model was restarted from a checkpoint.

3.5 JTLS-2020-14869 Cancellation Of Delayed Air Mission Crash

An ordered air mission was delayed during the allocation of resources. A notification message was generated when an external program was used to control the air tasking. This message was meant to tell the external program that the mission was delayed with a divert code. This scheduled message was not canceled if the mission was canceled. This produced a logic error because the message could not be generated for an unknown air mission.

This logic error occurred when the mission was automatically canceled at the end of the maximum delay interval, or when the player ordered a cancellation of the air mission. The event used to notify the external program was never canceled when the mission ended. The CEP was modified to correct this oversight.

3.6 JTLS-2020-14871 Aircraft End Maintenance Leaving Squadron Error

Errors are generated whenever a squadron must leave the game and has aircraft in maintenance.

It was originally thought that it was impossible to have a squadron leaving the game and have aircraft in maintenance. This can happen if the squadron has aircraft in maintenance that can not self lift from a sinking ship.

The logic error was removed.

3.7 JTLS-2020-14873 Missing CSP CAT Record

The database that was being tested was missing a database record for a specific Combat System Prototype (CSP) and Combat Arms Type (CAT). This could happen if the database build simply copied a CSP and did not perform a Deep Copy. The result of this was that the model held a value of 0.0 for CSP CAT MEAN TIME BETWEEN REPAIR. This is an illegal value. The data parameter must hold a value greater than zero.

Prior to reading in the CSP CAT data, the parameter CSP CAT MEAN TIME BETWEEN REPAIR is set to the database default of 0.25. The model then reads in the data from the database which will reset the parameter to the database specified value. If a CSP CAT record is missing, it cannot be read in, and the database parameter retains a value of 0.25, which will not cause a model crash.

3.8 JTLS-2020-14876 Crash Deleting Unit Destroy Target Task

A unit was given an order to destroy a target. The unit had to move to the target location. While the unit was moving toward the target, the user decided to make the unit a part of a Tactical Ground Formation move. The model canceled the move to target task and since the move was created in response to the target destroy order, the task to destroy the target was also moved from the task list. While writing a message for the pPayer explaining that the Target Destroy Task was canceled, the model crashed.

The model crashed when the model was attempting to inform the user of the current percent capability of the target. The target was not accessed properly. The fix was relatively simple. The code simply needed to properly access the target.

3.9 JTLS-2020-14878 Zero HRU Assessment Duration Error

A ship was under attack by a missile. The sinking ship generated lifeboat HRUs. After the HRUs were placed on the game board, they came under attack by another ship. During the event HRU.ASSESS.COMBAT, the duration for the HRU to fight ended up being zero and a logic error was generated. This occurred because the HRU did not have an HRU.STAND.AND.FIGHT.TIME defined.

When an HRU is activated, the model calls the routine COMPLETE.HRU.ACTIVATION. In this routine, the HRU to be activated will now be given a default value for its HRU.STAND.AND.FIGHT.TIME.

3.10 JTLS-2020-14882 Alert Status Improperly Conveyed To C4I Systems

When an air mission was on-alert, this alert status was supposed to be set on the JODA and passed on to real world systems. A valid take-off time was specified even though the mission was still on the ground. Testing revealed that the alert status was not properly set in the model so it was not handled correctly by the JTLS-GO Transaction Operational Interface (JTOI) linked to a real-world C4I system. The take-off time was filled incorrectly by the model.

In the model, a value for the alert flag was only accessed when air missions were created on the JODA. When created, all missions start with a posture of scheduled or in pre-launch. When a mission transitioned to an alert posture, the alert status was never changed to flag it as being on-alert. The code was corrected to designate the alert status on missions during initial order processing. The take-off time for missions was modified such that it is only filled when the mission was no longer on-alert and it began to fly.

3.11 JTLS-2020-14884 Variation In Computations Of Unit Radii

A user ordered an air mission to go on alert at its home base. The squadron for this mission was located far from the center of the airbase. When the mission attempted to go on alert, it checked to see if it was at the proper location before executing the alert task. This check failed, but the mission took off. It flew a short distance, landed closer to the center of the airbase, and then stood alert. To execute the alert task, the mission should not have required a move task.

