JTLS-GO

Version Description Document

May 2020

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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.1.8.0)**

[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 5.1.8.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 5.1.8.0 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

[Blank Page]

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 5.1.8.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 5.1.8.0 is a Maintenance release for the JTLS-GO 5.1 series.

JTLS-GO 5.1.8.0 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.

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 5.1.8.0 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 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 Air Services User Guide (*JTLS-GO Document 24, Version 5.1.2.0)

1.2.3 Updated Documents

- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.1.8.0)

- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.1.8.0)

- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.1.8.0)

- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.1.8.0)

1.2.4 New Documents

No new documents are delivered with JTLS-GO 5.1.8.0.

1.2.5 Delivered Software Components

JTLS-GO 5.1.8.0 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)

- 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 will be addressed in a future JTLS-GO release by only permitting one login per TRIPP user instance, such as a WHIP login prevents from users logging into the same WHIP 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 (JOBE)

- JTLS Geographic Information System (GIS) Terrain Building Program

Instructions for installing JTLS-GO 5.1.8.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 5.1.8.0 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 5.1.8.0, 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 and is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 5.1.8.0 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 5.1.8.0 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 252. 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 LICEN.SE version 2.1 or later) by Object Refinery Limited, http://www.object-refinery.com

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

- KML Operational Interface (KOI)

  The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth^{TM}. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using 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 252.

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 5.1.8.0 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 5.1.8.0 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.

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 5.1.8.0, 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 B will provide a summary of the data structure changes made to the data repository. No data structure changes have been made is this Maintenance release; therefore, 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 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 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.

1.5.3.2 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 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.

1.5.3.3 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 5.1.8.0 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-2020-14762 Insufficient Runway Canceled Alert Message

An air mission was ordered to stand alert at an airbase that was not its home base. The runway at the airbase was not long enough to support the aircraft. The alert canceled and a task cancellation message was generated, which listed incorrect values for the required and available runway lengths at the airbase.

The message definition field assumed that the length of the runway was in KM (the basis for the DISTANCE UOM), but the runway length was in FEET (the basis for the AIRDISTANCE UOM). The message definition file was changed to properly indicate that the data coming from the model was an AIRDISTANCE parameter, not a DISTANCE parameter.

3.2 JTLS-2020-14763 Write TACELINT Reports To File System

The TACELINT Message Service (TEMS) has the ability to connect via TCP to a receiving system, but this capability was insufficient during a recent exercise, due to unique infrastructure requirements. The TEMS required the ability to write its reports to the file system for local post processing and other manipulation.

The TEMS can now write its reports to the file system. This is enabled by selecting "FILES" for the service's C4I Connection Type in the Interface Configuration Program (ICP).

3.3 JTLS-2020-14764 JXSR Incorrect Reason For Denied Connection

The ICP port parameters for a JXSR were changed without stopping the currently running JXSR. A JXSR was started using a new Web Services Manager, which used the new port parameters. This second JXSR failed to receive a download from the JODA and reported a discrepancy with the protocol build for the current JODA. This JXSR also reported that it had connected successfully to the JODA.

The JXSR reports connection failures by providing a JDSP version string and a reason string. The reason string is properly filled with the details of the failure, and the JXSR uses this reason when reporting errors to its error log.

However, the JXSR function was missing the version parameter; therefore, Technical Control received the erroneous report that the connection failed due to a version mismatch. The version string was added to the parameter list for this function so the reason for the failure is now reported properly in the log. The function also was changed to disconnect from the JODA and stop the JXSR. This allows Tech Control to investigate and fix the connection problem before attempting to restart the JXSR.

3.4 JTLS-2020-14765 GIS Tool: Sea Lane Arc And Node Coordinates

The Sea Lane network was modified and saved. When the GIS Tool was restarted and the project opened, the Sea Lane network appeared on the map, but was re-scaled to a small size with all of the arcs and nodes clustered around the Equator and the Prime Meridian. This indicates that the arc and node coordinates were saved in radians, and not in decimal degrees.

