JTLS-GO

Version Description Document

August 2019

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

**JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.1.3.0)**

[Blank Page]

ABSTRACT

The Joint Theater Level Simulation - Global Operations (JTLS-GO®) is an interactive, computer-based, multi-sided wargaming system that models combined joint and coalition resource air, land, naval, and Non-Governmental Organization (NGO) environments.

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 5.1.3.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 5.1.3.0 is a Maintenance release of the JTLS-GO 5.1 series that includes an updated wespac51 demonstration database as well as updated repository data held in the repository51 database. There are no major Engineering Change Proposals (ECPs) included with this release, but there are a few minor ECPs that required no new data or data format changes, which are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

This publication is updated and revised as required for each Major or Maintenance version release of the JTLS-GO model. Corrections, additions, or recommendations for improvement must reference specific sections, pages, and paragraphs with appropriate justification and be forwarded to:

> JTLS-GO Development Team Leader
> ROLANDS & ASSOCIATES Corporation
> 120 Del Rey Gardens Drive
> Del Rey Oaks, California 93940 USA
> jtlsdev@rolands.com

[Blank Page]

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 5.1.3.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 5.1.3.0 is a Maintenance release for the JTLS-GO 5.1 series.

JTLS-GO 5.1.3.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations, called "wespac51". No database format modifications have been made for this release, but a static data error for allowable OTH-Gold Ship Types was corrected in Version 5.1.1.0. This does require a user to execute a correction procedure for all of their Version 5.1.0.0 scenarios loaded in Oracle. Information on this procedure can be found on Page 1-10.

Descriptions of minor Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0. Explanations of all Software Trouble Reports (STRs) corrected in this release are provided in Chapter 3.0. Outstanding STRs are provided in Chapter 4.0. Changes made to the JTLS-GO 5.1 engineering data repository are provided in APPENDIX C.

JTLS-GO 5.1.3.0 executes on the Red Hat Enterprise Linux Version 7.6 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 Portable Document Format (PDF) within a documents subdirectory.

1.2.1 Obsolete/Outdated Documents

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

1.2.2 Unchanged Documents

- *JTLS-GO Configuration Management Plan (*JTLS-GO Document 03, Version 5.1.2.0)

- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 5.1.2.0)

- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 5.1.2.0)

- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 5.1.2.0)

- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 5.1.2.0)

- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 5.1.2.0)

- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.1.2.0)

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

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 5.1.3.0)

- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.1.3.0)

- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.1.3.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.1.3.0)

- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 5.1.3.0)

- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 5.1.3.0)

- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 5.1.3.0)

- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 5.1.3.0)

- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.1.3.0)

- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.1.3.0)

1.2.4 New Documents

No new documents are delivered with JTLS-GO 5.1.3.0.

1.2.5 Delivered Software Components

JTLS-GO 5.1.3.0 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)

When operating the TRIPP capability in current JTLS-GO releases, users are not prevented from logging into an actively running TRIPP making a connection to the same Replay JXSR. A TRIPP, as documented, requires its own Replay JXSR to control and perform the replay of the recorded simulation events.

This situation as a consequence can have more than one user concurrently control the playback of the game, as the shared Replay JXSR will honor each of the user's playback requests and will then change what each connected user sees on their TRIPP instance. This issue will be addressed in a future JTLS-GO release by only permitting one login per TRIPP user instance, such as a WHIP login prevents from users logging into the same WHIP instance.

- Entity Level Server (ELS)

- JTLS Operational Interface (JOI) for both OTH-Gold and Link 16 generation

- Tactical Electronic Intelligence (TACELINT) Message Service

- KML Operational Interface (KOI)

- JTLS Transaction Interface Program (JTOI)

- JTLS Interface Network Navigator (JINN)

- JTLS Order of Battle Editor (JOBE)

- JTLS Geographic Information System (GIS) Terrain Building Program

Instructions for installing JTLS-GO 5.1.3.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 5.1.3.0 is not necessary. No other upgrade beyond installation of the compressed TAR files (or CD) is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario "repository51" serves as a repository of engineering level data. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your Oracle scenario table-space. With JTLS-GO 5.1.3.0, it is possible to access and copy records from the repository51 database into your own developed scenarios.

- The scenario "wespac51", which is based on the Western Pacific theater of operations and is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 5.1.3.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers:

- Operating system for the model: Red Hat Linux Enterprise Edition Version 7.6 (ES), 64-bit architecture.

> Security Enabled (SE) Linux must not be enabled on systems used to execute JTLS-GO or its components. Tests reveal that SE Linux-enabled systems cause frequent and random JXSR crashes and errors. Use of SE Linux to support JTLS-GO is currently not authorized.

- JTLS-GO 5.1 has been tested with the following versions of Linux 7:

    Red Hat Linux 7.6 - This operating system license must be purchased, but it has been approved by the Defense Information Systems Agency (DISA) for use by U.S. Government Agencies.

    Oracle Linux 7.6 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

    CentOS Linux 7.6 - A free version of Linux 7 that has **not** been approved by DISA for use by U.S. Government Agencies.

- There are no restrictions on the operating system used for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 5.1.3.0 has been tested on the following operating systems:

    Red Hat Linux Enterprise Edition Version 7.6.

    CentOS Linux Version 7.6.

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

- The JTLS-GO 5.1 series no longer uses Oracle Java, and has moved to the latest version of OpenJDK 8, which is OpenJDK 8 Version 222. We no longer deliver the Java Runtime Environment (JRE) within the JTLS-GO delivered software TAR files. Each user organization must obtain the latest version of the OpenJDK Red Hat Package Manager (RPM) and install the RPM on the servers used by JTLS-GO.

- JTLS-GO database tools require use of a certified Oracle database server and the full Oracle Client installation for runtime requirements. Additional installation details can be found in Section 1.5.2 of this chapter.

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

- TCP/IP is required for inter-process communication between the JODA data server and all user interface 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 Agency that can obtain source code and plans to re-compile JTLS-GO SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc.

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

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

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

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

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

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

- KML Operational Interface (KOI)

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

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

- JTLS-GO 5.1 implements SSH Tunneling between Apache and the services, and among the services. Rigorous testing should be done prior to use in any exercise, and particular attention should be paid to network performance under load.

1.3.2 JTLS-GO Information Assurance Compliance

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

One of the DoD IA requirements is that the software must implement a methodology that ensures that the end user keeps the software up-to-date and properly installs all security patches. 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.

Information available at the time of this release indicates that OpenJDK will not implement an expiration date. In order to fulfill the DoD IA requirement, Maintenance releases of JTLS-GO will be released as new versions of OpenJDK are released. In a future version of JTLS-GO, all delivered Java programs will check that the latest version of Java is being used. Each local installation will be able to override the warning generated when the program detects that the latest version is not properly installed.

JTLS-GO has completed the IA program mandates and the JTLS-GO 5.1 series of releases has been granted an Authority To Operate (ATO) on DoD systems. Contact the U.S. Government Program Manager, Mr. Don Weter (donald.e.weter.civ@mail.mil), for additional information.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 5.1.3.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO uses the Federation Object Model

(FOM), located in the $JGAME/data/hla directory. Federation testing of JTLS-GO is not complete, but initial tests with CAE's GESI wargaming system have been accomplished. Future plans include expanding the capabilities of the JTLS-GO-GESI federation, called "GlobalSim".

GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming, that helps commanders and all levels of staff prepare for a range of operational scenarios. The solution combines JTLS-GO with CAE's GESI constructive tactical entity-level simulation system as a federation. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The CAE GESI system is used to represent a virtual battlefield, including weapons, vehicles, aircraft, and ground forces.

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

All JTLS-GO software needed to run GlobalSim is included in this delivery.

The HLA RTI (Run Time Infrastructure) executive program recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 5.1.3.0 delivery. Users may obtain a full installation package of the RTI executive program from Pitch Corporation (www.pitchtechnologies.com). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

Significant database structure differences exist between the JTLS-GO 5.1 series and the previous JTLS-GO 5.0 series database structure.

To upgrade your JTLS-GO 5.0 scenario to JTLS-GO 5.1 compatibility, see instructions listed in Chapter 3.1 of the *JTLS-GO DDS User Guide*.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 5.1.0.0 formatted data from your Oracle database server into the JTLS-GO 5.1.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in Appendix B.11 of the *JTLS-GO DDS User Guide*. This procedure will reorganize the structure of the .gs and .scf symbols-related files.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 5.1.3.0, but end user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs.

1.4.3 Standard Repository Changes

The JTLS-GO 5.1 series of JTLS-GO is the first series in which R&A is delivering an unclassified data repository called "repository51". In future Major releases of JTLS-GO, APPENDIX B will provide a summary of the data structure changes made to the data repository. No data structure changes have been made is this Maintenance release; therefore, APPENDIX B is empty. Refer to Appendix B in the *JTLS-GO 5.1.0.0 Version Description Document*, included with this release, for data structure changes made for the JTLS-GO 5.1 series.

1.5 INSTALLATION

1.5.1 Installation Instructions

The *JTLS-GO Installation Manual,* a PDF file available for direct download, is part of this JTLS-GO delivery. It provides detailed instructions for installing a new version of JTLS-GO.

1.5.2 Oracle Compatibility and Installation

A full Oracle Client (not Instant Client) installation that matches your database server version is currently a requirement for running some JTLS-GO applications. The Oracle Instant Client is not sufficient for JTLS-GO applications because certain Oracle utilities, such as sqlldr, imp, exp, and tnsping, are missing. If you have applied a patchset to your database server, the same patchset should be applied to the Oracle Client installation. A 64-bit Oracle Client installation must be used.

The JTLS-GO scenario/database modification process also expects Oracle 11.2.0.1 or higher full Oracle Client installation. Some sites NFS mount their database server as Oracle Client; other sites prefer a full installation of the Oracle Client to a different directory that mounts to JTLS-GO (a simple NFS mount will suffice). Your system administrator can choose the appropriate installation.

