JTLS-GO Document 17

JTLS-GO

Version Description Document

November 2021

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.0.5.0)

[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 air, land, naval, and Non-Governmental Organization (NGO) environments.

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.0.5.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.0.5.0 is a Maintenance release of the JTLS-GO 6.0 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as some minor model functionality improvements implemented as Engineering Change Proposals (ECPs), These ECPs are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

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

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.0.5.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.0.5.0 is a Maintenance delivery for the JTLS-GO 6.0 series of releases.

JTLS-GO 6.0.5.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations called “wespac60”. Database modifications that were accomplished to upgrade the previous JTLS-GO database format to the JTLS-GO 6.0 series data format were summarized in the VDD for JTLS-GO 6.0.0.0. No data format changes were made between JTLS-GO 6.0.0.0 and this Maintenance release.

One Engineering Change Proposal (ECP) is delivered with this maintenance release and is described in Chapter 2.0. All software corrections are summarized in Chapter 3.0.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Obsolete/Outdated Documents

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

1.2.2 Unchanged Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.0.2.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 6.0.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.0.4.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.0.1.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.0.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.0.2.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.0.4.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.0.0.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.0.4.0)

- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 6.0.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.0.2.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.0.3.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.0.4.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.0.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.0.0.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.0.2.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.0.3.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 24, Version 6.0.0.0)

1.2.3 Updated Documents

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.0.5.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.0.5.0)

1.2.4 Delivered Software Components

JTLS-GO 6.0.5.0 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)

  The current JODA build number is 138.

- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:

  Apache Server (APACHE)
  JTLS XML Serial Repository (JXSR)
  Order Management Authority (OMA)
  Synchronized Authentication and Preferences Service (SYNAPSE)
  XML Message Service (XMS)
  Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service
- KML Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)

- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool

Instructions for installing JTLS-GO 6.0.5.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.0.5.0 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.

1.2.5 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data is called "repository60". Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space. With JTLS-GO 6.0.5.0, it is possible to access and copy records from the repository60 database into your own developed scenarios.
- The scenario “wespac60”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.0.5.0 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 7.9 (ES), 64-bit architecture.

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

  RedHat Linux 7.9 - this operating system license must be purchased.

  Oracle Linux 7.9 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

  CentOS 7.9 - a free version of Linux 7 that has not been approved by DISA for use by U.S. Government agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.0.5.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.9.

  CentOS Linux Version 7.9.

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

Although JTLS-GO 6.0 is only approved for use on a RedHat Linux 7.9 system, several users have already upgraded their JTLS-GO Linux servers to RedHat 8. There have been no reported problems running the JTLS-GO 6.0 series on RedHat 8 Linux servers, as long as the servers are not Federal Information Processing Standards (FIPS) enabled. Each user organization needs to decide whether to move to RedHat 8 with a FIPS-disabled environment. To summarize:

- JTLS-GO 6.0 series on RedHat 7.9 can be run in a FIPS-enabled or disabled mode.
- JTLS-GO 6.0 series on RedHat 8.0 must be run in a FIPS-disabled mode.
- JTLS-GO 6.1 series must be run on RedHat 8 and can be run in a FIPS enabled or disabled mode.

- The JTLS-GO 6.0 series no longer relies on the Open Java Development Kit (OpenJDK^{TM}) to be installed at the system level using the Red Hat Package Manager (RPM). Instead, the JTLS-GO series is delivered with the AdoptOpenJDK package, which is equivalent to the current version of OpenJDK. Using AdoptOpenJDK provides two benefits:

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

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

  JTLS-GO 6.0.5.0 is delivered using the AdoptOpenJDK 1.8 Update 312.

- JTLS-GO uses IcedTea to provide the Java Web Start capability to start the JTLS-GO web-enabled applications. JTLS-GO supports IcedTea version 1.8.4. The user has two possible avenues to access the supported version of IcedTea when starting a web-enabled application on the JTLS-GO Linux servers. The IcedTea 1.8.4 RPM can be installed on the JTLS-GO Linux servers, or the user can set their Browser to use the proper version of IcedTea, located in the bin_support/Linux64 directory for "jnlp" applications,

  Client machines should be setup to also use the supported version of IcedTea,

- JTLS-GO database tools require a certified PostgreSQL 11.8 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.0.5.0. There are several alternative methods available for obtaining the PostgreSQL 11.8 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.

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

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

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

- SIMSCRIPT III Version 3-5.2.0 (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc.

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

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

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

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

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

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

- KML Operational Interface (KOI)

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

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

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

1.3.2 JTLS-GO Cybersecurity Compliance

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

One of the DoD requirements is that the software must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. After a certain date, all Java related programs stopped working, whether connected to an open network or not. All JTLS-GO releases were closely linked to the Java expiration date.

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

- Within days of an Oracle Java security release, AdoptOpenJDK produces an equivalent version using infrastructure, build and test scripts to produce prebuilt binaries of the OpenJDK class libraries. All AdoptOpenJDK binaries and scripts are open source licensed and available for free.
- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a Maintenance release version (JTLS-GO 6.0.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the Maintenance released versions when distributed.

Contact the U.S. Government Program Manager, Mr. James E. Blank (james.e.blank.civ@mail.mil) to obtain the completed Cybersecurity paperwork.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.0.5.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

The solution combines JTLS-GO with CAE's GESI constructive tactical entity-level simulation system. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The GESI system is used to represent a virtual battlefield, including weapons, vehicles, aircrafts, ground forces and more.