The GIS Tool saves all network arcs and nodes following the ESRI shapefile standard. The .shp file that holds the point (for nodes) or polyline (for arcs) coordinates is a binary file that follows the standard. The standard is implemented by the OpenMap dataAccess.shape package and its sub-packages.

The GIS Tool uses these packages to read from, and write to, the arc and node shapefiles. The shapefile standard does not specify the units that the coordinates must be stored in; however, the OpenMap shape I/O packages assume that the coordinates are stored as latitude and longitude in decimal degrees. The user provided the Sea Lane arc and node shapefiles with bad coordinates and it was verified that the coordinates were stored in radians, not decimal degrees.

When the Sea Lane network was saved, the coordinates were written out correctly in decimal degrees to the .shp file, but the contents of the data structure that held the coordinates in memory were altered. If the user saves the network again, the altered coordinates are now written to the .shp file. When the user reopens the project, the arcs and nodes of the Sea Lane network will appear clustered around the Equator and Prime Meridian.

Note - If the user had refreshed the screen before the second save, the problem would have been noticed immediately, because the GIS Tool would have redrawn the Sea Lane network using the bad coordinates now stored in memory.

The cause of the problem could not be identified within the GIS Tool code. It may be possible that the problem is occurring somewhere within the OpenMap I/O packages, but time is needed to examine that code. As a stop-gap fix, additional code was added to the tool to immediately read the network after it is saved. This clears the network from memory and replaces it with the same network with decimal degree coordinates.

3.5 JTLS-2020-14766 Submerged Submarine Jammer Targets

When submerged, all jammer targets are supposed to be turned off. This was not always happening when a submarine submerged.

There was a logic error that jammers were being turned off using the same logic as an Air Defense Target. This was not appropriate and under some unusual circumstances, could cause a crash. This logic error was corrected.

3.6 JTLS-2020-14767 ATO Parser Misidentifies ICAO Name

The ATO Translator (ATOT) is not given a complete list of ICAO names which are discovered from an ATO message by the ATO Parser. The ATOT therefore neglects ICAO names for ATO missions that access an airbase by its ICAO name.

Under some circumstances, the Parser was picking up ICAO names and improperly interpreting them as Air Control names. This resulted in the ICAO not being placed in the ICAO linking table. If the ICAO linking table does not contain an ICAO from the ATO message, the user cannot properly link the ICAO to the associated airbase in the JTLS-GO scenario. The end result is that the ATOT will not discover an airbase that would be needed for mission orders that require the airbase by its ICAO name.

Under these special circumstances, the parser logic was corrected to properly identify the ICAO name and place it in the ICAO linking table.

3.7 JTLS-2020-14768 Order Graphics Missing Return To Home Leg

JTLS-GO 5.1.7.0 delivered an expanded capability to graphically display an objects ordered route based on the contents of an order. During detailed testing, several issues were uncovered with this new capability.

The following issues were found and corrected with the new order graphics capability:

- There was no option to display the last planned leg of an air mission depending on whether a Return Unit was specified or not. The new code will display the last leg for the mission to end at the Return Unit, if specified, or the launching squadron if no Return Unit is specified.

- There was no option to display the fact that an Air Mission was ordered to reverse its Ingress Route after completing its assigned tasks. The new logic checks for this option and graphically indicates that the mission will revere its Ingress Route.

- Not all orders were provided with the graphical definitions necessary to display the complete object routes when viewing the planned order. A complete review of all orders was conducted and graphical definitions were added where needed.

3.8 JTLS-2020-14769 Add IMT Location Preference To WHIP Layout

Users are now able to set a location format for individual IMTs that differs from the primary WHIP preferences. This location format was not being saved to Layouts.

The Location preferences for each IMT in a Layout are now saved and restored with that Layout.

Also, an error is now caught that caused a Layout to fail to load. This occurred when a Layout containing an IMT was recalled very quickly after it was saved.

