JTLS-GO

Version Description Document

October 2020

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.0.1.0)

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

JTLS-GO 6.0.1.0 is a Maintenance release of the JTLS-GO 6.0 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as some minor model functionality improvements implemented as Engineering Change Proposals (ECPs), These ECPs 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 6.0.1.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.0.1.0 is a Maintenance delivery for the JTLS-GO 6.0 series of releases.

JTLS-GO 6.0.1.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations called "wespac60". Database modifications that were accomplished to upgrade the previous JTLS-GO database format to the JTLS-GO 6.0 series data format were summarized in the VDD for JTLS-GO 6.0.0.0. No data format changes were made between JTLS-GO 6.0.0.0 and this Maintenance release.

Detailed descriptions of some minor Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0, and all software corrections are summarized in Chapter 3.0.

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 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

1.2.2 Unchanged Documents

- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 6.0.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.0.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.0.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.0.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.0.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.0.0.0)
- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 6.0.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.0.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.0.0.0)

- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.0.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.0.0.0)

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.0.1.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.0.1.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.0.1.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.0.1.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.0.1.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.0.1.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.0.1.0)

1.2.4 New Documents

The US Government has required JTLS-GO to produce a Department of Defense (DoD) Architecture Framework (DoDAF) document for the entire system. The DoDAF provides a foundational framework for developing and representing architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures across organizational, joint, and multinational boundaries.

It establishes data element definitions, rules, and relationships and a baseline set of products for consistent development of systems, integrated, or federated architectures. Each program or process delivered with JTLS-GO has been included in this new document. The *JTLS-GO DoD Architecture Framework*, JTLS-GO Document 22, Version 6.0.1.0, is delivered with this release for the first time.

1.2.5 Delivered Software Components

JTLS-GO 6.0.1.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)
- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)

- JTLS Symbols Application (JSYMS)
- Database Development System (DDS)
    - Database Configuration Program (DCP)
    - DDS Client User Interface (DDSC)
- ATO Translator Service (ATOT)
- ATO Generator Service (ATOG)
- ATO Retrieval Program (ATORET)
- JTLS Convert Location Program (JCONVERT)
- Count Critical Order Program (CCO)
- JTLS HLA Interface Program (JHIP)
- After Action Review Client (AARC)
- Scenario Data Client (SDC)
- Order Entry Client (OEC)
- Order Verification Tool (OVT)
- JTLS Object Distribution Authority (JODA)
- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:
    - Apache Server (APACHE)
    - JTLS XML Serial Repository (JXSR)
    - Order Management Authority (OMA)
    - Synchronized Authentication and Preferences Service (SYNAPSE)
    - XML Message Service (XMS)
    - Total Recall Interactive Playback Program (TRIPP)
- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation

- Tactical Electronic Intelligence (TACELINT) Message Service
- KML Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool

Instructions for installing JTLS-GO 6.0.1.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.0.1.0 is not necessary. No other upgrade beyond installation of the compressed TAR files (or CD) is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called "repository60". Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space. With JTLS-GO 6.0.1.0, it is possible to access and copy records from the repository60 database into your own developed scenarios.
- The scenario “wespac60”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.0.1.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Edition Version 7.8 (ES), 64-bit architecture.

  JTLS-GO 6.0 has been tested with the following versions of Linux 7:

  RedHat Linux 7.8 - this operating system license must be purchased.

Oracle Linux 7.8 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

CentOS 7.8 - a free version of Linux 7 that has not been approved by DISA for use by U.S. Government agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.0.1.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.7 and Version 7.8.

  CentOS Linux Version 7.7 and Version 7.8.

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

- JTLS-GO 6.0.1.0 no longer relies on the Open Java Development Kit (OpenJDK™) to be installed at the system level using the Red Hat Package Manager (RPM). Instead, JTLS-GO 6.0.1.0 is delivered with the equivalent AdoptOpenJDK package. Using AdoptOpenJDK provides two benefits:

  a. Only the JTLS-GO account on the system servers access this version of Java. An installation site can use the JTLS-GO servers for programs other than JTLS-GO without impacting the version of Java used by other programs.

  b. Security releases of AdoptOpenJDK software are produced on the same schedule as the Oracle OpenJDK security release procedure. An organization can expect to receive a bug release version of JTLS-GO within two-weeks of a new Java 1,8 security release. As long as a user organization installs all of the JTLS-GO bug releases, JTLS-GO can guarantee that the latest Java security release is being used on the servers. JTLS-GO no longer depends on system administration for implementing proper Java security update procedures.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.3.

  Note: Red Hat Linux version 7.8 continues to be distributed with IcedTea version 1.7.1. There is an available RPM for IcedTea 1.8.3, and this should be explicitly installed on the JTLS servers and client workstations. If this is not done, then HTTP unsecured communication cannot be used with Apache and Glassfish.

- JTLS-GO database tools require a certified PostgreSQL 11.8 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered

containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.0.1.0. There are several alternative methods available for obtaining the PostgreSQL 11.8 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* 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 III Version 3-5.2.0 (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc.

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

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

- The JTLS-GO DDS (Database Development System) application uses these open source libraries:

  JFreeChart, licensed under LGPL (GNU LESSER GENERAL PUBLIC LICENSE) by Object Refinery Limited, http://www.object-refinery.com

  JCommon, licensed under LGPL2.1 (GNU LESSER GENERAL PUBLIC LICENSE version 2.1 or later) by Object Refinery Limited, http://www.object-refinery.com

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

- KML Operational Interface (KOI)

  The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth™. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using the COP, C2PC, or other C4I systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

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

- JTLS-GO 6.0 implements SSH Tunneling between Apache and the services and among the services. Rigorous testing should be done prior to use in any exercise, and particular attention should be paid to network performance under load.