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

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the $JGAME/data/hla directory, Federation testing of JTLS-GO with CAE's GESI model have been accomplished. Future plans include expanding the capabilities of the GlobalSim federation.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.0.5.0 delivery. Users may obtain a full installation package of the RTI software from Pitch Corporation (www.pitch.se). For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

1.4 DATABASE MODIFICATIONS

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

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

1.4.1 JTLS-GO Using Legacy Default Symbol Set

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

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.0.5.0, but end-user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

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

1.4.3 Standard Repository Changes

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

The JTLS-GO 6.0.4.0 release, included a procedure needed to correct STR JTLS-2021-15237 which required changes to some post-update database triggers. If the following procedure was not accomplish with the release of JTLS-GO 6.0.4.0, it should be perfformed after the installation of this Maintenance release. Please:

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

1.5 INSTALLATION

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

2.0 ENGINEERING CHANGE PROPOSALS

The following Engineering Change Proposals (ECPs) were added to this Maintenance Release of JTLS-GO.

2.1 JTLS-2021-15314 Check For Duplicate Targets Unnecessary

Summary of Model Change Request

The check for duplicate Target names currently in the SIP/SVP code is time consuming and should be removed.

Design Summary

The DDS forms and triggers have been modified to prevent users from entering Target CCF NUMBERS or Target LONG NAMEs that match another Target in the database.

The Simscript code used to perform these checks is no longer necessary and has been removed from the sip.exe program. This will greatly increase the performance of the verification process especially when the database contains a large number of targets.

2.2 JTLS-2021-15320 ELS Changes For Link To VBS4

Summary of Model Change Request

A link to VBS4 is under development. This project connects VBS4 to JTLS-GO via the EODA. In support of this link, changes to the ELS were required to allow VBS4 to fire weapons from their units. The goal is for VBS4 to aim at other entities or at specific locations.

Design Summary

Prior to these changes, the ELS only accepted weapon fire from an externally-controlled air mission. These interactions had to be aimed at a specific entity. The ELS code now accepts weapon fire from all types of external objects. This capability allows the external model to aim at a specific entity or at a location.

These ELS changes apply to any connected external model, and are not specific to VBS4.

2.3 JTLS-2021-15324 New SVP Checks For Federating With VBS4

Summary of Model Change Request

When linked with VBS4, JTLS-GO scenario data verification requires additional SVP checks.

Design Summary

Several new Errors and one Warning were added to the SVP. The new Errors (5014 and 5015) and Warning (6020) involved the verification of the DIS Codes used by combat systems in units controlled by VBS4. They are only evaluated when the option to "Check Federation Data" is selected.

2.4 JTLS-2021-15355 Count Orders By Unit

Summary of Model Change Request

A count of orders sent to individual units in the exercise was required. The Count Critical Order program was modified to handle this function.

Design Summary

The Count Critical Order file was provided another capability to check for orders with the UNIT attribute set and provide a tally of orders sent to all units in the game. This capability is required rarely enough that it was not added to the Javamenu. The capability can be accessed from the command line and a robust usage help was provided to aid in this. If this capability becomes routinely used, the decision to exclude it from the Javamenu will be revisited.

During the integration of this capability, it was found that when searching for a count of a specific order, an existing capability, that the order by player terminal summary listed all orders sent rather than the specific type of order sent. That error was fixed.

3.0 SOFTWARE TROUBLE REPORTS

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

Chapter 4.0 describes STRs that remain outstanding from previous versions. These issues are being addressed and solutions will be included in future Maintenance releases in the JTLS-GO 6.0 series.

3.1 JTLS-2021-15286 Foreign Air Mission Tracking Lacking

Users are able to monitor foreign side units using the Foreign Units Intel IMT. Users also needed the ability to monitor and reference foreign side air missions quickly. The only way to do this was to configure a map to display air missions in an obvious way and use the Sitrep component to view mission data. This method was inefficient and tedious.

A Foreign Air Mission Intel IMT was created to allow users to monitor foreign air mission data in a tabular format.

3.2 JTLS-2021-15287 Object Name And Task Type Frozen IMTs

The IMT screen for object tasks had one frozen column for the task sequence number. It was hard for the user to find information when multiple columns were visible.

Changes were made so the Object Name and the Task Type are also frozen columns.

3.3 JTLS-2021-15288 Percent Capable And Posture On IMT Screens

The Naval Unit IMT screen did not contain the unit percent capable and posture data. These data are available on the general Unit Information IMT.

The percent capable column on the Supply Status IMT screen is very misleading. It contains data on the percent capable of the unit holding the supplies, but users often assume the values are the percentage of supplies at the unit.

Percent capable and posture were added to the Naval Unit IMT. Percent capable was removed from the Supply Status IMT.

3.4 JTLS-2021-15289 Downed Pilot HRU Created With Extant Name

When air missions are created using read order files, they can reuse the same mission name. This is ok if the previous mission with the same name has completed before the new one is launched. If the mission is shot down, a downed pilot HRU may be created. If a second same-named mission is destroyed, then a downed pilot HRU might be created with the same name as *the one from the first mission. This is a problem for the model, especially when restoring from a checkpoint.*

New code was added to ensure that HRU names are unique when an HRU is created for a downed pilot.

3.5 JTLS-2021-15290 Recce Mission Report Not Being Generated

A Reconnaissance Mission should generate a summary Recce report at the end of the mission. This report was not being generated. The problem was localized as an issue with whether the detected object met the minimum relationship reporting level.

The check to determine if a detected object met the IIP Minimum Relationship Reporting Level was not built correctly. It was the reverse of what it should have been. The problem was corrected.

3.6 JTLS-2021-15291 CEP Connecting To Wrong Service On GSS Socket

The CEP was crashing on a call to the Geo-Spatial Service (GSS) when a process other than the GSS had a listen on the GSS assigned port number. The crash occurred when the other process sent back garbage to a GSS request which could not be interpreted.

When the CEP first makes its connection to the GSS, it sends over the scenario name so that the GSS can verify that it is running on the same scenario. The reply back to the CEP is a zero to denote failure or a positive value to denote success.

This handshake code was modified to also send a 16 bit integer which the GSS will respond with and will act as a verification that the response is from the GSS. If the verification string is not sent, does not match, or additional data is sent then the CEP will close the GSS connection and perform the calculation with its in-built algorithms.