3.9 JTLS-2020-14770 Speed Leader Pointing In Wrong Direction

When a unit begins moving, there is a speed leader graphic which points in the direction that the unit is moving. When the unit began executing a move task to move from its current location to the first point in its route set, the speed leader was pointing straight up, which meant it had a UT.COURSE of 0. This made the unit appear as if it was heading in a different direction than the route it was following.

The first time the CEP attempts to calculate the course, the first route point in the unit's set is the unit's current location. Since it is considered an error for the current lat/long and the next lat/long coordinates to be the same point when calculating the course, the model would exit the routine and the unit's course would remain 0. Now, the model skips a route point if it is equal to the unit's current location in order to obtain an accurate course.

3.10 JTLS-2020-14771 SSM Selection Criteria Error

When the "Firer Select Missile" option is chosen on the Fire Missile order, the model assesses each SSM type available in the Unit/HRU and selects the "best" weapon to use against the specified target (Unit, Target, or Location). If a Target is specified in the order, the model calculates a Measure of Merit (MOM) based on the SSM weapon type and Target type subcategory.

The MOM was simply the single-shot PH*PK value for the weapon type against the specified Target type. The model then selected the weapon to fire with the greatest PH*PK value. This approach failed to take into account that some SSM weapons have more than one munition (warhead) which would make their lethality greater than a single munition weapon.

The MOM calculation was enhanced to take into account SSM types with multiple munitions. The combined PH*PK value, which is the probability of killing the target, is valid only for a single munition. The probability of at least one kill given one or more munitions is properly defined as:

$$1.0 - ((1 - PH*PK) ** Number\_of\_Munitions)$$

The model was changed to use this statistical formula to calculate the MOM for each weapon type vs the Target type.

3.11 JTLS-2020-14772 OVT Erroneously Reports Polygon Lines Cross

The OVT reported two segments of a polygon crossed. The polygon consisted of six vertices that were arranged such that no segments crossed. However, one of the vertices crossed the international dateline. When that vertex was moved across the dateline so that all six vertices were on the same side, the OVT no longer reported the error.

The C function responsible for checking whether two lines of a polygon cross did not handle the lines as arcs of a great circle. The function now calls the spherical geometry library (libspherical_geom) to perform the crossing lines check.

3.12 JTLS-2020-14773 SSM Launcher Set-Up Time Error

When a Fire Missile order is submitted to a Unit/HRU, the model checks the status of the SSM Target launchers to determine which are available (i.e. PREPARED), or can be made available, for launch. The model computes when the next earliest launcher will be available (immediately, if already PREPARED). The model was correctly assessing each launcher time, but the current game time was saved for comparison instead of the launcher time.

Consequently, all assessed launchers had the same next earliest available time when in reality, they did not. The model therefore could select a launcher to use that was not the next earliest to be available, delaying the launch of the missile unnecessarily in some circumstances.

The code was corrected to use the assessed available times instead of the current time to determine which launcher has the next earliest available time.

3.13 JTLS-2020-14774 ATOT XML File Duplicates

The ATOT overwrites the error and mission XML files after each translation. For each translation of the subsequent ATOs, the error and mission file data can no longer be seen.

The ATOT was modified to make backup files for the error and mission files for each ATO. The file names contain the ATO period identifier in order to keep them from being overwritten.

3.14 JTLS-2020-14775 Synapse Using Common Email Library

The Synapse has a feature for sending text data via email. The text data can be provided by the IMT or the Message Browser. Due to its unique method for sending email, it is difficult to trace and troubleshoot the emailing of this data.

The JDSP library of functions provides a common and standard set of functions for sending email. The Synapse feature for sending email has been changed to use this library.

3.15 JTLS-2020-14776 Mark Target Subcategory Error

Tunnel Class entities are not properly marked as being used within a scenario. This affects SVP error wording.

In the JTLS-GO scenario verification routine, each target class subcategory contains a variable that indicates if the subcategory is used in the scenario. This variable is referenced to decide if certain errors and warnings should be printed. The variable for Tunnel Class was using the Transportation Class variable. This was an oversight because of the similarity of the names. The Tunnel Class attribute was added and code corrected to use the new, proper variable.