Assigning the full Oracle Client installation location (or mount point) as the ORACLE_HOME in the JTLS-GO .cshrc file allows connecting to an Oracle database server (11.2.0.1 or higher - including 11gR2 XE) running on any Oracle-certified database server platform.

Between the free Express Editions (XE) of Oracle, the 18c XE has a large footprint and much more complex database architecture compared to the 11gR2 XE version. For test environments and scenario building purposes, or for collecting AAR data for a short period of time, installation and setup of the 11gR2 XE version would be much simpler.

The DDS application utilizes the Oracle GlassFish J2EE server, which, like the JTLS-GO WHIP Apache server, is delivered with JTLS-GO and requires no separate installation.

Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional details pertaining to the Oracle installation.

1.5.3 Special Installation Instructions

As mentioned above, JTLS-GO 5.1.0.0 was delivered with an old list of OTH-Gold ship types. This problem was corrected in JTLS-GO 5.1.1.0, but to properly implement this solution, users must execute the following additional procedures for each of their JTLS-GO Version 5.1.0.0 scenarios loaded in Oracle:

New OTH-Gold ship types were added to JTLS-GO, due to STR JTLS-2019-14238.

1. Execute the following command:

    cd $JTLSHOME/script/dds/version5.1/scripts/

2. Execute the following command:

    sqlplus yourScenario/OraclePassword @update_oth_gold_types.sql

3. Verify the related execution listing file under the $JDATA/scenario/ directory for errors.

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

5. Verify the .srw ascii file for your scenario.

To fix the problem renaming Force Sides, due to STR JTLS-2019-14267:

6. Reload the database.

Users who have already executed these procedures for their scenarios (including during installation of JTLS-GO Version 5.1.1.0 or 5.1.2.0) do not need to execute them again.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 5.1.3.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2019-14379 Add LC2IS XML Report Generator

Summary of Model Change Request

JTLS-GO has been using a Message Definition File (MDF) to generate Own Side Sitrep information that could be populated to the NATO Land Command Control Information System (LC2IS).

This methodology does not provide the robust information that is required to fully utilize LC2IS. LC2IS has very specific keywords for category, force, specialty, mobility, and qualification that it uses to compose a ground unit symbol. This information is not directly available in JTLS-GO. In order to provide additional information about personnel status, combat system status, and supply status the current ADAT-P3 format used in the MDF would need to be greatly expanded.

Design Summary

LC2IS requires a robust set of data to populate all the possible data sections. It would be more efficient to use a service to produce all the information, rather than have the Combat Events Program (CEP) do so.

The LC2IS Message Service (LC2MS) was developed as a web service to demonstrate the capability of feeding LC2IS via its XML file interface. The LC2MS is responsible for gathering the required unit, combat system, and supply information from the JTLS Object Distribution Authority and providing an XML file to the user, which can then be saved and input directly into LC2IS.

2.2 JTLS-2019-14412 New Message Format Specifications

Summary of Model Change Request

A Message Browser message that provides a location can print out the message in either a Lat/Lon format or an MGRS (Military Grid Reference System) format, based on the user's preference. All locations are printed out to the user's preferred coordinate system.

However, some messages may require that locations be specified in both Lat/Lon and MGRS. Given the current preference methodology, this is not possible. The Initial Imagery Report (IIR) is one such message. should be printed in an alternate coordinate system.

Design Summary

The MDFs support a LOCATION function with two or three arguments:

```
LOCATION(D1,D2,"format_string")
```

D1 is the latitude in decimal degrees, D2 is the longitude in decimal degrees, and the "format_string" is an optional format specification, handled as C string specifications.

The supported specifications are described in Table 12.4 of the *JTLS-GO Software Maintenance Manual*. If the user's preference is MGRS and the third argument is present, it is ignored. If the preference is Lat/Lon and the third argument is present, the specified format is applied. If the preference is Lat/Lon and the third argument is not present, a default format is applied.

Two format specifications for the LOCATION function have been added and are defined as follows:

- %M - Output the location in MGRS regardless of user preference.

- %L - Output the location in Lat/Lon regardless of user preference.

There are now five ways to call the LOCATION function, listed below. Note that Calls 1 and 2 are the calls that could be made before this ECP was implemented.

1. Call 1. LOCATION(D1,D2)

- If user preference is Lat/Lon, output the location using Lat/Lon as the default format.

- If user preference is MGRS, convert the location to MGRS and output accordingly.

2. Call 2. LOCATION(D1,D2,"format_string") and no %M or %L in "format_string"

- If user preference is Lat/Lon, output the location using the Lat/Lon format specified by format_string.

- If user preference is MGRS, ignore format_string, convert the location to MGRS, and output accordingly.

   Example: LOCATION(D1,D2,"%-2.2D-%-2.2M-%04.1FS%N %-3.3d-%-2.2m-%04.1Fs%E")

3. Call 3. LOCATION(D1,D2,"%M")

- Convert the location to MGRS and output accordingly, regardless of user preference.

4. Call 4. LOCATION(D1,D2,"%Lformat_string") or LOCATION(D1,D2, "format_string%L")

- Output the location using the Lat/Lon format specified by "format_string", regardless of user preference.

   LOCATION(D1,D2,"%L%-2.2D-%-2.2M-%04.1FS%N %-3.3d-%-2.2m-%04.1Fs%E")

LOCATION(D1,D2,"%-2.2D-%-2.2M-%04.1FS%N %-3.3d-%-2.2m-%04.1Fs%E%L")

5. Call 5. LOCATION(D1,D2,"%L")

- Output the location using the default Lat/Lon format, regardless of user preference.

This implementation has been applied to the WHIP Message Browser, the Message Delivery Program, and the Tech Tool.

2.3 JTLS-2019-14422 ELS Improved Timing of Moving Objects

Summary of Model Change Request

When JTLS-GO was linked with the MUSE/AFSERS simulation, the movement of entities was observed in detail. Sometimes it appeared as if the objects would take a short hop backwards and then continue moving forward. For moving objects, the Entity Level Server (ELS) always performed some dead-reckoning to simulate the movement between the larger moves made by the CEP. In addition to the dead-reckoning done by the ELS, the MUSE/AFSERS simulation also dead-reckoned the entities between location updates provided by the ELS. It was suggested that offsets in the dead-reckoning computations resulted in the irregular movement that was observed.

Design Summary

In an effort to synchronize these movements between the models, a new time attribute was added to all location updates from the ELS. The ELS recorded the time when an aggregate location update was received from the CEP. This time was accounted for in the dead-reckoning calculations in the ELS. A time was now published on the ELS Object Distribution Authority for when the ELS updated the location of an object.

2.4 JTLS-2019-14443 TACELINT Message Service - Send Email Reports

Summary of Model Change Request

The TACELINT message service needed to expand its ability to send reports to a new system that accepts reports via email.

Design Summary

The TACELINT message service was modified to use the Linux sendmail utility to send reports via email.

2.5 JTLS-2019-14466 Expand TACELINT Data Fields

Summary of Model Change Request

The PRM Record in the TACELINT Message was missing some data that could not be filled with default values.

Design Summary

The only way to solve the problem was to add some non-configuration managed data to the model. The format of the database has not changed, but the database builder can hand-edit the <scenario_name>.ncm file and place this new data in the file. The new data needed is:

- For every Sensor Type that has its ELINT Notation field filled in the database, the database builder can specify the:

  a. Pulse Duration as a text string

  b. Scan Type as a text string

  c. Scan Rate as a text string

- Similarly for every Jammer Type that has its ELINT Notation field filled in the database, the database builder can specify the:

  a. Pulse Duration as a text string

  b. Scan Type as a text string

  c. Scan Rate as a text string

The database builder should obtain the properly formatted text strings from the same organization that provide all of the other ELINT data needed by the TACELINT message

The wespac51 scenario is delivered with a wespac51.ncm data file which describes the format required to enter this needed data.

2.6 JTLS-2019-14467 Additional HRU Definition Flexibility

Summary of Model Change Request

Because of NATO requirements for a unique NIC for each country to feed LOGFAS, country-specific HUPs are built that specify which Combat Systems are required. For example, two separate HUPs are needed for an American 4 Man Mechanized Recce Team and a British 4 Man Mechanized Recce Team, because the Americans use Bradley Fighting Vehicles, while the British use Avenger APCs. This causes the number of HUPs that exist in an exercise to become unmanageable.

JTLS-GO requires some means by which to create an HRU without having to find a HUP for a specific country.

Design Summary

A new Create HRU option has been implemented, called "Create with customized HUP". This option creates an HRU with a customized list of Combat Systems, Supplies, and Targets. The user specifies the parent unit and HUP as normal, but only the HUP's general characteristics, such radius and covert capability, are used. The HUP's assigned Combat Systems, Supplies, and Targets are ignored. The user must specify the desired detailed capability for the HRU.

In order to simplify the option, it is possible to not enter the customized supply list on the order panel, but to have the model compute the needed supplies for the HRU.

2.7 JTLS-2019-14468 Create BDA IIR Message

Summary of Model Change Request

Users should be able to differentiate between Initial Imagery Reports (IIRs) that report Order of Battle (OB) information, and those that report Battle Damage Assessment (BDA).

Design Summary

The current IIR message 5502 was copied to create a message 5503. When generating an IIR, the model determines if there is any damage to report in the IIR. If there is, a 5503 BDA IIR message is generated. If there is no damage to report, a 5502 IIR message continues to be generated.

3.0 SOFTWARE TROUBLE REPORTS

This chapter summarizes Software Trouble Reports (STRs), which describe software code errors that have been discovered by JTLS-GO users or developers and have been corrected.

Chapter 4.0 describes STRs that remain outstanding from previous versions; however, because of the model enhancements included with JTLS-GO 5.1.0.0, no STRs identified in prior JTLS-GO series are considered valid. Uncorrected errors identified for JTLS-GO 5.1.3.0 are documented in this chapter and should be corrected in future maintenance releases.