3.7 JTLS-2021-15292 Link 16 Structure Selection Not Updated

The WHIP's Link 16 Module connects to a running Link 16 Message Service (L16MS). The user is able to assign a Link 16 Network Structure to the L16MS through a GUI element in the Link 16 Module. Controllers are able to create new Link 16 Network Structures during gameplay. The problem is that the WHIP needed to be restarted to see Link 16 Network Structures that had been created since the WHIP was launched in the Link 16 Module.

The Link 16 Module now properly updates with any Link 16 Network Structures that are created during the lifetime of the WHIP.

3.8 JTLS-2021-15293 L16MS Initialization File Misread

The L16MS reads an initialization file on startup. A piece of data was present in the initialization file that the service did not expect, so the service's read was corrupted after that piece of data.

The L16MS read code has been updated to match the format of the initialization file.

3.9 JTLS-2021-15294 SVP Warning 1707 Correction Option Nonfunctional

The first correction option for SVP Warning 1707, to open the target table so the user can change the target mobility to something other than STATIONARY does not work.

Warning 1707 identifies stationary sensor site Targets whose owning unit does not arrive at game start. The sensor site will turn on and function even though the unit is not in the game.

The sensor site had been created as a POT Target. The correction option was only designed to open the Target sensor site table when it needed to open the TUP or SUP POT table in this instance.

A new Warning 1709 has been created and documented for the situation if the target is created as the result of being a POT target.

3.10 JTLS-2021-15295 Mission Package Ingress Problems

An air mission package was created without an ingress route. Before an OAS mission was assigned to the package, a Manage Attack Package order was submitted to add an ingress route to the package, which immediately crashed the model.

The model incorrectly assumed that an OAS mission was already part of the package and crashed trying to access the cruise altitude of the aircraft type. The cruise altitude was being used to assign altitudes to route points that had no specified altitude.

Further examination revealed that the logic was purposefully propagating the cruise altitude or the previous altitude (i.e. the rendezvous altitude) to every succeeding ingress route point that did not have a specified altitude. This logic was in error because, during flight execution, the model interprets a zero altitude route point as a signal not to change the mission's altitude for the next segment. By purposefully filling every zero altitude route point with the previous mission altitude, the Player was unable to alter the altitude for an entire route without having to change each route point individually as listed in the Air Mission Tasking Orders IMT.

This inconsistency was corrected in several places in the model dealing with adding/deleting/modifying ingress and egress routes for packages and independent missions. The model will no longer fill route points that have no specified altitudes with the mission's previous altitude or cruise altitude. Hence, the Air Mission Tasking Orders IMT now displays zero altitudes with each Move task, unless an altitude was specified in the player order for that task.

Another error was found when an egress route was added to a package that had an ingress route. The existing ingress route was duplicated which caused the package to backtrack and repeat its ingress route. This code error was also corrected.

Several minor errors were found in the Player confirmation messages. When an egress route was deleted from a package, the message did not indicate this action had occurred, but did so for a deleted ingress route. The appropriate sub-message (sub 18) was present in the message template, but was not referenced from the underlying subroutine. The altitude's Unit of Measure did not print in several sub-messages. These problems were corrected.

3.11 JTLS-2021-15296 Too Many New WHIP Registrations

After a new WHIP is created using the ICP, the SYNAPSE sometimes sends too many registrations for this WHIP. It should only send one registration to the CEP for each new WHIP.

The SYNAPSE discovers new WHIPs that were created by the ICP in two ways:

1. Receiving a Player MIP (WHIP) create over the SYNAPSE connection with the JODA, or

1. Receiving WHIP creates by reading the CEP’s .cif file when it changes.

The SYNAPSE properly sends WHIP registrations back to the CEP when the Player MIPs are received from the JODA and removes the WHIP from its list of unregistered WHIPs.

However, when the SYNAPSE discovered WHIPs from the .cif file, it sent the registration to the CEP but does not remove the WHIP from the list of unregistered WHIPs. Until the WHIP is received from the JODA, the SYNAPSE kept sending the registration until a corresponding Player MIP was finally received from the JODA.

This has been changed so the SYNAPSE removes the WHIP from the list of unregistered WHIPs on the first registration following discovery from the .cif file.

3.12 JTLS-2021-15297 Automatic Clear Checkpoint

The CEP crashed trying to write unit combat system summary data during a checkpoint.

During a checkpoint, the CEP creates combat system summary files in the game/scenario_name/cbtsys_summary directory. The files are then moved into the checkpoint subdirectory. Under special circumstances, the checkpoint save directory needs to be cleared before writing the new data.

During the 'clearing' of the old data, the game/scenario_name/cbtsys_summary directory was removed when it should have been the game/scenario_name/checkpoint/xxxx/cbtsys_summary directory. This was caused by a mistake in the file specification during removal. The code was corrected.

During this process, the code that does the removal of the old save data was streamlined to make it easier to understand and process.

3.13 JTLS-2021-15298 Weather Front Data In JODA Incorrect

When a Weather Front is sent to the JODA, the weather condition, which should be an index between 1 and 21, is showing up as 88512, or 54632, or some other sequence of 4 or 5 digits.

The error was that the Weather Condition was being passed by reference between the CEP Simscript code and the CEP C code that manages the JODA API. The argument has been modified to pass it by value instead.

3.14 JTLS-2021-15304 SVP Warning 1227 Correction Nonfunctional

The correction to update the supply table for SVP Warning 1227 does not work.

Warning 1227 identifies shortages in SAM weapons. The non-functional correction was to insert a new supply record if one did not exist. The routine was modified to add an insert record option as a new Warning 1229.

The code that identifies if a record needed to be updated or inserted was also found to be incorrect. This problem was also corrected under this same STR.

3.15 JTLS-2021-15305 SVP Warning 1449 Cannot Open Supply Table

There was no option in the automatic corrections for SVP Warning 1449 to open the TUP/SUP supply table to review or make manual updates.

The new option was added.