The squadron was located within the unit radius of the airbase when using the radius defined by the TUP. However, when the unit radius was computed during runtime, it resulted in a smaller value because the airbase was at less than full strength. The mission needed a move task to place it within the shrunken radius of the airbase.

However, for consistency with owned units and targets, the model should not consider the unit strength when computing the radius of airbases and FARPs. If a squadron or a target is owned by one of these units, those objects should not fall outside the radius when the owning unit is damaged. The code was changed such that the calculated radius of an airbase or a FARP is always the radius defined by the TUP. For ground units, support units, and squadrons, the computation of unit radius continues to include a reduction in size due to changes in unit strength.

3.12 JTLS-2020-14886 Moving Unit Not Stopping When Task Canceled

A unit was given a Direct Support order. Because of the unit's location and the location specified in the order, the unit was required to move. While the unit was moving, the Direct Support task was canceled and removed from the unit's task list, and the unit received another Direct Support order to support a different unit.

However, before the replacement Direct Support order could be sent, the unit's MOVE ALONG ROUTE GROUND event was executed. A series of routines were called from this event that eventually led to the routine ADJUST AUTO MOVE ROUTE. A logic error was recorded in this routine because the unit's task list only held the currently executing Move task and the replacement Direct Support task had not yet been added to the list.

When a unit is given a Direct Support order and the unit must move, the unit will be given two tasks. The unit will first execute a Move task followed immediately by a Direct Support task. The Move task will have its PET AUTO MOVE FLAG set to Yes. The CEP correctly does this. If a MANAGE LAND UNIT TASKS order is sent to cancel the Direct Support task, it should:

1. Remove the Direct Support task, and;
2. If the current executing task is Move with its PET AUTO MOVE FLAG set to Yes, that task should also be canceled.

The CEP executed Step 1 but did not execute Step 2.

In order to correct this problem, the logic that cancels unit tasks was modified.

3.13 JTLS-2020-14888 Stop Glassfish Server If Times Out

The Java 1.8.0_262/265 release caused delays in stopping the Glassfish server and multiple residual Glassfish processes that could prevent Glassfish from starting.

A hard stop was added to the Glassfish server if the process of stopping Glassfish takes too long.

3.14 JTLS-2020-14890 UNIT WORKING SET Not Cleared After Event

A logic error was reported at the end of the ASSESS WEAPON DAMAGE event due to a non-empty UNIT WORKING SET.

In some cases where an automatic firing engagement is involved, ASSESS WEAPON DAMAGE calls routine EXTEND MISSILE EXCHANGE to find another naval unit to extend the engagement. This routine calls CHECK AUTOMATIC FIRING which fills the UNIT WORKING SET to find that naval unit. Normally the routine clears the working set when the search is done. Under some circumstances, however, the search is allowed to end early without clearing the set. This is what caused the logic error.

The UNIT WORKING SET is now cleared at the end of CHECK AUTOMATIC FIRING under all circumstances.

3.15 JTLS-2020-14892 SET.SC.PARAMETER Bad Message Trace

A user sent a SET.SC.PARAMETER order in which the "New Consumption Type" field was selected, and the show "ALL_DATA" option was also selected. The user received a bad message trace.

When selecting a new consumption type, there was a mismatch between the opening and closing tags. This mismatch was corrected.

3.16 JTLS-2020-14894 No Unit Authority Report When Using CONT WHIP

A user requested a Unit Authority Report for a US_WHIP from the primary CONTROLLER WHIP. The user received a message stating "THE US_WHIP WHIP is a Controller WHIP and does not have authority over any units." The US_WHIP is not a Controller WHIP, and the CONT_WHIP should have displayed the same report that would have been displayed by the US Player WHIP.

The Controller WHIP can now receive Unit Authority reports for WHIPs of other Force Sides. If any Controller WHIP attempts to request a Unit Authority report for a WHIP with the Force Side of CONTROLLER, that WHIP will receive a message stating that the "Controller WHIP does not have authority over any units."

3.17 JTLS-2020-14896 Ship Joining Formation Logic Error

A formation was created with four ships. After three of the ships arrived on station, the formation was ordered to move. The fourth ship lagged behind while attempting to catch up with the faster moving formation. During one of the slower ship's moves, a logic error was generated related to the time, speed & distance calculations required to adjust the ship's course to reach the moving formation's new location.

