JTLS-GO

Version Description Document

January 2019

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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.0.14.0)**

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

JTLS-GO 5.0.14.0 is a Maintenance release of the JTLS-GO 5.0 series that includes an updated SBDKOR50 Standard Database, as well as minor model functionality changes implemented as Engineering Change Proposals (ECPs), 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 5.0.14.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 5.0.14.0 is a Maintenance delivery for the JTLS-GO 5.0 series of releases.

JTLS-GO 5.0.14.0 includes the entire JTLS-GO suite of software and the SDBKOR50 Standard Database that supports a realistic scenario based on the current Korean Peninsula theater of operations. Database modifications that were accomplished to upgrade the previous JTLS-GO Standard Database to this current version are summarized in this chapter, as well as APPENDIX B. Detailed descriptions of minor Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS-GO 5.0.14.0 executes on the Red Hat Enterprise Linux Version 6 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Unchanged Documents

JTLS-GO 5.0.14.0 is a bug release of the JTLS-GO system. Other than date changes, the following documents have had no significant changes since the last release of JTLS-GO.

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 5.0.14.0)

- *JTLS-GO Air Services User Guide* (JTLS-GO Document 03, Version 5.0.14.0)

- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.0.14.0)

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.0.14.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.0.14.0)

- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 5.0.14.0)

- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 5.0.14.0)

- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 5.0.14.0)

- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 5.0.14.0)

- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 5.0.14.0)

- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 5.0.14.0)

- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 5.0.14.0)

- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 5.0.14.0)

- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.0.14.0)

- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.0.14.0)

1.2.2 Updated Documents

The following documents have been changed or improved since the last release of JTLS-GO.

- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 5.0.14.0)

- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.0.14.0)

1.2.3 Delivered Software Components

JTLS-GO 5.0.14.0 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)

- Database Development System (DDS)

- JTLS Symbols Application (JSYMS)

- ATO Translator Service (ATOT)

- ATO Generator Service (ATOG)

- ATO Retrieval Program (ATORET)

- JTLS Convert Location Program (JCONVERT)

- Count Critical Order Program (CCO)

- JTLS HLA Interface Program (JHIP)

- After Action Review Client (AARC)

- Scenario Data Client (SDC)

- Order Entry Client (OEC)

- Order Verification Tool (OVT)

- JTLS Object Distribution Authority (JODA)

- Web Services Manager (WSM)

- Web Services

  Apache Server (APACHE), version 2.4.27

  JTLS XML Serial Repository (JXSR)

  Order Management Authority (OMA)

  Synchronized Authentication and Preferences Service (SYNAPSE)

  XML Message Service (XMS)

- Web-Hosted Interface Program (WHIP)

- Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)

- JTLS Operational Interface (JOI) OTH-Gold, Link-16, and TACELINT

- KML Operational Interface (KOI)

- JTLS Transaction Interface Program (JTOI) supporting

  ICC 2.8.2 and 3.0.2

  NEC-CCIS

  TBMCS

- JTLS Interface Network Navigator (JINN)

- JTLS Order of Battle Editor (JOBE)

- Database Configuration Program (DCP)

- DDS User Interface (DDS)

- JTLS Terrain Building Program

Instructions for installing JTLS-GO 5.0.14.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 5.0.14.0 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.

1.2.4 Released Databases

This release includes the following sample unclassified databases:

- The scenario developed as the Korea Standard Database and named SDBKOR50 is a large-scale, seven-sided scenario database reflecting the approximate starting positions of units involved in the Korea Peninsula theater of operations. This example scenario was developed using unclassified data sources and is consequently not completely accurate. Discrepancies among actual units and their locations are not detrimental to the intended purpose of this database, which is to provide a recognizable and realistic scenario that demonstrates the simulation capabilities and supports JTLS training.

- The scenario DEMSDBKOR50, which is a reduced version of SDBKOR50, has also been updated.

- The scenario blank50 is the SDBKOR50 database with all force structure data removed, which can be used as a framework for building your customized database.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 5.0.14.0 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 6.9 (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.

A RedHat Linux documented loop-back interface problem continues to cause degraded performance issues on Red Hat/CentOS 6 systems. This issue negatively impacts JTLS services when executed on the same server as Apache.

While we expect Red Hat to eventually correct this issue, significant effort has been place on JTLS-GO optimization to minimize impact of the problem. Still we recommend users with more than 25 concurrent WHIP/TRIPP users to avoid this Red Hat/CentOS loopback issue by running Apache and the SYNAPSE on different physical or virtual servers. For users with significantly more than 25 concurrent WHIP/TRIPP users, consider not running other JTLS services, such as the JXSR and XMS, on the same server used by Apache.

See the *JTLS-GO Installation Manual*, Appendix D for some recommended system tuning changes that can mitigate these performance issues.

JTLS-GO 5.0 has been tested with the following versions of Linux 6:

> RedHat Linux 6 - this operating system license must be purchased.

> Oracle Linux 6 - this is a free and distributable version of Linux which has been approved for use by U.S. Government agencies by the Defense Information Systems Agency (DISA).

> CentOS 6 - a free version of Linux 6 that has not been approved by DISA for use by U.S. Government agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 5.0.14.0 has been tested on the following operating systems:

> Red Hat Linux Enterprise Edition Version 5 and 6.

> CentOS Linux Version 5 and 6.

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

Note that when using a Linux system as a client, if a dialog saying "Too many open files" appears, see the *JTLS-GO Installation Manual,* Section 5.5 for a work around. If this occurs on a server, see the *JTLS-GO Installation Manual,* Appendix D.

- Java 8 Update 202 is required for all platforms and must be used to support all workstations.

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

  One of the DoD IA requirements is that software executing on DoD systems must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. Java 8 has fulfilled this mandate by implementing an expiration date for its software. After a certain date, all Java related programs stop working, whether you are connected to an open network or not. Over the past few months, this restriction has been loosened because it caused numerous operational issues, but all versions of Java are issued an expiration date.

  JTLS-GO software releases are linked to the Java 8 security updates. This version of JTLS-GO is released using Java 8 Update 202. The next Java security update is due to be released on 16 April 2019. Until that time, an organization should plan on using Java 8 Update 202 for the reasons of optimal performance and security integrity.

- JTLS-GO database tools require use of a certified Oracle database server and the full Oracle Client installation for runtime requirements. Refer to Section 1.5.2 of this chapter for additional installation details.

- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of the supported versions of Red Hat Linux ES.

- TCP/IP is required for inter-process communication between the JODA data server and all user interface programs. The version of TCP/IP included with the supported versions of Red Hat Linux ES is sufficient.

- The Perl script language is used by the JTLS-GO system and game setup scripts. The version of Perl included with the supported versions of Red Hat Linux ES is sufficient. The Perl program is typically located in the /usr/bin directory. If Perl is installed in a another location, a link should be created from the /usr/bin directory to this program.

- SIMSCRIPT 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 if you are a U.S. Government organization that can obtain source code and plan 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 used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO component programs. The C Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.

• C++ Compiler: It is not necessary to use a C++ compiler to execute JTLS-GO. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS-GO HLA component programs. The C++ Compiler version delivered with the supported versions of Red Hat Linux ES is sufficient.

• The JTLS-GO DDS (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-2.0HLA 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™. 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 5.0 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 High Level Architecture Compliance

The JTLS-GO 5.0.14.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO has moved to a new Federation Object Model (FOM) located in the $JGAME/data/hla directory. Federation testing of JTLS-GO is not complete, but initial tests with CAE's constructive simulation system GESI have been accomplished. Future plans include expanding the capabilities of the JTLS-GO-GESI federation, called GlobalSim continue.

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is RTI-NG-Pro-v7.0 or Pitch pRTI Evolved 4.4.2.0. However, these programs are not included in the JTLS-GO 5.0.14.0 delivery. Users may obtain a full installation package of the RTI software from either vendor: Raytheon Company (http://www.raytheon.com) or Pitch Corporation (www.pitch.se). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

No database structure differences exist between JTLS-GO 5.0.14.0 and previous versions of JTLS-GO 5.0.0.0.

Significant database structure differences exist between the various versions of JTLS-GO 5.0.0.0 and any previous 4.1 series database.