3.16 JTLS-2021-15306 Assess Damage For Auto Fire Missile Crash

A missile was automatically fired at an enemy ship. It was a long-range weapon and, by the time it reached the target ship, the ship had moved outside of the weapon's search area. The weapon then found an alternate target nearby - the Naval Port in which the target ship had been located when the weapon was fired.

After damage was assessed, the model attempted to continue the automatic firing. The automatic firing logic assumes that its target was a naval unit, but the target was now a ground unit, and the model crashed attempting to access the unit's prototype. The prototype value was out of range and the model crashed.

If the hit target is changed in Assess Weapon Damage, the model now stops the continuation of the automatic firing continuation logic.

3.17 JTLS-2021-15307 Attempted Mining w/Non-Mine Weapon Crash

During a naval mining operation, a Set Targetable Weapon (TW) Parameter order was sent to change the mine being laid to a non-mine weapon with no associated Minefield Type. When the *next mine was about to be laid, the model crashed attempting to access the non-existent Minefield Type to determine the terrain eligibility.*

This situation also caused a crash attempting to come up from a checkpoint because the minefield target subcategory was zero.

The problem was solved by adding new code to prevent the Controller from changing the Weapon Effects of the Mine TW from MINE to any other Weapon Effects if either of the following two circumstances exist:

1. There are existing minefield targets that use the same Mine TW.
2. There are mine-laying tasks currently executing or scheduled that use the same Mine TW.

If one of the circumstances is true, the request to change the TW to a non-mine TW is ignored in the Set TW Parameter order. A non-compliance message with an explanation is sent to the Controller, including a list of the existing minefield targets or a list of the LAY_MINE tasks from all air missions, naval vessels, ground units, and HRUs.

3.18 JTLS-2021-15308 Order Group Verification Passes On Error

When verifying an order group, each order is verified individually. If an order fails, then the group fails as a whole. In one instance, an exception was being thrown during the verification of a single order. However, the error flag failed to be set, allowing the group to pass verification.

When an exception is thrown during the verification of an order group, the failure flag is now properly set.

3.19 JTLS-2021-15309 Cannot Select Unit From Map

The user was unable to select units from the map for the MANAGE TPFDD and REACTIVATE UNIT orders.

The REACTIVATE UNIT and MANAGE TPFDD orders have not been map-selectable for many years. Because filters have been added to display OUT_OF_GAME and DESTROYED units on the map, the ability to select those units on the map has been added.

3.20 JTLS-2021-15310 Magic Move Unit Near Enemy Rejected

A Magic Move order was sent to relocate a ground unit near an enemy unit. The new location was also near another unit from the same Force Side as the unit to be moved. The order was rejected because the enemy unit was too close, even though the new location appeared to be within the radius of the other unit from the same Force Side.

By definition, the magic move location cannot be within twice the TUP/SUP radius of the enemy unit, unless it is also within twice the TUP/SUP radius of a same-side or friendly-side unit. A flaw in the logic excluded same-side units from being considered when determining whether enemy units were too close. As a result, a unit could only be magic moved within twice the TUP/SUP unit radius of an enemy unit, if also within twice the TUP/SUP radius of a friendly foreign unit.

The logic was corrected to include both own-side and friendly side units in the determination. The non-compliance message was rewritten to accurately describe the situation.

Note: The TUP/SUP radius is NOT the same as the current unit radius displayed on the WHIP map. The current unit radius is usually less than the TUP/SUP radius due to attrition, posture, and terrain factors.

3.21 JTLS-2021-15311 TRIPP IMT Format

The IMT Format font size was not presented in the TRIPP Preference List.

IMT Format font size was added to the TRIPP Preference List.

3.22 JTLS-2021-15312 OMA Error Longitude On East Side of Terrain Layer

The OMA threw an error for a Naval Move order that contained a location that was exactly on the Eastern edge of a Terrain Layer. The location was being converted from a latitude/longitude to a Terrain Grid to verify the location was water. The error was a check to ensure that the grid was still on the specified layer, which it was not.

By definition, a Terrain Layer covers the area greater than or equal to its lower left corner and less than its upper right corner. Locations exactly on the Eastern or Northern edge are not considered part of the grid, with an exception made for the North Pole. The OMA was not properly following these specifications, with a longitude on the Eastern edge or a Terrain Layer being assigned to the Terrain Layer.

The code was modified so only locations less than the Eastern and Northern boundary of a Terrain Layer are considered part of the Terrain Layer. An exception was made for the North Pole, and code was also added so the 180W would be treated as 180E to force it into the proper Terrain Layer.

It was originally believed this issue may have been pointing to a problem with the Terrain Layer data. A check of the existing SVP checks showed that the situation of a Terrain Layer not entirely within its parent layer was already flagged. What was not being checked was having overlapping Terrain Layers. Even though this problem was not related to bad Terrain Layer data, the SVP was modified to check for overlapping Terrain Layers.

3.23 JTLS-2021-15313 Naval Unit Patrol Polygon/OPAREA Vertex

When given a Patrol Order, a naval unit should go to the closest patrol polygon or OPAREA vertex to enter the patrol area. The unit instead goes to the first vertex defined in the polygon definition.

This problem could only be replicated when the Patrol Order specified a patrol polygon and also specified an ingress route to the patrol area. When the order was submitted and accepted, the route drawn on the WHIP followed the ingress route. But the final leg to the patrol polygon was drawn from the last route point to the first vertex of the polygon, which was not the closest vertex.

However, when the model executed and the unit reached the last route point, the unit correctly moved to the closest vertex. The WHIP also redrew the last leg at this point so that it was drawn correctly from the last route point to the closest vertex. Incidentally, the same observations were made with naval formations.

Because the model was already moving the ship/formation to the correct vertex, the only problem to resolve was getting the WHIP to draw the final leg correctly. Since the WHIP always drew the final leg from the last route point to the first vertex of the patrol polygon, a fix involving redefining the vertex order of the polygon was considered. The vertices of the polygon are now checked when the patrol order is received by the model. If necessary, the vertex order of the patrol polygon are now reordered such that the vertex closest to the last route point is defined as the first vertex of the polygon.