1.3.2 JTLS-GO Cybersecurity Compliance

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

One of the DoD 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. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. After a certain date, all Java related programs stopped working, whether connected to an open network or not. All JTLS-GO releases were closely linked to the Java expiration date.

OpenJDK™ has not implemented an expiration date. In order to fulfill this DoD Cybersecurity requirement, JTLS-GO has moved to AdoptOpenJDK, a full OpenJDK Java environment with licensing alternations allowing an application to deliver the software. The following procedure has been established and approved by the JS/J7 Cybersecurity branch:

- Within days of an Oracle Java security release, AdoptOpenJDK produces an equivalent version using infrastructure, build and test scripts to produce prebuilt binaries of the OpenJDK class libraries. All AdoptOpenJDK binaries and scripts are open source licensed and available for free.

- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a Maintenance release version (JTLS-GO 6.0.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the Maintenance released versions when distributed.

Contact the U.S. Government Program Manager, Mr. Donald Weter (donald.e.weter.civ@mail.mil) to obtain the completed Cybersecurity paperwork and a current Authority to Operation certificate.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.0.1.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

The solution combines JTLS-GO with CAE's GESI constructive tactical entity-level simulation system. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The 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. JTLS-GO uses the Federation Object Model (FOM) located in the $JGAME/data/hla directory, Federation testing of JTLS-GO with CAE's GESI model have been accomplished. Future plans include expanding the capabilities of the GlobalSim federation.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.0.1.0 delivery. Users may obtain a full installation package of the RTI software from 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

Significant database structure differences exist between JTLS-GO 6.0.0.0 and the previous JTLS-GO 5.1 series database structure. There are no data format changes as a result of this Maintenance release.

To upgrade your JTLS 5.1 scenario to JTLS-GO 6.0 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 6.0.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.0.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario_name>.gs and databases symbol.scf file.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.0.1.0, but end-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

This means that the user can easily move in this new symbol set. Please follow the steps outlined in the *JTLS-GO DDS Users Guide*, Section B.13, Updating Scenario Symbol Set.

1.4.3 Standard Repository Changes

R&A has continued to improve and expand the unclassified data repository, “repository60.” The DDS comparison and synchronization function can be used to determine if any of the changes delivered are of use to a JTLS-GO user organization.

Database triggers were modified due to STR JTLS-2020-14976. In order to update your database, perform the following:

1. Unload your scenario using the JTLS-GO Menu, Options 1 -> 1 -> 5.
2. Load your scenario using the JTLS-GO Menu, Options 1 -> 1 -> 4.

1.5 INSTALLATION

The *JTLS-GO Installation Manual,* a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery, It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 11.8 required to operate JTLS-GO 6.0 series of software.

2.0 ENGINEERING CHANGE PROPOSALS

Normally, a Maintenance Release of JTLS-GO does not include any Engineering Change Proposals (ECPs). In order to meet various customer needsm some minor ECPs were added to JTLS-GO 6.0.1.0. These ECPs are summarized in this chapter.

2.1 JTLS-2020-14790 Split Formations With Embarked Units

Summary of Model Change Request

In JTLS-GO, ground units can be embarked on ships in naval formations. A formation can be directed to merge with another formation thereby forming a single formation carrying the embarked units. However, the single formation with embarked units cannot be subsequently split into its original parts. The current JTLS-GO Split Formation capability was expanded to provide full support of splitting formations as needed to support user requirements.

Design Summary

The Split Formation Order will be accepted and executed if all ships that contain a portion of a given embarked unit are part of the Split. An embarked unit will not be split between two different formations. If a Split is not legal, the user is notified of the problem and the problem can be corrected by either:

- Including all ships carrying a portion of the problem embarked unit in the Split, or
- Using the Cross Deck order to magically place the embarked unit on the ships that will be split.

2.2 JTLS-2020-14910 Moving Ship Probability of Hit Modifier

Summary of Model Change Request

Currently, there are numerous missiles that have a Probability Of Hit (pH) against ships as zero, because they do not have the ability to hit a moving ship. These missiles could be used against ships in port, but they cannot be used since the pH is zero. Some weapons can be used against ships in port and while moving, but not with the same pH. This problem is not simply limited to ships, it is also true for moving trucks or other moving objects. This ECP addresses these issues.

Design Summary

Users do not need to use this ECP. Your database as developed for JTLS-GO 6.0.0.0 will operate as it always had without any changes. To take advantages of this improvement, the following steps must be taken:

- Review all Surface Kill Lethality Probability of Hit (pH) data for Targetable Weapons that are labeled as TW MISSILE CAPABLE of YES. This ECP only applies to missiles. Remember that this pH data now represents the probability that the weapon can hit a stationary version of the ship or target. Change the pH as desired. For our ship example, the R&A database team altered the repository data to indicate that Tactical Land Attack Missiles (TLAMs) can hit ships. These parameters were set to zero in previous versions of JTLS-GO

- Add a new Targetable Weapon (TW) database parameter to the scenario's non-configuration managed data file. This new database parameter is called the TW MOVING OBJECT PHIT MODIFIER. If object A wants to fire a weapon at Object B, the model always checks if the weapon has a positive probability of hit and a positive probability of kill against Object B. If not, then Object A is restricted from firing.