To upgrade your JTLS 4.1 scenario to JTLS-GO 5.0 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

Prior to unloading your newly upgraded JTLS-GO 5.0.0.0 formatted data from your Oracle database server into a JTLS-GO 5.0.0.0 scenario ASCII file, you must execute the JSYMS program,

which will reorganize the structure of the .gs and .scf symbols-related files. See the *JTLS-GO DDS User Guide*, Appendix B.11 for instructions.

1.4.2 Upgrading JTLS-GO Default Symbol Set

The JTLS-GO Default Symbol Set delivered with this release has been modified to use the US 2525C symbol standard and well as the NATO Joint Military Symbol [APP-6(c)] standard. If you choose to propagate this new Default Symbol Set to any of your scenarios that use the 4.1 Default Symbol Set, see the *JTLS-GO DDS User Guide*, Appendix B.12 and B.13 for instructions.

You should not make any modifications to the Default Symbol Set, but you are free to copy the Default Symbol Set and create your own set to meet your needs.

1.4.3 Standard Database Changes

Additional Standard Database changes that are applicable to JTLS-GO 5.0.14.0 are described in APPENDIX B.

1.5 INSTALLATION

1.5.1 Installation Instructions

The *JTLS-GO Installation Manual,* included in the compressed TAR file that is part of this JTLS-GO delivery, provides detailed instructions for installing a new version of JTLS-GO. The *Installation Manual* can be found in the /documents subdirectory of the JTLS-GO TAR file.

> With the JTLS-GO 5.0 series, there have been significant changes to the directory structure which holds the files for the Online Player Manual (OPM). Users must remove all existing files from their JTLS 4.1 or earlier version $JGAME/{scenario} directory before setting up their scenario for execution with the JTLS 5.0 series. Prior to running Option 3 - "Setup System for a Specific Scenario" of the JTLS-GO menu, all users must remove the existing files in their scenario game directory. Failure to clean up the $JGAME/scenario directory will result in a program crash while generating the OPM (Option 4).

> If you are upgrading from JTLS 4.1 or earlier, you **must** unload and load your JTLS databases after installation of JTLS-GO 5.0.14.0, due to a change in one of the stored procedures used by the DDS.
>
> You do **not** have to unload and load your JTLS-GO databases if you are upgrading from JTLS-GO 5.0.0.0 or later.

1.5.2 Oracle Compatibility and Installation

A full Oracle Client installation (not Instant Client) 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 Oracle utilities, such as sqlldr, imp, exp, tnsping, etc., 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 install of the Oracle Client to a different directory that mounts (simple NFS will suffice) to JTLS-GO. 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 .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.

The new DDS application utilizes the Oracle GlassFish J2EE server, which, like the JTLS-GO WHIP Apache server, is delivered with JTLS 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 Disabling Certificate Authority

Industry standard programming practices strongly recommended that Web Start technologies be signed by a Certificate Authority (CA) to validate that an application being run comes from a trusted and registered source under the PKI. Keeping with high security standards, R&A has opted to sign its web enabled applications with a certificate from the COMODO, which uses the Public Key Infrastructure (PKI) to validate the digital signature. Local area networks, intra-nets and standalone systems that have no outside connection to the Internet must configure their installation to disable certificate revocation checks. If this step is not taken under a closed environment, the JTLS-GO web-enabled applications will experience a significant delay in start up while the PKI attempts to validate the certification on COMODO's servers via the Internet.

The following outlines the method that should be used to turn off certificate revocation checks. Again, this step should only be taken if the WHIP, TRIPP and/or DDS are going to be run in a closed environment.

Bring up the Control Panel. On Linux, typing 'jcontrol' in a terminal window will bring up the configuration panel. On Windows, go to the Windows 'Control Panel', select 'Programs' then click on 'Java'. Both Linux and Windows will display the same configuration panel.

Navigate to the 'Advanced' tab and in the list of configuration options, select the 'Do not check' radio button and click 'OK'. The image below illustrates the corresponding selection in the Control Panel.
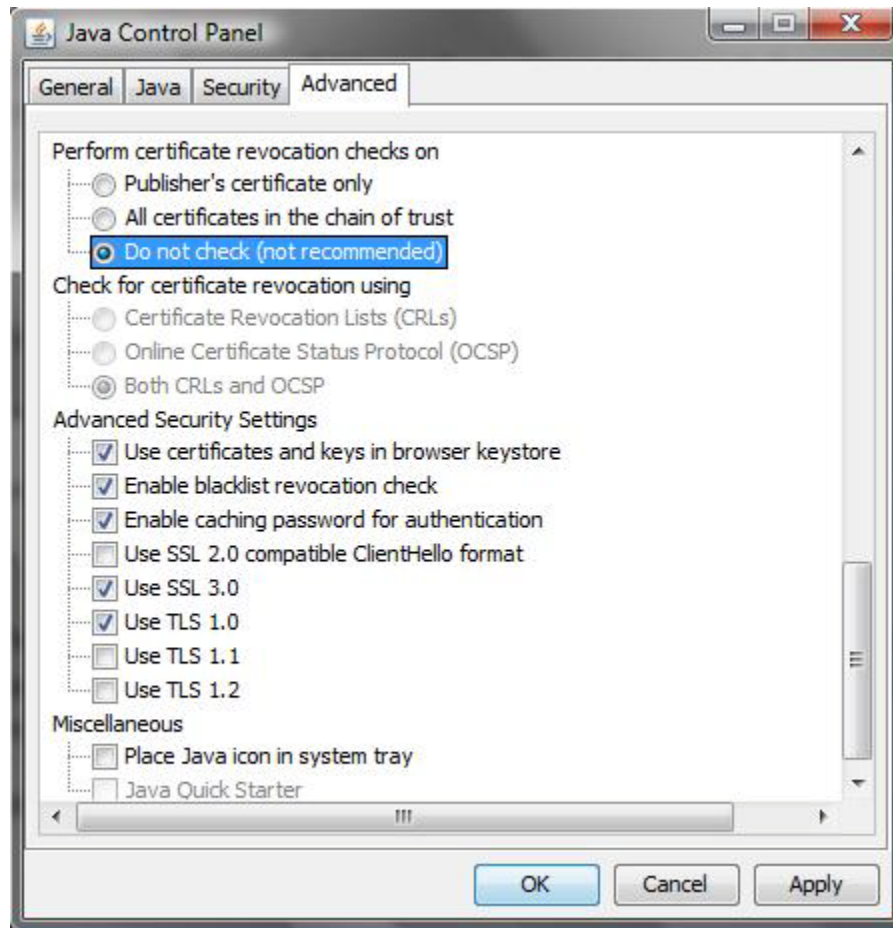
Figure 1.1   Control Panel

Completing this step will allow you to run the JTLS-GO web enabled applications without any external checks and launch the application without the validation delay.

1.5.4 Map Vector File Format Update

The format of the files used to display outline maps on the WHIP and DDSC was changed between JTLS 4.1 and JTLS-GO 5.0. Files in the new format are delivered with JTLS-GO 5.0. The files are located in $JTLSHOME/data/maps/vector and are in XML format.

For users who desire to use any custom files that are in the JTLS 4.1 format, a conversion script called "outlineConverter" is provided. The script can either convert individual files or a directory of files. The user can also choose to convert all types of outlines or only the shoreline outline (coded 9999 in the 4.1 format).

The script has 2 usages:

- To convert a single file:

  ```
  outlineConverter -a <map_file> [-c shoreline]
  ```

- To convert all files in a directory. The file extension is assumed to be .map.

  ```
  outlineConverter -d <directory> [-c shoreline]
  ```

The optional "-c shoreline" argument causes only the shoreline vectors to be converted. Rivers and national boundaries no longer follow hex-edges in JTLS-GO 5.0, but are represented as distinct lines with latitudes and longitudes. These terrain features can be individually turned on and off using the WHIP filter capability. Thus, the need to represent rivers and national boundaries in the vector maps is eliminated. The ability to represent these features, as part of the vector map, has not been removed from the WHIP, but the user should consider whether these old vector features should or should not be displayed by the vector maps.

The output files are placed in the same directory as the input files.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 5.0.14.0 as a result of implementing minor Engineering Change Proposals (ECPs).

