JTLS-GO

Version Description Document

April 2017

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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.0.4.0)**

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

JTLS-GO 5.0.4.0 is a Maintenance release of the JTLS-GO 5.0 series that includes an updated SBDKOR50 Standard Database, as well as major 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 5.0.4.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 5.0.4.0 is a Maintenance delivery for the JTLS-GO 5.0 series of releases.

JTLS-GO 5.0.4.0 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 Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS-GO 5.0.4.0 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 5.0.4.0 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 5.0.4.0)

- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 5.0.4.0)

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.0.4.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.0.4.0)

- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 5.0.4.0)

- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 5.0.4.0)

- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 5.0.4.0)

- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 5.0.4.0)

- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 5.0.4.0)

- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 5.0.4.0)

- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 5.0.4.0)

- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 5.0.4.0)

- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 5.0.4.0)

- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.0.4.0)

- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.0.4.0)

1.2.2 Updated Documents

- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.0.4.0)

- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.0.4.0)

1.2.3 Delivered Software Components

JTLS-GO 5.0.4.0 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)

    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 5.0.4.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 5.0.4.0 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 5.0.4.0 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 6.8 (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.

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 client, see the *JTLS-GO Installation Manual,* Appendix D.

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 5.0.4.0 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.

- Java 8 Update 131 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 the software 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 will stop working, whether you are connected to an open network or not.

JTLS-GO software releases are linked to the Java 8 security updates to meet this requirement. This version of JTLS-GO is released using Java 8 Update 131, which expires on 18 July 2017. A new version of JTLS-GO will be released prior to that expiration date.

- 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.

1.3.2 JTLS-GO High Level Architecture Compliance

The JTLS-GO 5.0.4.0 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 Gefechts Simulation (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 5.0.4.0 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 5.0.4.0 and JTLS-GO 5.0.0.0.

Significant database structure differences exist between JTLS-GO 5.0.4.0 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 symbology standard and well as the NATO Joint Military Symbology [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 5.0.4.0 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).

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, intranets 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.
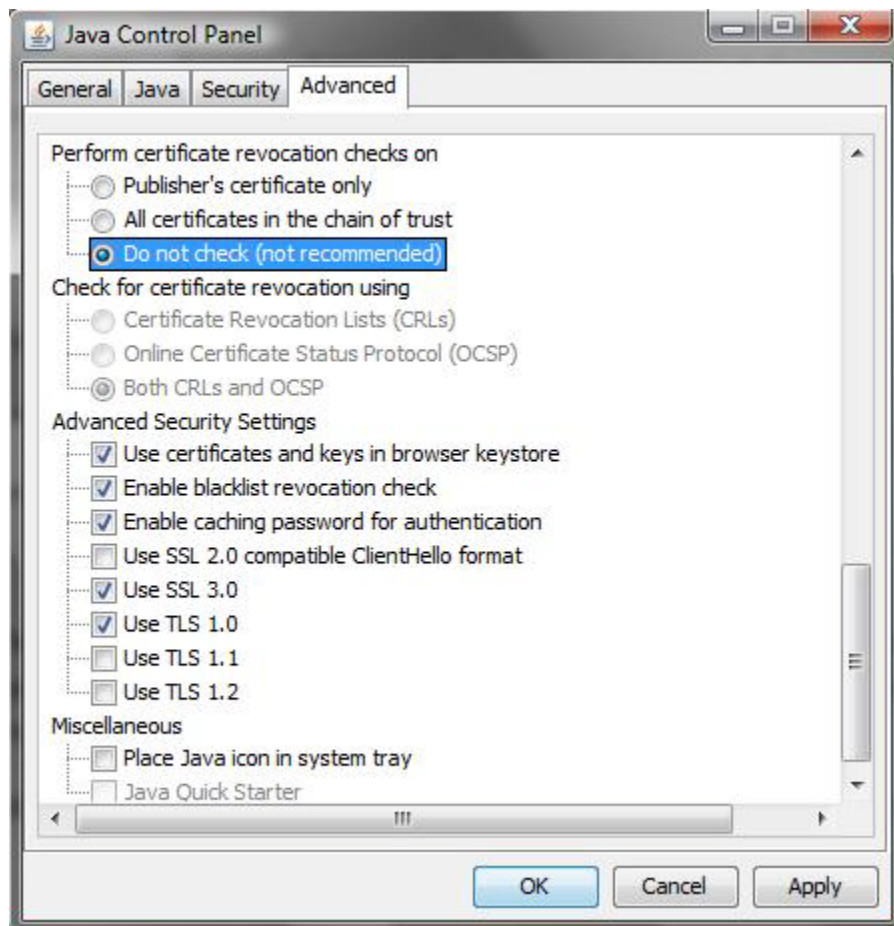
Figure 1.1   Control Panel

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 has changed. Files in the new format are delivered with JTLS-GO. 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 5.0.4.0, 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 5.0.4.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2017-12853 Parse Messages For Tabular Reports Framework

Summary of Model Change Request

Users expressed a desire to gather certain information en masse from the messages generated by the CEP. A couple of scripts were developed to gather the messages from specific checkpoints and invoke an XSL stylesheet to parse the messages for desired data. The result was a HTML page that could be viewed through the Apache server by the end-user. This ad hoc capability proved useful and there was a desire to leverage the work already done into a more formal approach.

Design Summary

The framework to generate HTML pages and provide them to the users via the Apache server already exists to support the ad hoc AAR queries. This same framework was used to implement the message parsing capability. A new script was written to get a scenario, checkpoints, and stylesheet from the user, gather the checkpoint messages, and invoke the stylesheet to parse the messages. A utility template was provided to perform common functions such as converting indices into vocabulary items, convert times from XML times to Date Time Groups, and convert decimal lat/lon representation into a text representation. Two sample stylesheets were provided that check for missile firing events and air defense engagement events.

2.2 JTLS-2017-12891 GIS Tool: Trim Nodes Over Portion Of Network

Summary of Model Change Request

The GIS Tool currently has the capability to trim nodes along an individual road/rail/river/shore barrier polyline, as well as over an entire network, given minimum arc length and maximum acceptable deviation constraints. Some users have requested an additional capability that allows them to choose a portion of a network and only trim the nodes off of that portion.

Design Summary

The user zooms and pans to the region on the map that contains the polylines to be trimmed. Next, the user right-clicks button on any part of the map away from the arcs and nodes. A menu will pop up at the mouse's cursor location and one of the menu items will be "Trim Nodes in Current View". When this item is selected, the "Enter Node Reduction Constraints" dialog will be displayed for the user to enter the minimum arc length and maximum acceptable deviation constraints. Clicking the dialog's Start button will start the process to trim the nodes of all polylines currently displayed on screen (note that if a portion of a polyline is displayed in the

current view, the entire polyline will still be trimmed). The existing trimming methodology is applied to these polylines.

2.3 JTLS-2017-12893 GIS Tool: Save Layers and Grids While Calculating

Summary of Model Change Request

Determining a terrain grid's type (e.g., open, forest, city, etc) and road coverage (good, poor or no road) is a computationally intensive process. Although there are many factors that affect the time it takes to determine these parameters per grid, it can generally be said that the time period it takes to calculate the parameters for 100 grids can be roughly measured in minutes, for 10000 grids the time to completion may be measured in hours, and for 100000 grids the time may be measured in days. Currently, the GIS Tool does not intermittently save the results, so if a catastrophic event occurs (e.g., power failure) all results up to that point will be lost. Therefore, it is desirable that the tool have the capability to intermittently save the results while the grids are being processed.

Design Summary

The tool processes the grids by terrain layer. When all of the grids for a layer have been processed, the tool will automatically save the results before proceeding to the next layer.

2.4 JTLS-2017-12921 ATO Parser Write Errors To Screen And Report File

Summary of Model Change Request

Retain all screen errors from the ATO Parser in an independent errors file on the server. This will allow the operator to forward the list of errors to air staff more easily. The suggested location for the errors file is a directory containing the errors reported by the ATO Translator (./ato/atot/ errors/). The name of the errors file should include the name of the ATO that was parsed.

Design Summary

Modify the Parser to establish a path and name for the errors file while running each Parse of an ATO.

Modify all routines that generate errors reports so the errors are written to two places: the screen display and a server errors file.

2.5 JTLS-2017-12932 GIS Tool: Functional Enhancements

Summary of Model Change Request

Three functional enhancements to the tool have been proposed and accepted for implementation:

1. Provide feedback to the user in terms of a running count of terrain layers and grids that have been processed as grid parameters are being calculated. Since these calculations may take hours, or even days, it would be useful to see how the process is progressing.

2. When the user creates a new terrain layer, provide a capability to auto-generate a name for the layer. There are situations where a JTLS scenario may require hundreds of layers, therefore an auto-naming capability would save the user the laborious task of typing the name of each layer.

3. Provide a capability for the user to find a terrain layer on the map. There already exists a search capability for networks, nodes and arcs. The user enters the name of the object and the tool places a red X over the first object that matches the name (for a found network the X is placed over the first node listed in the network). This enhancement adds terrain layers to the search.

Design Summary

Enhancement 1. Feedback provided through the console window. The grids are processed layer-by-layer. While the grids of a layer are being processed, a count of the number grids processed out of the total number of grids in the layer is updated and recorded in the console window. When all of the grids in the layer have been processed, a separate count of the number of processed layers out of the total number of layers is updated and written to the console window.

Enhancement 2. A Layer tab has been added to the Set Terrain Properties dialog. It contains a check box that turns auto-naming on when selected, and turns it off when deselected. The auto-generated name of a layer follows the following format:

```
<index>_<grid size><unit of measure>
```

where `<index>` is a four-digit index number assigned to the layer (the index number may have leading zeroes), `<grid size>` is an integer indicating the size of the grid, and `<unit of measure>` is DEG, MIN or SEC indicating that the units for `<grid size>` is, respectively, degrees, minutes or seconds. For example, if the user creates a new layer consisting of 5 degree X 5 degree grids, and the tool assigns it the index number 1, the name will be 0001_5DEG.

3. To find a terrain layer, the user clicks the Edit menu and then selects the Find Object... menu item. This causes the Find Terrain Object dialog to appear. The user enters the name of the terrain layer in the designated text field and clicks the Find button. If a layer with a matching name is found, a red X will appear over the center of the layer. The X remains over the layer until the user clicks the View menu and then selects the Refresh menu item.

2.6 JTLS-2017-12937 Extract DSA Detections From Messages Stylesheet

Summary of Model Change Request

During an exercise, the Intel Response Cell needed an efficient method to gather information detected by Recce missions. Since the AAR was not running, the only source of the information was the Non-Theater DSA Collection Reports.

Design Summary

The framework for mining messages for information already exists within JTLS 5.0, and a stylesheet was required to extract the requested information from the Non-Theater DSA Collection Reports. This stylesheet was created. In addition, since the report used some dual index vocabulary items (CSP CS Names) the stylesheet utility template that retrieves vocabulary items was expanded to allow dual indices. The reporting of the error in case of a failure of vocabulary translation was also improved.

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; however, because of the model enhancements included with JTLS-GO 5.0.4.0, no previously-identified STRs are considered valid. Errors identified for JTLS-GO 5.0.4.0 and corrected for future Maintenance releases in the JTLS-GO 5.0 series will be documented in this chapter.

3.1 JTLS-2017-12836 Road Arc Distances Generate SIP/SVP Errors

Roads from a GIS project were exported to JTLS format. The JTLS scenario that received the roads was a copy of the demsdbkor50 scenario. When the scenario was checked by the SVP, thousands of "651" Errors (road arc distance is less than the arc's great circle distance) were encountered.

The GIS Tool uses OpenMap's com.bbn.openmap.geo.Geo.distanceKM() method to calculate great circle distance. The SIP/SVP (as well as the CEP) uses the SIMSCRIPT routine GREAT.CIRCLE.DISTANCE to calculate great circle distance. Both Geo.distanceKM() and GREAT.CIRCLE.DISTANCE assume a spherical earth. Geo.distanceKM() takes the latitude and longitude of two points, converts them into 3D vectors, and uses vector algebra to find the distance. GREAT.CIRCLE.DISTANCE takes the latitude and longitude of two points and derives the distance by the spherical law of cosines.

While investigating this problem, a Java version of GREAT.CIRCLE.DISTANCE was discovered in the com.rolands.util.Distance class. The method, called Distance.greatCircle(), is literally a line-by-line translation of GREAT.CIRCLE.DISTANCE, with one small exception: for reasons unknown, one IF block in GREAT.CIRCLE.DISTANCE did not get entirely translated to Distance.greatCircle(). This STR adds the missing code to the method.

With the method corrected, a comparison was made between Geo.distanceKM() and Distance.greatCircle(). The test concentrated on points that were around 10 KM apart or less, under the assumption that the user will not want to trim nodes from a road network that resulted in nodes that were more than 10 KM apart - a reasonable trade-off between maintaining network detail and reducing computer resources. The results showed differences ranging from 100 to 400 meters. It should be noted that Distance.greatCircle() declared all real variables as floats. When they were declared as doubles and another comparison was made, the differences went down, in some cases down to single digits.

As a result, this STR keeps that modification in Distance.greatCircle(). At this point it is not known which method is more accurate, and this STR does not even attempt to address that issue. Given GREAT.CIRCLE.DISTANCE was created in 1983 and has been effectively used with each JTLS version release since then, and since it would be preferable that all the JTLS programs and tools

consistently use the same great circle algorithm, all calls to Geo.distanceKM() in the GIS Tool have been replaced with Distance.greatCircle().

3.2 JTLS-2017-12837 Magic Move Amphibious Unit Crash

A Magic Move order was submitted for a ground unit that was currently involved in an amphibious operation. The CEP immediately crashed.

The crash was caused by obsolete references to hexes at the end of the routine MAGIC.CANCEL.LIFT.DROP. These obsolete references have been removed and the routine functions correctly.

While testing, the Amphibious Operation Estimate incorrectly displayed a Pickup as an Assault and vice-versa. The definitions for Formation_Mission (AMPHIB_PICKUP and AMPHIB_ASSAULT) in the static vocabulary file did not match the integers defined in the Preamble for .PICKUP and .AMPHIB. They were reversed. The error was corrected in the static vocabulary file.

3.3 JTLS-2017-12838 MFP: Added Multiple SUB Checker

The Message File Parser did not check if multiple SUB numbers existed in the same block in a MTF file.

A Message File SUB checker class was created to examine SUB numbers to see if the same SUB numbers appeared in the same block in a MTF file.