Each time the lagging ship moved as it pursued the moving formation, the ship's next move location was calculated based on the bearing and the distance (at current speed) it expected to travel before the next scheduled move time. This methodology incorporated the radius of the Earth to accurately derive the next location.

Each time the formation moved, the lagging ship's route was recalculated toward the formation's new location. In this methodology, a great circle distance was calculated using an approximation of one nautical mile equals one degree. Ordinarily, the traveled distance is less than the great

circle distance because the ship has not yet completed its planned move. However, in this case, the traveled distance (based on speed and elapsed time), was slightly greater than the great circle distance which generated the logic error. The model self-corrected by using the great circle distance.

To correct this inconsistency in the two methodologies, the great circle distance calculation was changed to use the radius of the Earth instead of the nautical mile approximation.

3.18 JTLS-2020-14897 Replay JXSR Freezes When Playing Backward

Whenever the Replay JXSR is playing backwards to the start time of the run, it becomes stuck processing packets and becomes unresponsive and fails to pause the TRIPP.

The Replay JXSR could compute a Next Time Step in negative or zero days for the time it is to move toward. This could happen when the TRIPP is playing the download files backward and generally at a rate greater than 1:1. With a negative or zero time, the Replay JXSR would get stuck reading the first packet, a download packet, which would result in another negative or zero time step. This would cause the JXSR to again read the download packet and continue to loop.

The code for playing backward has been changed to prevent the Replay JXSR from reaching zero or negative scenario time.

3.19 JTLS-2020-14900 Missing Intelligence Report

The following message was received:

An error has been discovered in the routine CREATE.INTELLIGENCE REPORT.

The Specific Type was 92.

Please report the problem to the appropriate personnel.

The error is generated when a WHIP player receives an unsolicited intelligence report from another non-friendly side, and the routine has not been programmed to report the specific message type (in this case, Assigning Special Air Unit Characteristics). The routine was modified and tested.

During testing, another problem was identified in the order panel, which would not allow a Player to set only the capability for Strategic Air Missions to land during periods of conflict. This was also corrected.

3.20 JTLS-2020-14901 NEC CCIS JTOI CWIX Issues

A list of issues was reported for the JTOI interface to the Northern European Command, Command and Control Information System (NEC CCIS). The following issues only applied internally to the JTOI:

1. *The mission status for Alert missions that are canceled while still on the ground was incorrectly set to Terminated and not Canceled.*
2. *When a squadron was given a new home base, any active air missions from that squadron had their Departure Base set to that new home base when it should have been preserved.*

The following fixes were implemented for the JTOI-NEC CCIS issues:

1. The JTOI now checks to see if the air mission was canceled while airborne or on the ground. The appropriate mission status is then set.
2. The JTOI now only set the Departure Base for an existing air mission if that air mission does not yet have a Departure Base set.

3.21 JTLS-2020-14904 Shadowing Ship Did Not Alter Speed

A naval vessel was ordered to shadow another naval vessel. The shadowing vessel did not adjust its speed when necessary to maintain the specified shadow distance while following the other vessel.

The model automatically calculated the correct speed necessary to keep pace with the shadowed vessel. However, a sequencing error in the code bypassed the logic that changed the ship's current speed and task speed. Consequently, the ship did not adopt the new speed. The problem was fixed by re-sequencing the code to not skip the current speed and task speed updates. Also, new logic was added to update the ship's ordered speed so that the ship does not revert to its previous speed after its first move along the intercept route.

In addition, the warning message generated when the shadowing vessel cannot keep pace incorrectly recommended that the Manage Naval Tasks order be used to change the speed of the shadowing vessel. By design, the speed of a shadow task cannot be manually altered by the player. This action is not permitted by the model. The message text was changed to recommend canceling the shadow task and assigning a faster vessel to a new shadow task.

3.22 JTLS-2020-14907 JNLP File Not Removed From Downloads Folder

The Java Network Launch Protocol (JNLP) file downloaded from the browser to start the WHIP or DDSC was not being removed from the user's Downloads folder after the WHIP or DDSC application was exited.