2.1 JTLS-2018-13923 JOI - Provide Airmission Callsign in Table

Summary of Model Change Request

The OTHGold and Link16 modules provide tables to monitor the status of units and the track information they are producing. Users often refer to air missions by their callsign, but the tables did not include the callsign data for air missions.

Design Summary

A Callsign column has been added to the air mission table in the OTHGold and Link16 interface modules.

2.2 JTLS-2018-13978 Add Message Number To AAR Message Script

Summary of Model Change Request

The aar_message script gathers all messages, within the specified checkpoint range, generated during the game run. This can easily exceed 100,000 messages over the course of an exercise. The Saxon parser requires an excessive amount of memory to load this number of messages into memory in order to parse it. It is desirable to only load the messages that are required for the parsing.

Design Summary

An argument was added to the aar_message script so that the user can specify the message type number that should be collected and parsed. In addition, since each stylesheet only parses on type of message, in the absence of the argument the script will determine the required message number based on the stylesheet being used. If the message number is omitted from the command line, or the stylesheet is not recognized, then the previous default behavior of collecting all messages will occur.

3.0 SOFTWARE TROUBLE REPORTS

Software Trouble Reports (STRs) describe software code errors that have been discovered by JTLS-GO Users or developers and have been corrected. Chapter 4.0 describes STRs that remain outstanding from previous versions.

3.1 JTLS-2018-13921 Set Air Defense Range Unit of Measure

The Max Range vs Missile field in the Controller Set ADA Parameter order did not have a Unit of Measure (UOM) assigned. The absence of the UOM attributes requires the Player to enter the field as Kilometers rather than allowing them to choose a UOM that is meaningful for their data.

The order definition was modified to use a distance UOM. In addition it was also noted that the utility to set Altitude Zone ranges was also missing the UOM designation and was fixed accordingly.

3.2 JTLS-2018-13922 Crash Decreasing Ship To 0% Strength

While testing ship Air-Defense firing at Surface-Surface Missiles (SSMs), the number of compartments for a Ship Unit Prototype (SUP) was substantially increased. One of the ships of this SUP then suffered extensive damage from SSMs bringing it down to 0% strength and an incapable posture. The ship was then repaired via a Controller Change Unit Strength order.

The same ship was then brought back to 0% strength via a second Controller Change Unit Strength order. This second order resulted in a game crash when attempting to find an undamaged compartment to destroy and one was not found. The ship keeps track of the total number of damaged compartments, and also a list of all compartments with their damage status. There was a discrepancy between these two tallies.

When the code increased the number of compartments associated with the ship it reassigned objects such as small boats, targets, on-board supplies, and on-board combat systems among the new range of compartments. This same list also keeps track of the identifiers for the specific compartments. However, the list is ordered by the compartment number, which requires removing the items from the list and re-filing them when the compartment number is re-assigned. This was not being done which resulted in the list becoming unordered.

When the order to return the ship to 100% strength was sent, the lack of ordering caused some compartments to remain in a damaged state, but the number of damaged compartments was explicitly set to zero. This discrepancy then caused the damage logic, invoked with the second order, to have insufficient compartments available to damage resulting in the crash. The crash was fixed be removing items from the compartment list and re-filing them when we change the compartment number the item resides in was changed.

3.3 JTLS-2018-13924 JOI OTHGold Relationship Change Indicator

Users have the ability to change the side relationship and associated force code for OTHGold tracks. This change only affects the outgoing traffic from the OTHGold message service. The issue was that there was no obvious indicator when looking at the track summary tables that the relationship had been changed. The user had to select each track and open a submenu to determine if the relationship had been modified.

The OTHGold track summary tables now list the original side relationship of a unit in addition to the overriding relationship that has been assigned by a user.

3.4 JTLS-2018-13925 Naval Unit On Sealane Keeps Missing Turning Node

A naval unit that was on a sealane jumped back and forth past the node at which it was supposed to turn.

The problem was that the current sealane on which the naval unit was traveling had a maximum speed that the unit was exceeding. The routine that determined the maximum distance it could travel and the speed at which it could travel along the arc was used to determine that the unit could not make it to the node in its next move, but the logic threw away the computed arc speed and used the computed non-arc speed for the naval unit. This speed was faster than the arc speed and thus the unit sailed past the turning node.

3.5 JTLS-2018-13926 ADA Report Missing Advanced TW

The ADA Report showed the amount (in tons) on-hand of the standard targetable weapons used by an ADA target. The report did not show the amount of advanced targetable weapons on-hand used by the ADA target. This particular ADA target type used both weapon types.

The report was modified to show the weapon counts and the tonnage on-hand for both the standard and advanced targetable weapons (if any) used by any ADA target.

3.6 JTLS-2018-13927 IADS Nodes Initialization

When the CEP is started, all IADS networks are created and can be displayed on the WHIP. However, some of the nodes do not match the locations of their associated SAM/AAA, Sensor, and Communication targets.

The nodes of the IADS networks are initialized using their associated targets. When the targets are Prototype Owned (POT), they are not assigned their proper locations prior to when the nodes are initialized. The locations of POT targets are that of the owning unit. Therefore, the nodes associated with POT targets can be several kilometers from their actual location and tend to overlap other nodes and hide the arcs of other branches of the network.

The code was changed to update the node location from the actual, associated target locations prior to publishing the nodes.

3.7 JTLS-2018-13928 Targets Not Being Switched On

A ship was in the process of sinking, which means it had an Incapable posture, when the Controller sent a Change Unit Strength order to bring both it and its targets back to 100% capable. The ship and targets were brought back to 100%, but a Communication Center target still showed as being Destroyed. An analysis of the code showed that the targets were being rehabilitated before the unit's posture was changed. As part of the process of processing the targets the code checked if the unit was in Incapable posture and would not turn the targets on if it was.

The code to check if the unit should change posture was moved to execute before processing the rehabilitation of the targets. This also happens before subordinates are process, and if the unit switches to a Wiped Out posture its subordinates are removed. This necessitated keeping track of the subordinates before checking for posture changes, and processing the saved subordinates later.

3.8 JTLS-2018-13929 Spreadsheet Attack Order Location Does Not Work

The Spreadsheet Option for the Attack order from the Ground Unit IMT did not work and threw an exception.

The order xml file for the Attack order specified the attack location field as an optional field. The code was changed to a mandatory field. The same field specified BOTH land and water grids are valid for an Attack Location. This is incorrect. Only a land grid is a viable grid. This was corrected.

3.9 JTLS-2018-13930 Delayed Carrier Air Missions Never Fly

Air missions were launched from squadrons on board a carrier while the ship was in port. The missions did not launch, and were placed in a posture of Runway Delay. The ship was ordered to leave the port. When the carrier left the port, the associated air missions were never launched. They remained in Runway Delay indefinitely.

There was no code to launch any delayed missions after a ship had left port. This was an oversight during the development process. Changes were made to automatically begin the take off process for ship-board missions when the ship leaves port. This applies to all delayed missions from all squadrons on the naval vessel. As a result, the missions now take off immediately after the ship leaves port.

3.10 JTLS-2018-13931 New Targetable Weapon Record Creation Fails

Creation of a new targetable weapon record throws exceptions, caused by a database trigger.

The database trigger had a type mismatch against the related database schema field. That mismatch was corrected.

In addition, the default value for the tw_ext_artillery_bundle_time field of the tgtable_weapon table in the JTLS-GO database schema was modified to be half an hour to match the data in all standard JTLS-GO scenarios. Users can simply unload and reload their scenarios to remedy this issue.

3.11 JTLS-2018-13934 ATOT Refuel Missions Without Orbit Locations

A valid ATO refuel mission with an orbit location defined in the associated ACO message is converted into a refuel mission order by the ATO Translator (ATOT), but is always missing the location for its orbit.

The ATOT code for creating refuel mission orders was missing the code statements that insert the orbit location into the order for the mission.

3.12 JTLS-2018-13935 ATOT No Warning For Mobility Mission ICAOs

The ATOT is creating Mobility missions from the ATO with the string "NONE" as the stop unit whenever the unit's ICAO is unknown. No warning is reported to the user prior to when the ATOT writes the orders. The operator discovers the bad unit name only when checking the order group.