3.4 JTLS-2017-12839 Airlift/Amphibious Time Estimates In SITREPs

The Situation Report order produced messages which contained time estimates for units to complete airlifts and amphibious operations. The resulting estimates were not correct for units that were part of an airlift, or were participating in an amphibious assault or pickup.

The code to provide these estimates had not been updated since the model was changed to use object tasks to control most major functions. Significant changes were made to the estimation code in order to properly compute and report these time intervals. The messages for situation reports now report accurate time estimates for units which are being airlifted or are part of an amphibious operation.

3.5 JTLS-2017-12840 Improve Execution Speed Reading Road Networks

The availability of new terrain data has provided better road networks to complement the standard scenario data. The improved road network data contain many more nodes and arcs, because the road structures are much more detailed and complex. Reading these larger data-sets into the model took an unacceptably large amount of time.

Code changes were made to improve the time needed to read the new road networks. These changes involved the use of data mapping (hashing) to speed up the time needed to access an

individual network node. When using the new methods, the overall time to read the demonstration-level road data dropped from over two minutes to a nearly instantaneous amount of time.

3.6 JTLS-2017-12841 Change Sensor On Board Satellite

Users requested the ability to change the sensor used on a satellite to detect objects on the JTLS terrain.

The order to manage National Asset Passes was modified to add a "Change Sensor" option. This option allowed the Controller to change the on-board sensor while the satellite remains in orbit. The code was easily modified to add this feature. When used, it generates a message to notify the Controller that the satellite characteristics have changed.

3.7 JTLS-2017-12842 Secondary Blast To Pre-Launch Aircraft Crash

During an Air-Ground attack against an enemy airbase, an air mission on the ground in a Pre-Launch posture sustained secondary blast damage. The model calculated both the pre-damage percent capability of the mission and post-damage capability as the number of aircraft in the mission divided by the number that had originally launched. Since the mission had not yet launched, this was a division by zero which caused a crash.

The code was modified to check if the number of aircraft launched is zero, and if so to use the number of aircraft assigned to the mission as the divisor in the percent capability equations.

3.8 JTLS-2017-12843 One-Way Amphib Boats Pickup Return To Shore

An amphibious pickup order was submitted for a ground unit that owned amphibious vehicles. The vehicles (aka small boats) happened to be one-way boats. The boats traveled to the formation, unloaded a portion of the ground unit, and then went back to the shore. As a result, the boats became just another combat system that needed to be transported to the formation by formation assets, thereby delaying the completion of the overall operation. The one-way boats should have remained on the formation with the delivered ground unit portion.

The logic that determines the disposition of one-way boats after delivering their loads always returned the boats to the asset owner. In this test pick-up case, the asset owner unit was still on shore. Instead, the one-way boats should always be given to the receiving unit. The receiving unit is either on the shore (in an assault operation) or on the formation (in a pickup operation). The code was corrected to give one-way boats to the receiving unit whether the operation is an assault or a pickup.

3.9 JTLS-2017-12844 Support Unit Owned SAM/AAA Site Crash

While using a scenario that contained an Air Defense (SAM/AAA) site with enemy air missions and Rules of Engagement (ROEs) set to Fire, a game crash occurred. The support unit-owned ADA site fired, and after firing it determined it should order more missiles from its owning unit.

As part of this process, the code ensures that the missiles are available from the owning unit before placing an instruction for a implicit convoy with a Combat supply request. A Combat supply request is the highest priority request. However, the implicit convoy code realized that the unit being asked for supplies was a support type unit. When receiving a requisition, support type units only make supplies available above their own requirements. This is different than the original check, which was made when the SAM/AAA site fired. This second algorithm determined no supplies were available and therefore a backorder for the request was made. This was a backorder for a target with a request type of Combat. The code that initializes the backorder request with the JODA always expects Combat requests to be going to a unit versus a target. It therefore attempted to access information from the target, treating it like a unit, which resulted in a crash.

The code was changed to submit the implicit convoy request as a Mandatory request. This type of request will cause the unit to provide the supplies even if it means giving up some of its own combat systems. This will not be an issue because we have already ensured that the unit has what is needed available as supplies.

3.10 JTLS-2017-12845 Return To Base Excessive Crew Time Shelf Life

The message for a mission returning to base because of excessive crew time did not have a shelf life specified.

A shelf life of 12 hours was specified for the message. This matches the shelf life for the message of a mission being canceled by player order.

3.11 JTLS-2017-12846 Assault Followed by Pickup Left Unit on Shore

An amphibious assault order was executed and then followed with an amphibious pickup order for the same ground unit. When the pickup finished, the ground unit remained on shore.

Note: An amphibious pick-up order for a unit that had not been previously assaulted was transferred to the formation successfully.

The problem occurred only when an amphibious pickup was ordered after the ground unit had been put ashore by amphibious assault. If the unit had never been involved in an amphibious assault, the pickup operation functioned normally.

When the logic checked to see if the operation was complete, it failed to reset an attribute of the unit that indicated the "last load" had been picked-up. Consequently, the attribute still held a value of .YES when the next amphibious operation order was submitted for the same unit. As a result, only an initial wave (aircraft or small boat) were sent to pick up the unit and then returned because the operation was deemed complete as indicated by the attribute .YES value.

The problem was corrected by the re-setting the unit attribute to .NO when any amphibious operation has completed.

3.12 JTLS-2017-12848 ATOG Crash Deleting Runway Cut

While executing a game with an ATOG service running, the ATOG service crashed when a Runway Cut was repaired in the model. This happened because the base entity of the ATOG Runway Cut object was never defined, though it was being set and accessed.

The ATOG Runway Cut entity was expanded to include the common ATOG base entity. This provides the Runway Cut object with a name and an object class.

3.13 JTLS-2017-12849 SVP Crash Different CSP For HRU/TUP-subHUP

SVP Warning 1219 specifies that multiple HRUs using the same HRU Unit Prototype (HUP) will have different Combat System Prototypes (CSPs) assigned. The multiple HRUs can come from different sources, but the CSP they use is always based on the CSP assigned to a Tactical Unit Prototype. An HRU explicitly defined in the database is assigned a CSP based on its parent unit's TUP. The database can also contain TUP-subTUP and TUP-subHUP hierarchies, which exist to define a breakdown of a unit. The subHUPs in these hierarchies are assigned a CSP based on the hierarchy's TUP.

A database contained an HRU CSP conflict between an explicitly defined HRU and a TUP-subHUP hierarchy. This resulted in a crash of the SVP.

The code attempted to write out the data for the warning with the assumption that the situation was caused by an HRU vs HRU conflict and did not properly treat the cases for the TUP-subHUP hierarchies. The code was modified to handle all possible combinations of conflict. The data file that defines corrective actions was also updated to allow the deletion of the TUP-subHUP record when appropriate.

3.14 JTLS-2017-12850 Ground Unit Delay Move Crash

The CEP crashed during a delay move by a ground unit in Lanchester Combat.

The CEP code called a routine that was removed because of the hex-to-grid upgrade throughout the code base.

3.15 JTLS-2017-12851 Merge Formation Crash

A Merge Formations order was submitted to merge two existing naval formations. The order executed successfully, but the model crashed, attempting to notify the player at the conclusion of the process.

The crash occurred because the logic tried to send the player message to the owner of the merging formation, which no longer existed. There was no pointer to the "first ship" in the merging formation to obtain the unit IP which caused the crash. The error was corrected by changing the code to send the message to the owner of the receiving (merged) formation.

3.16 JTLS-2017-12852 Supplies From Target For Air Mission Crash

The model crashed when an air mission went through its resource allocation event and the supporting unit was taking supplies from a target to provide them to the air mission. The logic of the code subtracted the supplies from the unit when it should have been subtracting the supplies from the target. Besides being wrong, since the unit's supply category had not yet been accessed, this resulted in a crash.

The code was modified to properly subtract the supplies from the target versus the unit. Following this fix, the code already accessed the unit's supply category and gave the supplies to the unit.

3.17 JTLS-2017-12854 Ship Joins Formation Without Participating

A naval formation was executing an amphibious assault operation. A nearby ship (not part of the formation) was ordered to join the formation with the intent of expediting the operation using its small boats and helicopters. The ship joined the formation, but did not participate in the operation.

The join formation logic did not invoke the correct subroutine to cause the ship to contribute its assets to the ongoing amphibious operation. To correct the problem, three new routines were written to initiate small boat waves and helicopter waves when the ship joined the formation. The new logic pertains to both amphibious assaults and amphibious pickups.

3.18 JTLS-2017-12855 DDS Log Hierarchy Context Sensitive Menu

In the Logistics Hierarchy context-sensitive menu, the "No support unit" option is enabled even if the selected unit already has no support unit.

Code was added to check if the selected unit already has no support unit, and to disable the option to select "No support unit" in the context-sensitive menu if it does not.

3.19 JTLS-2017-12856 Ground Move Issues For Units Firing Artillery

The movement of units on the ground was not functioning as expected when the unit was ordered to fire artillery. A moving unit was ordered to fire artillery, but at the time when the firing was to occur, the model began executing an endless loop. Similarly, a unit was given a Defend order with a non-zero value in the field specifying the "Rounds Before Moving". When the unit was ordered to fire artillery, with more rounds than limited in the defend order, then the unit would attempt to automatically move to a new location. This auto-move always failed because the unit was unable to find a suitable location.

When the model started to fire artillery while the unit was moving, it sent a message reporting that there was a problem. After sending this message, it immediately rescheduled the fire event. This sequence cause the model to be trapped in an unending loop. The code was modified to

stop the immediate rescheduling of the fire event. The fire events were changed to be rescheduled after any movement task was completed.

For units with a defend task, with a limiting value to the number of rounds before moving, the code was never able to find a suitable destination because the code to check the legality of the movement was not complete. Checks were added to ensure that the movement was legal, and the unit correctly moved to a new location after it had fired the specified number of rounds of artillery.

3.20 JTLS-2017-12857 Apache Server Status Page Exposed To All Users

The Apache server status page contains detailed information about the web server, including information about current hosts and requests being processed.

Optional password protection of the server status page was added, which can be set using the ICP. If the password is blank, then no password protection is enforced.

3.21 JTLS-2017-12858 DDS Poor Performance During Large Changes

During large changes, like hierarchy copy and data repository data migration, poor performance caused the DDS to take excessive time to process the updates. It was most noticeable in the Command Hierarchy.

DDS Map filters are only rebuilt at the end of a download or update, improving DDS performance.

3.22 JTLS-2017-12859 Filter Target Symbols With Unit Symbols

Symbols for some Targets were appearing in the Map Filter under "Land Units" -> "Unit Symbols".

The processing for Unit symbols was incorrectly adding the Target symbols and was corrected to remove adding Target types when processing ground Unit filters.

3.23 JTLS-2017-12860 Extremely Long Amphib Unit Pickup Action

When a formation picks up a unit, the process of picking up the fist load took an extremely long time.

The problem was tracked down to the canceling of the unit's implicit resupply events. Within that code, the model properly canceled the implicit resupply event, and then proceeded to return the non-delivered supplies to the providing unit. As the model put back each of the supply categories, it asked whether there were any backorders held by the providing unit. If there were, the process of fulfilling those back orders was accomplished.

In this specific case, there were several supply categories that were part of the implicit resupply, and for each one this computation was conducted separately. This turned out to be a length computation.

The logic was changed. All of the supplies are returned to the providing unit and then the model asks only once whether there were any backorders that could not be filled.

3.24 JTLS-2017-12861 Air Mission Package Reported Delay

Almost every Air Mission Package would report that it was delayed leaving the rendezvous point. Seconds later the package was complete and would proceed properly. This message would cause the user to worry about a non-existent situation.

The computation of determining when the package should leave the rendezvous point was closely tied to the computation of the launch time and travel time of all air missions within the package. Due to simple round-off issues, the reported situation occurred.

The situation was easily solved by simply telling the package to leave the rendezvous point one second after the missions were supposed to arrive at the point. This allowed for the travel time round off and when the LEAVE RENDEZVOUS Event executed, the package was complete and no package delay message was generated.

3.25 JTLS-2017-12862 Logic Error Initializing Routes

Logic errors were being generated each time a formation route or convoy route was being initialized.

The route initialization process copies the entity's current route to a temporary route and then adds planned future tasks for the entity to the temporary route. Once the JODA is notified of the route, the temporary route is no longer needed and is deleted. During this delete process, the model reported a logic error when it could not delete the temporary route from the JODA. The route should not be deleted from the JODA, and so the model appeared to be working properly.

The code was fixed to ensure that the routine that was attempting to delete the temporary route from the JODA was not called and the logic error is now not generated.

3.26 JTLS-2017-12863 Inadequate Help Menu For Several Orders

The help menus for the Delay, Enter Port, and Set Model Times orders, and the Delay Route utility, did not properly explain model function and what could be expected after execution of the items.

Help menus were altered to properly explain model function.

3.27 JTLS-2017-12864 Colocated Nodes Adding Network Branch

If two nodes were collocated when adding a branch to a network or creating a new network, the DDS would fail to add arcs after the one attached to the collocated node.

If two nodes are collocated during adding a branch, the second node is skipped.

3.28 JTLS-2017-12866 Mission Package Automatically Cancels Crash

The model crashes while attempting to compute the estimated return time for the missions in an automatically canceled Air Mission Package

An air mission that is canceled because it belongs to an automatically canceled Air Mission Package may still contain a Rendezvous Task and the estimated return code did not consider this possibility. The code to compute the estimated return time now considers this possibility.

3.29 JTLS-2017-12867 Air Mission Package With ASAP Start Refused

An air mission package order was given a start time of ASAP. The model was stopped and restarted from an earlier checkpoint. The air mission package order was refused on the restart with push, but it was accepted when manually submitted.

The problem was that the order had the start time specified as ASAP. When the order is entered into the game, a time of ASAP is given a time of TIME.V or the current game time in the system. On a Push, the order is read in at the time of the checkpoint from which the model is being started. Thus the ASAP time was assigned a time much earlier than the time it was when manually submitted.

The problem was solved by setting the ASAP time for a pushed order not to the current game time, but to the time at which the order arrived in the model.

3.30 JTLS-2017-12868 Amphib Pickup Loses Combat System

An amphibious pickup operation failed to deliver a combat system that was in maintenance to the ship. Before the pickup began, the combat system (OTHER-TROOPS) was in the Maintenance column in the IMT. When the operation completed, the combat system was not present in the Maintenance or Available columns in the IMT in the embarked unit. The embarked unit had "lost" one of its combat systems during the pickup operation. Note this anomaly did not occur in most amphibious operations.