3.16 JTLS-2020-14777 SVP Warnings Not Ignored

Warnings 1275, 1276, and 1156 are not ignored when preferences are set to ignore them.

For every warning in the SVP, a check is supposed to be made to see if the user wishes to ignore all the warnings for those types. The check to ignore warnings 1156, 1275, and 1276 were missing. They was added and tested.

3.17 JTLS-2020-14778 Unit Orientation In OPM

The unit orientation in the OPM specifies NONE when it should specify the heading of the unit in degrees.

An error in the formatting of the output was identified and corrected.

3.18 JTLS-2020-14779 Weather Front Location Accuracy

The routine ORDER WEATHER PREDICTION creates a weather report and calls PREDICT WEATHER FRONT LOCATION to obtain an estimated location of the weather front at a given time. It makes use of the IIP ACCURATE WEATHER TIME. This parameter is used to determine a MAX ACCURATE time.

However, although MAX ACCURATE is set in ORDER.WEATHER.PREDICTION, it is never used anywhere else in the routine. The purpose of MAX ACCURATE is to determine at a given time whether the weather front's location will be 100% accurate or estimated. This STR will complete the implementation of MAX ACCURATE.

If the report time is less than MAX ACCURATE, the location of the weather front will be reported with 100% accuracy. Otherwise, the report will provide a random location of the front based on random speed and direction parameters associated with the front's movement. The logic assumes the front's accurate location at the MAX ACCURATE time and then randomizes a predicted location at the time requested in the weather report.

3.19 JTLS-2020-14780 Incorrect Determination Of Is Ship Emitting

The routine IS SHIP EMITTING was always returning a value of No.

The routine has been corrected and now properly checks for any emitting targets that the ship may have.

3.20 JTLS-2020-14781 ATO Parser Consumes ACO Message

The ATO Parser cannot accept and parse two or more Air Control Order (ACO) messages at one time. The user must merge the ACO messages manually into one ACO whenever the corresponding ATO depends on data from two or more ACOs.

The ATO Parser was modified to allow zero or more ACO messages during the Parse phase. This allows the user to specify one, two or more ACO messages for the parse processing of the ATO. The jtlsmenu launcher was also modified to accept user input of none or, one or more ACO message files.

3.21 JTLS-2020-14782 Unit/HRU Occupies Destroyed Shelter

An equipment shelter protecting a unit was destroyed by an enemy unit using the Destroy Target order. The unit's Combat System IMT showed the combat systems still sheltered in the equipment shelter. When the unit was ordered to move away from the equipment shelter, the IMT no longer showed the combat systems sheltered, as expected. Later, the Unit returned to the destroyed equipment shelter and the IMT again showed the combat systems sheltered.

There was no logic in the CEP to cause the sheltered unit or HRU to stop using the destroyed shelter. New code was written to remove protected combat systems and targets from an equipment shelter destroyed by an enemy unit. There was also no logic to prevent a unit (or HRU) from using a destroyed shelter. New code was written to exclude destroyed equipment shelters from consideration when a unit or HRU seeks protection.

3.22 JTLS-2020-14783 ATO Parser Tanker Mission Chits

When an ATO message contains a refuel tanker mission having several fuel chits for other missions, the Parser does not catch any listed receiver missions that do not exist in the same ATO.

The Parser was changed to check each mission listed in every 1REFUEL record of the current ATO. The check includes verifying that the listed missions are included in the same ATO message. When any are found to be missing, the operator will receive a warning report, including the names of the tanker and the receiver missions.

3.23 JTLS-2020-14792 Order Polygon Field Allowed Duplicate Points

When drawing a polygon area on the WHIP map that had duplicate point locations, the duplicate points continued to exist when the order was submitted or saved.

This problem was corrected by correctly purging duplicate points when the polygon was drawn, before the order was submitted.