The ATOT uses the provided ICAO to try and locate a unit either in the downloaded data from the JODA or from the list of linked ICAOs held by the ATOT's WHIP GUI, linking tables. If no unit can be found the ATOT wrote "NONE" for the name of the unit in the generated Mobility mission order. This code was changed to first write an error in the ATOT's WHIP GUI error page. The order will still have the bad unit name but the operator will have the opportunity to correct the ICAO before batch sending the associated order group.

3.13 JTLS-2018-13936 ATOT Multi-Target Attack Weapon Allocation

The ATOT is distributing all the weapons from the corresponding mission's load to the first target of a multi-target attack mission. Any targets beyond the first are not having any weapons assigned. The resulting order generated for the APM mission therefore needs to be edited by the operator before sending.

The algorithm for distributing the weapons of the mission's load to the mission's targets was not considering the additional targets. This was changed to account for additional targets of the attack mission.

3.14 JTLS-2018-13937 GIS Tool Unable To Save Trimmed Networks

A problem occurred when all the road networks in the GIS Tool project were trimmed using the "Trim nodes in current view" option. When prompted to Accept or Discard the reduced networks, and Accept was chosen, the tool did not save the reduced networks, but saved the original networks instead.

Although this is the same problem reported by STR JTLS-2017-13502, GIS Tool Failed to Save Trimmed Road Networks, the origin of the problem and its solution are different.

A polyline that forms a loop may be trimmed to a single point if the user provides minimum arc length and maximum acceptable deviation constraints that are large. This can occur if the minimum length exceeds the total length of the loop and removing nodes in the polyline satisfies the maximum deviation constraint. The tool throws an exception when the loop is reduced to a single point and this leaves all the networks in their original state. A check to handle this situation has been implemented to eliminate the exception. If trimming the loop results in a single point, the original loop is retained; otherwise, the trimmed loop is presented to the user.

3.15 JTLS-2018-13938 Crash Withdrawing Unit With Severe Attrition

A ground unit experienced severe attrition and the result of this situation was that its nominal travel speed was zero. When the unit attempted to find a withdraw route, it found a route, but when it attempted to determine if it was the best route, the model crashed.

The travel time of a ground unit with severe losses is assumed to be zero. When its nominal speed is zero, the time to accomplish a move is automatically set to FOREVER, an internal JTLS time of 1E20. The model could not correctly determine the best route to take when they had a time for FOREVER. The logic was corrected and the model can now determine that even if a route has a travel time of FOREVER, it is better than not finding a route at all.

3.16 JTLS-2018-13939 Repositioning Ships Removed During Magic Move

A Naval Formation was magic moved while two of its seven ships were attempting to reposition to their assigned station location. The two repositioning ships were inadvertently removed from the formation as part of the logic sequence.

The logic canceled the Change Station task under which the two repositioning ships were operating. Under most circumstances, when canceling a Change Station task, the desire is to remove the ships from their formation. This is not desired nor was it planned for the Magic Move Formation logic. Special code was written to insure the positioning ships are returned to the formation as part of the Magic Move Formation logic.

3.17 JTLS-2018-13940 Repositioning Ship Inaccurate Next Move Data

When a Formation moves, each of the ships in the formation determines whether it can make it to its proper station location by the time the formation is scheduled to move next. If the ship can make it to the newly computed station location, the JODA is updated with the ship's next move location and next move time. This data is used by the ELS to compute between move dead reckoning.

If the ship cannot reach the newly computed station location by the formation's next move time, it will start to move independently to its new station location. Under this circumstance, the model

was still updating the ship's next expected location and next expected move time. This information is not correct and caused the ELS to compute inaccurate dead reckoning information.

The problem was solved. The ship now reports its next location and next location time from the routine that handles independent ship moves.

3.18 JTLS-2018-13941 Ship Destination Incorrect When Formation Stops

If a formation stops, it updates the destination of each of the ships in the formation. The destination of the ships should be its final station location. This was not true. The destination was always being set to the location of the formation.

The issue was corrected. We do not believe that this had an operational impact on the model, but the data in the JODA was not correct and could confuse the Entity Level Server and other JODA clients.

3.19 JTLS-2018-13942 Change Station Task Incorrectly Canceled

A Formation ship was in a wait state, because it could not find a good station location given the movement of its formation. The formation finally moved out of the problem area, and the ship determined that it now had a good station location. The ship properly canceled its wait state and added a new CHANGE STATION TASK to its list.

When the model went to execute this new CHANGE STATION TASK, the model decided that the ship did not have a good station location. This led to the ship being in a non-planned state.

The issue was that the new station task was added based on the formation's planned next move. When the ship went to validate the CHANGE STATION TASK, the model checks based on the formation's current location. The model should have checked based on the formation's next location. This change was made.

3.20 JTLS-2018-13943 Message Format In Manage Mission Task Intel

An order to change a current mission's tasking was received via unsolicited intelligence. The resultant message that was generated had an error in the formatting which resulted in it failing to convert to XML and being delivered.

The error in the message was easy to identify and fix, however the message being generated did not match the order that was sent by the operator. There was a mismatch between integers used within the order and the integers used by the unsolicited intelligence code. This was fixed by switching the use of integers into the proper static defines within the code.

3.21 JTLS-2018-13944 AAR Missing Children Records

The AAR Client was not inserting any records into the AAR_DETECTION_ITEM or AAR_ENGAGEMENT_SURFACE_ITEM tables. The code was attempting to insert records into these tables for inserting the parent records. This is because of an attempt to fix a previous problem, where the AARC would attempt to truncate parent tables before the child tables were truncated. The solution to that problem (reordering the table definitions within the JDS Protocol) introduced this problem.

A more robust solution of identifying parent and children tables was introduced to solve both the problem of truncation and insertion ordering. The parent tables are recognizable within the JDS Protocol because they always have a unique ID attribute, whereas child tables do not.

3.22 JTLS-2018-13945 Crash Restarting Sinking Ship

A moving ship was attacked and started sinking. The speed of the ship was zero and the speed of the associated move task was zero. In an attempt to get the ship moving again, the operator sent a Manage Naval Task order switching the task speed from zero to 0.01. This resulted in scheduling the next event earlier than the current game time, which is a simulation time decrease crash.

The simulation time decrease was the result of a divide by zero when determining the new movement time. The code was modified to check for a zero speed, the attainable speed of the sinking ship, before performing the divide operation. The timing on the resultant event was also checked to ensure it is substantially in the future, otherwise it will be executed immediately to avoid any future issues with simulation time decreases.

3.23 JTLS-2018-13946 Memory Leak In Logistics Rollup Report

The Logistics Rollup Report reserves a local array, which is the number of Combat System Prototypes (CSPs) long, to track losses by the more robust CSP Combat System (CS) name. This enables a rollup report for a combined NATO Brigade to list losses of Challenger, M1A2, and LeClerc tanks versus reporting all the losses as heavy tank or a specific heavy tank. However, this local array was not being released at the end of the report, keeping memory no longer needed or being used by the CEP.

The code was modified to release the array memory.

3.24 JTLS-2018-13947 Cannot Add Insert/Extract HRU Task

The air operator is unable to add a task to insert or extract an HRU. The order goes through but the new task is never displayed, nor does the mission divert to perform the new task.

A mismatch between the order enumerations and the code was preventing the task from being fully implemented. The problem was fixed.

3.25 JTLS-2018-13948 Badly Formatted Message In Order ADA Report

The Order Air Defense Report was not generating properly if there were no air defense targets within the specified region. This prevented the message from being converted to XML and being delivered.

The formatting mistake was fixed.

3.26 JTLS-2018-13949 Old Shared Intel Overriding New Intel

A large multi-sided coalition is being modeled with the sharing of intelligence between the various force sides. Whenever intelligence is received from a foreign source and it is newer than what is known, it receives a current timestamp. This causes newer intelligence to be overwritten.

Consider a coalition of 3 sides, A, B, and C. Side A had detected enemy unit Tango at time 0545Z. It shares intel with side B every 2 hours and at 0600Z passes the intel on Tango to Side B. This is newer than anything side B has so it incorporates the intel and puts an 0600Z timestamp on it. Side B shares intel with side C every 2 hours on the odd hour. But before that happens Side A gets a new detection on Tango at 0645Z. At time 0700Z Side B passes the older piece of intel to side C. It is newer intel for side C, so side C incorporates it. Side C also shares intel with side A every 2 hours on the even hours. At 0800Z side C passes the original 0545Z intel to Side A, but with a timestamp of 0700Z. This is newer than Side A's intel time of 0645Z so Side A throws out its intel and incorporates this "new" intel.