Previously, the file was being removed from only the user's system temporary folder when it was downloaded there by the browser. The user's Downloads folder and one level underneath the user's system temporary folder were locations added to delete the respective JNLP file from upon exiting the application.

3.23 JTLS-2020-14909 Unknown Object Type In Mission Report Error

An "Unknown Object" logic error was encountered while generating a Mission report for an OAS Mission. The routine "SUMMARIZE.REPORTED.DAMAGE" which adds the summary pilot report of observed damage to a Mission Report, did not have a case established for damage to a ship's hull.

A new case and sub-message has been added to account for an object type of ship hull.

3.24 JTLS-2020-14915 SVP Option For Warning 1442

The routine EXECUTE.CONVOY.ROUTE was executing for very long periods of time.

In the case described above, the naval unit was trying to find a route to its support unit. In some cases the best solution is to remove the support unit for the unit. This could be because it is a naval unit which is out to sea or another type of unit that is secluded in a hard to get to area.

A new SVPR correction to Warning 1442 was added to allow the DDS user to use an automatic delete option to do so.

3.25 JTLS-2020-14917 Magic Move Submarine With Future Task Crash

A user magic moved a submarine indicating that future tasks should not be deleted, and the submarine had one future task to submerge. Under this condition, the Magic Move order crashed the game.

The problem was caused when the model was attempting to restart the execution of the non-deleted tasks. The model properly determined that the submarine's task list was not empty and tried to schedule an Execute Task Event for the first task in the task list. The problem was that the Submerge task is an instantaneous task and should not have been restarted in this manner. The logic issue was corrected and the submerge task occurs as it should at the proper time.

3.26 JTLS-2020-14919 Checkorder Not Checking AAR/DDS Reports

The checkorder script is the front-end for the Order Verification Program (OVP) and is responsible for all the low-end checking for the arguments sent to the OVP, such as ensuring the order file exists, the arguments are legitimate, and so on.

When requesting a verification of an AAR or DDS report, the checkorder expects an argument of "-r aar" or "-r dds" accordingly, which should then be passed to the OVP as "aar \$JGAME/data/aar/orders/" or "dds \$JGAME/data/dds/orders". However, what was being passed to the OVP was "r aar" or "r dds" which was preventing the OVP from finding the correct report file to verify.

The script was modified to pass the correct arguments to the OVP.

3.27 JTLS-2020-14921 SCP Correction Different Comparison Operators

The corrective actions defined for different SVP Errors and Warnings are defined in an XML file. Within this XML file, a single corrective action may have different executions depending on the type of object associated with the corrective action.

For example, an SVP error that informs the user a target category has some ELINT attributes set, but not all ELINT attributes are set, would have a corrective action to open the appropriate target category table to add or remove attribute values. The exact target category table would depend on if the error was referring to a Sensor Site or a Jammer Type. To accomplish this, two match statements exist for the corrective action: the first asking if the target category is a Sensor Site, and the second asking if it is a Jammer Type.

To reduce simple mistakes, a checker exists to validate the contents of the XML file. One of the items that this checker does is to ensure that the matches for a corrective item are unique. This check was not taking into account the comparison operator, so a match of is Target Category a Jammer Type was the same as Target Category is not a Jammer Type. This is limiting the ability for some corrective actions.

The checker was expanded to consider the comparison operator when determining if two match statements were equivalent or not.

3.28 JTLS-2020-14924 Setting Link16 Network Throws Error

When setting the Link16 Network Structure for a running Link16 Message Service (L16MS), the L16MS reconnects to the JODA and redownloads objects with new parameters. This re-download was being performed immediately upon receiving the request to set the Network Structure. This combination of events caused multiple responses to be passed back to the Link16 WHIP Module, which the module failed to process and threw an error.

Setting the Link16 Network Structure is now deferred and later processed on the main loop of the L16MS. This methodology handles requests and responses for the Link16 WHIP module correctly.

3.29 JTLS-2020-14926 Building Ship Movement Plan Crash

The model crashed when a Player submitted an order to create a ship movement plan. This plan was intended to perform a strategic lift.

The capability to create ship movement plans and perform strategic lifts has been temporarily disabled in the model. Changes were made to the unsupported code to prevent the crash. A message was added to tell the player that no action was taken for this order.