3.1 JTLS-2019-14348 JINN Fails To Open From DDS

When the JTLS-GO Order of Battle Editor (JOBE) Interface Navigator (JINN) attempted to access images for needed icons, the program crashed before the interface became visible.

The JINN code was attempting to load image files embedded in the jinn.jar file using unacceptable characters for the Java class loader, throwing an exception. The unacceptable characters were removed from the reference to the resources. Additionally, any exceptions that occur during loading are logged but otherwise ignored, allowing the JINN to start with the missing resources.

3.2 JTLS-2019-14349 Zero Range Accepted For ADA/Sensor Targets

The Controller Set Target Parameter order permits a Target Distance of zero to be entered for Surface-to-Air (SAM) targets and Sensor targets. The model incorrectly interpreted this order. This order does not mean to set the range to zero; instead, it is designed to indicate that the target should assume the maximum range associated with its defining prototype.

While investigating this issue, the Controller Set ADA Parameter order was submitted to change the range in a specific altitude zone for an entire ADA target class. The change was accepted and the range ring display updated correctly on the WHIP. However, an inspection of the underlying code revealed there was no logic to determine if the individual targets' effective range (TG RANGE) should be changed as a result. In addition, no logic was present to adjust the terrain grid tags that denote which grids are covered by each target's effective range (TG RANGE).

The CEP code was changed to properly interpret the order to set the Target Distance to zero.

New logic was added to update the TG RANGE and grid tags for each target within the class when necessary. If, as a result of changing one (or more) altitude zone ranges, the maximum range (over all of the altitude zones) changes as well, the TG RANGE is assigned the new maximum range value if:

- The old TG RANGE was equal to the old maximum range; or

- The old TG RANGE was less than the old maximum range and the old TG RANGE was greater than the new maximum range.

Under all other situations, the TG RANGE does not change.

Finally, if the TG RANGE value has indeed changed, new code was added to adjust the target's grid tags to reflect that change.

Note: The TG RANGE for an individual ADA target, also known as the "Limiting Range" when it is less than the max altitude zone range for the target class, has been added to the WHIP display in the context sensitive menu for a specific ADA target.

3.3 JTLS-2019-14358 SVP Error/Warning Values Truncation

Some values reported by the Scenario Verification Program (SVP) are less than the actual value needed to correct an Error or Warning.

When the SVP reports values in some Errors and Warnings, the output is truncated to some number of significant digits, so the amount reported is actually less than the actual number. Code as been added to numerous Errors and Warnings within the SVP to increase the least significant digit by one to account for any truncation problems.

3.4 JTLS-2019-14359 ICP Incorrectly Checks C4I Connection Type

The Interface Configuration Program (ICP) allows users to modify the C4I Connection Type of the JTLS-GO Operational Interface (JOI) services. The OTH Gold JOI is not capable of a UDP connection type, and the ICP would check for this value and disallow it for OTH Gold. The checker was incorrectly inspecting data in the ICP and either failing to report errors, or reporting errors when there were none.

The C4I Connection Type checker for OTH Gold in the ICP was modified to correctly read from the ICP's data.

3.5 JTLS-2019-14360 WHIP/DDSC UOM Preference Search

WHIP/DDS drop-down menus for Supply Category and Basic Unit Of Measure (UOM) showed a list of UOMs, but did not have a search function.

WHIP and DDS code were modified to implement UOM search capabilities, with a default search mode of searching by short name.

3.6 JTLS-2019-14361 Technical Coordinator Guide Update

The JTLS-GO Technical Coordinator Guide was out of date and did not discuss the multiple-sided WHIP configuration, the new Force Side selection check boxes, the new menu file available for a

multiple-sided WHIP with a Controller, and the new command authority options when discussing the configuration of WHIPs in the ICP.

The Technical Coordinator Guide was updated to reflect the latest changes to the WHIP tab of the ICP in JTLS-GO 5.1.

3.7 JTLS-2019-14362 WHIP Countrycode Preferences Not Saved

When WHIP preferences were changed, corresponding properties files were updated. However, the update did not proceed until either the preferences component window or the WHIP window were closed. Furthermore, when only Country Code preferences were changed, the countrycode.properties file did not update after the WHIP window was closed.

WHIP code was modified to save preferences properties files whenever a change is processed, rather than when the preferences component window or the WHIP window are closed.

3.8 JTLS-2019-14363 Improve Error 409

None of the corrections in SVP Error 409 worked.

Error 409 is used to compare an aircraft's required takeoff and landing distance against the runway length for the squadron's home base. Error 409 was built to present all the combinations of options for takeoff and landing lengths against runway lengths for POT or standard runway targets. This made the corrections confusing and presented corrections which were not applicable.

To clear up the confusion, Error 409 was removed and four new errors were created:

- Error 452 corrections are for aircraft takeoff length and TUP POT created runways.

- Error 453 corrections are for aircraft takeoff length and standard runway targets.

- Error 454 corrections are for aircraft landing length and TUP POT created runways.

- Error 455 corrections are for aircraft landing length and standard runway targets.

3.9 JTLS-2019-14364 WHIP Show Raw Message Data Option

The feature to show the raw message details of a message in the Message Browser using a keyboard-mouse combination did not always work properly.

It is possible from a WHIP Message Browser to view not only the formatted message but the raw message contents by pressing the Control and Alt buttons, and then clicking on a Message subject. This three click requirement did not always work.

The raw message detail feature was corrected by changing the default behavior of the keyboard-mouse combination. This enables displaying the message data using any combination of the three character requirement the user may attempt.

3.10 JTLS-2019-14365 Periodic Munition Report Issue

The Periodic Munitions Report displayed the weapon counts that had been expended during the previous reporting period, as well as Supply Categories available at units (e.g. airbases, FARPs, naval units, ground units, etc). However, the report included non-weapon Supply Categories. The report should include only Supply Categories that are sources for targetable weapons.

The routine that generates the Periodic Munitions Report had no logic to exclude non-weapon supply categories. New logic was added to filter out non-weapon Supply Categories from the report.

3.11 JTLS-2019-14366 DDS/JOBE Expand/Hide Map Side Panes

It was difficult to expand and hide the map side panes in the map components of the DDS and JOBE.

The WHIP's map component allows the user to expand and hide the side panes easily by using accessible buttons embedded in the pane dividers. This same solution was added to both the DDS and JOBE maps.

3.12 JTLS-2019-14367 Convoy Graphics Symbol Zero Crashes OPM

If the database does not specify a graphics symbol for any of the SLP convoy types, the Online Player Manuals (OPMs) will not generate. The process crashes.

It is legal but unusual to not specify convoy graphics symbols, and the OPM did not check for the situation. The check has been added and the OPMs are now properly generated.

3.13 JTLS-2019-14368 Simulation Time Decrease In Move Missile

A cruise missile was told to orbit prior to heading towards its assigned target location. It is believed that the missile arrived at its orbit area at the exact time it had to leave the orbit area due to fuel requirements. The computation resulted in the missile computing its next move time as a negative number, causing the CEP to crash.

This situation was accounted for within the logic. A check was made to ensure that the next move time will always be positive.

3.14 JTLS-2019-14369 MCP Greater Than Barrier Types Crash

The OPM generation procedure crashed if the number of Mobility / Counter-Mobility Prototypes (MCPs) is greater than the number of Barrier Types.

The identifier index for an MCP was used to access the name of a Barrier Type. There was no outward indication of this error because it was simply used as a hidden tag for use within the OPM. The OPM worked properly but the hidden tag name was supposed to reference the MCP name, but referenced the Barrier Type name instead.

The typographical error was corrected and the hidden tag name now references the MCP name.

3.15 JTLS-2019-14370 Minefield Create In Future Checkpoint Crash

If the user entered a Create Minefield order for the future, and did not specify a Political Country on the order panel, the CEP would crash saving the future target create. Political Country is an optional field, so the order was legal and the crash should not have occurred.

There are several order panels that the Controller can use for creating targets during game execution. The Minefield Create order panel is unique in that for various reasons, the order within the CEP needs to be initialized, meaning that the unfilled order fields contain a negative one value. This negative one value was placed in the Minefield's attribute TG TARGET CATEGORY. This caused the CEP to crash. The proper minefield creation checks were made throughout the code to ensure that this does not happen.

3.16 JTLS-2019-14371 Detected SSM Launch From ADA Target Crash

The CEP would crash attempting to send the SSM Report. The crash only occurs if the type of Air Defense Class has an identifying number greater than then number of SSM Types.

This issue was caused by the new JTLS-GO 5.1 capability to use an Air Defense Target to fire like an SSM. The SSM Report was accessing an SSM Type characteristic to determine the firing detection mode, but it was an Air Defense Target that was firing the weapon. The problem was corrected.

3.17 JTLS-2019-14372 ADA Fires SSM While In Port Crash

The CEP crashed when the user told a naval unit in port to fire an air defense weapon as a SSM.

The logic was accessing an SSM target's in-port firing allowed attribute, not the Air Defense site's in-port firing allowed attribute. This crash only occurred if the index number for the firing Air Defense Class was greater than the number of SSM Types in the database. If this was not true, the CEP would not crash, but the results of the check were most likely incorrect.

The logic was expanded. If an Air Defense site attempts to fire an SSM against a target while in-port, the Air Defense Class AD PORT FIRING ALLOWED attribute is checked. If a SSM site attempts to fire an SSM while in-port, the SSM Class SSM PORT FIRING ALLOWED attribute is checked.

3.18 JTLS-2019-14373 ADA Limiting Range for Created Target JODA Error

A Controller Create ADA Site order was submitted with a specified range less than the maximum altitude range for the ADA class. This limiting range ring was displayable on the WHIP via the context sensitive menu, but a JODA error was generated in the Web Services Manager.

A sequencing error in the logic attempted to publish the newly created ADA target's limiting range ring before the target itself had been updated. The JODA was not yet aware of the new target's creation and threw the error message. Later, when the JDS was notified of the target's presence, the attempt to publish the limiting range ring was successful. To eliminate the JODA error, the initial publish attempt was removed from the CEP in two routines.