For example, if the patrol polygon has four vertices (V1, V2, V3 and V4 defined in that order), and V2 is the vertex closest to the last route point, the vertex order will be redefined as V2, V3, V4 and V1.

3.24 JTLS-2021-15315 Naval Formation Route Deviates After Ship Repaired

A ship on station in a naval formation became INCAPABLE (started sinking). This caused the formation to stop moving. When the ship was repaired so that the formation could start moving again, the route changed as if an extra route point was added.

A formation that followed a route had its next location determined by the routine COMPUTE FORMATION NEXT LOCATION. If the formation was currently not moving the routine created a Route Point (with coordinates at the formation's current location) and added it to the formation's Route Owner route set. This Route Point became the last point in the set. The problem was that this Route Point needed to be inserted as the first point in the set, so that it could be processed properly by the next MOVE FORMATION event. This correction was made to COMPUTE FORMATION NEXT LOCATION.

The routine UPDATE SHIP SPEEDS also had to be modified. When a naval formation needed to change speed, the routine called ADJUST FORMATION SPEED. In the case, where a naval formation stopped moving because one of its ships started sinking, and that ship was repaired so that the formation could resume movement, two things needed to be done before ADJUST FORMATION SPEED was called:

1. The first Route Point (whose coordinates happened to be the formation's current location) was now redundant and needed to be removed and destroyed.

2. When the formation stopped, its ordered speed was set to zero. Now that the formation could move, its ordered speed needed to be restored.

Both items are now implemented in UPDATE SHIP SPEEDS. Because this routine is called by several other routines, a YES/NO flag was added to the argument list to ensure that items 1 and 2 are executed at the appropriate time. (YES means execute both items, NO means skip both items.) This flag is set to YES when the routine STOP SINKING PROCESS calls UPDATE SHIP SPEEDS; otherwise, all other routines that call UPDATE SHIP SPEEDS have the flag set to NO.

3.25 JTLS-2021-15316 JODA Windows Client Crash During TCP Connect

A Windows JODA Client that is embedded as a module of a main program will sometimes crash during the TCP connection with the JODA service.

The JDSP library for Windows uses an arbitrary limit of 16384 for any file descriptors for TCP sockets. This is an old limit that was fixed to a much smaller average memory size for PCs. More powerful Windows platforms may provide a descriptor number greater than this limit to a client that is using a large number of descriptors, such as the VBS simulation. When this happens, the JDSP library would receive a memory fault that was caused by accessing an array item beyond this limit.

The number of descriptors available on a Windows platform is dependent on the memory pool available to the system, which comes from the installed RAM. It is generally accepted that this number will not exceed 25000 on contemporary Windows platforms. This limit in the JDSP has been increased to 25k. A check was implemented to reset the connection attempt if a Windows platform returns a descriptor higher than this value.

3.26 JTLS-2021-15317 Windows JDSP Library Checks Standard Input

A Windows JODA client program, which does not use the terminal for input to the Console, will terminate if input is detected from an input source (keyboard).

Currently the JDSP allows both Linux and Windows programs to link with the Console library functions. These functions allow the client program to discover and process keyboard input. On Linux, the JDSP properly discovers and processes input, whether or not the Console functions are used. However, with a Windows client, detecting keyboard input occurs even if the Console library was not used. This causes the client to attempt to process the input from an uninitialized stream, which causes a crash.

The input detection code has been modified for Windows libraries to ignore keyboard input.

3.27 JTLS-2021-15318 Forbid Minefield Type Terrain Eligibility Change

A user attempted to update the TW drop down list for a specified ground unit by changing the terrain eligibility of the TW’s Minefield Type. A Controller should not be allowed to modify a Minefield Type's terrain eligibility during gameplay.

Changing the terrain eligibility of a Minefield Type can cause problems during game play. When changing the Minefield Type's terrain eligibility, all mines that are currently in play would have to be considered and altered.

This would be a huge change with numerous design implications. For this reason, the Controller is no longer allowed to change the terrain eligibility of a Minefield Type during game play. If this capability is desired in the future, an Engineering Change Proposal should submitted and funding acquired to properly implement changing the terrain limitation of a Minefield Type.

3.28 JTLS-2021-15319 Ground Unit Mine Order Empty Message

The message to reject a ground unit Mine Order that will take an infeasible amount of time was empty.

The message has been filled out appropriately.

3.29 JTLS-2021-15321 Remove TW Record Count Entry

Prior to JTLS-GO 6.0.0.0, the Targetable Weapon (TW) was a Permanent Entity. It is now a Temporary Entity and the TW record count entry needs to be removed from the rec_count file.

The routine READ RECORD COUNT TABLE DATA no longer reads the TW record count from the rec_count file in the CEP or SIP. The routine WRITE RECORD COUNT TABLE DATA no longer writes a TW record count to the rec_count file in the CEP.

The script record_count.dnl no longer writes a TW record count to the rec_count file.

3.30 JTLS-2021-15326 Crash When Two Mines Hit Ship

The model crashed when two mines from the same minefield, but in different depth zones, hit and damaged a ship.

The model was attempting to keep track of the mines that hit the ship and attempted to file the same weapon into the set used to save this information. This caused the crash. The model now accounts for this possibility.

3.31 JTLS-2021-15331 GIS Tool Improvements

User feedback has identified the following improvements for the GIS Tool:

1. *Allow automatic saving of a newly created terrain layer.*
2. *Do not save terrain layers automatically after deleting a terrain layer.*
3. *Allow the user to select only the unprocessed terrain layers from the Calculate Grid Parameters dialog.*

4. *Do not allow a terrain layer to be created if it crosses the International Date Line.*

The changes made to the GIS Tool are described below:

1. An Auto-save check box has been added to the New Terrain Layer dialog. If the check box is selected, the tool will automatically save the new layer after its grids have been generated. If the check box is clear, the layer will not be saved until the user selects "Save..." from the File menu.