  This logic has been altered to determine if Object B is moving. If it is, then the TW's database pH is multiplied by the new TW MOVING OBJECT PHIT MODIFIER. If the modified pH is zero, the Object A is restricted from firing on Object B with that TW.

  Continuing with the TLAM example, the TW MOVING OBJECT PHIT MODIFIER for the TLAM should be set to zero, which will stop it from hitting a moving object. Similarly, a new Naval Strike Missile will probably have a TW MOVING OBJECT PHIT MODIFIER of 1.0 indicating that the weapon is designed to hit moving ships.

The following rules are used to determine if an object is moving:

- The object must have a Move Event scheduled to occur.

- The firing object and the receiving object must be further apart than the Visual Detection Range. This rule allows "aimed" fire to occur even if an object is moving.

To add the new data to the Non-Configuration data file, a separate text file must be created containing the names of the TWs for which there is a difference between the pH against a stationary target and the pH against the target while moving. This text file belongs in the $JTLSHOME/data/scenario/<scenario_name> directory and is called <scenario_name>.ncm.

This text file will be read by the CEP and SIP. The file will be written during a CEP checkpoint in case the values have changed via controller order. A sample file is excerpted below:

```
TW.MOVING.OBJECT.PHIT.MODIFIER TLAM-E 0.0
TW.MOVING.OBJECT.PHIT.MODIFIER NSM 0.8745
TW.MOVING.OBJECT.PHIT.MODIFIER SS21.B.HE 0.6843
TW.MOVING.OBJECT.PHIT.MODIFIER DF21A.HE 1.0
TW.MOVING.OBJECT.PHIT.MODIFIER DF15.HE 9.999998
```

Note the line highlighted in "Yellow" is not needed. The database parameter is automatically initialized to 1.0.

2.3 JTLS-2020-14949 Improve Manage TPFDD Order

Summary of Model Change Request

The purpose of this Model Change Request is to include an option on the Time Phased Force Deployment Data (TPFDD) report for listing a specified unit and its subordinates. The Player's version of the report request order and the Controller's version of the order each needed to be addressed as part of this ECP.

Design Summary

The implementation for this ECP was straightforward. A new field was added to the Controller and the Player version of the order indicating whether the unit or the unit all of its subordinates should be included in the resulting report.

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. These issues are being addressed and should be solutions will be included in future Maintenance releases in the JTLS-GO 6.0 series.

3.1 JTLS-2020-14866 Old Altova Code Not Deleted

When the order checker was rewritten to use JAXB, the generated Altova code was deleted from the svn repository. However, the Altova directory, and some subdirectories and old Java code, were not deleted from the JSOURCE/ovp/ directory.

The Altova directory was removed from of JSOURCE/ovp/src/com/rolands/ovp/ directory.

3.2 JTLS-2020-14867 Undefined Force Side For Contamination Areas

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

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

3.3 JTLS-2020-14870 Delayed Air Mission Error Cancellation

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

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

3.4 JTLS-2020-14872 Aircraft Leaving Squadron End Maintenance Error

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

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

3.5 JTLS-2020-14874 Missing CSP CAT Record

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

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

3.6 JTLS-2020-14875 Crash Deleting Unit Destroy Target Task

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

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

3.7 JTLS-2020-14877 Zero HRU Assessment Duration Error

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

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

3.8 JTLS-2020-14881 Alert Status Improperly Conveyed To RW Systems

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

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

3.9 JTLS-2020-14883 Variations In Computations Of Unit Radius

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

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

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

3.10 JTLS-2020-14885 Direct Support Unit Does Not Stop Moving

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

However, before the replacement Direct Support order could be sent, the unit's MOVE ALONG ROUTE GROUND event was executed. A series of routines were called from this event that

eventually led to the routine ADJUST AUTO MOVE ROUTE. A logic error was recorded in this routine because the unit's task list only held the currently executing Move task and the replacement Direct Support task had not yet been added to the list.

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

1. Remove the Direct Support task and;

2. If the current executing task is a Move with its PET AUTO MOVE FLAG set to Yes, that task should also be canceled.

The CEP executed 1 but did not execute 2. The logic was adjusted to correct the problem.

3.11 JTLS-2020-14887 Hard Stop Glassfish Server If Timeout

The Glassfish domain stop-domain took excessive time to complete, resulting in multiple residual Glassfish processes that could prevent Glassfish from starting the next time it was used.

A hard-stop to Glassfish was added if the stop-domain process times out.

3.12 JTLS-2020-14889 UNIT WORKING SET Not Cleared At End Of Event

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

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

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

3.13 JTLS-2020-14891 SET.SC.PARAMETER Bad Message Trace

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

When selecting a new consumption type, the sub-message #11 tag and the group tag was written out twice, which resulted in a mismatch between the opening and closing tags. These tags are no longer duplicated.

3.14 JTLS-2020-14893 No Unit Authority Report For CONT WHIPs

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

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

3.15 JTLS-2020-14895 Ship Joining Formation Logic Error

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

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

Each time the formation moved, the lagging ship's route was recalculated toward the formation's new location. In this methodology, a great circle distance was calculated using an approximation of one nautical mile equals one degree. Ordinarily, the traveled distance is less than the great circle distance because the ship has not yet completed its planned move. However, in this case, the traveled distance (based on speed and elapsed time), was slightly greater than the great circle distance which generated the logic error. The model self-corrected by using the great circle distance.