3.30 JTLS-2020-14927 Non-POT Invalid Target Ranges Not Identified

Invalid ranges for non-prototype owned targets are not identified.

The SIP code was correctly catching both POT and non-POT targets that had invalid ranges. The actual problem was in the SVP-Template file for Warning 1605. It did not have the necessary entries to output the error messages to the DDS user. The missing entries were added and tested in the new SVP-Template.

3.31 JTLS-2020-14930 Large WSM Log/Error Log File WSM Crashes

The Web Service Manager (WSM) crashed with numerous different Java exceptions.

The JTLS web services logging and/or error log files became too large over time. To resolve this, the WSM was updated to optimize how log and error files are processed and displayed. Log file sizes are now unlimited and the allowable memory allocated for log file display on the WSM was expanded.

3.32 JTLS-2020-14936 OAS Crash At End Of Single Path Orbit

The model crashed when an OAS mission flying as an Armed Recce on a single pass orbit fired a weapon at the very end of its orbit path.

After the mission fired the weapon, it went into a weapon hold and then attempted to fly to the next point in the single-pass path orbit. Since the mission was at the end of the single-pass path, there was no next point. The logic was improved to have the mission go through its last point again after firing a weapon.

3.33 JTLS-2020-14937 Change Country Code Values

A user attempted to change a country code to a three character value, but the order to set Political Country attributes only allowed a maximum of two characters. For this scenario, the Country Code Standard was already using values with three characters.

The order panel was modified to allow Country Codes with up to three characters. The supporting code was modified to verify that the Country Code specified on the order matched one of the known values in the given Country Code Standard. An error message is now generated if the Country Code is not valid.

3.34 JTLS-2020-14938 Setting Shelter Priority Mislabeled Order Field

The order to Set Shelter Parameters allowed the user to change the priorities for objects placed in shelters. One such order field was incorrectly labeled as Combat Systems. The resulting message reported on the priorities for Combat Arms Types.

Shelter priority was held as an attribute on Combat Arms Types, and not on Combat Systems. The order utility and the supporting code were changed to correctly use Combat Arms when setting the shelter priorities. The utility panel was renamed to support this correction.

3.35 JTLS-2020-14940 Error 317 Not Functional

The SVP was unable to generate Error 317.

The first part of Error 317 checks each OAS TTG to ensure the load has weapons. The code was improperly written and the check was never performed. This problem was corrected.

Additionally, the SVP message was expanded because the original did not provide the user enough information.

3.36 JTLS-2020-14941 SET.SC.PARAMETER Message Change

If the user attempts to set the SC.SPECIAL.CAPABILITY of the supply category to the same SC.SPECIAL.CAPABILITY it has already, the user will receive a cannot comply message and the order will be ignored.

Added a new "Cannot Comply" message in response to this order.

3.37 JTLS-2020-14945 DDSC/SVP Unnecessary Verify Link

The SVPR was updated not to depend on a specific link to access the ~ data/scenario/<scenario-name>/verify directory. However, the link is still created.

The code that created the link to the directory was removed.

3.38 JTLS-2020-14954 JSAT Help Text Outdated

Some errors in the JSAT help manual were identified. The content, which addressed the steps to process database output, was not current.

The JSAT help pages were fully updated to match the current capabilities.

3.39 JTLS-2020-14957 Embarking Units Formation Unable To Split

A formation with several embarked ground units was ordered to split into two separate formations. Some of the embarked units were held entirely on the ships specified to form the new formation. The model did not allow ships with the embarked units to split. Only those ships without embarked units were allowed to split to form the new formation.

The model routinely checks to make sure that embarked units are not split between the new formation and the parent formation. If so, the Split Formation order is correctly rejected to prevent portions of a ground unit being spread across two separate formations.

In this case, however, the embarked units were entirely contained on the ships specified to form the new formation and should have been permitted to do so. A minor code error prevented those ships from joining the new formation. The logic incorrectly referenced the embarked unit instead

of the ship which caused the ship to be excluded from the split operation. The error was corrected.

3.40 JTLS-2020-14959 Mismatched Code For Offloading Convoy

A code error was found associated with the offloading of supplies from a convoy. One routine in the model called another routine with the wrong calling arguments. The error occurred in code which was designed to offload supplies from a convoy using Material Handling Equipment (MHE).

