JTLS-GO Document 17

JTLS-GO

Version Description Document

December 2018

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.1.0.0)

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

JTLS-GO 5.1.0.0 is a Major release of the JTLS-GO 5.1 series that includes an updated SBDKOR50 Standard Database, as well as major model functionality changes implemented as Engineering Change Proposals (ECPs), which are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

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

[Blank Page]

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 5.1.0.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 5.1.0.0 is a Major delivery for the JTLS-GO 5.1 series of releases.

JTLS-GO 5.1.0.0 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". Database modifications that were accomplished to upgrade the previous JTLS-GO database format to this current version are summarized in this chapter, as well as APPENDIX B. Detailed descriptions of Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS-GO 5.1.0.0 executes on the Red Hat Enterprise Linux Version 6.6 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

1.2.2 Unchanged Documents

JTLS-GO 5.1.0.0 is a major redesign of the JTLS-GO system and all of the documentation has been updated for this release.

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 5.1.0.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 5.1.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.1.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.1.0.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.1.0.0)

- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 5.1.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 5.1.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 5.1.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 5.1.0.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 5.1.0.0)
- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 5.1.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 5.1.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 5.1.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 5.1.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.1.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.1.0.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.1.0.0)

1.2.4 New Documents

The U.S. Government has provided guidance that JTLS must have its own Configuration Management Plan. This document has been written and is delivered for the first time with this release.

- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 5.1.0.0)

1.2.5 Delivered Software Components

JTLS-GO 5.1.0.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

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

- Database Development System (DDS)

  Database Configuration Program (DCP)
  DDS Client User Interface (DDSC)

- ATO Translator Service (ATOT)
- ATO Generator Service (ATOG)
- ATO Retrieval Program (ATORET)
- JTLS Convert Location Program (JCONVERT)
- Count Critical Order Program (CCO)
- JTLS HLA Interface Program (JHIP)
- After Action Review Client (AARC)
- Scenario Data Client (SDC)
- Order Entry Client (OEC)
- Order Verification Tool (OVT)
- JTLS Object Distribution Authority (JODA)
- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:

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

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service

- KML Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program

Instructions for installing JTLS-GO 5.1.0.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 5.1.0.0 is not necessary. No other upgrade beyond installation of the compressed TAR files (or CD) is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called "repository51". 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 5.1.0.0, it is possible to access and copy records from the repository51 database into your own developed scenarios.
- The scenario “wespac51”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 5.1.0.0 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:

  RedHat Linux 7.6 - this operating system license must be purchased.

CentOS 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 for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 5.1.0.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.6.

  CentOS Linux Version 7.6.

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

- JTLS-GO 5.1.0.0 no longer uses Oracle Java and has moved to the latest version of OpenJDK 8. 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 RedHat Package Manager (RPM) and install the RPM on the servers used by JTLS-GO. This version of JTLS requires OpenJDK 8 update 192. This is also known as OpenJDK 8u192 and an RPM is readily available for download by all user organizations.

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

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

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

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

- SIMSCRIPT II.5 (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS-GO SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc.

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

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

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

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

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

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

- KML Operational Interface (KOI)

  The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth^{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 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, as software that executes on DoD systems, must comply to the mandates of the IA program, along with all of the third party software used by JTLS-GO, such as Oracle and Java.

One of the DoD IA requirements is that the software must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. In previous versions of JTLS, 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 releases were closely linked to the Java expiration date.

Given the currently available information, OpenJDK will not implement an expiration date. In order to fulfill the DoD IA requirement, bug releases of JTLS-GO will be released as new versions of OpenJDK are released. In a future version of JTLS-GO, all Java programs delivered 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).

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 5.1.0.0 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 Gefechts Simulation (GESI) 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 called GlobalSim. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The CAE GESI system is used to represent a virtual battlefield, including weapons, vehicles, aircrafts, ground forces and more. Combining JTLS-GO and GESI brings together operational and tactical level constructive simulations to prepare commanders and staff to make timely, informed and intelligent decisions across the full spectrum of operations, including conventional combat, disaster relief, and operations other than war.

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

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 5.1.0.0 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 5.1.0.0 and the previous JTLS-GO 5.0 series database structure.