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

3.16 JTLS-2020-14898 Replay JXSR Gets Stuck When Playing Backward

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

The Replay JXSR could compute a negative or zero Next Time Step in days for the time it is to move toward. This could happen when the TRIPP is playing the download files backward and generally at a rate greater than 1:1. With a negative or zero time, the Replay JXSR would get

stuck reading the first packet, a download packet, which would result in another negative or zero time step. This would cause the JXSR to again read the download packet and continue to loop.

3.17 JTLS-2020-14899 Missing Intelligence Report

Received the following message during testing:

An error has been discovered in the routine CREATE.INTELLIGENCE REPORT. The Specific Type was 92. Please report the problem to the appropriate personnel.

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

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

3.18 JTLS-2020-14902 NEC CCIS JTOI CWIX Issues

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

1. *The mission status for Alert missions canceled while still on the ground was incorrectly set to Terminated and not Canceled.*

2. *When a squadron was given a new home base, any active air missions from that squadron had their Departure Base set to that new home base when it should have been preserved.*

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

1. The JTOI now checks to see if the air mission was canceled while airborne or on the ground. The appropriate mission status is then set.

2. The JTOI will now only set the DepartureBase for an existing air mission if that air mission does not yet have a DepartureBase set.

3.19 JTLS-2020-14903 Shadowing Ship Did Not Alter Speed

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

The model automatically calculated the correct speed necessary to keep pace with the shadowed vessel. However, a sequencing error in the code bypassed the logic that changed the

ship's current speed and task speed. Consequently, the ship did not adopt the new speed. The problem was fixed by re-sequencing the code to not skip the current speed and task speed updates. Also, new logic was added to update the ship's ordered speed so that the ship does not revert to its previous speed after its first move along the intercept route.

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

3.20 JTLS-2020-14906 WHIP JNLP File Not Cleaned Up

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

Previously, the file was being removed from only the user's system temporary folder when it was downloaded there by the browser. The respective JNLP file will now be deleted from the user's Downloads folder and one level underneath the user's system temporary folder upon exiting the application.

3.21 JTLS-2020-14908 Unknown Object Type in Mission Report Error

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

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

3.22 JTLS-2020-14912 Super WHIP SITREP

The perception Side for the SITREP on the Super WHIP was not visually consistent with the component perception that invoked the object information display.

A [SIDE.Name] label was created on the component's title bar, as well as color-coded line border to match the side for which perception information is being displayed.

3.23 JTLS-2020-14913 Lengthy Force Side Name Crash

The generation of the Online Player Manual (OPM) fails if a Force Side name has more than 15 characters.

The proper OPM file structure was not created for the Force Side with a long name. The issue was solved.

3.24 JTLS-2020-14914 SVP Warning 1442 Remove Support Unit

The routine EXECUTE.CONVOY.ROUTE. took a very long time to execute.

The naval unit was trying to find a route to its support unit. In some cases the best solution is to remove the support unit for the unit. This could be because it is a naval unit which is out to sea or another type of unit that is secluded in a hard to get to area. A new SVPR correction to Warning 1442 was added to allow the DDS user to automatically delete the support unit.

3.25 JTLS-2020-14916 Magic Move Submarine With Future Task Crash

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

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

3.26 JTLS-2020-14918 Check Order Script Not Checking AAR/DDS Reports

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

When requesting a verification of an AAR or DDS report, the checkorder expects an argument in a certain format, which should then be passed to the OVP in a different format However, what was being passed to the OVP were the arguments written in the original format, which prevented the OVP from finding the correct report file to verify.

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

3.27 JTLS-2020-14920 SVP Actions Different Comparison Operators

The corrective actions defined for different SVP errors and warnings are defined in an XML file. Within this XML file a single corrective action may have different executions depending on the type of object JTLS is dealing with.

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

To reduce simple mistakes, a checker exists to validate the contents of the XML file. This checker ensures that the matches for a corrective item are unique. This check was not taking into account the comparison operator ("is" versus "is not"), so a match of "is Target Category a Jammer Type" was the same as "is Target Category not a Jammer Type". This limited the ability to make some corrective actions.

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

3.28 JTLS-2020-14922 Generating Online Player Manual Crash

The SIP crashed while generating the OPM. The scenario contained an external event to schedule an Assess Weapon Damage (AWD) event for sometime in the future. The crash occurred when the SIP was printing data for this event for the OPM.

No targetable weapon was found when printing the AWD event in the OPM. The specific weapon was not identified because targetable weapons were changed from integers (permanent entities) to pointers (temporary entities) as part of code improvements for JTLS-GO 6.0. Corrections were made in the SIP to resolve this crash.

3.29 JTLS-2020-14923 Setting Link16 Network Error

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

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

3.30 JTLS-2020-14925 Building Ship Movement Plan Error

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

The capability to create ship movement plans and perform strategic lifts is currently not supported in the model. Changes were made to the unsupported code to prevent the crash. A message was added to tell the player that no action was taken for this order.

3.31 JTLS-2020-14928 Non-POT Invalid Target Ranges Not Identified

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

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

3.32 JTLS-2020-14929 Crash Reading Checkpoint Formation Data

The CEP crashed reading formation ASCII data when coming up from a checkpoint.

The problem was that formation ships being used in the current formation task, such as an amphibious pickup or assault, are placed in a set. To preserve these data, the set membership attribute was being written as a single digit.