The mismatched code was corrected to resolve this problem.

3.41 JTLS-2020-14962 Naval Formation Magic Move With Ship Routes

A formation that was moving fast had ships that were either falling behind or running into ground and getting stuck. A user tried to clean up the situation by Magic Moving the formation. All the units were magic moved to their proper formation location and showed a posture of formation. All their tasks had been canceled as normal. However, the ships continued to show routes from when they were trying to catch up to their formation location.

If a ship in a formation has a ground wait task when the formation is magic moved, the model calls CANCEL.OBJECT.MOVE. This routine cancels any move events scheduled for the unit, clears the object's route if it has one, and registers the route change. When the formation is magic moved, the ships in the formation no longer have unusual routes.

3.42 JTLS-2020-14968 Negative Fuel Percent In IMT

A very large amount of fuel (1,000,000 gallons) was added to an active air mission using the Controller Magic Air Operations order. The "Fuel" field in the IMT and SitRep correctly showed the increased amount. However, the "% Fuel" field displayed a negative percentage value.

In this particular example, 1 million gallons was added to a mission with 995 gals fuel at 48% remaining. The AC FUEL for the aircraft type was 2070 gallons. After adding the fuel, the Fuel field correctly showed 1000995. However, the "% Fuel" field incorrectly showed -17179%. The correct percentage should be 48357%.

The JODA Data System Protocol used a data type to hold the "% Fuel" value that was too small to accommodate the large percentage value. Consequently, the value translated to a negative percentage. To correct the problem, the data type size was increased, which was consistent with the "Fuel" field data type.

3.43 JTLS-2020-14970 Error 425 Will Not Generate

Error 425 is not generated when the error conditions are met.

When a combat systems effectiveness is affected by a units lack of fuel to support it, Error 425 is supposed to be generated. An error in the code was discovered and corrected.

3.44 JTLS-2020-14972 AAR Object Attack Report Exception

When attempting to view an Object Attack Report for a ship that has suffered hull damage, a WHIP/TRIPP exception occurred, informing the user of an attempt to compare a double to a text.

One of the values that appears in the result table for hull damage is the number of final hull breaches, which is the summation of the initial number of breaches plus the number of new breaches. Since its a summation, it is a number and this number is passed to the utility stylesheet to format HTML output. However, that stylesheet expects the input values to be strings causing the exception. The input value is now cast explicitly to a string before passing it to the HTML utility stylesheet.

3.45 JTLS-2020-14975 Unit Relative Position Not Maintained

Units assigned to a Group Move have a designated leader, while the remaining units are designated as followers. The positions of the followers relative to the leader at the end of the move are expected to be similar to their positions relative to the leader at the time the move started. This holds whenever the leader is given a location to move to.

However, this may break down when the leader is given a route and some of the followers are slower than the leader. In this case, the final position of a slower follower relative to the leader may not come close to resembling its relative position from the leader when the move started.

When the CEP receives the Move Order, the leader is given a Move task for every waypoint specified in the route. For example, if the leader is given five waypoints to visit, five Move tasks will be created and the leader will execute each task in succession.

The follower units are not given Move tasks until the leader starts to execute its first Move task. Each follower is given a single Move task. That move ends at a location that is offset from the leader's waypoint. That location is based on the bearing and distance that the follower is from the leader at the time the leader executes its Move task.

The followers do not receive their next Move task until the leader reaches its first waypoint (completing its first Move task) and starts to execute its second Move task (leading to the second waypoint). The follower's destination is calculated in the same manner as described in the previous paragraph. Note the following:

- The distance and bearing from the leader to the follower is now based on the leader's and follower current location at the time the leader starts to execute its second Move task.
- If the follower is slower than the leader, the follower will still be moving. The follower will cancel the Move task it is currently executing and execute the new Move task.
- If the follower is faster than the leader, it will have completed its Move task and will be waiting at its designated waypoint for the leader to reach its waypoint.

This process is repeated, waypoint to waypoint, until the leader reaches its last waypoint and completes its last Move task. When this occurs, the follower units will no longer receive individual Move tasks, and they will stop at the location designated by their last Move task.