To upgrade your JTLS 5.0 scenario to JTLS-GO 5.1 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 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 the *JTLS-GO DDS User Guide*, Appendix B.11. 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 5.1.0.0, 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

This version of JTLS-GO is the first version in which R&A is delivering an unclassified data repository called “repository51”. In future released versions of JTLS-GO 5.1.0.0, APPENDIX B. will provide a summary of the data changes made to the data repository, In the next major release of JTLS, tentatively labeled as JTLS-GO 6.0.0.0, the user will be able to automatically view changes by comparing the two different versions of the “repository51”.

1.5 INSTALLATION

1.5.1 Installation Instructions

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 a new version of JTLS-GO.

1.5.2 Oracle Compatibility and Installation

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

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

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

Between the free Express Editions (XE) of Oracle, the 18c XE has a large footprint and a 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 and requires no separate installation.

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

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 5.1.0.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-0055 Range Specifications As Polygons

Summary of Model Change Request

All range specifications that are currently restricted to a radius should also be specified as polygons. This should apply to Combat Air Patrol (CAP) missions and Rules Of Engagement (ROE), especially for Surface-to-Air Missiles (SAM). This is especially important because some SAM sites only protect small polygonal areas.

Design Summary

In previous versions of JTLS-GO, CAP missions could already protect a polygonal Operations Area (OPAREA). No changes were made to the model as part of this required,

This ECP did expanded the coverage definition of a SAM Site and Sensor Site to use either the already existing circular protection area or partial sector of a circle. a polygonal protection area. Before firing on or detecting an object, the model ensures that the object is within the physical sector capability of the Site.

When firing against a Theater Ballistic Missile (TBM), the model determines if the predicted impact point for the TBM is within the newly-added polygonal protection area given to an ADA site.

2.2 JTLS-0431 GPS and Infrared Jammers

Summary of Model Change Request

Jammers are currently assumed to be radar jammers that interfere or jam the ability of a sensor to detect objects. JTLS-GO does not currently represent the jamming of a weapon’s guidance capability.

The capability to define various types of weapon guidance mechanisms used by precision guided weapons should be provided, and also allow the opposing force to employ jammers capable of interfering with the weapon’s guidance system. Each weapon should have a guidance type, possibly both a long-range and/or terminal capability. Jammers should specify which type of guidance they are capable of jamming.

Design Summary

In previous versions of JTLS-GO, only an “Anti-Radiation Missile” (ARM) weapon guidance capability was represented. The types of weapon guidance capabilities was expanded to include Infrared (IR), Global Positioning System (GPS), LASER, Electro-Optical (EO), and Radar missile guidance.

The available types of jammers was also expanded from communications, PSYOPS, and radar jammers (which jammed radar sensors) to allow jamming of each of the new weapon guidance capabilities and ARM. If a weapon guidance jammer covers the impact location of the weapon using the jammed weapon guidance capability, then its probability of hit (PH) is adjusted.

Additionally, ARM weapons are only be allowed to fire on Targets that have a currently “TURNED ON” and have electronic emitter.

2.3 JTLS-2006-1653 Represent Unmanned Underwater Vehicles

Summary of Model Change Request

In naval operations, unmanned vehicles have become an important force multiplier in conducting maritime missions. These vehicles, operating at or below the sea surface, reduce risk to personnel by performing missions that manned vehicles often cannot. These relatively small, tactical assets are becoming more significant participants in asymmetric warfare and should be represented in JTLS-GO.

Design Summary

Previous versions of JTLS-GO did not represent small, unmanned vehicles employed from naval vessels or ground combat units. JTLS-GO did model Unmanned Aerial Vehicles (UAVs) as individual aircraft flown from squadrons to conduct reconnaissance, SEAD, or Offensive Air Support (OAS) missions. There is no similar modeling of unmanned vehicles that operate on or under the sea surface, or on land. Submarines, represented as naval units with crews, are the only assets in JTLS-GO that operate below the sea surface.