2. Automatic saving is turned off permanently. This will allow the user to recover the deleted layer in case it was inadvertently deleted.

3. A "Select Only Unprocessed Layers" check box has been added to the dialog. When this check box is selected a check mark will appear alongside each red terrain layer (those that have not had their grid parameters calculated). Deselecting the check box will remove the check marks alongside the red terrain layers.

4. The user creates a terrain layer by specifying the lower left and upper right corners of the layer. If the specified corners are such that the resulting layer will cross the date line, an error message will pop up in a message dialog, and the user will have to input a new pair of corners.

3.32 JTLS-2021-15335 Unit Attempt To Interdict Convoy Crash

A Ground Combat Unit had a zero distance move. This can happen easily under a variety of circumstances, but this time, the unit saw an enemy convoy and was trying to determine if the convoy was close enough to interdict.

To accomplish this algorithm, the model determines if the current unit move, as defined using a great-circle line, crosses the convoy's next move. The algorithm to determine if two great-circle lines cross crashed because one of the "lines" was not a line, but was in fact a single point.

This situation has happened before under different circumstances, and in each circumstance the model was changed to check if the move was a zero distance move before calling the routine to determine of the two great-circle lines cross. The problem was fixed by implementing two different code updates:

1. The model now checks if the unit has moved any distance before calling the routine to determine if the unit movement path intercepts the convoy's movement path. This is the equivalent fix that we have made in other parts of the code.

2. The routine that determines if two great-circle lines cross was also improved. If it determines that one of the two lines is in fact not a line, but a point, the algorithm passes back that there is no intersection before attempting the computation that crashes the game.

3.33 JTLS-2021-15340 Bad Message Stopping Replenish Supply Task

The user stopped an ongoing ship-to-ship replenishment task. The message indicating the list of supplies that were transferred prior to postponing the replenishment could not be displayed in the message browser.

The error occurred because a variable for the Supply Categories that were transferred was referred to by two different names. The wrong version of the variable was being accessed. This caused the generated message to include inappropriate references to the supplies that were transferred.

3.34 JTLS-2021-15347 Incorrect HRU Resource Allocation Procedure

The model attempted to allow an HRU that required a Small Boat to be created from a ground parent Unit that had the proper Small Boat, but from a different combat system than the HRU was expecting. The check was incorrectly formatted.

This caused a crash in JTLS-GO 6.1.0.0, but was wrong without causing a crash in 6.0.4.0.

The formatting error was corrected.

3.35 JTLS-2021-15351 LC2IS Country Codes In NATO-DiGraph

The JTLS-GO database was built with NATO-DiGraph Country Codes. When a LC2IS SIF, an importable XML file, is produced by the LC2IS Message Service (LC2MS), it includes the NATO-DiGraphs instead of the NATO-TriGraphs expected by the SIF.

A secondary error was discovered in the LC2MS interface when a few units became Wiped Out, which sets their Higher HQ unit to None. The interface is a command hierarchy tree by Force Side. The Force Side commander is identified as a unit on the Force Side without a Higher HQ, and the tree is constructed downwards from them. One of the newly Wiped Out units was picked up as the Force Side commander, and because it did not have any subordinates that was the only node in the tree.

The LC2MS already had code to translate the database Country Code format to NATO-TriGraphs, but there was an error in the function and it was always returning the input value - in this case NATO-DiGraph. The code was fixed to return the translated value rather than the input value.

Code was also added when searching for the Force Side commander to disregard those units that are Wiped Out. Within JTLS-GO, if the Force Side commander is Wiped Out command will transfer to another unit, so there is no issue with that situation.

3.36 JTLS-2021-15352 MDP E-Mail Delivery Failure

Messages that were designated to be forwarded to an e-mail account were not being delivered. The configuration specifies a header file to be used when the messages are sent as *attachments. This header file may reside in either the $JGAME/data/mdp directory or the $JGAME/data_site/mdp directory.*

When attempting to find the file a stringstream is used to construct the file name and check for it in the first directory. If that fails the stringsteam was cleared and a filename for the second directory was constructed and checked. If that failed the stringstream was cleared again and the default filename in the first directory was constructed and sent to the Sendmail module to use as the body of the message. However, when sent with the Sendmail module, messages were being rejected because the module insisted there was no body to the message.

The error was using the clear() method to reset the stringstream between constructing file names. The clear() method does not actually reset the text but clears any I/O error bits. As a result a file that resides in the $JGAME/data_site/mdp directory would first construct the filename and path with $JGAME/data/mdp, and when the file was not found there would append on $JGAME/data_site/mdp to that.

This, of course, was an invalid directory structure so the file was not found. Then the default path and file were added onto the stringstream and passed to the Sendmail module as the body of the message. Since this path now had 3 filenames all appended together the file was not found and no body was added to the message. The correct method to reset the stringstream is now used and the problem is solved.

3.37 JTLS-2021-15357 Crash Turning Off National ELINT Coverage

The Controller switched the National Asset for ELINT coverage to NONE while an air mission with an emitting sensor was airborne. The action of switching to NONE caused all emitter objects owned by targets and air missions to be destroyed on the JODA. However, the code was attempting to turn off the Air Mission, not its sensor, as a JODA emitter object. This caused a crash for accessing the wrong type of entity.

The code was modified to pass the proper entity, the emitting object rather than the air mission, to the routine that deleted the object.

3.38 JTLS-2021-15359 Dropped Tracks On Ship Detections

JTLS-GO Naval tracks were being dropped on the OTH-Gold Message Service (OTHGMS) periodically for incomplete information. A new track would then be recreated when complete information was again available in the OTHGMS. This was happening even though the information on the track was still valid within the game. The dropping and recreation of the tracks was causing historical information about the tracks to be lost, and was causing the NORCCIS operator extra work to clean the system.