It is believed that the assigning of a current timestamp when the intel is received from a foreign source was a coding mistake. As such it has been fixed and passed intel when now carry the timestamp of the original detection vs the timestamp of the receipt.

3.27 JTLS-2018-13950 Crash Reading Max Sorties From Checkpoint

The game crashed when reading a checkpoint value for a squadron's maximum sorties per day. Within the checkpoint file this was written as "*****", five stars, instead of an integer value. This is the behavior Simscript defaults to when it is told to write an integer in 5 digits but the value is larger than 5 digits. An examination of the situation found a Controller-created unit that had been assigned a value of 999,999 for the maximum sorties per day.

Within the JTLS database a squadron's maximum sorties per day is limited to 4 digits. But in the Create Unit order and the Set Individual Unit Parameter order this value can assigned any positive value up to 999,999. It was felt that the orders were the source of the issue and that they should impose the same restrictions on the value as the database. Therefore the orders were changed to accept a maximum value of 9,999. The same issue was found with the airbase/FARP/Naval unit aircraft capacity values and was fixed in the same way.

3.28 JTLS-2018-13951 Checkpoint Sealane Arc Boundaries Crash

The model crashed during a checkpoint restart while reading the national boundaries that a sealane arc crosses. An examination of the checkpoint file showed that one of the national boundary names had been split between two lines, resulting in the read of two data items instead of the single item. This threw the read off and other data did not align, resulting in the crash.

The code that wrote the checkpoint is configured to handle 15-character national boundary names, which is the DDS constraint on the name length. But the Manage Barriers and Boundaries orders allows the creation of barrier and boundary names up to 20 characters. A 17-character national boundary was created, which caused the write code to write too many boundary names on the line resulting in the splitting of the name.

To solve the problem the order was modified to limit the names of new national boundaries and barriers to 15 characters in accordance with the database. While investigating it was discovered that the same national boundaries appeared repeatedly for an arc, some as many as 256 times. This was tracked down to a duplicate call to the reading of the national boundary checkpoint data. The duplicate call was removed. Lastly, though it had not been a problem, protection code was added to the routine that initially determines the crossing of arc with boundaries to prevent the duplication of boundary names.

3.29 JTLS-2018-13952 Destroyed HRU Attempting To Fire Missile Crash

An HRU was instructed to perform a missile fire event in the future. Before the fire time arrived the HRU was destroyed. The event was not removed, and when the event time arrived it attempted to access the pointer to the HRU resulting in a crash.

The routine that handles the destruction of all HRUs was modified to search for any missile fire events belonging to the HRU and cancel them.

3.30 JTLS-2018-13953 JINN/JOBE Field Source Incorrect

Naval unit basic types were given an OTHGoldType unit type, but required a BasicType unit type.

The code was corrected to set the naval unit basic type from a valid data source.

3.31 JTLS-2018-13954 JINN Fixed Potential Issue With Deleting Units

When the JINN deleting a unit from the database (using the JOBE), it was not handling cases where that unit could be a faction leader, an embarked unit, or a ship in a formation. This could cause the JINN to fail to delete the unit if it was a part of a formation, or introduce bad data where the faction leader pointed to a deleted unit.

The JINN code was changed to use the Oracle-stored procedure delete_units.sql directly, because it properly maintains database integrity while deleting units.

3.32 JTLS-2018-13955 Checkpoint Individual Data Load Failure

Checkpoint data load failed if the user decided to load only the sea lane nodes and arcs instead of the full checkpoint data load.

The full checkpoint data loading process utilizes a workaround remedying a known Oracle SQL*Loader bug. However, this workaround did not exist in the related script, which was responsible for loading only the individual sea lane nodes and arcs. The same workaround which remedies this particular bug during the full data load was implemented in the scripts which are responsible for loading only the sea lane data and flight path nodes and arcs.

3.33 JTLS-2018-13956 Network Naval Move With Faction Side Change

A naval unit, which had a current task involving a sealane network move, was part of a group of units assigned to a new faction which was joining a different Force Side. As part of the code that switches Force Sides, the naval unit's current activity was stopped and re-submitted as a new task. However, the current location and destination location was also inserted into the unit's route.

When the new task started executing, it also inserted the current location and destination into the unit's route. Realizing this is a network move, the code then attempted to determine an offset from the network to use, and assumed that, other than the single start point and end point, all other points on the unit's route are network points. This caused a crash when the duplicate set of non-network start and end points was encountered.

The solution was to remove the insertion of the current location and destination after creating the new task. These are still inserted into the unit's route when the task starts to execute. This code change affects not just naval units, but also ground units, HRUS, and air missions.

3.34 JTLS-2018-13957 Satellite Movement and Collection Issues

Satellites appeared to collect intelligence over the next (future) leg of their movement, and satellites collected data for only a single Directed Search Area (DSA) during each movement step.

This intel collection was incorrectly done for a path which started at the current location and stretched to the next movement point. It was changed to utilize the path between the previous route point and the current location of the satellite.

For satellite collection, each Faction has a specified value for the time it should spend collecting on a DSA. If this collection time is less than movement time for a satellite, then the satellite can collect on multiple DSAs in a single move. For example, if a satellite moved every five minutes and the collection time was one minute per DSA, then the satellite could potentially collect on up to five DSAs per move. The code for this collection process was modified to correctly gather intel on multiple DSAs as allowed by the movement times.

3.35 JTLS-2018-13958 Allow Displaying Unit Without Children

In the Command Tree Filter in the WHIP, DDSC, and JOBE, if all of the child filters were set to the same value, it was propagated up to the parent.

The command tree filter was modified to not propagate changes from child nodes to the parent.

3.36 JTLS-2018-13959 ICC JTOI Detached Squadron Base

Detached squadrons created in ICC were being assigned a home base equal to their actual base. Their home base should have been the home base of their parent squadron. Also, these detached squadrons were being assigned a Unit Establishment equal to their Combat Ready value. Their Unit Establishment should have been equal to the Unit Establishment of their parent squadron.

Detached squadrons are assigned a home base equal to the home base of their parent squadron. Detached squadrons are assigned a Unit Establishment value equal to the Unit Establishment of their parent squadron.

3.37 JTLS-2018-13960 Prevent Caching WHIP/DDS User Config Files

The WHIP and DDS configuration files contain information that should not be cached on client machines.

The WHIP and DDS were modified to retrieve the configuration files directly from the server without caching them to local storage.

3.38 JTLS-2018-13961 Field Restriction Resources Not Generated

For purposes of internationalization (i18n), all readable text associated with a WHIP should have an associated resource. The end user may then modify these resources to display the WHIP in their native language. These resources include error messages generated by the OMA when an order fails verification. It was noted that such resources do not exist for field-restriction errors.

The i18n Property Generator was modified to create the resource associated with the field restriction help.

3.39 JTLS-2018-13962 Unable to Share Authority

The primary WHIP name for one of the Force Sides contains 23 characters. The ICP allows up to 25 characters. However, the Read Order File order only permits a WHIP name of 20 characters to be entered.

The Read Order File order was modified to accept WHIP names of up to 25 characters in length. In addition, the WHIP name field was modified from being a free format text field to being a JODA object field, allowing valid WHIP names to be selected from a list versus typing them in.

3.40 JTLS-2018-13963 Convoy Not Listed As Damage Object In AAR

A convoy had suffered damage from an Air-Ground attack. The event was submitted to the AAR with the corresponding damage record. However, the user could not find the associated event in the documentation for convoys.

Both Convoys and JDPIS were missing as potential victims for a Ground Damage event in the on-line documentation. The XML file describing the various events, and which is used to generate the documentation, was modified to include the two object types.

3.41 JTLS-2018-13964 Unit Renaming Fails If Referenced From HRU

Unit renaming fails in the DDS if the unit was set as the initial coalition support unit of an HRU record.

The unit renaming related packages were modified to include the High Resolution Unit (HRU) database table, if a unit was set as the initial coalition support unit of an HRU record. The users can simply reload their scenarios into Oracle database to re-create the modified database renaming packages.