Unmanned Underwater Vehicles (UUVs), Unmanned Surface Vehicles (USVs), and Unmanned Land Vehicles (ULVs) are essentially small, tactical assets that perform various dedicated missions. They can operate remotely or autonomously and are deployed when required. As such, it is not practical to model these Unmanned Vehicles (UVs) as full-fledged naval or ground units in JTLS-GO. To do so would populate the scenario database and WHIP display with many, ever-present, limited capability units that must be managed by players.

The most logical and expedient way to represent UVs in JTLS-GO is through High Resolution Units (HRUs). HRUs are currently used in JTLS-GO to represent lifeboats, small ground units, SOF direct action teams, and covert Human Intelligence (HUMINT) teams. An HRU that can move on the water is equipped with an organic Small Boat (e.g. a RIB or amphibious craft). HRUs are created by the player (or deployed) when needed from a parent unit and may rejoin the parent when no longer needed. They are capable of performing a wide variety of missions as ordered by the player, depending on their assigned combat systems and targets.

An HRU is defined by its HighRes Unit Prototype (HUP). The HUP specifies the types and quantities of combat systems, targets, and supplies that equip the HRU upon creation. As a result of this ECP, HUPs were modified to effectively represent particular types of UVs. New HUP attributes were due to the necessity to model control type (i.e. remote or autonomous), control distance limitation, the most critical combat system, and whether the created HRU is a one-time use weapon system.

An HRU that represents a UV does not have a personnel combat system assigned. Instead, it has a specific critical combat system that, if destroyed, renders the HRU completely ineffective. This critical combat system could represent an on-board artificial intelligence system (i.e. a piloting robot) or some other specified combat system. Each UV also has, if applicable, an organic Small Boat that is also present as a combat system to allow for attrition. The UV is allowed to own targets as is currently true for other HRUs.

An HRU that represents a UV is capable of performing reconnaissance and weapon delivery functions. The sensor targets assigned to the UV automatically allow specific functions to be carried out. Depending on the sensor type, the UV has the capability to perform reconnaissance, jamming, or IADS communications relay. If an SSM target is assigned to the UV, it is also able to fire one or more munitions (targetable weapons) at other vessels or land targets. If an ADA target is assigned, the UV is able to fire at airborne threats.

An HRU that represents a UUV or USV must be equipped with an organic Small Boat and a corresponding Combat System Amphib. Small Boats are already used in JTLS-GO for HRUs that are capable of moving along a waterway. New Small Boat attributes were necessary to model the behavior of the UUV or USV. Operational depth (which could be zero) and generated noise level were added to the Small Boat data structure.

In previous versions of JTLS-GO, HRUs were destroyed when the last personnel combat system was killed, even if the HRU had fully functioning other non-personnel combat systems. Because the UV HRU does not possess any personnel combat systems, it is destroyed when its designated critical combat system is destroyed. Additionally, if the UV's Small Boat Combat System Amphib is destroyed, the entire UV is considered destroyed.

UV functions that were considered by the design team, but were not be implemented in this ECP, are submarine decoys, submarine crew rescue, illegal contraband trafficking, and endurance.

2.4 JTLS-2007-2072 Allow Attach/Detach Targets

Summary of Model Change Request

Users who detach a unit cannot detach the targets. Detaching 30 percent of a unit removes 30 percent of its Combat Systems. Users cannot detach some or all targets owned by the unit from which the detachment is being taken. The Detach order should be modified to allow this capability.

Design Summary

JTLS-GO was enhanced to:

- Allow individual targets to be detached from a unit as part of the detachment process.
- Allow a target to be transferred from one unit to another.
- Allow a target to be moved from an owned status to an unowned status, and vice versa.

There were no changes to the attachment process. It continues to result in a unit transferring all of its combat systems, supplies, owned targets, and associated targets to the receiving unit.

2.5 JTLS-2008-10000 Database Repository

Summary of Model Change Request

JTLS-GO provides a set of standard databases, both as demonstrative representations of real world scenarios and as baselines from which users can build their own databases. These two purposes can be conflicting. For demonstration purposes the data in the database should be concise, but when considering a baseline data repository. the data should be robust.