When the combat system arrived at the ship, the logic assumed that the equipment item entity that holds the combat system of that type was already in existence. In this case, no other combat systems of the same type (operational or otherwise) had been delivered to the ship. The logic did not allow for the creation of the equipment item for the first time arrival of a combat system in maintenance. Instead, a logic error message was generated and the code that added the combat system was bypassed entirely.

In other words, the logic failed to allow for the delivery of a maintenance combat system before the arrival of operational systems. The situation occurred because the small boat wave carried only combat systems in maintenance. If the wave also carried operational combat systems, the error would not have occurred because the operational systems are always added first, which creates the equipment item that also holds in-maintenance systems.

To correct the problem, the logic error message was removed and the code was changed to create the equipment item when the maintenance combat system arrives for the first time.

3.31 JTLS-2017-12869 Delay Order Allowed Invalid Strength Levels

The Delay Order allows the specification of a strength level on a delay route that could fall below the threshold for a Delay Posture.

The CEP only accepts Delay Order route points where the strength level is above the TUP.THRESHOLD.DELAY.WITHDRAW for the TUP of the unit. If the unit's strength level is below the threshold, only the invalid delay route point is rejected. The order is still processed for remaining valid route points.

3.32 JTLS-2017-12870 GIS Tool: River Crossings Not Saved

One of the functions of the preprocessor is to determine where road and rail arcs cross river arcs. Where a crossing occurs, the road or rail arc's Type attribute is set to BRIDGE and the river's arc is flagged. The flag is necessary so that nodes attached to such arcs are not removed when the user runs the GIS Tool to trim river nodes. The results should be saved to the appropriate .dbf file. The road and rail arc .dbf files were updated but the river arc .dbf files were not.

The code to save the river arcs was missing and has been added.

3.33 JTLS-2017-12871 DDS Logistic Hierarchy Does Not Reload

While the DDS was running, the DCP was restarted. This wiped out the Logistic Hierarchy tree, and it could not be reloaded or a new Logistic Hierarchy started.

The code that integrated two different download methods was modified and proper handling for 'Data Reload' was added.

3.34 JTLS-2017-12872 CEP Crash during Amphibious Lift

The CEP crashed when moving and placing a unit during an Amphibious Lift. The unit has no Pending Execution Tasks, but the resulting message required a PET to determine the addressee.

The message providing feedback on the completion of unit placement determined the message recipient from the Player MIP of the unit, not any PETs the unit may/may not have.

3.35 JTLS-2017-12873 Duplicate Delete Calls In ATOG

The ATOG was displaying two error messages whenever a target was assigned or removed from pipeline support, which causes the CEP to delete the target with the JODA and then reinitialize it with its new owner, Force Side, and faction. The error messages specified that the ATOG was unable to find the object for deletion.

It was discovered that the ATOG was creating event handlers with the JODA twice. These handlers take care of the creation, update, and deletion of objects in the ATOG. Since there were two handlers, even though they were the same actual code, the first handler would delete the object and the second handler would be unable to find it and produce the error message. This was fixed by removing the duplicate creation of the handlers. While investigating this it was also discovered that the CEP was changing the target side twice, which resulted in the JODA object being deleted and reinitialized twice. This is why the ATOG displayed the error message twice. This duplication of action was unnecessary and was removed.

3.36 JTLS-2017-12874 DDS Command Hierarchy Reload Freeze

The 'Reload Data' function in the DDS Command Hierarchy stuck with a loop in the Logistic Hierarchy.

After GlassFish was restarted, a download flag for the supply category was reset to false and could never be set to true, because the Command Hierarchy was not interested in re-downloading the supply category.

When 'Reload Data' was selected, the Logistic Hierarchy rebuilt the tree but the download for the supply category meant that the rebuild would never end, locking up both the Command Hierarchy and the Logistic Hierarchy.

A static flag to end download for the supply category was added and set to 'true' after the first download is completed. Code that checks the flag to see if the download is completed before entering a loop was added.

3.37 JTLS-2017-12875 Amphibious Pickup Gains Combat Systems

An amphibious pickup operation completed successfully, but several extra combat systems were gained by the unit which raised its strength above 100%. Note this anomaly did not occur in most amphibious operations.

When the model determined that the last remaining combat systems had been picked up from shore, the empty unit's "flag" and icon were moved to the formation ship. Several waves were still inbound to the formation to deliver combat systems. The temporary receiving unit (detachment) already present on the ship was combined with the unit to await the inbound waves.

The logic then attempted to re-issue all combat systems based on the supplies present in the unit. Because some supplies that represented combat systems had been delivered earlier, those supplies were immediately converted to the equivalent combat systems and added to the unit. When the wave from the shore arrived carrying the same combat systems, they were also added to the unit. Hence, the unit had gained combat systems above its TOE.

To correct this problem, the called subroutine that re-issues all combat systems was removed from this process. It was deemed unnecessary to make that calculation at the time the last load

was picked-up. Also, calls to the routines that calculate the unit's TOE and combat strength were added at the end of the amphibious pickup or assault (i.e. after all waves of supplies and combat systems are delivered). Those calculations had been done before when the combat systems were re-issued and needed to be done at the end.

3.38 JTLS-2017-12876 Slow Down With Large Road Network

A new large road network was added to the demonstration database. The model slowed down more than expected.

The computation for the determination of whether a possible arc went outside of the defined allowable search region was improved.

3.39 JTLS-2017-12877 DDS Logistic Hierarchy Delete Record Exception

When deleting records continuously from the Unit Log Hierarchy table, the Logistic Hierarchy throws an exception.

Added code that does a check to an access index if it is not larger or equal to the array size.

3.40 JTLS-2017-12878 ASCII Checkpoint Hard-Coded Text For ACP/ATC

The code that saves the Air Control Prototype - Aircraft Target Class (ACP.ATC) parameter Auto Intercept Allowed was writing out the text YES or NO. While this is correct, the preferred method is to write the value from one of the fixed text arrays. These arrays are also used when reading in the data, removing any chance of a mis-match.

The code was changed to write out the parameter using the fixed text array value versus the hard-coded values. A double check was also made on the read code to ensure this is what was expected. That portion was correct except that the read code ensured that both the ACP and the ATC parameters were found. Nowhere else in the initialization or checkpoint read code do we check such things, as such an error indicates a corrupted database, so this "safety" check was removed.

3.41 JTLS-2017-12879 SSM Report Logic Error Message Issue

The SSM Report is generated when an SSM is preparing to fire and when an SSM is fired. In both cases, the code that reports who detected the preparation/launch generated a logic error and closed the message if the type of the detector was unknown. In both these cases (preparation and launch), the closing of the message was done improperly.

The code was fixed in both places to properly close the message.

3.42 JTLS-2017-12880 Repeated Glassfish Down Warnings

If the Glassfish server went down while the DDS login was displayed, the user was repeatedly presented a warning dialog.

The dialog code was modified so that it is only displayed once.

3.43 JTLS-2017-12881 DDS Unit Map Move Too Slow

There was a 4 sec delay between each unit move. This was to work around an issue with owned target positions not being properly displayed after a move.

JTLS-2017-12803 fixed what turned out the be the underlying cause of the target position errors. The delay was removed.

3.44 JTLS-2017-12882 IMT Negative Value For Troops In Amphib Pickup

Three small boats in an amphibious pickup wave were destroyed when they encountered an anti-invasion minefield. The combat systems and supplies carried by the boats were also destroyed. The IMT showed a negative value in the Available column for one of the combat system types carried by the small boat wave.

In the CEP routine that determines which combat system types were destroyed in the wave, an error in the logic deducted the losses from the Avail column (in the Equipment Item entity) for only one of the combat system types (in this case, OTHER-TROOPS). As a result, the Avail column for OTHER-TROOPS was decremented by the losses suffered by ALL combat systems in the wave combined. The total losses suffered by all the combat systems was greater than the number of OTHER-TROOPS Avail so the count became negative (-52).

The error was corrected by adding a subroutine call to access the Equipment Item corresponding for each combat system that was destroyed. The change ensured the losses were subtracted from the Avail column in the Equipment Item belonging to the correct combat system.

While testing this correction, it was observed that amphibious combat system losses belonging to the lifted unit in the wave were not tallied in the KIA column in the IMT. Also, those same losses were not included in the Unit Damage Report and the Controller Damage Report. All other types of combat systems lost in the wave (i.e. those combat systems carried by the destroyed small boats belonging to ships) appeared in the IMT KIA column and in both damage reports. Only the amphibious combat systems KIA and their crew KIA counts were missing. Note the Avail column correctly showed the remaining assets after losses.

The logic that updates the cumulative losses and creates the damage records for reporting purposes were omitted for amphibious combat systems that represented small boats owned by the lifted unit. The missing logic was added.

3.45 JTLS-2017-12883 WSM: SSH Tunnel Command Cannot Be Copied

When the Web Services Manager (WSM) failed to connect to the ssh tunnel, it displayed an error message that provided the correct command to establish an ssh tunnel; however, the displayed command and its arguments were too long to highlight or copy.

The code that displayed the tunnel error message was modified to allow the user to highlight and copy the correct ssh tunnel command into a terminal window.

3.46 JTLS-2017-12884 Inconsistencies With Road/Rail Nodes In Ocean

Road and Rail network nodes must not occur in an Ocean Grid. Road and Rail network arcs may cross ocean grids. Inconsistencies were found in enforcing this rule in both the Scenario Initialization Program (SIP) and the CEP.

The Scenario Verification Program (SVP) now checks the location of all rail and road nodes and provides an error if they are placed in an ocean grid:

- SVP Errors 660, 661, 657 and 660 were removed.

- SVP Errors 656 and 658 were simplified.

CEP checks during the Order MANAGE.MOVEMENT.NETWORK now ensure that this rule is consistent during road or rail network modifications.

3.47 JTLS-2017-12885 Manage Formation Task Order Inconsistencies

The Manage Formation Tasks order allowed the user to try and change the task number for the currently executing task 0. The order also allowed the user to attempt to change both the task associated unit and task associated target when it should be one or the other.

A constraint was added to prevent changing the location for the currently executing task 0, an error in the New_seq_number_specified constraint was corrected so that the user cannot change the sequence number of the currently executing task 0, and an error in the Both_Unit_and_Target constraint was corrected to prevent a change both the associated unit and associated target. Finally, a message was added to confirm a change in formation speed.

3.48 JTLS-2017-12886 Shore Artillery Failed to Engage Amphib Assault

An artillery unit did not fire on an enemy amphibious assault wave. The assault location was within the range of the unit's artillery combat systems. The ground ROE for the artillery unit was properly set. The unit had sufficient ammunition to carry out the engagement. The unit was not involved in an ongoing fire mission, nor was it involved in ground (Lanchester) combat.

Investigation confirmed that all necessary preconditions were met to permit the artillery unit to automatically fire at the incoming amphibious boats. Nonetheless, the unit's artillery combat

systems were prevented from firing because the model incorrectly concluded the systems were fully involved in Lanchester ground combat. Specifically, the combat system's Time Weighted Total, an attribute which tracks the remaining time in a combat assessment period, held a value of zero. Zero indicated none of the combat systems were available to direct fire because they were completely committed to ground combat. However, the unit had never been involved in Lanchester ground combat prior to this point, nor was it currently.

To fix the problem, the logic was changed to verify the unit is in a Lanchester battle and the artillery's Time Weighted Total is zero in order to stop the direct fire engagement. In addition, if the Time Weighted Total is zero and the unit is NOT in a Lanchester battle, the Time Weighted Total is re-set to a positive value based on the global Assess Combat Time value. The second code change was necessary because the logic expects a non-zero value to calculate the number of rounds to fire at the assaulting wave. Note these changes also apply to an amphibious pickup wave.

3.49 JTLS-2017-12887 Duplicate Sub Message Numbers

In the process of modifying a message for another STR, the programmer noted that one of the SUB message number had been duplicated, yet no errors were noted when the create_message script was run.

The problem of having two of the exact same SUB message numbers at the same level could have caused unpredictable results. The create_message script was modified to identify these possible errors. The first full run of the new script against all messages identified 5 problems. Each was fixed by correcting the message numbering, CEP code or a combination of both. The modified files are identified below.

3.50 JTLS-2017-12889 Player Orders Missing From Restart With Push

A user submitted several orders and then saved a checkpoint. The intention was to restart the game and push the same set of orders. This method should allow the user to repeatedly test the same conditions in the game. It was observed that some, but not all, of the saved orders were processed during the restart with pushed orders.

The restart and push of orders was found to be functioning properly. Some of the orders were not executed because they were being prematurely deleted by the model. For example, the player orders included a Magic Move Formation (MMF) followed by an Amphibious Pickup (APU) order. The Amphibious Pickup order was never processed during the push because it was deleted when the model processed the MMF order.

In several other cases, the model deleted some pending orders while processing the current order. Code changes were made to prevent these orders from being deleted during a restart/push. As is standard for this version of JTLS, the orders are now processed into tasks. These tasks may be canceled or postponed if the object is unable to execute the task. No extra deletion of orders is required since the task execution includes the required error checking.

3.51 JTLS-2017-12890 Controller Damage Report Underreports Losses

An amphibious assault wave encountered an anti-invasion minefield and was attacked by artillery simultaneously. The total small boat losses from both attriters were correctly shown in the ship's Unit Damage Report. However, the Controller Damage Report showed boat losses from only the artillery attack. Both reports correctly reported the combined combat system losses carried by the destroyed boats. Note that when only a single attriter (minefield or artillery) inflicted damage on the wave, the Controller Damage Report showed the correct small boat losses.

In this case, 13 boats were destroyed by mines, and 2 boats were destroyed by artillery in the same amphibious wave. Only the 2 boats destroyed by artillery were included in the Controller Damage Report, because of an error in the routine that reports small boats and combat systems lost for the engagement. The logic incorrectly assumed only one attriter per wave could inflict small boat losses (i.e only one small boat attriter could be present in the set of damaged objects for the ship). Based on that incorrect assumption, the code simply assigned the losses from the last attriter in the set to a temporary array for the report. The problem was corrected by accumulating the small boat losses from all attriters in the damage set.

3.52 JTLS-2017-12892 Statistical Errors in Air Mission Periodic Report

The periodic report for air missions was not properly reporting counts for some of the mission types.

Changes were made to fix the counts for missions delivering weapons in air ground attacks. Changes were also made to improve the code organization and to the format of the message. The summary data now appear first, followed by the counts for each service in the period report.