3.24 JTLS-2020-14793 TGF Move Crash When Unable To Cross River

If a Tactical Ground Formation (TGF) Move was entered, in which some of the units were incapable of crossing a river, the model could crash.

When a unit tried to cross a river and could not, it generated a tactical alert for the Player and then scheduled a Ground Wait Task with an end time equal to the unit's max move time. This gave the Player a chance to solve the problem. Units in a TGF can be given a Ground Wait task, but it should never end because the TGF Move logic will restart the move automatically. The end result was that a given unit was getting two Complete Unit Task events scheduled, which eventually crashed the model.

This STR fixes the problem and the model now checks whether the unit that must stop because of an uncrossable barrier is conducting a TGF move or an independent move. If a TGF move is ordered, no Complete Unit Task event is scheduled.

3.25 JTLS-2020-14794 Altitude Reverts While In Orbit Task

When using a quick order (or using the Manage Air Mission Parameter order) to change the altitude of an air mission, the altitude would change in the SITREP and in the "General Air Mission Information" IMT. The altitude would not change in the "Air Mission Tasking Orders" IMT.

After a period of time, the altitude of the air mission would also change back to the previous altitude before the order was sent.

When changing the altitude, the model would change the AM.ALTITUDE to the new altitude from the order. The model also changed the altitude of every route point in the Mission's route set. However, when a mission is in orbit, if the type of orbit is "Orbit Point", then it only has one single point in its route set. When UPDATE.ORBIT.ROUTE is called, the model created a new route point using the current PET's altitude. Since the model never updated the altitude of the current PET to the new altitude, the air mission ended up using the old altitude instead.

Now, the current object task is set to the new altitude along with the AM.ALTITUDE and the altitude of the route point. Hence, the Air Mission Tasking Orders now displays the new altitude and the mission's altitude never reverts back to the old altitude after sending a change altitude order.

3.26 JTLS-2020-14796 Coincident Arc Crash Moving To Same Location

An HRU was given an order to move to its current location. This caused the model to crash. The user had accidentally submitted the same order twice. The first move task completed normally and the crash occurred when attempting to compute a route for the second move.

The crash only happens if the identical location for the HRU is in a grid that either has a river, barrier, or national boundary. The logic is trying to determine if the move will cross the barrier, but the computation cannot handle the zero distance move. Before determining if the object will cross the barrier, the logic now checks if there really is any reason to move. If there is no reason to move, the task is marked as successfully completed. The move orders for all objects were tested to ensure they did not have similar problems. Ground units and Naval units did have similar problems and these problems were corrected in the same manner.

3.27 JTLS-2020-14798 HRU Combat Crash

An HRU was involved in combat with an ARU. After several high-resolution engagements, the HRU finally lost all of its combat systems. The CEP crashed at that moment.

Investigation revealed that the model was determining how many supplies had been lost in the final engagement from secondary blast effects. Because the HRU had no combat systems

available, its radius was zero. That situation eventually led to a division by zero error in the destroyed supply calculation which crashed the model. To correct the problem, a check was added to bypass the secondary blast calculation if the HRU radius is zero, which means the HRU cannot be covered by the blast.

3.28 JTLS-2020-14799 Change Mission Parameter Order Message

A Change Mission Parameter order was submitted with the Auto Assign Allowed flag set to YES to an OAS mission. The order was properly processed, but the resulting Player message text was wrong. The text stated the mission was now permitted to respond to a CAS request. The Auto Assign Allowed flag permits an OAS mission to detect and attack ground objects, which has nothing to do with fulfilling a CAS request order.

The wrong message text was printed because the wrong sub-message variable was referenced in the order routines. The problem was corrected.

3.29 JTLS-2020-14800 Bridge Operation Without Supplies Not Canceled

A unit was instructed to repair a bridge target, which requires that the unit has some bridging repair supplies. The unit had none, but the operation proceeded, and only when the operation should have completed did the unit discover it did not have the necessary supplies and refused to accomplish the task.