The above process has different effects on a follower's movement depending upon its speed relative to the leader. If the follower is as fast as the leader, or faster, the relative positions between the follower's waypoints and the leader's waypoints will be similar. If, however, the follower is slower than the leader, the relative position between the follower's Waypoint 1 and the leader's Waypoint 1 should be similar to the relative positions between the two units at the start of the Group Move. They begin to deviate from Waypoint 2 onwards. The amount of deviation depends on such factors as:

- The distance covered during the Move task the follower is executing before the next Move task arrives.
- The twists and turns in the leader's route.
- The number of waypoints in the leader's route.

At the end of the Group Move, the relative positions between the slow follower and the leader may not be the same as it was at the start of the move. For example, a slow follower that is abreast of the leader at the start of the Group Move can end up many kilometers behind the leader when the move ends.

This problem was resolved through the following changes:

When the leader reaches a waypoint, it now checks that all of the followers have reached their respective waypoints. The check is performed when the leader is ready to depart the waypoint. If at least one follower is still moving towards its waypoint, the leader now waits. Note that when the leader completed the Move task that got it to the current waypoint, it (and any unit that completes a Move task) had its Mission changed from MOVING to DEFEND, its Posture changed from MOVING to HASTY DEFENSE, and its Speed reduced to zero.

If all the followers are in position, the leader can now execute the Move task that will take it to the next waypoint. A Move task is created for each follower as before, and the followers proceed to the locations specified in their tasks.

When a leader waits for slower followers to catch up, and as each of these followers reaches its respective waypoint, the leader checks that all of the followers are in position. The leader only departs and gives the followers their next Move tasks when all of the followers have reached their waypoints.

In order to implement these changes, the PET ASSOCIATED OBJECT of the follower's Move task now points to the lead unit. For the leader's Move task, however, the PET ASSOCIATED OBJECT will be null. Note that for single unit moves the PET ASSOCIATED OBJECT remains null, and for

Tactical Group Formation moves the PET ASSOCIATED OBJECT continues to point to the TACTICAL GROUP FORMATION.

One additional change provides for a follower to be removed from the group if the player removes that follower's Move task using the Manage Land Unit Tasks Order. When this occurs the follower is removed from the leader's list of followers and the leader will no longer check where this follower is.

3.46 JTLS-2020-14977 Error 428 Will Not Generate

Error 428 was unable to generate.

Two code errors were found in the routine disembuni.sim:

- The routine was tracking combat systems by counting CS AVAILABLE and CS In MAINTENANCE. These variables are only used in the actual simulation. The correct method is to use unit TOE. The routine was corrected.
- The routine was calculating what percentage of the unit to put on a certain formation. Since the formation had absolutely no room for any of the unit, it used zero as the percentage. Therefore the weight of the unit was computed as zero, and the formation had plenty of room for a zero weight unit.

The formula computation was modified to also take into account the number of ships in the formation. Since this was a one-ship formation. 100% of the unit had to go on this ship and the weight put the formation over its limit.

3.47 JTLS-2020-14982 Formation Icons Displayed at Equator

Naval formation icons were displayed at coordinates 0,0 on Player WHIPs. The formations were active and visible to all Force Sides with neutral or friendly relationships. Formation icons should not be displayed on a perceived map view until a ship from that Force Side is ordered to join the formation.

An error in the model logic allowed formation icons to be displayed on the map for all Force Sides with a neutral or friendly relationship. Perceived views on the WHIPs therefore included all formation icons when no ships from the perception's Side were in the formation.

The model failed to make a distinction between formations that are active (in existence) and formations that should be displayed. By design, Force Sides with neutral or friendly relationships are made aware of active formations so that ships from different Sides may join. However, the logic incorrectly made the formation icons visible even when no ships from a foreign Side had joined.

The error was corrected by limiting the display of formation icons to just those from the owning side and to foreign Sides with ships in the formation. In other words, friendly and neutral Sides

that do not have any ships in the formation are permitted to know the formation is active, but do not perceive the formation icon on the map until one of their ships is ordered to join the formation.

3.48 JTLS-2020-14988 JOI OTHGold Unreported Errors

Generic connection errors were failing to be reported to the user. For example, when trying to connect to a service that was not running an error would occur but no popup was displayed for user attention.

Generic connection errors are now reported to users via popup messages.