3.19 JTLS-2019-14374 Fortify Scan Corrections

The Fortify scan discovered issues with possible XML External Entity vulnerabilities and possible unclosed stream resources.

Code was modified to mitigate the reported issues.

3.20 JTLS-2019-14375 Add To Dump Crash

The CEP crashed when a unit, that was being removed from the game, attempted to add its supplies to an available supply dump in the area. The issue is that the CEP collected all targets within range of the dying unit and then accessed these targets. One of the accessed objects was not a target, but was a unit. The only way this could happen is if the set used to hold the objects was not empty when the process logic started.

The dying unit owned a stationary SSM target. The CEP needed to find a new owner for the target and the surround object set was filled with all applicable units. The set was never cleared. When the dying unit, which was not within range of any units, needed to dump its supplies, the CEP looked for suitable supply dumps in the area. Since the surrounding object set was not cleared, the CEP crashed.

Several steps were taken to fix this problem and review the code to determine if there were any similar problems:

1. The logic when reassigning the stationary SSM site was corrected.

2. All instances of using the SURROUNDING OBJECT SET were reviewed and several other issues were uncovered. Some memory leaks were found and fixed. In addition, some inefficient code was corrected.

3. There are currently several places in the code that the CEP checks if the SURROUNDING OBJECT SET is being used before attempting to use the set. This code did not always properly handle the situation. Under these circumstances a centralized routine was created to properly catch and solve the issue.

3.21 JTLS-2019-14376 IMT Bad UOM Supply Category Designations

Some IMT fields had their Supply Category specified as Special Supply Categories, causing the amount displayed as either "0 n/a" or "0 Error". Because there was now not a one-to-one correspondence between the Special Supply Category and the actual Supply Category, the translation could not be made properly.

The generic Supply Category UOM function was removed from the references to Class III and Class IV type supplies within IMT definition files. These references are now hard-coded "WET_WEIGHT" for Class II supplies and "DRY_WEIGHT" for Class IV supplies within all IMT definition files.

3.22 JTLS-2019-14377 Catastrophic Kill Events

An extremely large number of Catastrophic Kill entities were being created, and had a persistence time of 30 days, causing the number of entities to grow and negatively impacting CEP performance. Since the entities were not being used in any intelligence reports, it was decided to reduce the amount of time they persisted to one second. This terminated the growing number of these entities, but because the events to destroy old entities were not adjusted for the new destruction time it did not clear the existing entities.

Changing the Catastrophic Kill Time should result in an adjustment to both existing entities and new entities.

The code was modified so that if the Controller changes the Catastrophic Kill Time, the execution time for all existing Wreckage Disappear events are also adjusted. The help for the Controller order was modified to inform the user of this effect.

3.23 JTLS-2019-14378 JHIP Multiple Objects Selected For Display Crash

The JTLS HLA Interface Program (JHIP) crashes when the user clicks the Object List button, after the Graphical interface has allowed the user to also select more than one object types from the main window.

The graphical library used to render the list of objects types from the HIP can allow the user to select more than one object type. When this happens, the HIP may also be directed to display the list of objects for the more than one object type that has been selected. The result caused the HIP to exit because it executed an assert command intended to ensure the number of object types was less than two. This problem was corrected.

3.24 JTLS-2019-14380 HRU Has No Mines To Lay Crash

The CEP crashed when a user gave an HRU an order to lay mines and the HRU had no mines to lay.

Naval Units and Ground Units check whether they have mines prior to attempting to execute a lay mine task, but HRUs did not have a similar check. HRUs now do the same checks as Ground Units and Naval Units before attempting to lay mines.

3.25 JTLS-2019-14381 No Arc In Optimal Road Route Crash

The CEP crashed when attempting to create the planned route for an optimal road path that was found. The database had either a very large time for a node delay or a very large specified arc distance. Although a route was found, the estimated time to complete was the maximum value that could be held in a real variable. This caused the crash.

The code was changed to allow an optimal route, even if travel time is estimated to be infinite. The computed travel time will then become an execution issue and will not crash.

3.26 JTLS-2019-14382 Crash When Moving Unit In TGF Dies

A unit in a tactical ground formation (TGF) died while moving. When the unit died, it was removed from the TGF, and the logic did not check for this possibility, causing the CEP to crash.

The logic now checks for this situation.

3.27 JTLS-2019-14383 Mine Laying Premature Ending Message

An HRU was executing a mine-laying task when it ran out of supplies to continue. A message was generated to notify the player the task had ended prematurely, but the message was a Naval message instead of a Ground "cannot comply" message.

The emplace mine logic was modified to generate a Ground message for both ground units and HRUs, and a Naval message for naval units when the mine-laying task ends prematurely.

A similar problem was found when a naval vessel does not have enough supplies on board to lay all the specified mines at the beginning of the mine-laying task. Under this situation a Ground message was generated instead of a Naval message. The mine lay check logic was modified to generate a Naval message if the tasked unit is a ship.

3.28 JTLS-2019-14384 No JOBE Directory Causes JINN Start Failure

If there is no JOBE directory under the scenario directory, the JINN fails to start, incorrectly reporting that the JINN is already running and the scenario is locked.

The servlet was not checking if the JOBE directory existed. If the directory did not exist, lock file creation would fail, which the previous logic treated as if the lock file already existed. The servlet now checks to ensure the JOBE directory exists.

3.29 JTLS-2019-14385 Diverted Mission Crash-Lands Multiple Times

An air mission, which was heading home, encountered problems with its primary landing site. Due to runway cuts, there was insufficient length for the aircraft to land. The mission attempted to divert, but no alternative landing sites were available.

The mission proceeded to crash-land at its current location. No aircraft were destroyed in the first attempt to crash-land, but it repeated this crash-landing process five more times until the mission was destroyed. Only one attempt to crash-land should have been made.

The landing process used a random draw to determine how many aircraft in a mission were destroyed during a crash-landing. It was normal for a mission to survive a crash-landing when there was still a partial runway at the landing site. However, when the mission survived the crash-landing with at least one aircraft, the decision-making tree in the code never allowed the mission to just land. Instead, the mission was caught in an execution loop where it went back to heading-home, then proceeded to crash-land again.

Code changes were made to complete the mission after the first attempt to crash-land. For clarity, changes were also made to the Mission Report to reflect the lost aircraft counts.

3.30 JTLS-2019-14387 DCP Ports In Use Message Issue

When the assigned Database Configuration Program (DCP) ports were in use and the DCP started, a message popped up warning the user that some ports in this range were currently in use. Selecting "Ok" on the message should stop the DCP configuration saving process, but saving continued until the message popped up again.

The code was modified to stop Glassfish startup if assigned ports are already in use.

3.31 JTLS-2019-14388 Encode/Decode XML Files

XML standards specify a small set of characters that must be escaped when using them as attribute or node values. The Scenario Initialization Program (SIP) generated a corrupted XML file for LOGFAS, because it did not escape one of these characters. The character itself originated in the description fields of NIC and RIC data in the JTLS-GO database.

An encoder and decoder were added to the Simscript XML code to encode ASCII characters that require escape sequences in XML during a file write, and to decode XML escaped sequences during an XML file read. The data held by the SIP/CEP is still the unescaped ASCII characters.

3.32 JTLS-2019-14389 XMS Withholds Messages From Super WHIP

A Super WHIP subscribing to the Controller side and any other side will not receive side broadcast messages.

The XMS receives a request from a Super WHIP containing the name of the WHIP and some optional other parameters, but without other Side information. For a standard WHIP registered to a single Side, Side broadcast messages are gathered in a reply using the primary subscribing Side for the WHIP. However, for a multi-sided Super WHIP, the XMS needs to gather broadcast messages for all Sides under which the WHIP is subscribing. This required modifications to the XMS processing code as well as the XMS Apache module.

The URL request from a WHIP may now have a "_sides" parameter, allowing the WHIP to specify a list of perception Sides for the request for messages. The XMS was modified to process this sides list while gathering the reply messages. With this change, a Super WHIP will now send the URL request with the associated side perceptions to get all corresponding messages.

3.33 JTLS-2019-14390 Reformat Spreadsheet Text Field Crash

The Reformat Spreadsheet Program (RSP) was crashing when saving a default space-allowed text field to a control file, and then using that control file to process a spreadsheet. The problem was that the RSP was not saving the text field within quoted strings, but was attempting to read it as if it was a quoted field.

The immediate error of not saving the space allowed default field in quoted strings was solved. There were also a number of minor issues that were addressed at the same time, such as informing the user where the spreadsheet file should be located, allowing the spreadsheet file name to be mixed case, and verifying the force side name provided to the RSP for accuracy.

3.34 JTLS-2019-14391 Order Duration Error Prints Improper Range

An order duration field had a minimum value of 30 Minutes and a maximum value of 99 Days. When the user entered 5 Minutes into the field and attempted to submit the order, the OVT replied with an error stating "Duration Value 05M is Not Between The Allowed Min of 05M and Max of 05M.", instead of providing the proper minimum and maximum values.

The error was tracked to the OVT library, which uses a rotating set of texts to send replies when converting the duration as decimal days into a text. The purpose of the rotating set of texts is to avoid a message that requires three conversions from overwriting the subsequent answer, but an error in the rotation caused the three different values in the full reply to all point to the same value. This was fixed.

3.35 JTLS-2019-14392 Javamenu Count Critical Order Names List

When an Order Type is selected in the Count Critical Orders Menu in the Javamenu, the user enters an order name in the text field. The program did not verify whether the order name was legal.

A script was created to extract existing order names from the order XML files and create a list that would be used on the Count Critical Orders Menu instead of a text field.

3.36 JTLS-2019-14393 JODA Limits Object Name Length To Twenty

Whenever a display command is entered in a JODA terminal for a SUPPLY_CATEGORY object, the 25 character name is truncated to 20 characters.