The incomplete status seen in the OTHGMS interface was tracked down to the updating of perceived unit information. As an update is processed each part of the update, such as vector, location, posture, or strength, are checked against the unit to determine if it is the most recent information. This check is performed because the collection and processing of the intelligence occur at different times, and for some sensors happen more quickly than for other sensors.

If the update location was more recent than what was already known within the update process, the location data was stored on local variables and these were used to update the JODA location. This was followed by checking if the COP location was more recent and, if so, the local variables were again used in a JODA update for the COP location. This was done without assigning the local variables. Thus, if a COP sensor made a detection, but before that detection was processed a non-COP capable sensor made and processed a detection that involved location, the processing of the COP capable sensor would set the COP location to 0N,0E.

For the OTHGMS the 0N,0E location is considered to be a lack of a location, resulting in incomplete track information and the dropping of the track. This was solved by assigning the local variables before they are used in the JODA update. This same situation was found in the processing of HRU and Air Mission detections and was fixed.

3.39 JTLS-2021-15361 Maintaining Link 16 Picture

A dropped mission track number would be reused fairly quickly by an unrelated newly acquired mission, which caused confusion, and that if a track was lost and quickly reacquired it was often assigned a different track number.

These shortcomings provided an unrealistic COP picture to the Training Audience, because in real world operations a Link 16 detector would cycle through all its track numbers before reusing them, and operators within an AOC or its sub-nodes would resolve temporarily lost tracks.

Objects now retain knowledge of their most recent Link 16-assigned track number and network for each side. If a track is reacquired by the same network, it can be assigned the same track number.

Track numbers from dropped tracks are placed into an unassigned queue instead of releasing them entirely. If a detector runs out of track numbers, it will release the unassigned track numbers and start using them again. This is a capability that already existed for TBM tracks, but was not being used for Air Mission tracks.

Previously, if a detector ran out of track numbers it would release all unassigned track numbers whereas now it only releases unassigned track numbers associated with its track blocks.

3.40 JTLS-2021-15363 OPM Regeneration Error Reading Checkpoints

A checkpoint existed in the .tim file, which tracks the time of all checkpoints, for which the reason exceeded 82 characters on the line. The reason code was likely the result of a hand edit of the file. But the SIP crashed when it encountered the line because it had opened the file with a recordsize of 80 characters.

The open statement was modified to specify a recordsize of 6000, which is a standard recordsize used in JTLS-GO when dealing with checkpoints. The CEP code, which did not have a recordsize specified and would have defaulted to 132 characters, was also modified to use a recordsize of 6000.

3.41 JTLS-2021-15365 Allow Fire Missile Range Worldwide

A Player attempted to use the Fire Missile order to shoot a long distance on a specific range and bearing but was limited to 1500 KM by the WHIP. This is an insufficient distance for many missiles, particularly TBMs. Whether or not a TBM would be shot on a range/bearing is immaterial to whether the model should support the capability.

The restriction resides in the XML file that defines the order and was switched from 1500 KM to 40,075 KM, allowing a range/bearing to hit anywhere on the world. It was also noted that the Manage Pending Fire order, which allows the attributes of a fire mission to be changed before firing, was also limiting the range to 1500 KM and was changed to 40,075 KM.

3.42 JTLS-2021-15367 Link 16 Source Unit Not On Network Crash

The model crashed when a Link 16-capable sensor owned by a unit with a JU number attempted to access the Link 16 network of the unit, which was not assigned to a network. The situation happened as a Controller was assigning JU numbers, track blocks, and network assignments to units and the detection occurred before the final step.

The code was modified to ignore units that are not on a Link 16 network when determining a JU number to assign to the detection.

3.43 JTLS-2021-15370 Map Networks/Routes Displayed Same Color

Unit ground routes and rail networks were both set to be displayed in brown on the map. It made it difficult to differentiate them on the map when a rail car unit's route that went over a railroad were both set to be on.

The rail networks' map display color was changed to be "darkgray" in the colors file to make them visually distinguishable from ground routes.

3.44 JTLS-2021-15373 JSAT GUI Twenty-Four Satellite Limit

The twenty-four satellite limit on JSAT .tle files is no longer necessary.

The satellite count check was removed from the method SelectAssetPanel.fillAssetTable().

3.45 JTLS-2021-15374 Targets With Multiple Elements

A Target designed to consist of a single element has multiple elements.

The Target types Bridge, Facility, Jammer, Runway, Sensor Site and Communications Target with the type COMMAND should only have a single target Element. Due to the design of the TUP/SUP POT target tables, it is possible for the user to enter values greater than 1 for these targets.

The new SVP ERROR 239 has been added to identify the situation for correction.

3.46 JTLS-2021-15375 Personnel Mean Time To Repair Too Large

The mean time to repair for some personnel combat systems is larger than the faction maximum treatment time at the unit.

The time it takes for personnel to recover should be less than the unit maximum treatment time at the unit as defined in the faction data. Otherwise, this will cause personnel to be considered WIA and then evacuated before they have used their full maintenance period.

The new SVP Warning 1162 has been added to identify the situation for correction.

3.47 JTLS-2021-15376 IIR Generation Cannot Be Stopped

A Recce air mission was equipped with two nearly identical real time imagery sensors, which caused two Imagery Interpretation Reports (IIRs) to be generated for the same Directed Search Area (DSA) at approximately the same time. An Emission Control order was sent to turn off one of the sensors. The Mission Emitter Load Screen IMT and Player message confirmed the sensor was turned off. However, two IIRs continued to be generated from the mission.

In this particular situation, the scenario database incorrectly included a day and night version of the same sensor in the aircraft's default load, with the only difference being the night degradation factor. Only one of the sensors were necessary. The model was correct to generate a separate IIR for each sensor, but with slightly different detection results. The repository database was reviewed and found to be correct: The aircraft type included only one sensor in the default load.