3.53 JTLS-2017-12894 Improve Log Reports By All JTLS Services

The following log report activities have been identified as necessary for good information assurance:

- *The Synapse service should report explicit logins and logouts for each WHIP.*

- *The JODA service should report client connections and disconnections.*

- *All services should report log events using Zulu time (GMT).*

- *All services not already doing so should report their own startup and shut down events.*

The described modifications have been made to the specified JTLS services.

3.54 JTLS-2017-12895 Inconsistencies In Pipeline Nodes And Targets

Pipeline network algorithms require the nodes, Supply Storage Area Targets, Pumping Station Targets, and Operating Units to follow strict colocation and flow rules. Inconsistencies were found in these rules in the Scenario Initialization Program (SIP). The CEP pipeline-filling code was operating inefficiently because it was not taking advantage of the rules set in the SIP.

The SIP was updated with mutually-exclusive checks for distances between nodes, pumping station targets, and supply storage area targets. The SVP was improved to enhance the DDS' ability to correct SVP errors and warnings on pipeline networks. The CEP algorithm for locating appropriate pumping stations during pipeline operations was also improved.

3.55 JTLS-2017-12896 Perceived/Detected Unit Has No Name

It was possible to detect a squadron on board a naval vessel without assigning it a name. The result is that the user could attempt to create an order that referenced an unnamed foreign squadron, causing several problems within the model.

The model now will not create a unit detection without assigning a perceived name to the unit.

3.56 JTLS-2017-12898 Patrol Polygon Not Shown With Object Routes

When an object was actively patrolling a polygon, the boundary of the patrol area was not visible on the WHIP when the object route was displayed. The patrol polygon was displayed on the WHIP when the object was moving towards the patrol area, but it disappeared when the object began the patrol task. This issue exists for patrolling air missions, ships, naval formations, and HRUs.

When an object begins to patrol a polygon, the points which define the polygon are held on the object task. In most tasks, the data held on the task are translated into route points for the movement of the object. In this case, the polygon points are not route points, but they should be displayed as such for visualization on the WHIP. Code changes were made to insure the polygon points were properly included in the displayable route.

3.57 JTLS-2017-12899 JOI OTH-Gold Service/Module Synchronization

The WHIP module and OTH-Gold service lacked the ability to synchronize filter settings between multiple WHIP modules.

The OTH-Gold service now holds the filter settings and informs any active WHIP modules of the current settings. The WHIP module interface was also updated to allow the user to connect directly to the service and monitor the connection status of the service. Prototype filtering was changed by splitting the prototype tables into tabs, as well as providing a filter/search mechanism.

3.58 JTLS-2017-12901 CEP Code Not i18n Compliant

Several places within the CEP code used hard-coded text strings for display in legacy Player messages. This implementation approach was not i18n compliant because the terminology was not accessible for translation to different languages.

The solutions applied depended on the specific situation. In most cases, the hard-coded text was removed from the CEP code and moved to the message template, which made the text accessible for language translation. In some cases, new sub-messages were created to accommodate conditional text display. Where applicable, new terms were added to the static vocabulary file so that integer values are passed in lieu of hard-coded text from the CEP code to the message processor and translated to the corresponding text in the message templates.

3.59 JTLS-2017-12902 GIS Tool: Arc/Node ID Field Not Set Properly

Arcs and nodes have an ID field holding a unique identification number of that object. Two problems were observed regarding the setting of this field:

- *When the user edits a node and selects the "Extend New Arc" option, a new arc is created with one end anchored to the edited node and the other end connected to a new node offset a short distance from the anchor node. It was discovered that the new arc's ID field and the new node's ID field were not set.*

- *When the user edits an arc and selects the "Insert Node" option, a new node is created and located at the midpoint of the arc. The arc, itself, is replaced by two new arcs linked by the new node. It was discovered that the new node's ID field and the new arcs' ID field were not set.*

There are Layers that allow the user to interact with the arcs and nodes. Each Layer is tailored to a specific type of network. For example, the RailLayer allows the user to only handle arcs and nodes belonging to railroad networks. When the user selects the "Extend New Arc" option the Layer's newArc() method creates A new arc and node. Likewise, when the "Insert Node" options is selected, the Layer's newNode() method creates new node and arcs. Both methods should use the ArcFactory and NodeFactory to generate, respectively, the new arcs and nodes.

It was found, however, that the ArcFactory and NodeFactory were not consistently used. In many cases the arcs and nodes were instantiated directly by invoking their constructors, followed by a series of set methods to set these objects' fields. Inevitably the call to set the ID fields were missing in many cases. The factories set various fields including the ID field. The affected Layers and methods have been updated so that only the ArcFactory and NodeFactory create the new arcs and nodes.

3.60 JTLS-2017-12904 WSM: Added Restart Option To Apache

Web Services had to stop and then start in order to restart.

A "Restart" menu item was added to the WSM popup menus. This function only applies to Services that are currently executing.

3.61 JTLS-2017-12905 Standoff Air Mission Unit Does Not Fire

An air mission, told to attack a specific object, travels all the way to the object before firing its weapons. Multi-Target Attack missions fire from the specified firing location and move in only as far as needed.

The problem was solved. The order to attack an object has a Time on Target (TOT) specified. This time is considered to be when the mission should arrive on the target. Since the model has no idea what weapon load will be selected when it computes the launch time for the mission, it has no way of automatically computing a firing location and making sure the mission arrives at the firing location at the specified TOT.

The solution does not change the launch time of the mission, it is assumed that relatively short range weapons will be used when performing an Attack object task. The problem was simply solved by having the mission attempt to fire from its current location when it executes the FIRE WEAPON task. The model determines the mission is not close enough, and that instant it computes the best firing location for the mission. The mission then moves to that firing location and fires all weapons in range. If more air-ground weapons exist, a new firing location is computed and the mission travels to that point and fires the weapons that are in range.

This process continues until all air-to-ground weapons are dropped or fired.

3.62 JTLS-2017-12906 Improve CEP Instructions For Removing Lock File

When the CEP detects a lock file from a previous or concurrently running CEP, the execution stops and a message about the lock file is displayed to the user. This message give little information about the lock file or how to remove it.

The message generated by the CEP was modified to more clearly explain how to remove the lock file if necessary.

3.63 JTLS-2017-12907 DDS Formation "Find Prototype" Exception

Right-clicking on a formation opens a context menu that included the option "Find Prototype". However, a formation does not have a prototype, and when "Find Prototype" was selected, the DDS threw an exception.

The "Find Prototype" option was disabled when the selected object is a Formation.

3.64 JTLS-2017-12909 OPM Does Not Display Aircraft Shelter Priority

On the Equipment Shelter OPM page, a table is displayed for each category listing the priority summary for each target subcategory. The Aircraft Class priority was always listed as None meaning No Data Available.

Normally subcategory data, such as Radius, is based on the Aircraft Target Class object. Users wanted increased detail and desired to have Shelter Priority based on Aircraft Class instead of Aircraft Target Class. The code incorrectly looked at Aircraft Target Class for which no data existed. The problem was solved.

3.65 JTLS-2017-12910 Crash Reading Supply Storage Target Supplies

The model crashed reading in database-specified supplies for a Supply Storage Area target.

The problem was fixed. There were no Supply Storage Targets with specific supplies in the Standard Database, so this problem was not identified earlier. Standard Database was changed to ensure the problem was properly solved.

3.66 JTLS-2017-12911 CEP Crash during Offload Unit Task

A CEP crash occurred after an air mission successfully completed a mission that encountered problems during an offload unit task. The Offload Unit task had been intentionally modified to change the offloading unit. The purpose of this testing was to address management of erroneous changes to offload unit task, typically due to a player MANAGE.MISSION.TASKS order.

The CEP code was modified to include air mission removal/cleanup of any lifting mission links or sets whenever a lift was unexpectedly halted. Messages were inserted into the Air Mission Report that explained the final disposition of the airlifted unit if the lift was halted. A Task Automatic Cancellation message was created to alert the player when an OFFLOAD UNIT task had to be automatically canceled by the CEP.

3.67 JTLS-2017-12912 Terrain Layers Not Properly Sorted

When the GIS Tool exports terrain layers to the .tl file, the layers should be sorted by decreasing grid size. This was not always the case.

The previous sort code has been replaced by a new GridSizeComparator that performs the sort correctly.

3.68 JTLS-2017-12913 Check-SVP Script Needs To Allow "or" Condition

The SVP-Template XML file had added a new attribute that included the "or" condition, and displayed an error message when the checksvp script was run, because the style-sheet did not have a process for the "or" condition.

An " or " process for @match in the check-action template in the stylesheet was added. "'or'" was also changed to " or " in the SVP-Template XML file, because it picked up words in which 'or' was included, such as "Aircorridor".

3.69 JTLS-2017-12914 Air Speed UOM Inconsistencies

Air Speed UOM inconsistencies were discovered in the CEP. Air Speed and any subsequent conversion must be strictly consistent with a starting UOM of KM per day.

The code base and documentation were investigated to ensure that Air Speed basic UOM consistently uses KM per day. Cases of incorrect conversions or documentation errors were found and corrected. The errors were due to much earlier versions of JTLS use of KM per hour as the basic UOM for Air Speed. An SVP check was created to ensure that there is a LOGFAS-required .UOM.AIR.SPEED entry with a short name of M_HR in the database.

3.70 JTLS-2017-12915 WHIP Launch Message Delay Reason Incorrect

When an Air Mission experienced a Launch Delay, the Air Mission Launch Message in the WHIP incorrectly provided the reason for the delay.

The CEP code incorrectly provided the reason for any launch delay as a "Fuel Delay". The launch delay algorithm incorrectly checked for zero fuel supplies, instead of any fuel supplies present. During a Launch Delay, if any fuel supplies were present, the delay reason always returned "Fuel Delay".

3.71 JTLS-2017-12916 Artillery Auto-Move Issues

An artillery unit was given a Defend order with the Rounds Before Moving field set to 10. A Fire Artillery order with Count Towards Move set to Yes was then sent to the unit. The order specified 10 rounds to be fired at three different aim points. The model generated three separate Fire Missions of 10 rounds each as expected.

When the first Fire Mission executed, a Counter Battery Status Report intended for the side that detected the fire was erroneously received by the firing side.

After the first Fire Mission completed, the artillery unit began to automatically withdraw to a new location to avoid counter battery fire. There was no start time displayed on the Task IMT. While en route, the unit unexpectedly executed the second Fire Mission while continuing to move.

When the unit finished withdrawing to its new position, it was out of range of the remaining Fire Mission, which was automatically canceled.

The Counter Battery Status Report received in error had been directed to the wrong Force Side. Instead of being sent to the ground function for the top unit on the detecting side, the message was sent to the ground function for the top unit on the firing side. The error was corrected.

At the same time, the Counter Battery Status Report shouldn't have been generated at all because the detecting side had a friendly relationship with the firing side and wouldn't have responded with counter battery fire. (The ground ROE is checked when the Request for Fire event is executed after detection). The logic was modified to only generate the message if the detecting side views the firing side relationship as enemy, suspect, or unknown at the time of detection.

The Withdraw order failed to initialize the start time attribute when the unit began moving along a specified route. This missing code was added to the Withdraw order routine.

The unexpected Fire Mission execution while moving occurred at the Hourly Unit Processing event. The logic failed to recognize the unit was moving and could not fire. The logic was corrected to first check the UT MOVING AUTOMATICALLY flag before scheduling the waiting Fire Mission to execute.

The algorithm that randomly determines how far the unit should move was modified to shorten the distance to twice the unit's radius to prevent the unit from moving out of range in most cases. Also, the Withdraw order logic was replaced with the more appropriate Move task logic.

The logic that checks for the presence of enemy units along a candidate move route before selecting the new location was improved. The old logic discarded the new location from consideration if there were any enemy units whose radii overlapped the moving unit's radius. The new logic instead checks the direct fire ranges of the enemy units at each route point. If the moving unit would be within the direct fire range, the new position will not be selected as the destination.

While investigating and correcting those issues, the logic that determines if the detecting radar is jammed was found to be incomplete. The missing code was added to reduce the effectiveness of the radar when jamming is present.

3.72 JTLS-2017-12917 Terrain Grid Calculation Speed Degrades

Determining the parameters of a grid (terrain type, road coverage and elevation/depth) is a computationally intensive process. It generally takes between 1 to 2 seconds to process a single one minute grid. When a sizable number of grids are processed the processing time per grid may start around the 1-2 second range, but as more grids are processed one after the other, the per grid time gradually increases. When tens of thousands of grids are involved the processing times eventually balloon to 2-3 minutes per grid. This is indicative of a memory leak.

A hash map used to temporarily hold information related to the terrain polygons in a grid was not being cleared after each grid was processed. This caused the map to increase in size every time a grid was processed and resulted in memory resources being used up. The fix clears the map after each grid is processed.

3.73 JTLS-2017-12918 ATOT Not Showing Service Port In Status Display

The status display for the ATOT does not show the service port information.

The routine that composes the status data for display was modified to add the service port number and label to this display.

3.74 JTLS-2017-12919 Jtlsmenu Reworked To Show Available Scenarios

When jtlsmenu showed the list of available scenarios on loading and unloading the database, the long name of a scenario, such as demsdbkor50, would run into the name of the next scenario. Additionally, there were no "quit" option for jtlsmenu scripts, except for the load and unload database options.

The jtlsmenu database options were modified to display the available scenario names so that they do not run into each other, and a "quit" option was added to each script.

3.75 JTLS-2017-12920 Problems in Requisitioning of Supplies by Ships

A considerable amount of computing time was being spent finding routes for convoys that were trying to resupply ships at sea. This issue often resulted in convoys which were stuck because they were unable to find a route to their destination.

Some new rules were set up to control how ships are allowed to requisition supplies. Changes to the code affecting the requisitioning of supplies by ships, and the delivery of supplies from support units from and port units, were made according the following rules:

When a ship is at sea, it can only take supplies from his support unit when the ship is within the implicit resupply distance from that unit. When a ship is in-port, it can take supplies from the port for non-personnel and non-combat system supply categories. For personnel and combat systems, those supplies can only come from the ship's assigned support unit(s). If the ship needs some supplies, but those supplies are not available for delivery to the ship, then these short-quantities are reported to the logistics player.

When a ship leaves a port, then any implicit resupplies will continue, but ongoing deliveries will be canceled. The amount of delivered supplies will be pro-rated based on the amount of time allowed for the delivery. Also, when a ship arrives in-port, any casualties and remains on the ship will be immediately transferred to the port unit.

3.76 JTLS-2017-12922 ATO-T Incorrect Slide Names For ACO Areas

The ATO Translator generates WHIP slides for each area defined in the ACO. The names of the slides should be the same as the names used for the areas in the ACO. The Translator is adding a prefix of "DSA_" to each slide name.