The type of supplies required for an engineering target repair depends on the specific type of target. For bridges, the supply is any category that has a special capability of bridging. For other targets, the special category required is Class IV (construction material). Though supplies are not used in the process, the existence of the appropriate supply is required.

The issue was the routine that was checking for the supplies had a conditional check to see if the target was a bridge. The conditional check was not implemented properly and so the repair was incorrectly started. Once corrected, the bridge repair task was canceled due to a lack of supplies.

In addition, the Design Team determined it was unnecessary to check for available supplies at the end of a repair task. This check was removed from the model; thus, as long as the engineering unit has the needed supplies at the start of the task, the task will completely successfully.

Finally, the task cancellation message was updated to distinguish between missing bridging material and missing CL IV construction material.

3.30 JTLS-2020-14801 Out Of Gas Unit Provided Direct Supplies Crash

A ground unit ran out of gas. The user decided to magically provide the unit gas, and then gave the unit a second move order in an attempt to restart the unit moving. The model crashed when the unit reached the end of the second move provided by the user.

The issue started to form when the user decided to magically provide supplies to the stopped unit by directly accessing the On Hand Supplies for the unit. The proper way to magically give supplies to a unit is through the Login Build Order. Directly accessing the on-hand quantity is reserved for fixing errors and bypasses all logic associated with giving supplies to a unit. One of the bypassed algorithms was to restart the unit and indicate that it was no longer out of gas.

To solve this problem the user submitted the second move order. At this time the unit now had the second Move Task, a Wait Task (telling it to wait for fuel), and the original first move in its task list. When the original move task was put in a wait state, the model automatically requisitioned supplies for the waiting unit. When those supplies arrive, the logic noted that the unit was waiting for gas, and restarted the unit moving.

The issue is that the unit was already moving. This resulted in the unit having two move events and in fact, the unit was moving twice as fast as it should. When it finally got to the end of the last route, there was still a moving event waiting to execute. When it did the model crashed because of a lack of route points.

The solution was to properly set the unit's out of gas flag when the user submitted the second Move order resulting in the postponement of the gas wait task.

3.31 JTLS-2020-14802 Order Graphics Field Type Expansion

Various order field types that included targets, JDPIs, polygonal areas, mission names and other general object types were not supported by the connected field graphics.

The connected field order graphics were expanded to support the various field types.

3.32 JTLS-2020-14803 Order Calendar Editor Did Not Display Properly

Time-specific order fields pop up a calendar editor, and on the initial pop up the calendar did not display correctly. The editor was not completely built before being displayed the first time.

The editor is now completely configured prior to displaying the calendar pop up, and will now always display properly.

3.33 JTLS-2020-14804 Empty Emission Control Air Mission Order

When sending an Emission Control Order for an Air Mission, users are allowed to send it with only a reference and the air mission. There is no requirement that the Sensor and/or the Jammer List be specified as well.

The user is not allowed to submit the Emission Control Order for an Air Mission if there are no sensors or jammers specified.

3.34 JTLS-2020-14805 Order Checker Updated To Use JAXB

The order checker has been using Java codes generated by Altova, based on the order schema, but JTLS-GO needs to move away from Altova. Because Altova generated a number of redundant Java codes, that were difficult to manage and not easy to extend, JTLS-GO now uses JAXB.

JAXB Java object classes were generated and extended classes were created. The order checker uses them to validate order XML files.

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 5.1.8.0 DATABASE CHANGES

No database structure changes were made for JTLS-GO 5.1.8.0.

APPENDIX C Version 5.1.8.0 REPOSITORY CHANGES

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

C.1 Added New Air Defense Class:

- PANTZYR-SA(SP)

C.2 Added New TW:

- SA-22 (57E6)-SA

C.3 Supply Category Changes:

- Added PERSONNEL to SPECIAL CAPABILITY COLUMN for Supply Category NON.COMBATANTS

- Added New Supply Category - CL.IX.BRIDGING.REPAIR (Special Capability Column - Bridging)

- Added New Supply Category - CL.IV.BARRIER (Special Capability Column - Barrier)