JTLS-GO should provide a way to quickly and easily pull data from or load data into a single large repository of data, from which users will be able to build exercise scenarios using data tailored to the exercise parameters, and which will have the capability to synchronize the data during or after the migration in order to maintain a comprehensible repository.

Design Summary

The Data Repository System is introduced as an added enhancement to the Database Development System. It is a tool that can migrate data from one database to another, enabling database builders to easily put data from various scenarios into a single repository, and grab pertinent data from the repository and copy it to other exercise databases.

With this migration tool, users are able to maintain their own repositories as large databases that can either be demonstrative gaming scenarios containing data representing multiple conflict regions, or used only as a mean to hold extensive data, thus alleviate the need to use standard databases as baselines to build other exercise scenarios.

2.6 JTLS-2009-10195 Non-Squadron Units Fly UAVs

Summary of Model Change Request

This request is similar to and has been subsumed into JTLS-2013-11748 Better Representation Of UAV Use By Ground Units, Section 2.10. Refer to that design for all details.

2.7 JTLS-2009-10414 Create Aircraft Target Class ROE For SAM Sites

Summary of Model Change Request

Aircraft can currently be directed not to intercept certain types of aircraft based on the Aircraft Target Class (ATC) attribute. JTLS-GO should provide the same capability for Surface-to-Air Missile (SAM) sites in which Rules Of Engagement (ROEs) to Fire or Not Fire can be set according to the same ATC attribute.

For example, a SAM site could allow firing on Cruise Missiles or Theater Ballistic Missiles (ATC Missiles), or SU-27s (ATC Advanced Fighter), but hold fire on UAVs (ATC UAV) or Helicopters (ATC Rotary).

Design Summary

Before allowing a SAM Site to fire on a missile or aircraft, the object must meet the following three requirements. The object must be:

- Within the Altitude Zone and Range restrictions of the SAM Sites engineering capability. Note that in previous versions of JTLS, this engineering capability was represented as a circle, but as a result of JTLS-2013-11722 Non-Circular Sensor And Air Defense Ranges, this capability is represented as a sector.
- Within the ROE fire range restriction of the SAM Site. This will not change as a result of this ECP. The SAM target will continue to obtain its range restriction from either its owning unit or for those sites without an owning unit from the site’s associated unit.
- On the list of allowable ATCs. This is the essence of the ECP and is new to the model as the result of implementing this new capability.

2.8 JTLS-2013-11683 Multiple Line Delete In DDS

Summary of Model Change Request

Allow the user to highlight several rows of a table and delete all of the records that are highlighted. The preference is to use the standard Shift and Control keys to highlight the rows the user wants to delete.

Design Summary

Multiple line capability was not be simply limited to the deleting of records, but also included other multiple line capabilities desired by users. Three multiple line capabilities are delivered as part of this ECP:

- Multiple line or record delete from a table.
- Multiple line attribute update.
- Multiple line Scenario Verification Program (SVP) correction.

2.9 JTLS-2013-11722 Non-Circular Sensor And Air Defense Ranges

Summary of Model Change Request

All sensors have a range and are assumed to cover a circular area. JTLS-GO does not have the ability to represent coverage areas other than circles, such as side looking radars or SAM/AAA sites that can only fire within a specific sector. This means we need to represent off-set areas and partial sectors for a circle for both sensors and SAM/AAA sites.

Design Summary

The Sensor Type target prototype and the Air Defense target prototype each had an additional attribute, call the coverage sector added to the prototype definition. ST COVERAGE SECTOR was added to the Senor Type target prototype and AD COVERAGE SECTOR was added to the Air Defense target prototype. This attribute defaults to 360 degrees, resulting in the model operating exactly as it does now. The user is able to set a specific target’s direction using a new user order.

2.10 JTLS-2013-11748 Better Representation Of UAV Use By Ground Units

Summary of Model Change Request

Small-scale, tactical Unmanned Aerial Vehicles (UAVs) are becoming commonplace assets within ground units in the real world, and the capability to launch tactical UAVs from non-squadron ground units to fly reconnaissance and attack missions should be represented within JTLS-GO.

Design Summary

In previous versions of JTLS-GO, small tactical UAVs are implicitly represented by the organic tactical intelligence collection range of their owning ground unit. An increased range reflects the capability of the unit to gather intelligence using tactical UAVs.