The ATO Translator routine for generating WHIP slides was modified to prevent this routine from adding the prefix "DSA_" to the slide name.

3.77 JTLS-2017-12923 WHIP/DDS Debug Log Not Printed To Console

When the Java Logger was used to log debug and error reporting in the WHIP, TRIPP, DDS, and JOBE on Windows, the output was not being printed to the debug Console.

The Console was modified to ensure that the debug and error reporting is logged correctly to the console on Windows.

3.78 JTLS-2017-12924 ATO-T Interface Issues and Improvements

Multiple issues and improvements were requested for the ATO Translator Module:

- *The protection radius field was only editable by the drop-down editor. The request was to allow operators to type directly into the field.*

- *When linking airbases, multiple JTLS units were able to be assigned to the same ICAO.*

- *Editing Target Alternatives was difficult because the length of many target type groups was larger than the area they were displayed in.*

- *Parsed ATO/ACO files were unable to be manually refreshed by operators. The only possible way was to open a new ATO Translator module.*

- *On the Mission Types screen, there were no selectable Jammer Types.*

The associated issues were solved:

- The protection radius field has been made into a fully editable field.

- Airbase links must now be unique. If an operator attempts to assign a unit that is already assigned, they are presented with a message and the link is reverted to the previously set value.

- Target Alternative editing is now done through a popup, rather than directly in the table. This allows for more space to view long target type group names.

- A menu item has been added to allow operators to manually refresh the parsed ATO and ACO file lists. A bug was fixed that prevented the ATO and ACO selection lists from updating with new files.

3.79 JTLS-2017-12925 ATO Parser Air Routing Reverse Course Indicator

The ICC indicator for Reversing the sense of an ACO defined path in F058 specified MTF was not considered by the ATO Parser. The Parser would ignore the indicator (RC) and simply build a route that follows the direction for which the path was created in the ACO.

The Parser was modified to detect the reverse indicator (RC) in the route records of an ATO. With the proper sense for each point of the route the forward or reverse is properly applied by the Parser is creating routes in the mission orders.

3.80 JTLS-2017-12926 IMT Incorrect Air Mission Scheduled Time

An Air Mission received a change to the end time for an orbiting task. The Scheduled Posture Change in the Air Mission IMT did not update the new posture change that would coincide with the new task end time. The Air Mission performed correctly, and the Air Mission Tasking Orders screen updated correctly.

The CEP code did not update the scheduled posture time in the JODA when the task end time changed. This was corrected.

3.81 JTLS-2017-12927 CEP Infinite Loop During Unit Move In Combat

A unit entered combat during a scheduled move. The unit correctly assumed a DEFEND posture and stopped the move. The CEP entered an infinite loop attempting to reschedule the move.

The CEP code rescheduled the attempt to move again immediately, rather than after the ASSESS.COMBAT.TIME days. The problem was fixed.

3.82 JTLS-2017-12928 Check For Unit And Support Unit Route

During tests with the standard database over long periods of game time, a number of convoys are generated which were unable to travel between a unit and its support unit. The convoys leave from the support unit, but are blocked from reaching their destination.

The Scenario Verification Program (SVP) was checking to confirm there were suitable network nodes near the involved units. However the check did not search for a suitable route for a convoy to follow between the two units. Additional checks were added to the SVP to ensure that there was a good route to follow between the unit and its support unit. If no route was available, then a new Warning 1456 was generated to report this problem.

3.83 JTLS-2017-12929 ATOT Converting Locations Errors

When the ATO Translator computes real number conversions between decimal and degree locations for Latitude and Longitude, there is occasionally a small 0.2 (degree) second error in the generated orders for some locations.

The ATO Translator reads ATO data, such as locations from an XML ATO file. The locations are converted inside the Translator and later written into order files. The default C++ precision for writing real numbers to a file is 4 decimal places. This was changed to 8 in order to maintain significant digits when the order files are written.

The initial XML ATO file that is read by the Translator is written by the ATO Parser, so this change was necessary for it as well.

3.84 JTLS-2017-12930 Small Island Terrain Grids Without Depth

If the tool classifies a grid as a Small Island but no depth is available for the grid, the tool sets the depth to -50. This needs to be changed such that if no depth is available, the tool re-classifies the grid as Open terrain with zero depth.

Only the method TerrainGrids.determineParams() needed to be modified to incorporate the revised logic.

3.85 JTLS-2017-12931 ATOT Not Using User Air Mission Times

The ATO Translator Reports warnings for air missions that start and end outside of the allowed ATO time period. The user may adjust the period allowed start and end times but the Translator does not properly adhere to the adjusted times.

The ATO Translator reads the user configured start and end adjustments and applies them to the ATO period, allowed start and end times for specific mission types. However, the Translator was only reading the minute value of the user supplied time. This caused little or no adjustment to the period time and the warning report persisted on subsequent translations.

The Translator was modified to account for the other portions (Days, hours and seconds) of time in order to correctly read the user configured time values.

3.86 JTLS-2017-12933 General Code Cleanup

Several routines had old, unused code and old variables defined.

A general search for old variables was conducted and the variables were removed from the source code. No functional changes resulted from this STR.

3.87 JTLS-2017-12934 Crash Damaging Associated Pumping Station

The CEP crashed when a pumping station associated with a pipeline node was damaged. The crash occurred in the generation of the Network Damage Report.

The crash was caused by the failure to set the local variable for the network and then attempting to access the network name from this unset variable. This error was fixed. During testing a number of other minor errors were noted and fixed. These include errors in the message format

for both the Network Damage Report and the Pipeline Node Damage Report, listing the damaged target multiple times in the Network Damage Report due to multiple rounds being accessed, and omitting Supply Storage targets associated with the network from both reports.

3.88 JTLS-2017-12935 Sealane Usage Too Restrictive

It was very difficult for a naval unit to believe the use of a sealane was in their best interest. Ships almost routinely rejected a sealane in preference for a direct route.

The sealane optimization logic was re-coded. The logic now works as follows:

The model determines if a straight direct route is even feasible. It may not be because it crosses land, goes through water that is not deep enough, or enters a grid with a mandatory sealand flag set to Yes.

If the straight direct route is not feasible:

- The model will use sealanes, no matter how far it needs to search.

- If no sealane exists, the move will start out as a direct route and stop when it runs into a problem grid.

If the straight direct route is feasible:

- The model computes an estimate of the time to travel the direct route, adjusting the time based on the grid's depth zone movement multiplier data.

- A new controller changeable data parameter has been added to the model. This parameter is called the SEALANE.COMPARISON.MULTIPLIER. This parameter is initialized to 2.0 but can be changed by the Controller using the SET.OPTIMIZE.ROUTE.PARAM Order panel. The larger the parameter, the wider the area the model will search for a feasible sealane.

- The model will search for sealanes as long as the time it takes to travel to the next node of the sealane is less than the SEALANE COMPARISON MULTIPLIER times the direct route estimated time.

- If a sealane path is found, the ship will use the sealane path. If no sealane path is found, the ship will start along the direct straightline path.

3.89 JTLS-2017-12936 CEP Memory Leak Clean Up

Several places in the CEP did not properly reclaim memory that was no longer used by the model. Over time, available system memory decreased.

The CEP memory leak errors were corrected.

3.90 JTLS-2017-12938 Message Lists Wrong Load Alternative

The Launch Capable Message for an air mission with a user-requested load would list the load alternative as the number of valid alternatives plus one.

An attribute on the mission load was added with JTLS 5.0 to keep track of which load alternative was chosen. The issue was that this attribute was not being set when the user specified a load. The code has been modified to set and report this attribute value.

3.91 JTLS-2017-12939 Targetable Weapon Name Not In Messages

A Ground Order Received Have Complied Message in response to an ordered fire mission was not printing out the name of the weapon that was to be fired.

The problem was tracked to code that was printing out the integer for the weapon with 3 digits. The scenario has over 1,000 weapons and the weapon that was fired had an index that was over 1,000. In this case the code writes out the index as *** since it can not fit it into 3 digits. When the WHIP Message Browser attempts to translate this to a weapon name, it fails and does not write out anything. The code was fixed to write the weapon name in 5 digits, and other places in the code with the same issue were fixed.

3.92 JTLS-2017-12943 OPM Minefield Clearing Rates Incorrect

The DDS Operator enters the number of mines that can be cleared in a frequency of mines per day. On the minefield definition page within the OPM the value is listed as mines per hour. Even after the conversion from mines per day to mines per hour, the values in the OPM are too large.

Within the OPM the code was taking the mines per day value from the DDS and multiplying it by 24.0 to determine the mines per hour. In fact the code should have been dividing the mines per day number by 24.0, not multiplying it. The implemented correction removed the wrong multiplier and reports the number as mines per day to maintain consistency with the DDS presentation. The column for the table in which the data appears was modified to state the rate was mines per day.

3.93 JTLS-2017-12944 Missing Messages In Message Browser Filters

A WHIP user brought up a Message Browser and specified in the filters that they desired only one type of message. In addition to their desired message type, they also had a Manage BE Facility Order Complied/Cannot Comply message in the Message Browser.

This has been a recurring issue when a new message is added to the CEP but is not added to the static_voc.xml file within the broadcast or directed message groups. It was felt that a robust approach was needed to not only correct this instance of the problem, but to also prevent the problem in the future with other messages and to maintain consistency between the Configuration Managed files.

Every message should appear within the static_voc.xml file as either a directed and/or a broadcast message. Every message should also appear in the whip_msg_alias.xml file, which instructs the XMS how long to hold onto and continue delivering various message types. To this end the message generation process was expanded to check that messages do appear in these two files, that they do not appear an excessive number of times, that the message title used in the two files is consistent, and that invalid messages do not appear in the files. After introducing the checks all inconsistencies were corrected.

3.94 JTLS-2017-12947 No Error Message When OPM Will Not Open

There is no error message to show that the OPM will not open because no default browser had been set.

An error message explaining this situation was added to the WHIP code.

3.95 JTLS-2017-12949 User Version Of NATO EnSitrep And OwnSitrep

A user requested that JTLS-GO use their versions of the NATO OwnSitrep and NATO EnSitrep messages. These are very similar to the configuration managed versions except with some minor character adjustments. Since these reports exist solely to support NATO exercises, it was felt that we should be delivering their versions of these reports.

The reports were integrated into the configuration managed baseline along with a small change to the NATO_C2_Unit_Posture vocabulary item where the JTLS Air Operations posture was previously translated to the text OPERATE but is now translated to DEFEND. This was because the text was used in two LC2IS fields, first in its full form of OPERATE and then as a subtask of OPER. The subtask was not a recognizable value for LC2IS.

3.96 JTLS-2017-12950 Change To Insert/Extract Task Rejected Crash

An Air Mobility mission was sent to perform a series of inserts and extracts of HRUs. Once the mission was created, changes to the delivery item list were sent to add an extract of an HRU. This order was flawed and was rejected by the model, but the model crashed during the generation of the rejection message.

The code that handled the generation of the message was used for both HRU Insert/Extract operations and for ARU airlift/airdrop missions. The code checked the mission's lift link attribute, which holds the ARU being airlifted/airdropped, and attempted to access its name. In the case of an Insert/Extract type mission, this results in a crash since the ARU link does not exist. The code was modified to take into account the non-existence of this link.

3.97 JTLS-2017-12951 Crash Scheduling Execute Naval Task Event

The user entered an order for a naval unit, to occur before the next task for that unit was scheduled to occur. The model canceled the event, telling the naval unit to execute its next task, and was attempting to reschedule the event for an earlier time when the crash occurred.

The issue was a simple coding error, which was corrected. A check of all code was made to see if any other examples of this coding error existed in the model. There were no other examples of this error, but several examples improper coding standards were found and corrected.

3.98 JTLS-2017-12952 Improper Message UOM Rendered By MDP

The Message Delivery Program (MDP) is inserting a category name for Units Of Measure (UOM), rather than the actual name for the Units Of Measure while it processes and sends logistics messages. The category also does not correspond with the configured UOM. Typically the messages are received with a category of NAVAL_SPEED in place of TONS, for example. It was also discovered that Date Time Groups were improperly written to the messages.

The MDP service uses an XML library to parse and apply a style sheet to the basic CEP-generated message. Whenever a UOM is to be inserted into the CEP message, the MDP looks up the correct UOM name from another library, which manages all configured UOM names and UOM categories. This UOM library was not correctly reading the UOM configuration file for the scenario. This was corrected in the library so the names are stored and accessed properly.

A problem was also discovered with another library used by the MDP. In the local UOM configuration for each instance of the MDP the library used the category name rather than the name for the UOM during initialization. This was modified to use the correct UOM name.

3.99 JTLS-2017-12953 Utility Delete Order Exception

If a player attempted to send or verify an order from the Order Group Editor that had a utility that had been deleted without first opening the order, a Java Exception would be output to the debug console, but with no other indication of a problem.

If the utility is missing, it is now left out of the order string, causing the OMA to reject the order and open the order panel, indicating the missing utility.

3.100 JTLS-2017-12954 Command Hierarchy Report Duplicate Name

The Command Hierarchy Report showed the Unit Short Name, twice instead of showing the short name and long name.

The report was corrected to show the short name and long name.

3.101 JTLS-2017-12955 OVP Checkorder Did Not Check Blank Content

The checkorder script did not check if <help><line> has blank content.

Added a blank check to the code.

3.102 JTLS-2017-12956 Ship Ignored New Sealane Arc

A new sealane arc was created using the controller Manage Movement Networks order. Then, a Naval Move order was submitted to a ship with a single destination that would cause the ship to incorporate the new arc in its network move. The ship did not use the new sealane arc. Instead, the ship took a much longer route using existing sealane arcs to the destination.

In the logic that establishes a sealane route, the depth of the arc is compared to the minimum depth required by the naval SUP. The depth of the new arc was zero, which caused the model to reject the arc from consideration. Obviously, the depth of the new arc should not have been zero. The problem was found in the routine that adds a new arc to the game. The arc's depth attribute was not initialized and therefore held a value of zero. Another routine that calculates the arc depth based on the shallowest grid depth along the arc's path was added to the routine to ensure the arc's depth is initialized with a non-zero value.

In the same routine, two other arc attributes (bearing and true distance) were found to be uninitialized. New code was added to compute and assign proper values to those attributes. Note these two changes also were added to rail, road, and river network arcs.