In previous versions of JTLS-GO, we packed these set membership attributes so they were only a single digit, but the new Simscript compiler introduced with JTLS 6.0.0.0 does not support packed attributes, so these values became pointers. When the attribute is set it is now up to a 15 digit pointer, and instead of attempting to write out too large a value within the designated space, Simscript will instead write asterisks (""). On a read, the CEP would attempt to read the asterisk as an integer and crash.*

The write statement was switched to write 15 digits, which will hold a 64-bit pointer, and the read statement was switched to check the read-in value against zero before filing in the set.

The rest of the checkpoint code was checked for any other instances of this error. Though none were found, there were places where the pointer was being written as 10 digits and places where the read-in value was checked if it was greater than zero (versus not zero). Both of these issues could cause problems on large scenarios and were fixed.

3.33 JTLS-2020-14931 Change Country Code To Three Character Value

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

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

3.34 JTLS-2020-14932 SET.SC.PARAMETER Crash

A user attempted to set the SC.SPECIAL.CAPABILITY to "NONE", but the Supply Category already had a SC.SPECIAL.CAPABILITY of "NONE". The CEP attempted to send a message stating that there would be no change. The game crashed when the CEP attempted to access the SPECIAL.SC.NAME.ARRAY using an index of zero.

The SET.SC.ATTRIBUTES order has been changed to write the special capability using an integer value instead of using the SPECIAL.SC.NAME.ARRAY (which was text). The message definition file has been updated to use the Supply_Special_Capability to translate from integer to text.

The user would previously receive a message detailing the supply category parameter modifications, even after receiving a "cannot comply" message which states the user's order "will be ignored." Now, once the user receives a "cannot comply" message, the CEP exits the routine and does not send another message detailing the user's changes.

3.35 JTLS-2020-14933 Set Shelter Priority Mislabeled Order Field

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

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

3.36 JTLS-2020-14934 Unusable Redirect Cruise Missile Order

A Player was unable to use the Redirect Cruise Missile Order. Some cruise missiles existed in the game, but the order panel did not allow the Player to select one of them.

The order panel was using an old attribute name when filling the data for the cruise missile object. As a result, no missiles were listed on the order. The attribute name was changed from "mission_name" to "reference_name" and the order field was correctly filled with the names of existing cruise missiles.

3.37 JTLS-2020-14935 OAS Crash At End Of Single Path Orbit

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

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

3.38 JTLS-2020-14939 OAS TTG Check Not Functional

The SVP was unable to generate Error 317.

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

The SVP message was also incomplete and did not provide enough information. This was also corrected.

3.39 JTLS-2020-14942 Unused Order Filling Code In WHIP

When opening an order in the WHIP from an object's context-sensitive menu, the WHIP had unused and non-applicable handlers to fill the order unit field with the object's higher headquarters name.

The unnecessary code was removed.

3.40 JTLS-2020-14943 Invalid Combat System Special Subcategory

The SIP crashed in routine CHECK.DISCODE.USAGE.

The cause of the crash was a combat system that had a SPECIAL.CAPABILITY but did not have a valid SPECIAL.SUBCATEGORY. This can happen if PSQL is used to modify the field in the database, if the user has hand-edited the file and made a mistake, or if the data is bad as the result of upgrading a previous database that had the same error.

The code was modified appropriately to prevent the program from crashing and tested.

3.41 JTLS-2020-14944 DDSC/SVP Created Unnecessary Verify Link

For JTLS-GO 6.0.0.0, the SVPR was updated not to depend on a specific link to access the ~/data/scenario/<scenario-name>/verify directory. However, the link is still created.

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

3.42 JTLS-2020-14946 Shared Order Group Fixes

Users have the ability to share Order Groups to other WHIPs on their side. Several capabilities were lacking for the original implementation:

1. *Users could not quickly tell which Order Groups they had shared, or which WHIPs they had shared the Order Groups with.*
2. *Shared Order groups were not sorted in the Shared Groups tree.*
3. *Clearing a shared order did not clear the ownership of the order panel.*

Shared Order Groups now have a different icon in the Order Groups tree, are now sorted in the Shared Groups tree, and clearing a shared order resets the ownership of the order panel back to defaults.

3.43 JTLS-2020-14955 JSAT Help Text Out Of Date

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

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

3.44 JTLS-2020-14958 Mismatched Code For Offloading Convoy

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

The mismatched code was corrected, which resolved this problem.

3.45 JTLS-2020-14960 Updated Screen Definition Checker

The screen definition checker did not check if a DRM attribute does not exist in the cols/tc/help node.

Some error messages did not correctly validate the mandatory attribute check.

Additionally, the script name to invoke this checker program was not descriptive.

The screen definition checker code and dds.function.type file were updated.

Changed the script name to a more descriptive name.

3.46 JTLS-2020-14961 Naval Formation Magic Move Ships With Routes

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

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

3.47 JTLS-2020-14963 Center World View Near Equator

When selecting World View in the WHIP map, the map was not centered on the equator.

The map now centers on the equator when selecting World View.

3.48 JTLS-2020-14964 Super WHIP Route Visible To Other Sides

In a Super WHIP, a route for a Force Side was visible on a map set to another Force Side's perception. The former Force Side's unit had also not been detected by the latter Force Side.

This was corrected to only display routes for objects of the same side perception of the map in a Super WHIP, when the map perception is not set to Controller. Detected units also cannot have their route be shown by the detecting side.

3.49 JTLS-2020-14965 Script Created Broken Links

The jinn_sipprogrm script created an Data Requirements Manual (DRM) link for the JOBE packaging for the OPM in $JTLSHOME. However, the JOBE directory structure differs from $JTLSHOME, so the link was broken.