The JODA executes a built-in print routine when it displays an object to the command console. The print routine uses a default string length whenever a string is printed such as the name of a SUPPLY_CATEGORY. The default length was set to 20 characters, and would cause the print routine to truncate the name string. This has been changed to 40 characters, which is the limit for any string variable in JTLS-GO.

3.37 JTLS-2019-14394 Transport Unit Convoy Cancel Message Issues

A Transport Unit order was submitted for a truck convoy. The convoy traveled to the pickup location. When the convoy attempted to load the unit, the convoy was canceled because it did not have enough capacity to carry the entire unit. The player message was confusing because it included two irrelevant text lines: "The convoy is already carrying other supplies" and "Formation does not have sufficient lifting capability".

The irrelevant text lines were removed from the cancel message. Also, a new message was created for the situation where the convoy is already carrying supplies and can't load a unit to transport.

Note: The logic calculated the correct numbers and types of trucks required to load the unit before leaving the support unit. The convoy should have been able to load the unit upon arrival. This inconsistency is under investigation.

3.38 JTLS-2019-14395 Coalition Support HRU Water Attribute Not Set

When an HRU stops performing Coalition Support and jumps off a ship, it does not properly set the attribute called HRU FEET WET, indicating that the HRU is in the water.

The issue was fixed.

3.39 JTLS-2019-14396 Air Refuel Instructions Lack Transit Point Option

The Air Refuel Instructions Utility for the Air Refuel Order offers Alert, Orbit, and Landing Point tasks. Unlike other mission types, the Transit Point task is missing from the Utility and should be added.

The missing Transit Point task was added to the Refuel Mission Instructions Utility.

3.40 JTLS-2019-14397 Added Game Info To WHIP Debug Console

When the WHIP Debug Console and Exception Display Frame were open, their frame title bars did not show either the scenario name or game color.

A border that containing the scenario name and game color was added to the Debug Console and the Exception Display Frame.

3.41 JTLS-2019-14398 TRIPP Step By Packet Event Too Granular

The TRIPP's capability to step by packet event was too granular for users to see incremental changes between steps. As such, users would click many times and not see any changes to the map and other information available on the TRIPP.

The step feature was changed to step by increments of game time seconds. This allows users see more changes between steps.

3.42 JTLS-2019-14399 Starting HIP Without EODA Crash

Starting the HIP, when configured to subscribe and publish entities from the ELS, but with the EODA button deselected, will result in the HIP crashing.

When the HIP is configured to subscribe to SUPPLY_CATEGORY objects from the JODA, it will try to merge the SUPPLY_CATEGORY data with EODA objects, which may also be part of the HIP subscription set. If an operator deselected the EODA as a data source, the HIP would still attempt to reference the corresponding EODA objects, resulting in a crash.

The subscription login in the HIP has been modified to exclude the SUPPLY_CATEGORY objects whenever the EODA is excluded as a data source.

3.43 JTLS-2019-14401 AARC Crashes In File Save Mode

When the AAR Client (AARC) was run in a mode to save AAR information to data files for later loading into a database, the AARC would crash without saving any data.

When checkpoint recovery and filtering capability was added to the AARC, two calls were introduced to access the database. These calls should have been inside a check to ensure database mode, not data file storage mode, was being used. The code was changed to properly perform this check.

3.44 JTLS-2019-14402 SVP Aviation Fuel Check Ignores Ground Units

The SVP verifies that aviation-related units have aviation fuel on-hand, but omits ground units that possess aircraft (e.g. UAVs).

Ground unit types with a non-zero UT AIRCRAFT TYPE attribute were added to the check in the SVP logic.

3.45 JTLS-2019-14403 MISREP Improper UOM Constant Use

The MISREP contains a summary of pilot observations after an OAS mission is complete. In that section of the MISREP, estimates of supplies destroyed are presented, including fuel, other liquids, and dry goods. The message misused a Unit of Measure (UOM) global constant that hard-coded the measurements to only gallons and tons.

The global constants WET.UOM and DRY.UOM were removed from the routine that generates the pilot summary. The MISREP was corrected to use the UOM functions to display the proper units of measure as chosen by player preferences.

3.46 JTLS-2019-14404 Isle Of Man Country Code

The Isle of Man Country Code was not an entry in the Configuration Managed Country_Codes.xml file, which permits C2 interfaces to translate between the database country code standard and other standards.

The data for the Isle of Man was added to the file.

3.47 JTLS-2019-14405 Terrain Layer Argument Type Mismatch

In the CEP processing of terrain grid data, a subroutine calculates the number of higher grids within a single grid. The terrain layer arguments passed from the calling routines were integers, but the subroutine defined them as pointers. Although this mismatch did not cause a crash, use of pointers when integers will do is inconsistent, inefficient, and misleading.

The variable definitions were changed from pointer to integer type.

3.48 JTLS-2019-14406 Provide BE Facility Initial History

If a Unit or Target is included in another Force Side's BE Facility, the object is automatically detected at the beginning of the game. The database indicates that the BE Facility's Force Side knows about the object, so it should be immediately detected at game start.

However, the information about the Unit or Target is not added to the detected history information from the BE Facility. The first time that the owner of the BE Facility goes to collect on the BE Facility, the resulting message indicates that this is an initial observation and all of the objects seen are new. This is incorrect and confusing.

When the initial object detection is provided to the Force Side of the BE Facility, the BE Facility's detection history records are now also initialized and filled.

3.49 JTLS-2019-14407 JSAT Error Reading Global File

The JTLS Satellite Service (JSAT) did not compute orbit points for some satellites.

The scenario global file contains the start time for the scenario, but the JSAT was not updated to account for a past change to the format of the global file. This caused the JSAT to use an incorrect start time and made it unable to compute the orbits for some satellites.

The JSAT code was modified to correctly read the scenario global file, and the orbit computations now utilize the proper start time for the scenario.

3.50 JTLS-2019-14408 Missing Delivery File

JTLS-GO installation files include many configuration and initialization-specific files. A new file was added to allow the user to perform analysis of a simulation run. The file is located at the $JGAME/data_site/ directory, but was missing from the installation files.

When JTLS-GO is delivered, an automated script assembles all the necessary installation files, by accessing a list of files and directories that should be included in the installation files of the release. The list of files used by this script was missing the "user_collection.example" file, so the automated script was not including this file in the complete set of installation files. This file has been added to the list for the automated script.

3.51 JTLS-2019-14409 Unneeded CSP Argument

The routine EMBARKED AS SUPPLIES had an argument that was no longer needed. The variable represented the Combat System Prototype of the embarked unit.

The unneeded variable was removed.

3.52 JTLS-2019-14410 Create HRU Rejection Message Missing Name

If an HRU cannot be created, the CEP sends out a rejection message. This message was missing the name of the HRU if the HRU was part of the initialization database.

The name is now included in the message.

3.53 JTLS-2019-14411 Improve IIR Format

Several changes to the IIR message were needed, such as that BE Targets need to be reported as individual items.

Changes to the IIR message were made.

3.54 JTLS-2019-14413 Unconfigured JSAT Startup GUI Crash

To run the JSAT graphical user interface (GUI), used to generate the orbital data for satellites to be represented in the CEP, the JSAT service must already be running. A user tried to start the JSAT GUI before the service was configured to run. This caused a crash of the JSAT GUI.

As part of its normal operation, the JSAT GUI read the configuration file for the JSAT service, which is produced by the ICP. Because the JSAT service was never configured to run, there was no JSAT configuration file, causing a crash trying to read a non-existent file.

The code was modified to first check that the configuration file existed before attempting to read that file. If the configuration file is missing, the program displays an error message, and then exits cleanly.

3.55 JTLS-2019-14414 JOI Country Code Outgoing Setting In ICP

The default value in the ICP for Country Code Outgoing for JOI services was set to "NONE". This was an invalid value for the field and should have defaulted to a valid value.

The default value for Country Code Outgoing has been changed to "ISO-3166-1-TRIGRAPH", which is the country code format used to build the default databases.

3.56 JTLS-2019-14415 Ship SITREP Has Bad Ship C3I Type

Clicking on some ships on the WHIP would throw an Exception for a Ship C3I Type vocabulary item being out of range.

The Ship C3I Type is no longer used; instead, the new OTH_GOLD_Type vocabulary item should be used. The Ship C3I Type was still referenced in the WHIP SITREP definition file and in one IIR message. The Ship C3I Type was replaced with the new OTH_GOLD_Type.

3.57 JTLS-2019-14416 Satellites Not Collecting On Some DSAs

An orbiting satellite was not collecting on Directed Search Areas (DSAs) which were directly under the path of the satellite.

The orbital points for the satellite were straddling, but not falling within, the DSAs. The CEP only performed collection when the satellite moved to a new location. Since satellites move quickly, the points which define the movement route were far apart. Because the positions of the route points were not located within the DSAs, no collection took place.

The code was modified such that satellites moved in a time interval which is consistent with the range of their on-board sensor. The interval between points was computed such that there were at least three points covering the area of the range of the sensor. This computed interval ranged from a minimum of one second, up to a maximum of one minute. The orbital data were generated as before, but with an appropriate interval between the points. The user no longer specifies the point time interval on the JSAT GUI.

3.58 JTLS-2019-14417 MDP - Allow Emails Without Domain Extension

The MDP service and MDP GUI did not allow users to input email addresses that were missing the domain name extension (e.g., ".com"). This limited users with varying network and email configurations to take advantage of their setups.

The MDP Service and MDP GUI were modified to allow email addresses to be input without domain name extensions.

3.59 JTLS-2019-14418 DCP Help Typo Fix

Typos existed in the DCP Java Help.

Typos in the DCP Java Help were corrected.

3.60 JTLS-2019-14419 Batch Mode CEP TRIPP Directory Traceback File

Whenever the CEP is run in batch mode, it tends to leave a TRACEBACK.LIS file in the TRIPP directory of the scenario for which the batch is being run. Having a stray file in the TRIPP directory causes the Logging JODA to terminate.