3.42 JTLS-2018-13965 Naval Patrol Occurring Outside Designated Area

Naval Formations and individual ships were executing their patrol tasks while the vessels were outside of the specified patrol areas. This behavior was not in agreement with their received orders. Patrols should only take place within given OPAREAs or Polygon Areas.

When a Naval unit or Formation was ordered to patrol an area, it may have been required to first move to that area from its current location. An ingress route may be submitted with the Patrol order to follow a specific path. However, if the patrolling object was given extra tasking, or if some of the pre-defined movement tasks were deleted, the Naval Unit or Formation could prematurely begin its Patrol task before it reached the specified area.

To prevent this problem, the CEP code was modified to better plan the movement to a Patrol area. The execution process for the Patrol task was modified to check if the object was properly located. If so, then the Patrol can begin, otherwise one or more Move tasks will be added before the Patrol can start.

3.43 JTLS-2018-13966 Import Order Groups Failed If Option Selected

If an Order Group was imported, and the user chose the option to rename the group/orders to prevent overwriting a group or order with the same name, the import failed.

The Order Group label was being set too late in the process of importing, throwing an exception when an attempt was made to sort the order group tree.

3.44 JTLS-2018-13967 OAS Alert Spreadsheet Order Default To Home

The OAS Alert order in the Spreadsheet Module defaults to "NONE". This entry is rejected by the OVT. A correct default entry should appear. The single entry OAS Alert Order Panel defaults to 'blank' and correctly associates that to home base.

The Spreadsheet Module OAS Alert order code in the Air Squadron IMT was changed to default the OAS Alert Unit to the homebase of the squadron launching the aircraft.

3.45 JTLS-2018-13968 Model Crash Redirecting Cruise Missile

The Player was redirecting a number of cruise missiles to new targets, specifying that their orbit location should not be canceled. None of the cruise missiles had an orbit station in their original orders. However, the instruction to not cancel their orbit resulted in an orbit at their current location being inserted, and the missile remained at that location until it was almost out of fuel, and then attempted a move for a time in the past. This resulted in a crash.

The code was modified to properly handle the situation where a non-existent orbit location should remain unchanged. Code was also inserted into the next move to prevent an attempt to schedule anything in the past. During testing of these fixes it was also noted that the new cruise route was not being updated with the JODA. This was also fixed.

3.46 JTLS-2018-13969 Embarked Units Missing From Report

In earlier versions of JTLS, the Amphibious Operations Report (generated by the Amphibious Status order) showed ground units embarked on ships in a formation. Now, if no amphibious operation was currently executing, no embarked units were included in the report. The embarked units section of the report showed only if the formation was executing an amphibious assault or pickup order. The missing information was useful when submitting a Cross Decking order.

The missing section of the Amphibious Operations Report was bypassed by a logic branch introduced during debugging and never removed. The if-check skipped the embarked units section unless the formation was executing an assault or pickup order. To restore the missing section, the spurious if-check was removed.

3.47 JTLS-2018-13970 Small Boats Not Capable Used in Amphib Ops

Small boats that are not amphibious capable were allowed to participate in amphibious operations. Any small boat type belonging to a ship in a formation was considered eligible to carry portions of ground units in an amphibious assault wave or amphibious pickup wave. Small boat types that are not amphibious capable should be excluded from amphibious operations.

A new criteria was added to the logic that creates and sends small boat waves in an amphibious operation. The new criteria checks whether the small boat type is amphibious capable before creating an amphibious wave. If the small boat type is not amphibious capable, no amphibious wave is created using small boats of that type.

3.48 JTLS-2018-13971 Stealth Aircraft Early Detection And Engagement

Stealth aircraft were being detected too easily. This resulted in unexpected engagements with enemy air defense (AD) sites.

Stealth aircraft were prematurely detected by enemy air search sensors and then fired upon by air defense sites. These detections were made by VISUAL sensors (meaning collection with human eyes) even though the aircraft was flying at over 30 thousand feet.

This problem resulted from two issues. First, there was a code error where detections were made for air missions. The code was not utilizing the Sensor data parameter for the Maximum Detection Altitude. Corrections were made to prevent the detection of missions which were flying above the maximum detection altitude.

Secondly, there were errors in the scenario database. For this testing, all of the VISUAL sensors had a Maximum Detection Altitude of 100 thousand feet. This was an unrealistically large value for this data parameter. Users are advised to use realistic values for the Sensor Maximum Detection Altitude for their sensors. It is suggested that users also verify that the Recognition and Identification Altitude parameters are consistent with the Maximum Detection Altitudes for their Sensors.

3.49 JTLS-2018-13972 Cross-Deck Order Embarked Asset Error

A Cross-Deck order was submitted to transfer 50% of an embarked ground unit from one ship to another. The dry portion of the weight was transferred to a different unit that was already embarked on the new host ship. This discrepancy was shown in the automatically generated Amphibious Operations Report. The wet portion was transferred to the correct unit on the new host ship.

The logic that transfers the combat system dry weight to the new host ship incorrectly assumed there would not be another unit already on the ship. The logic correctly created a new embarked asset for the transferred unit portion on the new host ship and filed it last in the embarked unit set. However, the logic then grabbed the first embarked unit asset in the set and transferred the dry weight to the wrong embarked unit asset. The problem was corrected by changing the logic to grab the last embarked unit in the set which is always correct no matter how many embarked unit assets are on the new host ship.

3.50 JTLS-2018-13973 Ship Hull Numbers Not Updating On COP

The Controller was updating hull numbers for OPFOR naval units, but the changes were not appearing on the Common Operational Picture (COP).

Hull number is a common attribute within the JDS Protocol, meaning that all sides should know about it. The error was that when it was changed the update was only sent for the Controller and owning side. The code has been modified to generate the update for all sides.

3.51 JTLS-2018-13974 HRU KIA Not Showing Up In Report

The personnel summary section of the Logistics Rollup Report was not listing Killed in Action (KIA) numbers from HRUs. The IMT screens also did not have this information.

Other numbers from HRUs, such as available and TOE, were being included. The additional information for KIA was added. As for the IMT screens, when a new JODA download was executed, such as after a game restart, the numbers were not initialized. However updates to the HRU combat systems would include the KIA. Now both the initialization and updates include the KIA count.

3.52 JTLS-2018-13975 DDS Screen Definition Files Errors

Logic and format errors were discovered in the DDS screen definition files.

Errors in the DDS screen definition files were corrected.

3.53 JTLS-2018-13976 Controller Aircraft Kills Not In AAR Tables

Controller Kill Aircraft orders do not result in a record in the AAR Table for Other Aircraft Losses.

The calling argument list between the CEP Simscript code and the C code which sends the JODA packet was inconsistent and missing the killer receiver number and killer type. There are 3 types of other aircraft losses, Controller kills, external model kills, and crash landings. Only the Controller kills have a killer receiver number and type, the Player order that caused the kill. These data were not being passed to the calling tree, requiring numerous routine argument lists to be modified to handle it.

For those non-Controller losses the records were also being rejected by the AAR Client since the zero reference was considered an unwanted record. It was decided this should not happen, and now events that have unresolved references, usually as a result of filter settings, will still be inserted with the reference zeroed.

There was also an error in the AAR retrieval of the other losses records, which was fixed.

3.54 JTLS-2018-13979 Manage Attack Package Order Help

The help associated with the Rendezvous Point field of the Manage Attack Package order was misleading. This field is filled when the Package Timing Option is Time on Task. The help stated that the field is the rendezvous location for a Reference Point package. The Package Timing Option of Reference Point Time is entirely separate and different from the Time on Task option. The help should have stated the field is the rendezvous location for a Time on Task package.

The help was changed accordingly.

3.55 JTLS-2018-13980 Crash While Rearming Air Mission

The model crashed while an air mission was attempting to rearm its weapons from an airbase. The mission landed at a different location than its home base. This unit also did not have any of the required supplies to rearm the mission.

The model crashed during the rearm task because it was attempting to send a message which used the name of the unit providing ammunition to the mission. The original order for the mission had no specified unit, only a location where the mission was to get rearmed. The crash occurred because no suitable unit was given. Changes were made to search for an appropriate unit when none was given for the rearming task. If no unit was found, the model sends out a suitable rejection message.