The jinn_sipprogrm script no longer creates a DRM link. Instead, the script that copies the DRM PDF documents when user creates the OPM was modified to upload the relevant DRM pages to the OPM.

3.50 JTLS-2020-14966 DDSC Problem with Validation

Certain dtabase tables specify column values that must be unique, and the DDSC has a built-in validator to ensure the uniqueness of certain column values during editing.

However, one of the columns in the faction_tc_assets table, the sma_rr_name, can be NULL, and when the sma_rr_name is NULL, the DDSC validator allowed data that were not unique.

During validation, the DDSC interpolates a field whose value is null as "field_name = ''", and PostgreSQL cannot accept this value. Therefore, the code was changed in the DDSC to specify "field_name IS NULL", as required by PostgreSQL.

3.51 JTLS-2020-14967 IMT Negative Fuel Percent

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

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

The JODA Data System Protocol used a data type to hold the % Fuel that was too small to accommodate the large percentage value. Consequently, the value translated to a negative percentage. To correct the problem, the data type INT_16 was increased to UINT_32, which was consistent with the Fuel field data type.

3.52 JTLS-2020-14969 Error 425 Will Not Generate

The SVP was unable to generate Error 425.

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

3.53 JTLS-2020-14971 AAR Object Attack Report Hull Damage Exception

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

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

The input value is now cast explicitly to a string before passing it to the HTML utility stylesheet.

3.54 JTLS-2020-14974 Unit Relative Position Not Maintained

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

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

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

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

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

- The distance and bearing from the leader to the follower is now based on the leader's and follower current location at the time the leader starts to execute its second Move task.

- If the follower is slower than the leader, the follower will still be moving. The follower will cancel the Move task it is currently executing and execute the new Move task.

- If the follower is faster than the leader, it will have completed its Move task and will be waiting at its designated waypoint for the leader to reach its waypoint.

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

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

- The distance covered during the Move task the follower is executing before the next Move task arrives.

- The twists and turns in the leader‘s route.

- The number of waypoints in the leader's route.

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

This problem was resolved through the following changes:

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

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

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

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

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

3.55 JTLS-2020-14976 Stationary Targets Moved With Owning Unit

When updating a unit location in the DDS, that unit's STATIONARY owned targets were updating their locations as well. It was decided that STATIONARY targets should remain at their current location even when their owning unit changes location. Some of these STATIONARY target locations were obtained from real-world databases, and automatically altering their location was inappropriate.

The database triggers responsible for updating unit-owned targets have been modified to exclude STATIONARY targets on a location update. If the unit is moved too far from its owned stationary targets, the SVP will generate an error and the user can decide the proper solution to the problem.

3.56 JTLS-2020-14978 Error 428 Will Not Generate

The SVP was unable to generate Error 428.

Several immediate problems were found in the routine **disembuni.sim**. The first involved how combat systems and supplies were tracked. There were also problems in the calculation if the formation had absolutely no room to embark a unit.

Since the process of embarking and distributing units in the CEP works well, it was decided to bring the SIP process in line with how it is done in the CEP. The CEP routine was duplicated for the SIP and the necessary changes to the routine and the preamble were accomplished and tested successfully.

3.57 JTLS-2020-14979 Check For Valid UOM Supply Category

The checkimt script was improved to ensure a valid UOM category group is used for Supply Categories.

The script was modified to add such a check.

3.58 JTLS-2020-14980 Various DDS SVPR Corrections

Various DDS SVPR corrections and enhancements were needed:

1. *A user attempt to use SVPR corrections for Warning 1443 to set the unit ICAO to null failed.*
2. *The user could not delete combat system aircraft using the SVPR correction for Error 430. A complete list of combat systems, instead of just aircraft, should also be added.*
3. *The correction option for Warning 1412 needs to display all combat systems.*
4. *Error 308 uses the name SHIP where referring to Ship Unit Prototype names.*
5. *Errors 415, 421 and 430 all appear to be identical.*
6. *There are no options in the SVPR to display the locations of arcs that have a zero distance on the map.*
7. *Warning 1221 indicates that a HUP is short of supplies and by how much, but the displayed on hand and required values are the same.*

1. The syntax used in the SVPR Template tried to use the format " to set the field to a null value. The correct method is to set the field value to NULL using the text word NULL. The template was modified.

2. Deletion of the combat system for both TUPs and SUPs was tested and working properly. The display of the combat systems was changed to report all prototype systems for both TUPs and SUPs.

3. Warning 1412 triggers when a Support Unit has no tanker trucks to use for transporting wet supplies. The corrective action only brings up a blank combat system window for the user to insert a tanker combat system. It would be better if the user could view all the combat systems for the support unit TUP, to determine if another mistake has caused the problem, such as that the Unit has a tanker but is not properly represented in the combat system table. The corrective action was modified to display the entire combat system table.

   Warnings 1411, 1412, and 1413 are in the same routine. These had the same problem and were also modified to display the entire combat system table for the support unit TUP.

4. Error 308 generates errors for all Target Type Groups displaying name and subcategory if the weapon load has no capability of damaging the category and subcategory combination. The code uses generic names when displaying the error for all categories. The generic code in this instance is SHIP. The code was specifically modified to modify this generic name to SUP when generating the errors for the Ship Unit Prototypes.