The CEP leaves behind a TRACEBACK.LIS file when execution has stopped. The file is left in the directory from which the CEP was started. From the script program that executes the various processes during a batch run, the order of commands was out of order with respect to cleanup of files from the previous CEP run and starting the CEP for the next run in the batch series.

3.61 JTLS-2019-14420 Crash Destroying Perceived Unowned Runway

The CEP crashed attempting to tell the user it would not accept an order for an ARU to destroy a runway that was perceived as unowned.

Unlike an HRU, an Aggregate Resolution Unit (ARU) is not allowed to accept an order to destroy a Target owned by a foreign side unit. The concept is that the ARU must fight and kill the foreign unit that owns the Target before the Target can be killed.

A user told an ARU to destroy a runway. The ARU's Force Side perceived the runway as unowned, but it was owned. When the logic checked whether the destroy target order was legal, it asked "Is the Target owned?" The answer was "Yes", and so the order was rejected, and while attempting to tell the user why the task was being canceled, the CEP crashed. The crash was caused by accessing the relationship between the ARU and the ARU's perceived Force Side for the Target. The ARU perceived the Target as unowned, and the concept of "relationship" does not exist between a side and the unowned side.

The problem was solved by not asking whether the Target was owned, but asking whether the ARU perceived the Target as owned. The task was then accepted and the ARU moved to the location of the runway, and entered combat with the owner of the runway.

3.62 JTLS-2019-14421 Batch Run Periodic Reports Lost

While running in Batch Mode, the CEP is told when to complete a run when the user enters a Stop Checkpoint Order. If the Stop Checkpoint Order includes the option to generate ending period reports, those periodic reports are lost.

The issue was that the periodic reports were generated and placed in the next checkpoint directory instead of the stop checkpoint directory. When the batch process started the next run, the next checkpoint directory was cleared and the messages were lost.

The checkpoint logic was reorganized and the periodic reports are now generated prior to closing out the current message directory and starting the message directory for the next future checkpoint.

3.63 JTLS-2019-14423 Spherical Library Incorrect Value

The Spherical Library code, which provides functions for determining great circle arcs, intersections, etc was using an incorrect value for the number of degrees per Radian. This would result in errors up to 4 meters in some great circle calculations.

The variable in question was changed to use the correct value.

3.64 JTLS-2019-14424 User Collection Script Batch Run Data

At the end of each run iteration during a batch run of the game, the user collection script is called. This is a script that users can modify to provide any extra data analysis and collection that they require. Nothing was provided to this script to help determine at what step in the batch run the user collection script was called.

The user collection script is now provided with the name of the scenario being run and the number of the last checkpoint that was successfully taken in the batch run.

3.65 JTLS-2019-14425 Newly Created Unit Command Authority

When a new unit is created, its Higher HQ is assigned as the Force Side commander. When the TPFDD order is submitted for the new unit, the unit will copy the command authority of the Higher HQ specified on the TPFDD order. However, if the TPFDD order does not change the Higher HQ, leaving it as the Force Side commander, the command authority is not copied, and the unit arrives in the game with only the authority of the primary player to control it.

The code was changed so that when the unit is created, it copies the command authority of the Force Side commander. If this Higher HQ is later modified with the Manage TPFDD order, this command authority will be discarded and the unit will copy the command authority of its new Higher HQ.

3.66 JTLS-2019-14426 SIP Verification Crash

If a database includes an Error 971, which indicates that a River Arc could not be assigned an appropriate barrier type, the verification process could crash attempting to check whether a support route existed between a unit and its support unit.

The crash occurred because the SIP was trying to determine whether a cargo truck could cross the bad river arc. The river arc was between a unit and its support unit. The SIP was changed to consider the possibility that the river arc, which needed to be crossed, was not assigned a barrier type. The improved logic concludes that the cargo truck cannot cross the river arc

3.67 JTLS-2019-14427 HRU Mine Clearing Not Working

HRUs did not recognize the existence of a minefield in the designated clearance area.

The code that determined if a minefield was in the clearing area did not properly interpret the minefield type mask that was passed into the selection routine.

3.68 JTLS-2019-14441 JTOI TBMCS/NECCCIS Read Color File

The color file read by the JTOI was changed, causing errors for the JTOI when trying to read the unexpected format.

The TBMCS and NECCCIS JTOIs have been updated to read the new color file format.

3.69 JTLS-2019-14442 JTOI ICC Airbases

The JTOI was not setting the reason code and limiting factor for airbases upon their creation in ICC.

The JTOI now sets the reason code and limiting factor for airbases when they are created in ICC.

3.70 JTLS-2019-14444 Delete Old LOGFAS Profiles On Setup

Performing a setup using the SIP generates LOGFAS profiles. The profiles that are generated depend on settings in the database. When a user removes profiles from the generation list in the database and performs a setup on an existing scenario, previously generated and out of date profiles would remain on the system.

The setup process now removes any existing LOGFAS profiles before the SIP is executed. This prevents stale LOGFAS profiles from remaining on the system.

3.71 JTLS-2019-14445 ASCII Save Incomplete Combat System Name

The ASCII save logic was not saving the full Combat System names. The maximum name length was changed to 25 characters and two places in the code were not saving the full 25 character names. This means that the Combat Systems would be lost on a restart.

The write was corrected to output the full and maximum combat system name length of 25 characters. A thorough review of the code was conducted and two other places were found to be outputting fewer than the required 25 characters.

3.72 JTLS-2019-14446 Error 414 SSM Supplies Availability

SVP Error 414 uses only the information from the last targetable weapon when building the correction data.

Error 414 looks at all the weapons in the SSM targetable weapon list to determine if the SSM has enough supplies for any of the SSM targetable weapons to fire at least one shot. As reported, if it cannot find enough supplies for any of the targetable weapons, it only reports the information for the last targetable weapon in the list.

Error 414, which was renumbered 1466, will now report a warning for every targetable weapon in the SSM targetable weapon list for which the unit does not have enough supplies to fire at least one shot.

New Error 449 was added. It will report if the SSM does not have any supplies available for any of the targetable weapons in the list.

Error 450 was renumbered to 447 and Error 448 was renumbered to 450 for consistency in the numbering sequence.

3.73 JTLS-2019-14447 HUP Combat System With No Explicit Weapon

An HRU Combat system was not functioning properly.

An attrition capable Combat System referenced in a High Resolution Prototype that did not have a HIGHRES EXPLICIT TARGETABLE Weapon.

A new WARNING 1117 has been added to the SVPR. It checks each attrition capable Combat System assigned to a HUP to ensure the Combat System has a HRU EXPLICIT TARGETABLE WEAPON assigned. Attrition capable Combat Systems are those that have DIRECT, INDIRECT or LONG.INDIRECT attrition types.

If the error is generated the user will have the choice to:

     a. Delete the Combat System from the HUP.

    b. Assign a EXPLICIT TARGETABLE WEAPON to the Combat System.

    c. Change the Combat System attrition type to NONE.

3.74 JTLS-2019-14448 DDS And SVPR Minor Corrections

Several minor changes were needed to the DDS and SVPR.

1. The TARGETABLE_WEIGHT field was realigned to be next to the TARGETABLE WEAPON SUPPLY CATEGORY field on the DDS Targetable Weapon Screen.

2. An automatic correction function was added to SVP Error 725, to automatically set the terrain depth to support the SUP, provided the terrain type is OCEAN, SMALL_ISLAND or DUAL_CAPABLE.

3. A new correction option was added to Error 403, to move the runway to the airbase location.

4. Two new correction options were added to Warning 1201. The first correction will automatically set the TUP CARRY CAPACITY WET to what is needed based on the BASIC LOAD values. The second correction is only necessary if the user attempts the first correction and receives an error about the carry capacity has to be less than the storage capacity. The second option will set the storage capacity to twice the carry capacity. The second option applies only to TUPs.

3.75 JTLS-2019-14449 Political Countries Are Permanent Entities

While upgrading JTLS-GO 5.1 to JTLS-GO 6.0, the new Simscript compiler checks found an error in which a Permanent Entity called "Political Country" was being referenced as a Temporary Entity. This would only cause an issue if the CEP was placed in the debugger in a routine that had the incorrect reference type.

A thorough review of both the SIP code and CEP code was made looking for routines with this issue. Two routines in the SIP were found and corrected.

3.76 JTLS-2019-14450 Logic Error Gathering Grid Lines

A logic error occurred when attempting to find the grids along a straight line path needed by a naval unit.

The logic error was caused by computing a path between two adjacent 10-degree grids close to the grid border. The logic was computing the bearing to each of the four corners to the large 10-degree grid. The logic should have computed that the path would leave the grid to the North, but it was coming to the conclusion is had to leave the grid to the East.

This incorrect computation was caused by the algorithm using the Great Circle Initial Bearing function. Since the grid was so large, the initial bearing was different enough than the actual rhumbline bearing that the computation came to the incorrect conclusion. The gather line grid algorithm was changed to use the rhumbline bearing computation, and not the great circle initial bearing computation.

The rhumbline bearing computation is only used to compute which grid side the path should leave. All other computation still use the great circle algorithms, resulting in all objects traveling along a great circle path.

3.77 JTLS-2019-14451 Change SUP Cruise Speed To Max Speed

The CEP would not allow the Controller to set a Ship Unit Prototype's Normal Cruise Speed to the same value as the SUP's Maximum Speed. The database allows the two values to be equal.

The code insisted that the Cruise Speed had to be less than the SUP's Maximum Speed. The code was changed to insist that the Cruise Speed had to be less than or equal to the SUP's Maximum Speed.

3.78 JTLS-2019-14452 Combat System No Surface Kill Lethality

A problem occurred when attempting to order a Combat System to fire.

Even though the Combat System was capable of firing the desired weapon, the weapon did not have a Surface Kill Lethality (SKL) assigned. The SVPR code was modified to check this situation and generate Error 117.