However, a model error did not exclude sensors that were turned off when more than one sensor was present on the mission. To correct this problem, new code was added to skip the intelligence Existing Update if the sensor associated with the update data event is turned off.

3.48 JTLS-2021-15381 ATO Parser Not Adding Route Locations

When a mission includes polygon orbits at the beginning or the ending of its complete list of tasks, the ATO Parser should include point locations for the mission to fly just prior to and just after the polygon, which are the center point of the polygons. This is necessary for timing to correspond more closely with the ICC system.

The Parser was modified to include an additional task point at the start of its tasking whenever the first task is a polygon orbit. The Parser was also modified to include an additional egress point just after its tasking and whenever the last task is a polygon orbit. This allows for better tracking of missions generated by the ICC.

3.49 JTLS-2021-15382 SVP Crash After 5.1 To 6.0 Database Conversion

The customer had a JTLS-GO 4.1 scenario database that needed to be converted to a JTLS-GO 6.0 database. When the conversion process generated the intermediate 5.1 database, an SVP check was run. All SVP errors were fixed before converting from 5.1 to 6.0 (although numerous warnings were not fixed). The SVP crashed when the 6.0 database was checked.

The crash occurred in the routine CHECK DISCODE USAGE. Two issues were discovered.

1. One check found a Target that was a ship whose sub-category index far exceeded the bounds of the array TWO USED TYPES. Access to this array is allowed only for 16 Target Categories, and access should not be allowed for Ships, Combat Arms, Vehicles, Aircraft, and Supply Types.

2. This issue concerns the original IF statement between TG OWNING FEDERATE and .JTLS.ONLY. TG OWNING FEDERATE holds the index that references a FEDERATION AVAIL MODEL entity and should not be used to directly compare with the define-to-means for federated models, such as .JTLS.ONLY. However, the attribute FAM MODEL can be used for this comparison. It would be appropriate to use FAM MODEL(TG OWNING FEDERATE(THE TARGET)) instead.

   The IF statement was modified to correct the issue.

3.50 JTLS-2021-15384 ATO Parser Including Mission Location Tasks

The ICC System often includes potential locations for a mission tasking in the ATO message. These locations are in addition to the planned tasking for the mission but are not included in the routing record for the mission. The Parser is creating the WHIP slides for each of the potential locations but is also including them in the tasking in the order for the mission. The Parser should only be including the task locations listed in the routing for the mission.

The ATO Parser was modified to withhold any locations listed in the mission description that are not included in the mission's routing record. The Parser will continue to create the WHIP slides however. In this way, just prior to sending the order, the operator may decide to plot the slide on the WHIP and use the slide geometry to select new points and modify the tasking for the mission.

3.51 JTLS-2021-15387 ATO Translator Mobility Mission Orders

The ATO Translator creates Mobility mission orders for ATO missions which represent movement of supplies for the exercise. However, the Mobility orders created by the ATOT have air drops of supplies at each air transit point along the tasking. This causes the mission to run out of supplies before completing the mission.

The ATOT was modified so it will not generate transit points with supply drops.

3.52 JTLS-2021-15388 WHIP Command Authority Check Failure

On specific WHIPs, the context order menu was failing to appear for all units owned by the WHIP's force side. The WHIP's unique ID was being stored in a data type on the WHIP that did not match the data type in the rest of JTLS-GO. This caused the value to be misread, which caused command authority checks to fail.

The WHIP now holds its unique ID in a matching data type, so that the value is used consistently with the rest of JTLS-GO.

3.53 JTLS-2021-15390 Create Satellite OPM Page From Checkpoint Error

Technical control attempted to generate OPM pages from a checkpoint during an exercise. The program had a logic error while reading the satellite data from the checkpoint.

The CEP was writing the satellite data for the checkpoint using inconsistent fields from the order called National Asset Pass. In JTLS-GO 6.0, this was modeled as an external event.

The problem was that the name of the satellite was sometimes stored in the order directive, and sometimes in the first text field. The code in the CEP and SIP was modified to consistently use the first text field in the order.

3.54 JTLS-2021-15391 ICP Classification Color Text Misleading

The ICP allows users to change the color of the WHIP's classification banner. The demonstration text for this setting read as UNCLASSIFIED, which may be confusing because it may not match the current classification of the scenario.

The WHIP classification banner color setting's demonstration text has been changed to read as CLASSIFICATION.

3.55 JTLS-2021-15392 OTHGold Module Multiple Error Popups

The OTHGold module queries the service regularly for updates. If the update times out, the module warned the user but the update timer was not canceled. This caused multiple warnings to be generated for the same error.

The OTHGold module now stops the refresh timer and disconnects from the service, as well as warns the user, when a server request times out.

3.56 JTLS-2021-15394 WHIP Unit Combat Map Filter Does Not Set

The WHIP's map filter settings includes the ability to filter units based on their combat status. This filter failed to be set for specific sides and would only work when changing the setting for all sides.

The unit combat filter has been fixed so that individual side settings can be applied.

3.57 JTLS-2021-15398 No Message For Ground Unit Out Of Fuel

When a ground unit conducting a move ran out of fuel, there was no message to inform the Player if the unit failed to submit a requisition for fuel.

A message is now generated informing the Player of the ground unit's current activity trouble.

4.0 REMAINING ERRORS

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

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

4.1 DDSC – TMU Line Mode Changes Multiple Grids

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

4.2 DDSC – Multiple Types In DDS History Table

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

4.3 WHIP - Pipeline Not Shown On IMT

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

4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

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

APPENDIX A. ABBREVIATIONS AND ACRONYMS

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

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

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

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

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

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

OJCS	Organization of the Joint Chiefs of Staff
OMA	Order Management Authority
ONC	Operational Navigation Chart
OPM	Online Player Manual
OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
SDB	Standard Database
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)

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

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

APPENDIX B. VERSION 6.0.0.0 DATABASE CHANGES

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

APPENDIX C VERSION 6.0.0.0 REPOSITORY CHANGES

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