5. Errors 415 and 430 were determined to be the same. Error 415 was removed. Error 430 is reported when an unit has a aircraft combat system record but does not have an aircraft type. Error 421 is the opposite. It is reported when the unit does not have an AIRCRAFT combat system record but does have a aircraft type specified.

   It was also determined that the situation to trigger Warning 1464 could not happen. It was also removed from the code.

6. An option to locate the pipeline arc on the DDS map has been implemented for Error 953. Error 953 is also used the checks for sealane arcs and air corridor arcs. The code for all programs were updated.

7. The problem is caused by rounding. Only 4 significant digits to the right of the decimal were displayed. The number has been increased to display six digits to the right of the decimal for Warning 1221.

3.59 JTLS-2020-14981 Formation Icons Displayed At Equator

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

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

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

The error was corrected by limiting the display of formation icons to just those from the owning Side and to foreign Sides with ships in the formation. In other words, friendly and neutral Sides that do not have any ships in the formation are permitted to know the formation is active, but do not perceive the formation icon on the map until one of their ships is ordered to join the formation.

3.60 JTLS-2020-14983 Drawing Tool Line Changes

The WHIP Drawing Tool did not allow the user to change arrowhead lines to normal lines after changing the direction of arrowheads.

The problem corrected by adding logic to change arrowhead lines back into lines.

3.61 JTLS-2020-14984 DDSC Turn Off Network Node Names On Map

The DDS shows network node labels on the map. The node names were taking up so much space, the user could not see the network to zoom in on where the breaks are. The user needs the ability to turn node names on and off on the DDS map.

The ability to turn network node names on or off on the DDS map was added. The button was placed next to the new network button.

3.62 JTLS-2020-14986 DDSC Network Filter Failed To Show Node Names

When a network editor button is selected before the network filter checkbox is selected, the selected network displays node names on the map. However, if a network editor button is not selected before the network filter checkbox is selected, the network filter failed to show node names on the map.

The DDS map code was modified to display node names when the map filters are changed.

3.63 JTLS-2020-14989 JOI OTHGold Unreported Errors

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

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

3.64 JTLS-2020-14990 DDSC Deleting Faction Country Problem

When a user tried to delete a faction country record from DDS Client's Faction Country table, nothing happened, and the record remained intact.

This was caused by the migration from Oracle to PostgreSQL. The Glassfish server's deleterecord servlet is now fixed to process the identification properly.

3.65 JTLS-2020-14991 DDSC SVPR Correction Option Not Working

For certain types of SVP errors, such as Error 112, the correction solutions find a location on the map. However, clicking on such options did not work.

This was due to the fact the location data provided for the correction option contained leading spaces. The DDSC was modified to ignore leading or trailing spaces during a location search.

3.66 JTLS-2020-14992 Warning 1300 Options

The corrective options for Warning 1300 are incorrect or do not work. The option to display the ADA altitude zone record does not work ,and the option to change the PK can be improved.

Warning 1300 is triggered when an ADA has a valid PK for a Surface to Air Lethality, Aircraft Target Class, Altitude Zone combination but has zero range capability in the altitude zone. If the altitude range data record actually exists but has a zero range then the record will be displayed. If there is no record, then the assumption in the altitude zone is considered zero but there is no record to be displayed and the corrective action will do nothing.

Option 1 is working as it should. We are looking at the possibility of displaying a "Record Not Found" warning to provide user feedback when the record does not exist. Option 2, to display the table record for the PK data, was reworded to indicate the value should be changed to zero or the record deleted, and a third option, to automatically delete the SAL PK record, was added.

3.67 JTLS-2020-14993 ATOG Shared Foreign Runways

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

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

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

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.

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 6.0.0.0 DATABASE CHANGES

Refer to Appendix B in the JTLS-GO Version 6.0.0.0 VDD. No database format changes were needed to this Maintenance release.

APPENDIX C VERSION 6.0.0.0 REPOSITORY CHANGES

The following changes were made to the JTLS-GO 6.0 repository.

C.1 New Aircraft Class Added

- AW159

C.2 New Aircraft Loads Added

- AW159.DEF
- 8SEA.VENOM.AL
- 20MARTLET

C.3 New HUPs Added

- SHIPPING.CONTAINER.SSM

C.4 New Naval Units Added

- SEVMORPUT_RU
- ALEX.TVARDOVSKIY
- ALEX.ZRYACHEV_RU

C.5 New Sensors Added

- SEAMASTER.400_SAA
- SEAMASTER.400_SGN
- SELEMAR_VGN
- NS-100_SAA
- NS-100_SGN
- SEAWATCHER-100_SGN
- SPN-760_VGN
- SEASPRAY.7000E_AGS
- GLOBUS.II.(IR)_LAA

- GLOBUS.II(FPS129)_LAA

C.6 New SSMs Added

- C705(1)
- C705(2)
- SHIPPING.CONTAINER.VLS(4)

C.7 New SUPs Added

- 150CLASS.LCT_TR
- AKIN_BU
- ALEMDAR_TR
- AWAJI_JP
- BADAU_ID
- BALTYK_PL
- BARTOLOMEU.DIAS_PT
- BAYRAKTAR.LST_TR
- CABRINI_IT
- CALIRAYA_PH
- CANTABRIA_ES
- CARGO.SHIP_SSM
- CASTELO.(OPV)_PT
- CLURIT.(20MM)_ID
- CLURIT.(30MM)_ID
- COSAR_RO
- DIANA_DK
- DOGAN_TR