As a result of this ECP, a single UAV aircraft type is allowed to fly from each ground unit. The UAV needs to be represented as an aircraft combat system in the ground unit. Only one UAV type is permitted for each unit. This restriction implies that only the most capable UAV type would be explicitly modeled in each unit. Such a restriction is consistent with current doctrine that typically assigns only one primary UAV type to any given ground unit. Any remaining UAV types would be presumed to be less capable or more tactical and would be represented implicitly in the unit's tactical intelligence range as previously modeled.

2.11 JTLS-2014-12336 Represent New Laser Technology

Summary of Model Change Request

Laser Weapon Systems (LWS) for ground, naval and aircraft are progressing through test and development in US military systems. Laser/energy beam uses unique technology, but produces weapons effects similar to a kinetic mass munition. The LWS is distinctive in that it has no constraints of munition flight and very minimal consumption, reload logistics, or weight.

The ability to represent new Laser Weapon Systems (LWS) should be incorporated into JTLS-GO. The LWS in JTLS-GO will have capability to fire on and degrade or kill enemy assets in dual Surface-Surface and Surface-Air domains using the same Air Defense (AD) target.

Design Summary

The LWS naval weapon can easily be modeled as a new type of targetable weapon (TW) using the existing JTLS-GO TW data structure, In addition, Targets and Lethality Tables already can realistically model LWS in the operational environment. What needed to be changed to fully implement the intent of this ECP was the ability for an Air Defense Target to fire either on aircraft or on other surface objects. The Air Defense Target capability was expanded to provide for this ability needed to properly represent the LWS naval weapons.

The ECP did not consider issues such as power output requirements, or th non-lethal aspects of LWS, such as blinding or disrupting effects. These issues need to be addressed as part of a separate ECP not delivered as part of JTLS-GO 5.1.0.0.

2.12 JTLS-2015-12458 Multiple TADIL Link-16 Feeds

Summary of Model Change Request

There are two reasons clients have provided for needing the representation of two independent Link-16 feeds:

1. Multiple TADIL JOI feeds are needed to properly simulate the real world Command Control Communication and Computer Information (C4I) system structure. In the example C4I system structure, the Navy C4I system gets all Navy Link-16 data from ships and naval surface search radars, while the Air C4I system gets all Link-16 information from other intelligence sources and radar sites. After that, both Navy and Air C4I systems sends their own Link-16 format data to the national C4I system. Currently, JTLS only represents the single Link-16 feed. This means that during a JTLS exercise, the individual Link-16 feed to separate systems cannot be represented; therefore, the exercise audience cannot make it an exercise audience responsibility to handle the technology of merging the Link-16 feeds for the national systems.

2. Some users hold exercises in which two or more sides are fully functioning staffs and each side needs to have its own independent Link-16 feed to the C4I system that represents the national level Common Operational Picture (COP).

Design Summary

Each Link-16 capable asset must now indicate the Link-16 network and Link-16 feed to which the assets is connected. This can be done as part of the database building process or established by the Controller after game execution starts.

The JTLS Operational Interface (JOI) also must identify the Link-16 network and Link-16 feed for the C4I system to which it is linked. The JOI will only pass information to its assigned C4I system if the Link-16 network and feed of the collected information matches the Link-16 network and feed assigned to the JOI.

2.13 JTLS-2015-12485 Allow WHIP to Display GeoTIFF Background

Summary of Model Change Request

CADRG maps are no longer used for many purposes and it is becoming more difficult to obtain these maps for use with JTLS. GeoTIFF maps are more commonly available and the WHIP and DDSC should be allowed to use a GeoTIFF background in addition to the CADRG background.

Design Summary

Support for displaying GeoTIFF map files in the WHIP and DDSC has been added. Similar to the previously delivered capability for displaying CADRG files, the WHIP and DDSC search for a "GeoTIFF" directory within the user’s home directory and within several locations on the Linux file server.

To simplify the design, the user is required to provide a separate data file for some GeoTIFF files to indicate the upper left and lower right latitude and longitude coordinates.