3.103 JTLS-2017-12958 Insert/Extract Missions At Incorrect Altitudes

When an insert/extract mission executed its primary task, the air mission failed to perform the task at the correct altitude. For extract missions, the aircraft dropped to a low altitude, completed the extract, then continued on to their next task at that low altitude. For insert missions, the aircraft maintained flight at the higher altitude as was used during its movement. The air mission never dropped to a reasonable altitude before performing the insert. Both of these cases occur for mobility missions which were ordered to perform an insert and/or extract from an airborne location.

For both the insert and extract missions, the order instructions allow the player to specify an altitude for the operation. This altitude was not used to control the mission altitude during execution. For the extract case, the mission reduced altitude before performing the extract. The model computed the altitude for this case by using the middle of the first altitude zone. The air mission properly reached this lower altitude for the extract, but it never returned to a normal flying altitude for the next movement task. For inserts, the altitude of the mission never changed to support that task.

Code changes were made to ensure that mobility air missions go to the order-specified altitude before conducting an insert and/or extract. If no altitude was specified on the order, then the model computes the altitude for these tasks using the middle of the first altitude zone. Once the insert/extract task has been completed, the air mission is returned to the altitude it had prior to the task.

3.104 JTLS-2017-12959 TBMCS-Linked Data Comprehensive Checks

When JTLS is connected to the Theater Battle Management Core System (TBMCS), the data held in the two systems must match for the connection to function properly. A number of additional checks were needed in the SVP to ensure that the data for both JTLS and TBMCS were compatible.

As part of the initial integration process with TBMCS, the SVP contained a basic set of data verification checks, but not all of the shared data were inspected for compatibility. To evaluate it with more completeness, checks were added to compare the data for squadrons, aircraft, missiles and their launchers. These additional checks support the data compatibility when JTLS and TBMCS are linked.

3.105 JTLS-2017-12960 TCU Failing To Produce New Grid-Based Terrain

The Terrain Conversion Utility (TCU) program was failing to produce the new grid based terrain files (tl and tl_gd) if the user did not have full scenario ascii file set but choose to modify a previously loaded scenario in the Oracle database.

During the modification process, the new scenario directory would be empty and therefore the default world map tl and tl_gd files would be used instead of properly generated ones using the legacy hx and ter_def files. The modification SQL script was modified to backup the legacy terrain data prior starting the modification process and save the legacy data into hx and ter_def ascii files under the empty scenario directory, so that the tcu.exe will find them during modification and properly generate the new tl and tl_gd terrain file pair.

3.106 JTLS-2017-12961 Objects Moving On Tunnel Arc Detected

Units, HRUs, and Convoys moving along a road/rail arc that is designated a tunnel arc were detected by foreign air missions, Units, and HRUs. Objects currently on a tunnel arc should be shielded from detection by such foreign assets.

New code was added to several routines in the CEP to prevent objects that are currently on a tunnel arc from being detected by foreign assets. The logic that determines which objects are covered by sensors carried by air missions or satellites was modified to ignore objects on a tunnel arc.

Also, the logic that allows tactical lookouts from a moving Unit or HRU to detect objects was changed to exclude objects on a tunnel arc. In the reverse situation, the logic that checks for other side detections of moving Unit, HRUs, or Convoys using tactical lookouts was changed to exclude those objects on a tunnel arc.

The logic that collects tactical intelligence for the Unit Tactical Intel Report and the Civil Affairs Report was changed to ignore nearby Units or HRUs if on a tunnel arc.

Finally, the Controller Area Report collection logic was modified to exclude Units and HRUs that are on a tunnel arc.

3.107 JTLS-2017-12962 DCP, ICP Caps Lock Warning

The DCP and ICP did not display a warning to the user when Caps Lock was on, which might lead users to enter an incorrect password.

A Caps Lock warning was added to the DCP and ICP interfaces.

3.108 JTLS-2017-12963 Improve OMA Condition Check Message

The failure message generated by the OMA when an object fails its condition checks was confusing. These condition checks are part of the XML order definition file. The purpose of the message is to indicate that the checks had failed and provide the user with text which described the checks. These checks are related to the status of the object, such as whether it is active or not.

For binary checks both sides of the condition are listed separated by the conditional operator (== for instance). For unary checks the unary text was listed followed by the condition (IS-TRUE FARP18.ACTIVE for instance). For clarity condition should be surrounded by parenthesis and rendered as IS-TRUE(FARP18.ACTIVE).

The parenthesis were added to the unary checks in the OVT Library.

3.109 JTLS-2017-12964 Air Mission Invalid Perception Prevents Release

An OAS mission was tasked to attack a ground unit. After completing its ingress route, the model calculated where the mission should fly to fire weapons based on the weapons' range and the perceived location of the ground unit.

While enroute to the firing position, the ground unit moved to a different location. When the mission arrived at its firing position and started to execute the Fire Weapon task, it correctly determined the ground unit had moved and detected its new location some distance away. However, the new location of the ground unit did not update on the WHIP map. Also, the firing position for the mission did not change to take into account the new location of the ground unit. Consequently, the mission did not create a Move task to fly within range.

Instead, the mission repeated the same Fire Weapon task from an invalid firing position. As before, the location of the ground unit was correctly detected but the mission did not relocate. This process repeated until a random draw caused the ground unit to not be detected, at which time the mission canceled the Fire Weapon task and went home.

When the new location of the ground unit was detected by the mission, there was no logic to update the perception of the mission's force side. The WHIP map did not update because the

perceived location of the unit had not been updated by the model. A missing subroutine call was added to correct the problem.

At the same time, there was no logic to calculate the mission's new firing position based on the newly detected location of the ground unit. The model continued to attempt to fire weapons from the old firing position based on the old location of the ground unit. The missing logic was added to update the Fire Weapon task location so that the mission executes a Move task to the new firing location before attempting to execute the Fire Weapon task.

3.110 JTLS-2017-12965 CEP Crash On Restart Due To bar_pt File

The CEP crashed upon restart from a checkpoint. The problem occurred during the reading of the Barrier Point checkpoint file data. The code that writes the Barrier Point checkpoint data only expected 999 points or less in a single barrier. A barrier with over 999 Barrier points was written incorrectly.

The CEP Checkpoint write code was corrected to allow 10 integer digits for the sorting number of barrier points.

3.111 JTLS-2017-12966 Target Destruct Message Misdirected

A ground unit was ordered to destroy a facility target which had an associated unit from a foreign side. The unit accepted the task and moved toward the target. Upon arriving, the unit could not find the specified target. The task was canceled and a message was sent to the target's associated unit WHIP terminal instead of the ground unit's WHIP terminal.

The message was sent to the target's associated unit interface program in error because the wrong subroutine call was used. The call was replaced with the correct subroutine call, so that the tasked unit's WHIP message browser received the message. The same error was also present when the unit was tasked to repair a target and could not find it, and this error was corrected as well.

At the same time, it was discovered that the messages concerning the status and estimated repair time of the target (when successfully destroyed by the ground unit) were sent to both the destroying unit's and the owning-side's WHIPs. Again, the wrong subroutine call caused the messages to be sent to both sides, and the call was changed to send only to the target's associated unit WHIP message browser.

After the ground unit completed the target destruction task, the target side and Controller side views on the WHIP updated to show the destroyed target. However, the ground unit's side view did not show the target was destroyed. The model had failed to update the target perception for the side that destroyed the target. The missing update was added to correct the problem.

The ground unit that destroyed the target did not appear on the target side WHIP after completing the target destruction task. However, the target side received messages that

identified the destroying unit. This inconsistency was corrected so that a unit update was sent to the target side to display the icon on the map.

When a ground unit was ordered to destroy a target that had been destroyed earlier by the same unit, the task started executing with a blank end time and never completed. The problem was the unit's perceived strength of the target had not been updated at the time the unit destroyed the target. The missing code was added so that the task is canceled and the player is notified.

3.112 JTLS-2017-12967 Automatic Supply Calculation Corrections

Several issues were discovered during testing of the updated Automatic Supply Calculation process.

The following changes were made to the Automatic Supply Calculation process:

- A busy cursor was added to display while calculations are in progress.

- The carry and storage capacities of the prototype can be adjusted if needed so that it can hold its basic load.

- Unit calculation data were added so that they are committed to the database.

- An issue that would cause units to be listed when "List prototypes" was selected was corrected.

- Improvements were made to error message logging.

- Default values and tool tips were added to the user interface.

3.113 JTLS-2017-12968 Faction Side Change Order Not Working

The order called Faction Side Change allowed the Controller to change the Force Side for one or more units. A patrolling naval unit was ordered to change sides. The order was processed, but not all aspects of the ship data were correctly represented in the JODA. The object continued to move but it had no route and no tasks. A number of JODA errors were also generated.

The order to change sides affected the faction leader and also any subordinate units, including owned and associated targets, owned HRUs, and any subordinate air missions. Each of these object types were tested while this problem was addressed.

Most of the data errors occurred because of changes in how object data are saved by the CEP and then updated on the JODA. To process the side change, the objects are deleted and then re-created on the JODA. If, for example, the delete was not done for the correct side perception, then the subsequent creation caused errors. A number of code changes were made to fix the numerous JODA errors and to correctly initialize the objects on the new force side.

3.114 JTLS-2017-12969 DDS SVPR File Access Issues

The method the SVPR was using to access files on the Glassfish server was causing a Java Exception on the server.

The file access was modified to use the FileCache to access files.

3.115 JTLS-2017-12970 Consolidation Of EventDelegator Code

A copy of EventDelegator code existed in a number of projects.

The EventDelegator code was refactored to the commonui_jar project so only one version exists.

3.116 JTLS-2017-12971 Glassfish Server Error Logging

When the Glassfish server would return an error code to the DDSC or TRIPP, the cause of the error was not logged in the server log.

Logging of errors was added to the server log.

3.117 JTLS-2017-12972 New Glassfish Domain Unexpected Messages

The messages returned from executing the commands to build a new Glassfish domain are expected to be in a specific format. Sometimes, due to errors running the command, the returned value was not in the expected format, causing an error in the DCP.

The DCP was modified to handle messages that do not conform to the specific expected format.

3.118 JTLS-2017-12973 Apache 2.4.25 Upgrade

The JTLS installation of Apache was not the latest version.

The Apache server was upgraded to the latest version (2.4.25).

3.119 JTLS-2017-12974 Glassifish 4.1.1 Upgrade

The JTLS installation of the Glassfish server was not the latest version.

The Glassfish server was upgraded to the latest version (4.1.1).

3.120 JTLS-2017-12975 DDSC Classification Update

Previously, the DDSC started a timer when launched to retrieve classification data from the server every 10 seconds. This was done in case the classification data might be changed. There was another timer created also for the DDSC to set the relevant information including the classification in the DDSC window's title bar every 10 seconds. This was very inefficient because the classification data was rarely changed in the database.

In stead of re-retrieving the classification data every 10 seconds, the DDSC monitors the classification data for updates only; and reset the DDSC window's title bar for updated information when the classification data changes.

3.121 JTLS-2017-12976 Arcs Disappear After Nodes Trimmed

After a road network was trimmed of nodes, the reduced network was missing numerous arcs.

When a network is trimmed, it is broken up into polylines and each individual polyline is trimmed. When nodes are trimmed from a polyline, a new set of arcs have to be generated that connect the nodes in the reduced polyline. The ArcFactory is responsible for generating these arcs and it is supposed to assign a unique ID number to the new arc.

The ID numbers were not generated properly, and occasionally an ID number would be generated that matched an already existing ID. This caused a new arc to replace an existing arc and caused the gaps in the network. A check has been implemented in the ArcFactory to insure any generated ID number is unique. An examination of the NodeFactory revealed a similar problem and a corresponding check has been put in place.

3.122 JTLS-2017-12977 Arriving Unit Does Not Have Basic Load

Units arrive at game start without their full basic load. The SVP does not raise this as a error or warning.

Units should normally arrive in theater with their full basic load, and if they do not, they should have an Initial Support Unit to draw from when they arrive. The SVP has been modified to provide two separate warnings. The first warning identifies units who arrive in theater without their full basic load and have no Initial Support Unit to draw from. The second warning is for the Initial Support units that do not have enough supplies on hand to give to units they are responsible for.

3.123 JTLS-2017-12978 ICAO Code Has Too Many Characters

Attempting to update the NATO ICC database with 5 character International Civil Aviation Organization (ICAO) codes from JTLS caused problems.

By definition, ICAO airport codes are always 4 alphanumeric characters. In JTLS any Unit type, Airbase, Squadron, FARP and Naval Unit that can support air operations may be assigned an ICAO code. Occasionally a database builder will assign ICAO codes that are not exactly 4 characters and do not actually exist.

A new warning has been added to the SVP to identify ICAO codes that are not exactly 4 characters. This will aid the database builder in identifying incorrectly-formatted ICAO codes and determining if that may have an impact on any external system, such as ICC or TBMCS.

3.124 JTLS-2017-12979 Techtool Copy Order No Relative File Paths

The Techtool copy order function required absolute file paths.

The Techtool code was modified to allow either absolute or relative file paths.

3.125 JTLS-2017-12980 ATOT DSAs Not Set For Periodic Collection

The ATO Translator generates DSA orders for the CEP during the translation of each ACO and ATO messages. When the Translator writes fields for DSA orders, it uses selected fields that configure the resulting DSAs to be one-time collection areas.

The ATO Translator was modified to include the specific order fields during order generation, which will configure the DSA order for multiple collection, with a frequency of 15 minutes between collection looks. With the DSA order configured using these fields, the resulting DSA areas will persist after the first data collection and until the end of the ATO period.

3.126 JTLS-2017-12981 HTTP Resource Closing Improvements

Issues in how some HTTP resources were managed may cause higher-than-desired system resource usage.

How HTTP resources are managed and closed was modified in JTLS.

3.127 JTLS-2017-12982 Whip Map Resizing Issue

When the WHIP map window was resized, the size of the filter panel was increased, rather than the map itself.

The WHIP map resizing code was corrected.

3.128 JTLS-2017-12983 XMS Does Not Remove Unit With Subordinates

The XMS received an update from the JODA that attempted to delete a unit. A full unit hierarchy for all sides is held by the XMS for look-ups such as with key words. In the case of this update the XMS would not delete the unit and simply reported that the unit would not being deleted due to the associated subordinates. Units would generally only be deleted if they had no subordinates. Stopping the delete left the XMS' hierarchy chain unmodified.

The XMS was modified to allow units to be removed without giving a log warning. However, any subordinates of the removed unit are changed to become direct subordinates of the parent to the removed unit, if any.

3.129 JTLS-2017-12984 Air Mission Message Not Sent To WHIP

When an air mission was created, a message confirming the launch was not received by the player who ordered the mission.