3.56 JTLS-2018-13983 Modify Package Missing Messages

A Manage Attack Package order with the Modify Attack Package option selected was submitted to change the Package Timing Option from Airborne on Call to Time on Task. The Primary Attack (OAS) mission did not have an assigned target. The order was accepted by the CEP with no apparent results.

The logic determined that the order was in fact illegal because the attack mission did not have an assigned target. The order was rejected by the model, but no message was generated to inform the player. The routine that checks the legality of the order was missing code to generate the rejection message. Two other such player messages that should have been generated for different reasons were also missing. All three missing rejection messages were added.

3.57 JTLS-2018-13985 Target Associated Unit Not Updated

The owning and associated units for a stationary SAM target were the same ground Unit. The Unit was destroyed, leaving behind the SAM target undamaged. The Target IMT correctly showed no owner, but still showed the same (destroyed) associated Unit for the target. The target should have been assigned a new associated unit.

Note: This situation didn't apply to owned targets that were destroyed with the Unit.

The routine that determines the new associated (support) unit when the owning/associated Unit is destroyed incorrectly bypassed the logic that assigned a new associated Unit. The coding error was corrected so that the SAM target is assigned a new associated Unit.

3.58 JTLS-2018-13988 MISREP Always States "No Damage Observed"

The final mission report generated after an OAS mission completes includes a section of pilot observations. The section always ended with the statement "No Damage Observed" when, in fact, damage was listed in the report.

A code error was corrected to display the statement "No Damage Observed" only if there was no damage reported in the MISREP.

3.59 JTLS-2019-13989 Assign Command Authority To New Units

Whenever a new unit is created, it is necessary for the Force Side's primary WHIP to additionally send a Share Command Authority order for the unit, or the unit cannot be tasked by Players. Command authority should be set automatically to avoid this additional step. This would make the behavior consistent with other CEP functionality which creates new units, such as ordered detachments, civilian air traffic squadrons, and air and amphib lifted detachments.

The logic for the Manage TPFDD order was modified so that if the Higher HQ of a unit that has not yet arrived is being changed, the code will destroy any existing shared authority for the unit and provide it the same shared authority as the new higher HQ. This is consistent with other unit creation code that duplicates the command authority of the higher HQ when creating a unit. The one exception is squadrons created by the Civilian Air Traffic order which were, and still are, given the authority of the homebase versus the squadron they are being split off from.

3.60 JTLS-2019-13994 OVT Does Not Perform Field Restriction Checks

The OVT Library was never programmed to handle field-restriction checks (dependency checks between two or more fields) that are defined in the order XML files. The most common two types are ensuring that a target category index and subcategory name are consistent, and ensuring that the unit on a fire artillery order has a caliber to shoot the specified weapon.

A Read Order File was constructed to create JDPIs, which require a target category index and subcategory name. There were errors in the combination which the OMA/OVT did not catch and which resulted in multiple model crashes.

The OVT was expanded to perform most field restriction checks. The exception is the checks for a unit caliber and targetable weapon pairing. This specific type of check uses a JODA table structure which the OVT has no code to handle. Since this specific type of check is obsolete in JTLS-GO 5.1, and violation of the checks does not result in model crashes, it was decided to not expend resources fixing this specific type of check.

All other field-restriction checks, and specifically those dealing with target category index and subcategory name, have been implemented.

3.61 JTLS-2019-14005 Message Problems In Mission Reports

Mission Reports were generated incorrectly for some air missions. Some of the remarks were incomplete. This caused the message data to be terminated prematurely. The associated message was not available on Message Browser in the WHIP.

This problem was caused by an error in the method where mission messages were generated. The data for the messages was temporarily stored in a buffer, and that buffer was inadvertently

over-written by a different part of the code. This caused some of the remarks to have improper content which subsequently resulted in conversion errors when generating the message for the browser. A small code change was made to correctly format the message data. This problem only affected Mission Reports for missions which had more than one intercepting air mission.

3.62 JTLS-2019-14012 Weapons On Ships Wrongly Firing While In-Port

When a ship was in port, it was incorrectly firing weapons at incoming missiles. The weapons were fired from on-board air defense (ADA) targets. However, the ADA target class was not allowed to fire while the ship was in-port.

The model was not checking to see if the ship was in-port before it fired its weapons. A code change was made to correct this oversight. Firing from an in-port ship was controlled by the data attribute for Air Defense called AD PORT FIRING ALLOWED.

3.63 JTLS-2019-14014 WHIP Name Truncated In Reports

The WHIP name was truncated in the "Query Reporting Players" and the "Players in Game" reports when the name was more than 15 characters in length.

The message templates for both reports were formatted to accommodate only 15 character WHIP names. To correct the problem, the format size was expanded to 25 characters (the maximum allowed WHIP name length) in both message templates, including some clean-up. A search of all messages that write a WHIP name didn't find any other truncation errors.

3.64 JTLS-2019-14020 Incorrect Display of Mission Package In IMT

When an air mission was removed from a mission package, the air mission IMT continued to show the mission was assigned to the package.

The IMT definition was using the air mission name to fill the cell. The JXSR is only capable of informing the WHIP that a value is now null for object ids (aka JODAs). Since mission packages are not map objects in the WHIP, implemented a new table cell renderer specific to mission package name.

3.65 JTLS-2019-14021 Air Mission Damage Not Recorded In AAR

There were only 2 damage records for air mission-inflicted damage in the AAR database. The Controller Damage Reports showed extensive damage inflicted by air missions.

In order to perform filtering on records the AARC must have knowledge not just about the record to be filtered, but any record that it might refer to. For example, an Air-Air engagement involves both a killer and a victim air mission. The AARC maintains knowledge of all air missions because the Air-Air engagement event might have a filter that states only records that involve a killer mission from a specific side should be filed.

For AAR records that exist in the CEP as objects, such as units, air missions, JDPIs, convoys, etc, the AARC always maintains knowledge about them. But some records, specifically engagements and detections, are so numerous that the AARC expires and forgets about them after a certain amount of time.

The AARC was improperly retrieving the creation time of these records as game start, which always resulted in their immediate expiration. Thus, for missiles, whose damage event record was filed at the time of firing but whose damage item records were filed at the time of impact, the damage event record did not exist causing the damage item records to be recorded w/o an associated damage event. This broke the link between the record and the firing air mission.

As stated above, the damage event records were filed at the time when an air mission fired its weapons. But if cruise missiles are fired, it could be hours between the firing event and the impact/damage assessment. It thus became necessary to re-write the CEP code so that both the damage event record and the damaged item records are input into the CEP together at the time of impact. The air mission itself will still record that it has fired its weapons with the AAR, but the actual firing event will not be inserted until impact of the weapons.

3.66 JTLS-2019-14023 Manage Air Mission Tasks Package Crash

An air mission package with Package Timing Option of Airborne on Call was orbiting. The Primary Attack mission Targeting Option was Unplanned Type. A Manage Air Mission Tasks order was submitted to add a multi-object attack with time NOW to the Primary Attack mission. The Primary Attack Mission began to move toward the target. The remaining package missions remained in orbit Holding. The player tried to force the remaining missions to follow by sending a Manage Attack Package order to change the Package Timing Option from Airborne on Call to Time on Task with a new rendezvous point (closer to the intended targets). The remaining missions added the new Rendezvous task at the top of their task lists, as expected. The CEP then crashed.

The situation that ultimately led to the crash was precipitated when the player attempted to convert an Airborne on Call package to a Time on Task with a new rendezvous location AFTER the strike mission left the rendezvous location. The logic attempted to execute the new rendezvous task for the remaining missions, but the strike mission had no rendezvous task, which the model had not anticipated and therefore crashed.

This situation would never have arisen if the player had added the Attack task sequentially AFTER the currently executing Air Hold task. Selecting NOW for the attack and then sending a new rendezvous order led to the crash. If the attack had been added after the Air Hold task and then the package was changed to Time on Task (without the new rendezvous), the entire package would've moved to execute the attack as desired.

Beyond that procedural issue, new validity checking was added to the Manage Air Mission Tasks order. The new logic rejects the order if the player tries to add an Attack task before the rendezvous/air hold/ground alert task. A non-compliance message notifies the player that the

Attack (i.e. Fire Weapon task) must always be added AFTER the rendezvous/air hold/ground alert task.