3.49 JTLS-2020-14994 ATOG Not Considering Shared Foreign Runways

For air missions selected from squadrons at an air base that does not have runways of its own, but does have access to a foreign runway, the ATOG does not create the missions.

During initialization, the ATOG collects all the runways in a list for each base if they are owned by the base. Later in the initialization process, a runway is identified as the primary runway at the base and this runway may be foreign. The ATOG was only considering the runways in the list of owned runways while determining if missions had sufficient runway length and availability time (following any repairs) for takeoffs and landings of the aircraft used by the squadron at the base.

The logic for calculating the runways available to the base (and squadron) has been changed to consider the primary runway, which may be loaned to the base from a foreign unit.

3.50 JTLS-2020-15002 Jtlsmenu Improperly Launches OVT GUI

The "jtlsmenu" script starts the Order Verification Tool (OVT) GUI, but will not fork the GUI as a separate process. The "jtlsmenu" script remains in a waiting state for the Order Verification Tool GUI to complete before allowing the user to select any other option.

The Jtlsmenu script code was changed to branch execution whenever the OVT option is selected and this allows the "jtlsmenu" process to execute separately. The "jtlsmenu" process therefore continues, and will allow the user to start other processes.

3.51 JTLS-2020-15004 Air Mission/Convoy Ordered To Move Same Unit

Currently the CEP allows a Transport Unit Order and Mobility Mission Order to specify the same unit to be transported/lifted. If one order is sent after the other order executes and completes moving the unit, there is no problem. However, a problem can occur if one order is sent while the other order is still executing.

A check is now made whenever a convoy or mobility mission starts its Resource task:

1. For a convoy required to move a unit, a check is made to find a mobility mission that has already been assigned to lift the same unit. If there is such a mission, the convoy is canceled.
2. For a mobility mission required to lift a unit, a check is made to find a convoy that has already been assigned to move the same unit. If there is such a convoy, the mission is canceled.

3.52 JTLS-2020-15009 Order Checker Did Not Catch Error

The order checker did not catch field names with the "fill", "with", "all", or "not-equal" elements.

The order checker code was corrected.

3.53 JTLS-2020-15011 Unable To Display Range Rings For Satellites

A user attempted to display range rings for sensors on satellites using the context sensitive menu on the WHIP. For satellites created by orders, the menu option was selected, and the range ring was visible. For satellites created from the database, the menu item to display range rings did not exist so nothing could be displayed.

This issue was caused by errors in the CEP code. In the CEP, objects called range items were not properly initialized on the JODA when satellites were created from the database. A code change was made to correct this problem.

3.54 JTLS-2020-15023 Changes To Air Mission DSA Processing

DSAs could be assigned to Reconnaissance missions that belong to an Air Mission Package using the Change Mission Parameter order. However, the Change Mission Parameter order should stop DSAs from being assigned to an air package mission.

Since other issues with DSA processing were found while testing, the logic for processing DSAs has been updated to solve these issues:

1. Recce missions that belong to an Air Mission Package can no longer be assigned DSAs using the Change Air Mission Parameter Order.
2. Missions that do not have a mission type of Reconnaissance can no longer be assigned DSAs.
3. The routine REPORT.NON.EXISTENT.DSA no longer prints an empty message when listing the non-existent DSAs that were attempted to be assigned to the Air Mission.

3.55 JTLS-2020-15028 Changed Aircraft Type Not Updating Ranges

When a Controller changed the type of aircraft at a unit, the range ring for the aircraft combat system was never updated.

The code to update the unit range data was only called from a place in the code which looked at changes to the unit's combat systems. It was not utilized when the aircraft type was changed. Modifications were made to correct this oversight.

3.56 JTLS-2020-15034 OVP Added JODA Retrieval Value Check

The Controller Kill Aircraft order was brought up in the WHIP but the JXSR did not know what to do with the request because it had no retrieval information in the order XML file. The order checker did not catch this.

The OVP code was modified to add the JODA retrieval value check.

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.

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 5.1.10.0 DATABASE CHANGES

No database structure changes were made for JTLS-GO 5.1.10.0.

APPENDIX C Version 5.1.10.0 REPOSITORY CHANGES

No changes were made to the JTLS-GO 5.1 repository.