3.79 JTLS-2019-14453 SUP Symbol Field Does Not Match ICC

ICC maintains a set of predefined values for ship symbols. When initializing Ship Types in ICC, the JTOI provides those symbol values to ICC. The values held in JTLS-GO did not match the values defined in ICC, causing Ship Types to fail to initialize in ICC.

The JTOI has been modified to map the incorrect values to those expected by ICC. The static vocabulary file has also been updated to reflect the correct symbol values.

3.80 JTLS-2019-14454 Error 405 Will Not Always Generate

SVP Error 405, reporting that a unit does not have enough basic load to support its combat systems, will not always generate.

A problem was found in the positioning of the checks used to bypass warnings within the associated program. The checks were re-positioned to allow Error 405 to be generated when applicable.

Modifications were made to the code and the SVPR corrective actions were added to Warnings 1425 and 1426 and Error 405.

3.81 JTLS-2019-14455 Air Mission Name On Own-Side WHIP

An air mission started to use a detected name instead of its real name on its own-side WHIPs.

When an air mission takes off, its owning side automatically starts to detect the air mission. If for any reason, the air mission's owning side loses that detection, when the detection is regained, the mission stops using its real name. This problem was corrected.

3.82 JTLS-2019-14458 Implicit Resupply Due In Numbers Incorrect

When a requisition is submitted and the requisition results in the generation of an implicit resupply event, the Due In values for the requisitioned supplies were not correct.

As soon as the requisition is submitted, the CEP should increase the Due Ins for the ordered supplies. Only when the supplies are delivered should the Due In values be decreased. Several issues were found and corrected in this process:

1. If the supplies were back-ordered, the Due In values were doubled, because the back order process added the back-ordered supplies to the unit's Due In values. The Due In values had already been recorded when the requisition was submitted.

2. Implicit Resupply Events were being scheduled even if zero supplies were available to send to the receiving unit.

3. When using the Push Order capability, the Due In values were not being updated.

4. LOGIN Builds were not represented in the Due In values.

5. A Directed Resupply Order resulting in the implicit delivery of supplies to the receiving unit did not update the receiving unit's Due In values.

3.83 JTLS-2019-14459 Formation Jumps When Given Speed Change

The user gave a formation a speed change order and the formation jumped to its next move location immediately.

The logic was changed to properly compute the location of the formation at the moment the change speed order was executed. The formation is now immediately moved to that location, assigned the new speed, and will continue with its move tasks at the new speed.

While testing and implementing the fixes to this STR, all other speed change code was reviewed and is now handled consistently. The Air Mission Speed change code had a small computational

error for the mission's current location, which was corrected. Independent Naval Unit speed changes are now handled using the same methodology as Air Missions and Formations.

3.84 JTLS-2019-14460 Improve SVP Warning 1341

SVP Warning 1341 needed better corrective actions.

SVP Warning 1341 has been improved to tell the user if an aircraft load is identical to another aircraft load in the database, and whether the load is a default load for at least one aircraft class. The corrective actions for Warning 1341 have also been expanded.

3.85 JTLS-2019-14461 ATOG Service Internationalization File

The ATOG Service was loading its internationalization (i18n) file, which defines the format of all messages sent to the user, from the wrong directory. The directory specified was $JGAME/data/ i18n/default/atog instead of $JGAME/data/i18n/default/services. Accessing the language specific i18n files which resides under $JGAME/data_site had the same issue in the directory path.

The directory path was fixed for both the i18n default resources and the i18n language specific resources. Since the ATOG also has default resources in its source code, the issue would not have been noticed unless a user was attempting to load a different language.

3.86 JTLS-2019-14462 ICP Incomplete TEMS Header Footer List

The ICP configuration for the TACELINT Message Service allows the user to select a header and footer format. These message formats exist in the $JTLSHOME/game/data and $JTLSHOME/ game/data_site directories. The ICP was only loading formats from the $JTLSHOME/game/data directory.

The ICP now loads all formats in the $JTLSHOME/game/data and $JTLSHOME/game/data_site directories when choosing the desired TACELINT Message Service header and footer format.

3.87 JTLS-2019-14463 Scenario Setup Wrong Directory

When running the "Setup System For A Specific Scenario" option of the JTLS-GO Menu, the user has the option of choosing whether the scenario files should be copied from the $JDATA/ scenario/<scenario> directory to the $JGAME/<scenario>/checkpoint/0000 directory.

No matter what option is chosen, the setup is then meant to use the files in the $JGAME/ <scenario>/checkpoint/0000 directory to prepare the system, maintaining a clear delineation between database development, in the $JDATA/scenario/<scenario> structure, and a running game. However, the setup was reading files from the database development directory structure instead of the game directory structure.

The code was modified to properly set up the directory structures to ensure that the proper directory is used to access the scenario files.

3.88 JTLS-2019-14465 SVP Protection Code Bad Entity Names

The routine that validates entity names was always returning a value, even when the entity name was invalid. This could mask database problems.

The routine was modified to allow the return of a NULL value, indicating a bad entity name. There were two situations in the database where this was causing a crash, because of an invalid Special Subcategory assigned to truck Combat Systems. Since this specific error is caught and reported elsewhere, the code was modified to handle the NULL value.

4.0 REMAINING ERRORS

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

As JTLS-GO 5.1.0.0 represents a major release of new functionality, remaining outstanding errors from the JTLS-GO 4.1 series and earlier have been considered to be obsolete and no longer relevant to JTLS-GO and have been removed from consideration for correction at this time. In future Maintenance releases, outstanding errors related to JTLS-GO will be listed in this chapter, with information provided regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

4.1 DDSC – TMU Line Mode Changes Multiple Grids

When using the line mode in the TMU, more grids than the ones the line passes through are changed. This can also cause a warning about trying to change multiple layers to appear.

4.2 DDSC – Multiple Types In DDS History Table

If records for more than one table type are selected in the DDS History table, "History Details" will display details for only one type.

4.3 WHIP - Pipeline Not Shown On IMT

A pipeline being operated by a non-detected unit is not shown in the pipeline IMT.

4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

When using the CADRG map projection, if the width of the map is less than the height the zoom tool does not work correctly.

4.5 WSM - Many Messages Cause Lockup

If a service produces a large number of log or error messages in a short period of time, it can cause the WSM to lockup.

APPENDIX A ABBREVIATIONS AND ACRONYMS

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

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

BDA	Battle Damage Assessment
BDE	Brigade
BN	Battalion
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CA	Civil Affairs
CADRG	Compressed ARC Digitized Raster Graphics
CAP	Combat Air Patrol
CAS	Close Air Support
CAT	Category
CCF	Central Control Facility
CCP	Command Control Prototype
CCU	Controller Change Unit
CEP	Combat Events Program
CMDR	Commander
COP	Common Operational Picture
CP	Combat Power
CS	Combat System
CSP	Combat System Prototype
CTAPS	Contingency Tactical Air Planning System
CTG	Commander Task Group
CTRL	Control keyboard command
DCA	Defense Counter Air
DCL	Digital Command Language
DDS	Database Development System
DISA	Defense Information Systems Agency
DIV	Division
DMA	Defense Mapping Agency
DoD	Department of Defense
DOS	Days of Supply
DPICM	Dual Purpose Improved Conventional Munitions
DS	Direct Support
DSA	Directed Search Area

DTG	Date Time Group
EC	Electronic Combat
ECM	Electronic Counter Measure
ECP	Engineering Change Proposal
EEI	Essential Elements of Information
ELINT	Electronic Intelligence
ELS	Entity Level Server
EODA	Entity Level JTLS Object Data Authority
ETA	Estimated Time of Arrival
FARP	Forward Arming and Refueling Point
FLP	Fire Lethality Prototype
FLOT	Forward Location of Troops
FOL	Forward Operating Location
FWL	Frederick W. Lanchester (originated a differential equation model of attrition)
GAL	Gallon
GCCS	Global Command and Control System
GRTE	Ground Route
GS	General Support
GSR	General Support Reinforcing
GUI	Graphical User Interface
HARM	High-speed Anti-radiation Missile
HE	High Explosive
HELO	Helicopter
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HQ	Headquarters
HRU	High Resolution Unit
HTML	Hypertext Markup Language
HTT	High Resolution Unit Target Type
HUP	High Resolution Unit Prototype
ICM	Improved Conventional Munitions
ICP	Interface Configuration Program
ICPLogin	Interface Login Program
ID	Identifier
IFF	Identification Friend or Foe

IIP	Intelligence Information Prototype
IMT	Information Management Tool
INFO	Information
INTEL	Intelligence
JCATS	Joint Conflict And Tactical Simulation
JDPI	Joint Desired Point of Impact (formerly DMPI: Desired Mean Point of Impact)
JDS	JTLS Data System
JDSP	JTLS Data System Protocol
JEDI	JODA Entity Data Identifier
JMCIS	Joint Maritime Combat Information System
JMEM	Joint Munitions Effectiveness Manuals
JODA	JTLS Object Distribution Authority
JOI	JTLS Operational Interface
JPL	Jet Propulsion Laboratory
JRSG	Joint Rapid Scenario Generation (formerly JIDPS: Joint Integrated Database Preparation System)
JSDF	Japan Self-Defense Forces
JTLS	Joint Theater Level Simulation
JTLS-GO	Joint Theater Level Simulation - Global Operations
JTOI	JTLS Transaction Operational Interface
JXSR	JTLS XML Serial Repository
KIA	Killed In Action
KM	Kilometer
KNOTS	Nautical miles per hour
LA	Lethal Area
LAN	Local Area Network
LAT	Latitude
LB	Login Build (JTLS order type)
LDAP	Lightweight Directory Access Protocol
LDT	Lanchester Coefficient Development Tool
LOG	Logistics
LOGIN	Logistics Input
LOGREP	Logistics Report
LONG	Longitude
LOTS	Logistics Over The Shore