3.67 JTLS-2019-14024 Intermittent Missing Order Fields

An order field would sometimes be missing from the order panel.

The field had its visibility set to false. When and where this is happening is unclear. A check was added when building an order field panel to determine if the field is not visible and if so, to force the field to be visible. Added additional synchronization to the field panel recycle bin.

3.68 JTLS-2019-14030 Model Crash Ship Movement With Air Mission

The model crashed when a ship with a mission in Aircraft Delay posture moved. The model was attempting to determine if the scheduled launch times for any of the air missions should be adjusted due to the move. It does this by checking the location and timing requirements of the air mission's first primary task. In this case the primary task for the air mission had been canceled, leaving no primary task, which resulted in the game crash.

An air mission should always have a primary task, and the player should not have been allowed to delete the only primary task associated with the air mission. Without a primary task, the air mission has nothing to do, and if this is desired the proper action is to cancel the air mission. Code was added to prevent the player from deleting the only primary task of an air mission, unless it is the mission's current task which should cause the mission to head home. This should prevent the crash situation from occurring again, but to be safe a logic error was also put into the code to catch and ignore the mission if it has no primary task.

4.0 REMAINING ERRORS

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

As JTLS-GO represents a major release of new functionality, remaining outstanding errors 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.

4.6 Strategic Airlift Does Not Work

Users should not submit Strategic Airlift Orders

4.7 Tactical Ground Formation Attacks Do Not Work

Users should not submit Tactical Ground Formation Attack orders. Tactical Ground Formation moves work fine.

4.8 Materiel Handling Equipment (MHE) Does Not Work

The use of Materiel Handling Equipment (MHE) has been temporarily disabled in the model.

4.9 Port Amphibious Operations Do Not Work

Users should not submit an order to do an amphibious assault or an amphibious pickup with the Port Option. The user should simply indicate that "Beach" operations are desired or Magic Move the unit off of the formation at an appropriate time.

4.10 Multiple Point Ground Group Move Results Not Consistent

A Move order was submitted to a ground unit with a specified route consisting of three points. Four other ground units were specified as part of the group move (not a TGF move). After the lead unit completed its third Move task and the other units finished their movement, the position of the other units varied significantly from their starting locations relative to the lead unit before the first move began. The reason this happened is that when the group leader starts its move, it makes note of the relative position to each of the units in the group. Not all units were complete with their first move; therefore, the relative positions did not match the starting relative positions. To solve this problem, insure that all units have completed a previous group move before giving a second group move order.

4.11 OVT Does Not Perform Checks For Caliber/Weapon Pairing

The OVT Library was never programmed to handle field-restriction checks (dependency checks between two or more fields) that are defined in the order XML files. A Read Order File was constructed to create JDPIs, which require a target category index and subcategory name. There were errors in the combination which the OMA/OVT did not catch and which resulted in multiple model crashes.

The OVT was expanded to perform most field restriction checks, except for checks for a unit caliber and targetable weapon pairing. This specific type of check uses a JODA table structure which the OVT has no code to handle. This specific type of check is obsolete in JTLS-GO 5.1 because caliber structures were removed, and violation of the checks does not result in model crashes, so this check will not be fixed in JTLS-GO 5.0.

In addition, if the calibers that can fire a specific targetable weapon are modified by the Controller, those changes are not being processed by the JXSR unless a full JODA download is performed.

APPENDIX A. ABBREVIATIONS AND ACRONYMS

Terms are included in this Appendix to define their usage in JTLS-GO design, functionality, and documentation.

AAA	Anti-Aircraft Artillery
AADC	Area Air Defense Commander
AAL	Air-to-Air Lethality
A/C	Aircraft
ACP	Air Control Prototype
ADA	Air Defense Artillery
AEW	Airborne Early Warning
AFB	Air Force Base
AG	Air-Ground (Air-to-Ground)
AI	Air Interdiction
AIM	Air Intercept Missile
AIREF	Air Refueling
AKL	Area Kill Lethality
AMMO	Ammunition
AO	Area of Operations
AOC	Air Operations Center
APC	Armored Personnel Carrier
ARECCE	Armed Reconnaissance
ARTE	Air Route
ARTY	Artillery
ASC	Automatic Supply Calculation
ASCII	American Standard Code for Information Interchange
ASW	Anti-Submarine Warfare
ATC	Aircraft Target Category
ATGM	Anti-Tank Guided Missile
ATK	Attack
ATO	Air Tasking Order
ATORET	Air Tasking Order Retrieve Program
ATOT	Air Tasking Order Translator
AWACS	Airborne Warning And Control System
AZ	Altitude Zone

BADGE	Bilateral Air Defense Ground Environment (used by Japan Defense Agency)
BAI	Battlefield Air Interdiction
BDA	Battle Damage Assessment
BDE	Brigade
BN	Battalion
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CA	Civil Affairs
CADRG	Compressed ARC Digitized Raster Graphics
CAP	Combat Air Patrol
CAS	Close Air Support
CAT	Category
CCF	Central Control Facility
CCP	Command Control Prototype
CCU	Controller Change Unit
CEP	Combat Events Program
CMDR	Commander
COP	Common Operational Picture
CP	Combat Power
CS	Combat System
CSP	Combat System Prototype
CTAPS	Contingency Tactical Air Planning System
CTG	Commander Task Group
CTRL	Control keyboard command
DCA	Defense Counter Air
DCL	Digital Command Language
DDS	Database Development System
DEMSDB	Demonstration Standard Database
DISA	Defense Information Systems Agency
DIV	Division
DMA	Defense Mapping Agency
DoD	Department of Defense
DOS	Days of Supply

DPICM	Dual Purpose Improved Conventional Munitions
DS	Direct Support
DSA	Directed Search Area
DTG	Date Time Group
EC	Electronic Combat
ECM	Electronic Counter Measure
ECP	Engineering Change Proposal
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
JDA	Japan Defense Agency
JDPI	Joint Desired Point of Impact (formerly DMPI: Desired Mean Point of Impact)
JDS	JTLS Data System
JDSP	JTLS Data System Protocol
JEDI	JODA Entity Data Identifier
JMCIS	Joint Maritime Combat Information System
JMEM	Joint Munitions Effectiveness Manuals
JODA	JTLS Object Distribution Authority
JOI	JTLS Operational Interface
JPL	Jet Propulsion Laboratory
JRSG	Joint Rapid Scenario Generation (formerly JIDPS: Joint Integrated Database Preparation System)
JSDF	Japanese Self-Defense Force
JTLS	Joint Theater Level Simulation
JTLS-GO	Joint Theater Level Simulation - Global Operations
JTOI	JTLS Transaction Operational Interface
JXSR	JTLS XML Serial Repository
KIA	Killed In Action
KM	Kilometer
KNOTS	Nautical miles per hour
LA	Lethal Area
LAN	Local Area Network
LAT	Latitude
LB	Login Build (JTLS order type)
LDAP	Lightweight Directory Access Protocol
LDT	Lanchester coefficient Development Tool
LOG	Logistics

LOGIN   Logistics Input

LOGREP   Logistics Report

LONG   Longitude

LOTS   Logistics Over The Shore

LR   Long Range

M&S   Modeling and Simulation

MAPP   Modern Aids to Planning Program

MB   Megabyte

MCP   Mobility Counter-mobility Prototype

MCR   Model Change Request

MG   Machine Gun

MHE   Material Handling Equipment

MIP   Model Interface Program

MOGAS   Motor Gasoline

MOPP   Mission-Oriented Protective Posture

MOSAIC   NCSA user interface software

MOTIF   X Window System graphical interface

MP   Maneuver Prototype

MPP   Message Processor Program

MSC   Major Subordinate Command

MSG   Message

MTF   Message Text Format

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
SDB	Standard Database
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)

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

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

APPENDIX B. Version 5.0.14.0 STANDARD DATABASE CHANGES

No new changes were made to the Standard Database (SDB) for this maintenance release.

APPENDIX C. Version 5.0.14.0 DATABASE CHANGES

No changes were made to the JTLS-GO 5.0.14.0 database for this maintenance release.