A message was generated which indicated that the mission was launch-capable. This message was sent to the WHIP which had authority of the home squadron. A minor code change was made to send this message to the player which ordered the mission. Subsequent tests confirmed that the message was received by the interface programs responsible for both the squadron and the air mission.

3.130 JTLS-2017-12985 Techtool Order Copy Improvements

Several improvements needed to be added to the Tech Tool Order Group Copy utility.

The various improvements implemented include:

- The use of the scenario provided when starting the Tech Tool as the Copy To scenario.

- Two new menu items added to disconnect the current connection and to disconnect and clear the fields (a full reset).

- Adding a reload button to retrieve any new order groups added to either scenario while connected.

- Improved Error/Warning messaging and printing warning and error messages in the message window of the Tech Tool.

- Improved performance in handling the copies, by copying orders as a batch in one open connection versus opening a connection per file copy, then closing before copying the next file and adding a SFTP capability to allow file access to any networked machine that can be secure shelled in to create directories and retrieve files remotely.

3.131 JTLS-2017-12986 Ability To Change World Map

The shape files for the world map in the DDS and WHIP were not able to be changed.

The ability to specify the location of the world map shape file was added.

3.132 JTLS-2017-12987 OGE Unable To Delete After Filtering

After filtering orders in the Order Group Editor (OGE), the delete order function was nonfunctional.

The issue that was causing the orders to not be deleted was corrected. Note that there is still an issue with the display not updating correctly.

3.133 JTLS-2017-12988 OGE Unable To Delete Orders After Drag/Drop

After rearranging orders via drag and drop in the Order Group Editor were unable to delete orders.

Corrected the issue to allow delete after drag and drop.

3.134 JTLS-2017-12989 Improve Ability To Size/Hide Map Side Panels

The divider and arrow buttons between the map and filters is small making it difficult to resize or hide the panels.

Added a larger divider and arrow buttons.

3.135 JTLS-2017-12990 OGE Exception Prevention

In JTLS 4.1, there was a case of a non-alphanumeric character in the reference field causing an Exception when opening the order in the Order Group Editor.

While this issue has not been observed in JTLS-GO 5.0, null checks were added as a precaution.

3.136 JTLS-2017-12991 WHIP Map History Shared Among All Instances

The WHIP map history was being shared among all map instances, causing inconsistent behavior and leading to Exceptions after maps were closed.

WHIP code changes were made so that each map instance has its own history.

3.137 JTLS-2017-12992 Order Tabs Data Incorrect Indication

When opening an order with tabs, the indication of which tabs have data was incorrect until the tab was selected.

Code was corrected to properly indicate which tabs have data when the order is opened.

3.138 JTLS-2017-12993 OGE Utility Context Menu Missing Functions

The Order Group Editor context-sensitive menu for Utilities had "send" and "check" options, which had no function.

"Send" and "check" were removed from the context menu when processing a utility.

3.139 JTLS-2017-12994 TRIPP Time Displayed Incorrectly

When the TRIPP time slider was near either end, the time was partially cut off.

The positioning of the time display was changed so that it displays correctly at all slider positions.

3.140 JTLS-2017-12995 DDS Server Tracing Capability

Additional capability to trace DDS server execution was required.

Added the ability to turn on DDS server tracing from the Glassfish admin console. This is intended for developer use only.

3.141 JTLS-2017-12996 Improved WHIP/DDS Startup Time

The Menu bar initialization would create duplicate menu items.

The WHIP/DDS menu bar was modified so that menu items are not created more that once.

3.142 JTLS-2017-12997 Improved DDSC Command Hierarchy Copy

When the DDSC performed a Command Hierarchy copy, a stored procedure called "copy_my_record" was used to copy each unit and/or target; and afterward ensured that the hierarchy of the copied records would mirror that of the originals. This two-stepped approach was necessary because "copy_my_record" duplicates many fields from the original records, including the higher headquarters/owning unit. Although this worked, this approach was cumbersome and inefficient.

The process is changed so that instead of using the "copy_my_record" stored procedure, the DDSC now directly insert the records in the appropriate tables, going down the hierarchy with the desired parent specified so there is no need to reparent.

3.143 JTLS-2017-12998 Systems Inappropriately Cross Barrier

A unit was able to cross a river, even though the mobility attributes indicated that the unit should not be permitted to cross. Controller orders were used to make the barrier un-crossable for some systems in the unit.

The Mobility Counter Mobility Prototype (MCP) parameters controlled the time it took to cross the barrier and whether such crossing were permitted. The order to set MCP parameters did not correctly assign the delay time. Additionally, the code used to compute the time for the unit to move across a segment on the route did not correctly identify the MCP characteristic which determined whether a barrier could be crossed. Code changes were made to correct both problems such that units now cross or do not cross barriers as expected.

3.144 JTLS-2017-12999 Disappearing DDSC Blinking Layer

Occasionally, when the DDSC was launched, the blinking layer was removed from the Map. When this happened, if the user turned on the blinking filter for Map objects, the corresponding filtering would no longer work.

The code is changed to prevent the Blinking Layer being removed.

3.145 JTLS-2017-13000 ATOT Incorrect Order File Names

When the ATO Translator creates each of the order groups for the WHIP, it adds the names of the contained orders using the full SYNAPSE permission name for the order file. In other words, the system file name for each of the order files contains a WHIP name and dollar sign as a prefix to the file. This is not necessary for the WHIP to discover each of the orders in the order group.

The ATO Translator was modified to store just the necessary portion of the file name for each order contained in the group.

3.146 JTLS-2017-13001 Air Mission Offload Supplies At Wrong Location

A Mobility Mission order was submitted to execute an Air Transport Mission to deliver supplies. The Mission Instructions specified geographic coordinates (not an airbase) to land and offload the supplies. The mission loaded the supplies and then failed to fly to the specified coordinates. Instead, the mission offloaded the supplies at its current airbase and terminated.

This problem did not occur if an airbase/FARP or naval vessel is specified in the Mission Instructions.

The problem was in the routine that determines whether the mission is at the desired location to perform the offload task. The logic failed to properly handle a task that specified geographic coordinates instead of an airbase unit. The logic incorrectly used the mission's current airbase location when no offload airbase was specified. Consequently, the routine concluded the mission was already at the desired offload location and unloaded the supplies at its current airbase.

The logic was changed to use the latitude/longitude specified in the task when there is no associated airbase specified in the task. This change now causes the routine to conclude the mission is not at the desired task location and needs to fly to the specified coordinates.

3.147 JTLS-2017-13002 Postponing Single Path Orbit Task Crash

If an air mission is executing a Single Pass orbit and the task is postponed for any reason, when the mission attempts to restart the Single Pass orbit, the model crashes.

The problem was that the single pass orbit task does not remember the tasked route once the task has started. The logic was changed to save the task route when the task is postponed.

3.148 JTLS-2017-13003 Ships Can't Move Between Dual Capable Grids

Ships are being stopped from moving from one dual capable grid to another dual capable grid.

The model incorrectly determined that there was an implicit shoreline between the two grids. The logic issue was corrected.

While testing this fix, it was also discovered that the model did not consider a land grid that has a depth but no elevation a dual capable grid. This problem was also fixed.

3.149 JTLS-2017-13004 Incorrect Contamination Type In NBC Report

When contamination occurred in an area, the model generated one or more NBC Contamination reports. These reports always indicated that the contamination was Nuclear even when Chemical weapons were used.

The code to generate the NBC Contamination Report used a static (fixed) value to specify the type of contamination reported in the message. Changes were made to ensure that the report contained the proper type of contamination when the message was generated.

3.150 JTLS-2017-13005 5.0.0.3 Documentation Updated

JTLS 5.0.0.0 documentation needed to be updated to reflect changes in system function and requirements.

The JTLS 5.0.0.0 documentation suite was updated to reflect changes and code corrections.

3.151 JTLS-2017-13006 Rejection Message For Accepted Order

A Mobility Insert/Extract Mission was en route to drop-off an HRU. A Manage Air Mission Tasks Order was sent to change the HRU in the Offload Task to a different HRU. The order was processed and the Offload Task showed the updated HRU name in the IMT. However, a rejection message was generated that implied the requested change was not accepted.

In the routine that processes air mission task changes, the HRU name change was properly executed. Next, the logic attempted to make the same change for a JDPI name, but incorrectly referenced the HRU name again instead of the JDPI variable. Because a JDPI cannot be carried on a mobility mission, the attempt to modify the Offload Task was deemed invalid and the rejection message was generated. This simple typographical error was corrected. Note this change also corrects the inability to change the JPDI name for other tasks, such as Fire Weapon.

3.152 JTLS-2017-13007 Unit Damage Report Inconsistent Messages

An error was reported by the XMS for message number 7500, the Unit Damage Report. The error indicated that a message was sent as a broadcast message but it should have been sent as a directed message.

The unit damage report is generated in several places in the model. In most instances, this message was sent to the player as a directed message. However, in one case, this message was being sent as a broadcast message. This error occurred for the mayday message that was sent by a sinking ship. For consistency, the code was corrected such that a directed message was used in all places where message number 7500 was created.

3.153 JTLS-2017-13008 Inaccurate Grid Calculation Infinite Loop

When a target owned by a naval unit was moved as part of the move unit procedure, roundoff caused an error in computing the grid from which the unit was to be moved.

A more accurate method of deciding the location grid of the owned target was implemented.

3.154 JTLS-2017-13009 OVT Library Functions Overflow

The OVT library has a built-in hash table for storing objects received from the CEP. When the CEP provides an object reference number (JEDI) that exceeds the maximum value for the hash index variable, the functions used to access the object from the hash table fail. The OMA service, among others, uses the OVT library and had reported the hash index failure in its log after running the game forward for several days.

The OVT library functions which use the hash index variable were modified to hold a value large enough to support any JEDI number from the CEP.

3.155 JTLS-2017-13010 Civilian Air Order Feedback

When selecting the 'Generate Orders' button on the Civilian Air Missions screen of the ATO Translator, there is no feedback provided informing the user if the order generation was successful.

The program was never designed with a notification window stating the orders generation had completed. The software has been updated to provide a simple pop-up window when the order file generation has completed.

3.156 JTLS-2017-13011 ICAO On Wrong Force Side

Two air missions in an ATO had a 4 character ICAO called 'MRCS', one departing from MRCS and one arriving at MRCS. The first mission (departing from MRCS) translated without issue; however, the second mission was deleted.

The ATO Translator should not assign an ICAO code to a mission unless it is on the same Force Side. The source code has been modified to require the departure or arrival ICAO to be on the same Force Side as the air mission.

3.157 JTLS-2017-13012 AMSNLOC Record Processed Incorrectly

When processing the AMSNLOC record for ALERT missions, erroneous error messages are generated.

ATO message records must be processed in a fixed format, taking into account optional fields that may or may not be there. In this case, the processing of the AMSNLOC record was incorrect for ALERT missions. The code was modified to correctly process the entire record.

3.158 JTLS-2017-13013 Inappropriate Road Type In Ocean

OCEAN terrain has a POOR ROAD type. It should only be considered NO_ROAD type.

The terrain type and road type of terrain grid (32, 11) of the default WORLD_LEVEL terrain layer file (accessed by the JTLS GIS Tool) has been changed to OCEAN and NO_ROAD, respectively.

Warning 1900 has been added to the SVP. It will notify the user of any OCEAN hexes that have a POOR or GOOD road type associated.

3.159 JTLS-2017-13014 Transfer Mission Difficult To Manage

The Transfer mission current design has the transfer taking place at the end of the specified Ingress route. If no Ingress route is specified, the transfer takes place as soon as the mission starts to execute. This makes it difficult to control and in fact can cause problems if the new home is too far away and fuel is needed.

The logic was changed to solve these issues. If no Ingress Route is specified, the model automatically creates an Ingress route half way between the mission's current home location and the new home location. After the mission is past the half way point it will transfer to the new location and created any needed detachment.

Thus if the mission is canceled prior to reaching the half way point, it will return to its original location. If it is canceled after passing the half way point, the mission will return to its new location.

3.160 JTLS-2017-13015 Consider Airbase Fuel Incorrect

When an air mission needs fuel, the model gives priority to finding an airborne tanker. If an air mission is not air-to-air refuelable, or there are no airborne tankers available, the model determines if it makes sense to find an airbase. The logic to do this was not as good as it could be.

The improved logic determines if the mission is in the middle of a tactical operation for which stopping at an airbase makes no sense. Refer to the *JTLS-GO Analyst Guide* for a complete description of the logic and examples.

3.161 JTLS-2017-13016 Can Equipment Be Sheltered Crash

A unit was forced to withdraw, including the possibility that some things were vacating an Equipment Shelter, and killed its stationary unprotected targets before the enemy could overrun the location. The model crashed because it was incorrectly attempting to do two things at once.

There is a set of objects in JTLS called the SURROUNDING OBJECT SET. The model was using this set to determine if there were any friendly units in the area to protect the stationary targets. It was also using the set to determine if there were any objects that could enter the Equipment

shelters because of the vacated items. This dual use caused the crash. Logic was added to ensure this situation was properly handled.

3.162 JTLS-2017-13017 Air Mission Posture Set To Zero

An air mission was told to change its mission type while it was shadowing another mission. This is perfectly legal, but the end result was that the mission lost its SHADOWING posture, causing a crash.

The mission posture should have remained SHADOWING. The logic error causing this situation was corrected.

3.163 JTLS-2017-13018 Order Push Pass Intelligence Does Not Occur

The new Manage Pass Intelligence Order is not saved to the Critical Orders File. The result on a replay involving a Push of orders, the intelligence information passed from one side to another is not executed. Note that restarting the model from a checkpoint after the order has been entered works just fine. This error only affects replaying along with the pushing or orders.

The order file was changed. The order is now a part of the Critical Orders File.

3.164 JTLS-2017-13019 Magic Move Unowned Mobile Target Crash

Users created a mobile SAM target during a game, but simply associated it with a unit. After creation, the exercise audience wanted to move the target, and the Controller created a Move Target external event. The model crashed attempting to pick up the target from the game board.

The problem was caused by an incorrect assumption that a mobile target will be owned by an ARU or an HRU. This was not the case and it is perfectly legal; although unusual, to have an unowned mobile target. The code was corrected and the incorrect assumption removed from the Target Move External Event.

3.165 JTLS-2017-13020 Resting Task Incorrectly Canceled

If the user cancels a resting task, the model does not properly cancel the event that would stop the task automatically. The end result is that the model will crash when the automatic cancellation occurs.