2.14 JTLS-2015-12497 Add Control Linkage Range to Appropriate Aircraft

Summary of Model Change Request

Positive command and control of Unmanned Aerial Vehicles (UAVs) is sometimes limited to line of sight or other communications distance restrictions. JTLS-GO should be enhanced to apply operational range constraints based on communications/line of sight distances to aircraft. These enhancements will introduce a control linkage range that will prevent an aircraft from flying beyond that distance.

Design Summary

Communication range restrictions are now imposed on certain aircraft types, primarily UAVs. A new aircraft attribute has been added to an Aircraft Class definition. This new attribute AC CONTROL DISTANCE has the following meanings:

- A value of zero indicates that the aircraft has no control range limitations.
- A value greater that zero indicates that the aircraft must stay within the specified range of its home unit. The model does not allow the UAV to fly beyond this distance from the launching unit. A range ring can be displayed over the launching unit on the WHIP.
- A value less than zero indicates that the aircraft must simply stay above the radar horizon from its home unit. As the air mission travels further and further away from its home unit, its altitude will increase, until it reaches the maximum operation altitude specified for the aircraft class.

Although terrain masking is applied in the JTLS-GO detection algorithms, terrain masking is not a factor in determining the control range of a UAV. This decision was made considering the short ranges and low altitudes of small tactical UAVs relative to terrain grid sizes made the check unnecessary.

2.15 JTLS-2016-12703 Multi-Side Functionality for WHIP/TRIPP

Summary of Model Change Request

A Player WHIP only allows viewing and interacting from one Force Side at a time. This is also true of a Controller WHIP, which provides the ground truth perspective and interactions of a theater of battle or a humanitarian relief scenario. A Controller WHIP does not provide the frame of reference of each individual Force Side's perceptions. Additionally, a Controller WHIP does not allow interaction with forces organically from the underlying Force Side, to manage its combat systems and supplies and order its units. These restrictions serve a well defined capability.

It is desired to include an additional capability for a specialized WHIP that will allow tailoring it whenever there is a need to interact or order units of different Force Sides, such as all Coalition Forces, all Opposition Forces or all Situational Forces, all capable of being done from one WHIP.

To fulfill that need now, a WHIP representing each desired Force Side would need to be accessed to view Force Side perception information or send orders to its units.

Design Summary

A single WHIP or Total Recall Interactive Playback Program (TRIPP) is permitted to view different perceptions on the map, Command Hierarchy, and other WHIP component interfaces individually. The multi-sided WHIP includes the ability to submit orders to units of different Force Sides.

Technical Controllers are able to easily configure a multi-sided WHIP from the Interface Configuration Program (ICP).

2.16 JTLS-2016-12705 Robust Filtering for AAR Collection

Summary of Model Change Request

The JTLS-GO After Action Review Client (AARC) is responsible for collecting AAR objects from the JODA and inserting the information into database tables. This information includes objects, object events, and interactions. For interactions, such as object detections and attrition events, the amount of information can become extremely large, slowing down the saving of data during checkpoints.

In an attempt to limit the amount of data saved, a filtering capability currently exists. This filtering capability can turn off the saving of information to specific tables, or can be used to turn off the saving of specific object events. No capability exists for robust filtering, which might include only saving interaction information if one of the objects involved belongs to a specific force side. For example, we may only care about detections of ALLIANCE and OPFOR units, not target detections or detections of units on other sides. This ECP outlines a filtering capability to allow user-definable robust filtering of AAR objects.

Design Summary

The filtering mechanism was created to provide the AAR team to define the JODA information that should be saved in the AAR database. In this way, additions or changes to the AAR data will not necessitate new code to support filtering. This type of filtering already exists in JTLS-GO for object selection within orders. The order filters provide for the specification of an object type, and comparison of any attribute of the object, including compound object references.

To implement the user defined filtering capability, an interface for the Technical Controller is provided for the generation and modification of the AAr filter file. This interface has been incorporated into the ICP, allowing the Technical Controller to turn on/off individual tables, or setup the robust comparison filters.