- DOORMAN.JSS_NL
- FLYVEFISKEN_DK
- FLYVEFISKEN_LH
- FPB-57.NAV.I_ID
- FPB-57.NAV.II_ID
- FPB-57.NAV.IV_ID
- FPB-57.NAV.V_ID
- GARDNO_PL
- GODETIA_BE
- GUR(T2-1400)_TR
- HATERUMA_JP
- HNO.ALESUND_NO
- HOLLAND_NL
- HOLM_DK
- J.D.WITT_NL
- JOSE.RIZAL_PH
- KASOS_GR
- KATITINGAN.30MM_PH
- KATITINGAN.40MM_PH
- KOGALNICEANU_RO
- KORMORAN.II_PL
- KRAIT_ID
- LA.CONFIANCE_FR
- LINDAU_LH

- LINDORMEN_EN
- MACHITIS_GR
- MAMRY_PL
- MANDAU_ID
- MPAC.MK.I_PH
- MPAC.MK.II_PH
- MPAC.MK.III_PH
- MRD_DK
- MSF_DK
- MUSCA_RO
- MV.R.FISHER_GB
- OKEANOS(209-1500)_GR
- PARI.(20MM)_ID
- PARI.(30MM)_ID
- PROJECT1407_RO
- PULAU.RENGAT_ID
- PULAU.ROTE_ID
- RASMUSSEN_DK
- RIZAL_PH
- ROTTERDAM_NL
- RUZGAR_TR
- SAMPARI.(40MM)_ID
- SAMPARI.(57MM)_ID
- SIBARAU_ID

- SMARDAN_RO
- TAGBANUA_PH
- TAIGEI_JP
- TARLAC_PH
- TEJO_PT
- TETAL.I_RO
- TETAL.II_RO
- THETIS_DK
- TRIDENTE(TYPE214)_PT
- TULCEA_RO
- TUZLA.PB_TR
- TYPE.404_DE
- TYPE.702(KSV90)_DE
- VIDAR_LG
- VIDAR_LH

C.8 New Supply Categories Added

- CL.V.SS-RDR-LR-C705
- CL.V.SS-SRBM-DF15A.CHEM
- CL.V.SS-SRBM-DF15A.HE
- CL.V.SS-SRBM-DF15A.ICM
- CL.V.SS-SRBM-DF15B.CHEM
- CL.V.SS-SRBM-DF15B.HE
- CL.V.SS-SRBM-DF15B.ICM
- CL.V.SS-SRBM-DF15C

C.9 New UOMs Added

- C705
- DF15A.CHEM
- DF15A.HE
- DF15A.ICM
- DF15B.CHEM
- DF15B.HE
- DF15B.ICM
- DF15C.HE

C.10 Created New FACTION

- PHILIPPINE_MILITARY
- Added Philippine Ground, Naval, Support and Airbase Units

C.11 Updated all Combat Systems that are an Ambulance:

- Added Special Capability - Cargo
- Special Subcategory - Ambulance

C.12 New TW Added

- C705

C.13 New SKL Added

- C705_SKL

C.14 New PH Added

- C705_PH

C.15 New PK Added

- C705_PK

C.16 Renamed TWs

- DF15ARP.CHEM renamed to DF-15B.CHEM
- DF15ARP.HE renamed to DF-15B.HE
- DF15ARP.ICM renamed to DF-15B.ICM
- DF15AIP.HE renamed to DF-15C.HE

C.17 Renamed SKLs

- DF15IP.HE_SKL to DF15C.HE_SKL
- DF15RP.HE_SKL to DF15B.HE_SKL
- DF15RP.ICM_SKL to DF15B.ICM_SKL
- DF15AIP.HE_SKL to DF15C.HE_SKL

C.18 Renamed PHs

- DF15AIP.HE_PH to DF15C.HE_PH
- DF15ARP.HE_PH to DF15B.HE_PH
- DF15ARP.ICM_PH to DF15B.ICM_PH

C.19 Renamed PKs

- DF15AIP.HE_PK to DF15C.HE_PK
- DF15ARP.HE_PK to DF15B.HE_PK
- DF15ARP.ICM_PK to DF15B.ICM_PK

C.20 Renamed AKLs

- DF15IP.HE_AKL to DF15C.HE_AKL
- DF15RP.HE_AKL to DF15B.HE_AKL
- DF15RP.ICM_AKL to DF15B.ICM_AKL

C.21 Deleted TWs

- DF15IP.CHEM

- DF15IP.HE
- DF15IP.ICM
- DF15RP.CHEM
- DF15RP.HE
- DF15RP.ICM
- DF15AIP.CHEM
- DF15AIP.ICM

C.22 Deleted AKL

- DF15IP.ICM_AKL

C.23 Deleted SKL

- DF15IP.CH_SKL
- DF15RP.CH_SKL
- DF15IP.HE_SKL
- DF15IP.ICM_SKL
- DF15RP.HE_SKL
- DF15RP.ICM_SKL
- DF15AIP.CH_SKL
- DF15AIP.ICM_SKL

C.24 Deleted PHs

- DF15A.CH_PH
- DF15IP.HE_PH
- DF15IP.ICM_PH
- DF15RP.HE_PH
- DF15RP.ICM_PH

- DF15AIP.CH_PH
- DF15AIP.ICM_PH

C.25 Deleted PKs

- DF15IP.HE_PK
- DF15IP.ICM_PK
- DF15RP.HE_PK
- DF15RP.ICM_PK
- DF15AIP.CH_PK
- DF15AIP.ICM_PK