The resting task is now properly canceled. No changes were needed for the postponement of a resting task, since they cannot be postponed.

3.166 JTLS-2017-13021 Crash Entering Withdraw While Moving

While a unit is moving, it may be stopped when it comes in contact with enemy units. It is possible that the longer the unit stays in the battle, while attempting to move, the greater the

likelihood that it may need to go into a delay or withdraw. When this happens, it is possible for the unit to obtain two move events. This situation can lead to a crash.

The logic error was corrected and under the specified circumstances, the moving unit does not get two move events scheduled.

3.167 JTLS-2017-13022 Terrain Masking Infinite Loop

An air mission that was flying close to Greenwich Mean Longitude in a larger grid was being detected by a radar that was north of the air mission. The radar was in a smaller grid and exactly on the grid line. This unusual set of circumstances caused a problem with the algorithm that collects grids between two points. The algorithm never stopped attempting to calculate the grids to be crossed.

The algorithm was improved to account for this situation.

3.168 JTLS-2017-13023 TRIPP - Object Attack Report Error

Accessing the Object Attack Report from a context menu threw an exception. Additionally, if the object was a target, the damaged object type was incorrectly set to "Unit" in the order panel.

The TRIPP menu definition file was incorrectly referring to the damaged object type group by its label. References were changed to the correct group name.

3.169 JTLS-2017-13026 Crash Recovering From Checkpoint

The game crashed coming up from a checkpoint as an HRU deployed its lookouts and came across a convoy that had no trucks. The crash occurred when the code attempted to reserve an array to hold the various vehicle assets of the convoy, which resulted in an attempt to reserve a zero dimension array.

The HRU was the result of a detachment from a support unit. The detachment logic splits assets, including vehicles, between the two units. It attempts to provide the detaching unit with all vehicle assets from available, i.e. non-deployed convoy, combat systems. If this does not fulfill the detachment's full vehicle requirements, the code then attempts to transfer convoys over. If transferring full convoys will provide excessive assets to the detachment the code will then split convoys to provide the assets.

Before splitting a convoy, the code should have checked if the convoy had the required assets. In other words, there would be no point in splitting a convoy with only tanker trucks when the detachment needed cargo trucks. But this check was not being done, and convoys were being split that did not have the necessary vehicle assets, resulting in newly created convoys without any assets. This has been fixed by performing the check to determine if the convoy has the necessary assets before splitting it.

3.170 JTLS-2017-13027 AAR Report Orders Fail Schema Validation

Several AAR Report Orders were failing schema validation because they contained obsolete fields or were missing the spreadsheets element.

Obsolete fields were removed from the AAR report orders and spreadsheet elements were added where missing.

3.171 JTLS-2017-13028 ATO Viewer Displaying Other Player Missions

A Red player was viewing the full mission profile on the ATO Viewer for detected Blue Missions.

The ATO Viewer's filter was changed to disallow detected air missions from being shown in the display. In addition, compile time warnings were cleared out.

3.172 JTLS-2017-13029 Incorrect Text Variable Checkpoint Write

Within the DDS database, the value for the variable ACP.MT.DEFAULT.OPTIMIZATION can be assigned the value "OPTIMIZED" or "FLIGHT_PATH". This was read in and assigned by the CEP to an integer, depending on if the value was equal to "FLIGHT_PATH" or not, never checking if the value was "OPTIMIZED". When writing a checkpoint, the value was either written out as "DIRECT_PATH" or "FLIGHT_PATH". This did not affect the read of the checkpoint, since the code only checks if its "FLIGHT_PATH" or not, but it did affect the ability to load a checkpoint back into the DDS.

The code was modified to use the standard text array read methodology, which verifies that the value is valid. The array that held the values was also wrong, using the value "DIRECT_PATH" instead of "OPTIMIZED" and this was also changed.

3.173 JTLS-2017-13030 Air Mission Does Not Leave Rendezvous Point

A legal air mission package formed up exactly on time, but never left the rendezvous point.

The problem was caused by an extremely accurate calculation in when the package should leave the rendezvous point. All missions had arrived, but none had actively gone into an air hold state. They needed to move for the first time after their arrival to put them actively in an air hold. When it was time to leave the rendezvous point, the model determined the missions were not executing a task, so the Air Hold task was never canceled.

The logic was changed so the Air Hold task is canceled whether it is or is not executing at the time the package should leave the rendezvous point.

3.174 JTLS-2017-13031 Hasty Defense Consumption Not Set

When a unit is in a posture of hasty defense, its supply consumption parameters were not properly set.

A unit in Hasty Defense will continue to consume supplies as if they are in a moving posture. They will start to consume supplies using the Defend posture data as soon as the Hasty Defense is complete.

3.175 JTLS-2017-13032 Restart Stopped Unit

A unit stopped for gas and submitted a request for fuel. Before the fuel arrived, the user canceled the unit's move task. When the fuel arrived, the model crashed. The unit could not find why it needed the fuel since it held no tasks.

The logic was improved to consider this possibility.

3.176 JTLS-2017-13033 Perceived Side For Ground Mission

The model could not determine the perceived side for an air mission on the ground. This generated a logic error. The model simply said the perceived side was the true side.

The model can now indicate the perceived side for an air mission on the ground.

3.177 JTLS-2017-13034 ICP User Set JVM Max Heap Size

There was no way to easily set the Java Virtual Machine's (JVM) maximum allowable memory usage (heap size) by the WHIP and TRIPP.

Users can now set the JVM's maximum heap size in the Interface Configuration Program (ICP) under the Options menu. Given WHIP and TRIPP memory requirements, it allows the heap size to range from of a minimum of 1 GB to a maximum of 10 GB.

3.178 JTLS-2017-13035 National Asset Pass Satellite Problem

A user attempted to manually create a satellite using the National Asset Pass (NAP) order. The orbit route contained just two points: the first location was for a time in the past and the second point corresponded to a time in the future. No satellite was created on the JODA. When the user tried to add points to the route, the order was rejected because the satellite did not exist. When the user tried to create a duplicate satellite, the order was rejected because the model said that a satellite already existed with the same name.

The initial order to create the satellite should have been rejected because the satellite route must have a minimum of two route points in order to create a valid object in the JODA. The code associated with the NAP order was modified to prevent these problems from occurring. If an order is received with insufficient route points, then the satellite is removed from the game and an error message is reported to the controlling workstation. The modified code also prevents the case where a satellite exists in the model, but no such object exists in the JODA. These changes were done in the CEP, and no changes were required in the JSAT service or in the JSAT GUI.

3.179 JTLS-2017-13036 Infinite Loop In Routine

A routine was given a polygon in which every random point generated was not in the polygon, and therefore it remained indefinitely stuck in a loop.

The routine RANDOM.POINT.IN.POLYGON would become stuck in an infinite loop when one side of a polygon is on a meridian (the longitude coordinates of its end points are the same) on the western-most or eastern-most sides of the polygon. The correction to this problem required a modification to CONSTRUCT.SPHERICAL.POLYGON that added a check for a side on a meridian. If this is the case, code was added to insure the subsequent side adjoins the previous side.

3.180 JTLS-2017-13037 DDSC Table Columns Ordering

When certain tables (such as Force Side, Force Side Relationship, Squadron, or Naval) are shown, the ordering of their columns can be random, rather than according to the order of columns specified in their respective screen definition files.

The code was fixed so that the orders in which the columns are defined in the screen definition files are preserved when the tables are shown in the DDS client.

3.181 JTLS-2017-13038 Invalid Mode 3 Squawk Entry

An order to change a Force Side's Mode 3 squawk was rejected, because the starting block number for Mode 3, 0000, was already in use.

The value 0000 for Mode 3 is reserved for use as a null value. The first number assigned to any block of Mode 3 numbers should be 0001. A new SVP error was added to notify the database builder of this problem.

While researching and implementing this solution, a error was discovered in the reporting of the number of Modeling Parameter Errors and Warnings generated by the Mode 3 verification routine. It was necessary to rearrange the order of processing to correct the problem.

3.182 JTLS-2017-13039 OTH-Gold Report Command Failed to Update

The OTH-Gold interface allows the user to modify the Report Command of the message. However, this field was failing to update when modified by the user.

A fix was added that causes any updates made to the Report Command field to be accepted.

3.183 JTLS-2017-13040 Force Side Attribute Messages

The messages for Force Side attributes are improperly worded.

Several changes were made to the message text associated with Force Side attributes changes, such as periodic report times, Mode 3 block numbers, and removal of convoy symbols.

3.184 JTLS-2017-13041 Unit Load Canceled Causes Crash

The model crashed when the user canceled a Unit Load task. The mission doing the load was the only aircraft lifting the unit.

There was a logic error in that two different routines tried to put the unit back on the ground. This logic error was fixed.

3.185 JTLS-2017-13042 Mission Stops When Canceling Base Fuel Task

When the user used the Magic Air Ops order to magically give a mission fuel, and then canceled the mission's Base Fuel task, the mission would not move again.

The resulting situation happens by design. If an air mission cannot get home, or to an appropriate base, without getting fuel, the mission simply stops. This is by design, so the user can fix the problem. The user community thought this was better than having the mission crash because of lack of fuel.

However, the mission had a problem because it thought its next refuel location was at 0N 0E. When the base fuel task was canceled, its next refuel location attributes were incorrectly set to 0N 0E. Now, when the task is canceled, the mission looks at its task list and determines the exact location that it will expect to get fuel.

3.186 JTLS-2017-13043 Checkpoint Mission Inventory Owner Crash

The model crashed on a checkpoint attempting to write out the mission's list of supplies being carried.

When an air mission is told to lift supplies, it arrives at the load location and takes supplies from units and supply storage targets in the area. It keeps track of the object from which it took the supplies, so if the user cancels the missions, the supplies can be given back to the correct unit.

The crash was caused by a mission that had taken Food Supplies from two different units and a Supply Storage target warehouse. A stop checkpoint was taken and all worked as expected. The model was restarted from this stop checkpoint, and the next time a checkpoint was taken, the model crashed attempting to write out the mission's supply list again.

The code to read in the stop checkpoint caused the mission's supply list to become corrupted. The read problem was fixed.

3.187 JTLS-2017-13044 Logic Error In Gather Line Grids

A logic error was being generated in Gather Line Grids when the set of grids into which the line grids were to be placed was not empty. The model cleared the grid set, generated the logic error, and then continued to gather the grids covered from one location to another.

Although the automatic solution of clearing the grid set worked correctly, the JTLS programming standard calls for correcting all logic errors as they are found.

Two places in the code were found in which the grid line set was not cleared. The first was in the routine that computed jammer transmission loss, and the second was found in the routine used by naval units to compute the best grid to enter when shadowing a foreign ship.

Each of these issues was corrected by clearing the grid set when the computation was complete.

3.188 JTLS-2017-13045 ICC Damage Report Unknown Weapon

The damage report sent to ICC did not contain information concerning which weapon caused the damage.

The weapon information was made available to the ICC Damage Report.

3.189 JTLS-2017-13046 Controller Killed Aircraft Crashed CEP

An airdrop unit mission was flying to the pickup location to load a ground unit. While en route, a Controller Kill Aircraft order was submitted to destroy four out of the seven aircraft in the mission. The model crashed attempting to cancel the mission.

The model crashed because the cancel airlift/airdrop logic tried to remove the mission from the Lifting Mission Set when the mission was not in the Set. The mission was not in the Set because the mission was en route to the initial pickup location. The cancel logic had failed to consider that the mission might be in such a situation. The logic was changed to check for the presence of the mission in the Lifting Mission Set before removing it.

3.190 JTLS-2017-13047 Interceptor Avoided ADA Site Unnecessarily

A DCA mission was intercepting a foreign side mission. The DCA mission entered an AIR_HOLD posture when it determined that an ADA site was a threat at the mission's next move location. However, the DCA mission's force side had no knowledge of the ADA target type (i.e. SA-2F battery), other than its location. The DCA mission did not have sufficient intelligence information to conclude the ADA site was a threat and therefore should have proceeded to its next location.

The SAM/AAA avoidance logic that applies to intercepting missions failed to consider the current intelligence detection level for the ADA target before assessing the threat based on the target's characteristics. At the Localization and Classification levels, the ADA target subcategory is not known. That information is not realized until the detection level reaches Recognition. The logic was changed to allow missions to proceed if the detection level is Localization or Classification (Note that missions are always allowed to proceed if the force side relationship is viewed as Neutral or Friendly). If the detection level is Recognition or Identification, the attributes of the ADA target (i.e. perceived strength, range, and PH/PK lethality), as well as the presence of jamming or terrain masking, are considered when assessing the threat.

3.191 JTLS-2017-13048 ICC Naval Unit SAMs Failed to Create

SAM units in ICC that have Naval Units as parents were failing to be created. This was because the Naval units were created in ICC using the Naval Unit's UIC as a key. The SAM was attempting to retrieve the Naval unit using the Name Prefix instead of the UIC.

SAM units with a Naval parent will now refer to their parents by UIC.

3.192 JTLS-2017-13049 Interceptor Continually Breaks Off Intercept

A DCA mission continually intercepts and breaks off an intercept, because the intercepted mission moved out of the protection zone.

There were really two problems uncovered. First, the DCA mission was given an Ingress Route to its orbit area, but was allowed to intercept on its way to the orbit location. This is normal. The problem was that the mission's reference point, around which the protection zone is drawn, was not set until the mission started to head towards the orbit location. When the break-off decision was being made, the mission was looking at a reference point of O,O. This problem was solved by setting the reference point as soon as the mission takes off.

The second problem was that the logic to determine whether an air mission could legally be assigned to intercept was using the mission's current location instead of the mission's reference point. This problem was also solved. The assignment logic and the continuation of intercept logic are now identical and the mission does not get into the observed situation.

While testing, we also discovered that a mission that becomes a DCA mission was not getting an assigned reference point. This problem was also solved.

3.193 JTLS-2017-13050 SITREP Crash Querying Convoy Status

A WHIP operator put in a SITREP request for a transported unit and the model entered debug.

The simulation entered a section of code where it was accessing the variable UT.BEING.TRANSPORTED to provide a proper message for the requested situation report. The value of the variable was not one that was expected, this caused the CEP to crash.

Unable to precisely duplicate the problem, a default LOGIC.ERROR message was added to notify systems personnel should this happen again, and added a message to the WHIP operator explaining the situation.

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.

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

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

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

APPENDIX B. Version 5.0.4.0 STANDARD DATABASE CHANGES

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

APPENDIX C. Version 5.0.4.0 DATABASE CHANGES

No changes were made to the JTLS-GO 5.0.4.0 database.