LR	Long Range
M&S	Modeling and Simulation
MAPP	Modern Aids to Planning Program
MB	Megabyte
MCP	Mobility Counter-mobility Prototype
MCR	Model Change Request
MG	Machine Gun
MHE	Material Handling Equipment
MIP	Model Interface Program
MOGAS	Motor Gasoline
MOPP	Mission-Oriented Protective Posture
MOSAIC	NCSA user interface software
MOTIF	X Window System graphical interface
MP	Maneuver Prototype
MPP	Message Processor Program
MSC	Major Subordinate Command
MSG	Message
MTF	Message Text Formats
MUREP	Munitions Report
MUSE	Multiple Unified Simulation Environment
NCSA	National Center for Supercomputing Applications (University of Illinois)
NEO	Noncombatant Evacuation Operations
NFS	Network File Server
NGO	Non-Governmental Organization
NIS	Network Information Service or Network Information System
NM	Nautical Mile
NTSC	Naval Telecommunications System Center
OAS	Offensive Air Support
OBS	Order of Battle Service (formerly UGU: Unit Generation Utility)
OCA	Offensive Counter-Air
OJCS	Organization of the Joint Chiefs of Staff
OMA	Order Management Authority
ONC	Operational Navigation Chart
OPM	Online Player Manual

OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)
SIP	Scenario Initialization Program
SITREP	Situation Report
SLP	Sustainment Log Prototype
SOF	Special Operations Forces
SP	Survivability Prototype

SQL	Structured Query Language
SR	Short Range
SRP	Start/Restart Program (a JTLS-GO component)
SRTE	Sea Route
SSM	Surface-to-Surface Missile
STR	Software Trouble Report
SUP	Ship Unit Prototype
SVP	Scenario Verification Program
SYNAPSE	Synchronized Authentication and Preferences Service
TADIL	Tactical Digital Interface Link
TCP/IP	Transmission Control Protocol/Internet Protocol
TEL	Transporter Erector Launcher
TG	Target entity attribute prefix
TGS	Terrain Generation Service (formerly TPS:Terrain Preparation System)
TGT	Target
TMU	Terrain Modification Utility
TOE	Table of Organization and Equipment
TOT	Time Over Target
TOW	Tube-launched Optically-tracked Wire-guided missile
TPFDD	Time-Phased Force Deployment Data
TTG	Target Type Group
TTL	Target Types List
TUP	Tactical Unit Prototype
TW	Targetable Weapon
UBL	Unit Basic Load
UIM/X	GUI builder tool
UNIX	POSIX-compliant operating system
UNK	Unknown
UOM	Unit Of Measure
USA	United States Army (U.S. and U.S.A. refer to United States and United States of America)
USAF	United States Air Force
USCG	United States Coast Guard
USMC	United States Marine Corps
USMTF	United States Message Text Format

USN	United States Navy
UT	Unit entity attribute prefix
UTM	Universal Transverse Mercator
VIFRED	Visual Forms Editor
VMS	Virtual Memory System
VTOL	Vertical Take-Off and Landing aircraft
WAN	Wide Area Network
WDRAW	Withdraw
WEJ	Web Enabled JTLS
WHIP	Web Hosted Interface Program
WIA	Wounded In Action
WPC	Warrior Preparation Center
WPN	Weapon
WT	Weight
XML	Extensible Markup Language
XMS	XML Message Service

APPENDIX B Version 5.1.3.0 DATABASE CHANGES

No database structure changes were made for JTLS-GO 5.1.3.0.

APPENDIX C Version 5.1.3.0 REPOSITORY CHANGES

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

C.1 Added All Brigade Units (BN and CO) for:

- 1 AD - 1ABCT Units to Ft Bliss, TX

- 1 AD - 2ABCT Units to Ft Bliss, TX

- 1 AD - 3ABCT Units to Ft Bliss, TX

- 1 CAV - 1ABCT Units to Ft Hood, TX

- 1 CAV - 2ABCT Units to Ft Hood, TX

- 1 CAV - 3ABCT Units to Ft Hood, TX

- 101AA - 1IBCT Units to Fort Campbell, KY

- 101AA - 2IBCT Units to Fort Campbell, KY

- 101AA - 3IBCT Units to Fort Campbell, KY

- 10MTN - 1IBCT Units to Fort Drum, NY

- 10MTN - 2IBCT Units to Fort Drum, NY

- 10MTN - 86IBCT Units to Fort Drum, NY

- 11ACR - Units to Fort Irwin, CA

- 173 Airborne Bde Units to Italy

- 1AD - 1ABCT Units to Ft Bliss, TX

- 1AD - 2ABCT Units to Ft Bliss, TX

- 1AD - 3ABCT Units to Ft Bliss, TX

- 1ID - 1ABCT Units to Fort Riley, KS

- 1ID - 2ABCT Units to Fort Riley, KS

- 2ACR - Units to Rose Barracks, Vilseck, GE

- 3ACR - Units to Fort Hood, TX

- 3ID - 1ABCT Units to Fort Stewart, GA

- 3ID - 2ABCT Units to Fort Stewart, GA

- 3ID - 48IBCT Units to Fort Stewart, GA

- 4ID - 3ABCT Units to Ft Carson, CO

- 82ABN - 1IBCT Units to Fort Bragg, NC

- 82ABN - 2IBCT Units to Fort Bragg, NC

- 82ABN - 3IBCT Units to Fort Bragg, NC

C.2 Verified All Brigade Units (BN and CO) for:

- 25ID - 1SBCT Units are in Korea

- 25ID - 2IBCT Units are in Korea

- 25ID - 3IBCT Units are in Korea

- 25ID - 4IBCT Units are in Korea

- 25ID - CAB (Combat Avn Bde) Units are in Korea

- 2ID - 1SBCT Units are in Korea

- 2ID - 2SBCT Units are in Korea

- 2ID - 81SBCT Units are at FT Lewis

- 2ID - CAB (Combat Avn Bde) Units are in Korea

- 4ID - 1SBCT Units are in Korea

- 4ID - 2IBCT Units are in Korea

C.3 Added SUP:

- ASAHI.DDG_JP

C.4 Added Naval Units:

- ASAHI_JP (ASAHI Class)

- CHARLESTON_US (INDEPENDENCE Class)

- CINCINNATI_US (INDEPENDENCE Class)

- INDIANAPOLIS_US (FREEDOM Class)

- MINNEAPOLIS_US (FREEDOM Class)

- SHIRANUI_JP (ASAHI Class)

- ST.LOUIS_US (FREEDOM Class)

- TULSA_US (INDEPENDENCE Class)

C.5 Added Combat Systems:

- GBL130.MINE.LAYER-ENG1 (China)

- GCZ110.CEV-ENG1 (China)

- GCZ111.CEV-ENG1 (China)

- GDG130.EXCAVATOR-ENG2 (China)

- GZM003.BRIDGE.VEH-ENG1 (China)

- HZ.PPB.BRIDGE-ENG1 (China)

- KM720.LWB_TRK.CGO (SK)

- KOHKIDOHSHA.HMV-VEH-NA (Japan)

- M860.SEMI.TRLR-EQUIP-TO (US)

- MRZR4-VEH-NA

- MSLP.STLR.VAN-EQUIP-TO

- NORINCO.DITCH.DIGGER-ENG1 (China)

- NTK62.GPMG.7.62M-CREW.WPN (Japan)

- OPFOR.CRANE-EQUIP.SP

- PRC.DEMO.TROOP (China)

- RNGR.SNIPER.M110

- ROK.DEMO.TROOP

- ROK.SOF.DEMO

- ROK.SOF.RIFLEMAN.K2

- ROK.SOF.SPT.TROOP.K2

- SKW464MR.WRECKER_EQUIP.SP (Japan)

- SX2190.DUMP-ENG2 (China)

- TYPE60-1.TRACTOR_EQUIP.SP (China)

- TYPE654.ARV_EQUIP.SP (China)

- TYPE73.6X6_TRK.CGO (Japan)

- TYPE75.DOZER-ENG1 (Japan)

- TYPE89.AMBULANCE-VEH-LA (China)

- VN1.12.7MM_APC.LA.OW (China)

- VN1.30MM_APC.LA.OW (China)

- VS21.ARV_EQUIP.SP (China)

- VS31.ARV_EQUIP.SP (China)

- WZ501.NBC.RECON-CHEM (China)

- WZ8581.MAINT.VEH-EQUIP.SP (China)

- ZBD09.HJ73C-IFV-ATSHHATC (China)

- ZJX93.MAINT.VEH-EQUIP.SP (China)

C.6 Added TUPs

- CIV.AFF.BN_US

- CIV.AFF.CO_US

- CIV.AFF.TM_US

- HHB.FA.BDE_US

- LMADIS.JAM_US

- PAT.BN.MNTCO_US

- PAT.PLT_US

- PATRIOT.BN_US

- RGR.STB.BN_US

- SIG.CO.FABDE_US

- SPT.CO.FABDE_US

C.7 Updated Combat Systems in TUPs:

- MLRS.BN.16L_US

- MLRS.BTRY.8L_US

- MLRS.PLT.4LH_US

C.8 Renamed TUP:

- PATRIOTBDEHQ_US to ADA.BDE.HHB_US

- RANGERRGTHQ_US to RANGERRGTHHC_US

C.9 Deleted TUP:

- RGR.RIFLECO+_US

- RGR.RECONPLT_US

- RGRRGTHQC2TM_US

C.10 Created New Jammer:

- LMADIS.COM.JAM

C.11 Created New Sensor:

- MPQ65_LAM

- OPS48_SGN

C.12 Created New TW:

- AGM88E.HARM

C.13 Created New UOM:

- AGM88E.HARM

C.14 Created New Supply Category:

- AGM88E.HARM

C.15 Added Sensor to A/C Load:

- APG77_AGK to FA22.DEF Load

C.16 Added Russia Side to Scenario:

- Migrated RUSSIA Force Side from scenario

C.17 UOM Verification

- Verified the Storage Floor Area is in Square Kilometers for all Personnel CS