2.17 JTLS-2016-12737 UAV Turns Into Weapon

Summary of Model Change Request

JTLS-GO should be enhanced to model Unmanned Aerial Vehicles (UAVs) that are flown as expendable aircraft. These UAVs are essentially airborne weapons that are destroyed if they attack a ground target, but return home if no attack occurs.

Design Summary

In previous versions of JTLS-GO, there was no capability for a UAV (or any aircraft type) in JTLS-GO to accomplish a one-way mission. In other words, a UAV could not launch, fly reconnaissance for a period of time, and then terminate with a self-destructive attack on a ground target, or alternatively return home without attacking any target.

An expendable UAV (e.g. Harpy), is now represented as a new aircraft class that models a loitering munition. A new aircraft attribute indicate whether the UAV represents a one-way asset. The UAV is explicitly flown as an air mission that carries a single weapon. It is subject to air intercept and ADA engagements while flying. The aircraft is immediately destroyed after it attacks a ground target. If the UAV mission is not assigned a target to attack, it will return home after its off-station time or upon receiving further orders.

The expendable UAV is controlled using existing air mission orders. The player is able to launch the UAV as a Reconnaissance, Offensive Air Support, or a SEAD mission. The Manage Air Mission Tasks order can be used to add, modify, or delete mission tasks. If the player adds a Fire Weapon task, the mission is automatically destroyed after task execution. The mission is capable of loitering and gathering intelligence information, if it carries the proper ground search sensors.

2.18 JTLS-2017-13199 Remove Combat System Prototype

Summary of Model Change Request

The JTLS-2008-10000 ECP, to develop a Data Repository, has been implemented in JTLS-GO 5.1. The Data Repository includes as a design goal the provision of an easy method to synchronize data that exists in two different databases. Data synchronization involves the identification of similar records, the identification of differences between these similar records, and the implementation of the database builder’s decision to keep the data as is or migrate the data from the repository into the scenario database.

The identification of similar records, and thus the entire synchronization process, is being hindered by the representation of the Combat System Prototype (CSP) structure, so the CSP structure should be removed from the JTLS-GO database.

Design Summary

The implemented ECP does not completely remove the CSP structure from the JTLS-GO database, but moves the majority of data from the CSP Combat System (CS) structure to the Combat System (CS) structure. In addition, for consistency, the design also remove portions of the Sustainment Logistics Prototype (SLP) data to the Supply Category (SC) structure.

The majority of data structure changes made for JTLS-GO 5.1.0.0 were required as a result of implementing this ECP. The most significant difference is that user organizations need to represent different Combat Systems as much as possible. The delivered “repository51” contains over 3000 different Combat System from which to choose when building scenario Order of Battle (OOB) data. Scenario databases should only include those Combat Systems needed for the specific scenario OOB.

2.19 JTLS-2017-13200 Improve Grid Movement Optimization

Summary of Model Change Request

JTLS-GO provides the capability to move entities either on a movement network or off-network on a direct path. A JTLS-GO scenario may incorporate multiple road, rail, and river networks, as well as a single worldwide air network and a single worldwide sea lane network. Off-network movement can be generalized to a direct route that follows the great circle from a starting location to a designated destination.

Entity off-network movement on the underlying terrain grids, particularly when an entity encounters an impediment to movement that requires the entity to detour around the obstacle and continue to its destination, needs to be improved.

Design Summary

The current direct route approach to off-network planning and movement has been replaced with an approach similar to planning and movement on a movement network. The approach taken by this change request is based upon methods applied to the cross country movement (CCM) problem. Although planning for the CCM problem is a land navigation problem, the basic principles can be carried over to the sea and air realms, as well.

If an object cannot find a network route to its assigned movement location and a direct route is not feasible due to intervening terrain, the model now created a grid-based network to find a path around the intervening terrain.

2.20 JTLS-2017-13201 Link-16E

Summary of Model Change Request

The Link-16 format has been expanded. This expanded format is named Link-16 Enhanced Throughput or Link-16E. Currently JTLS-GO does not support many of the new fields associated with the Link-16E format. The purpose of this ECP is to add some of the new data that is part of the Link-16E format.

Design Summary

The following Link-16 message formats have been updated to support the new Link-15E format:

- J2.2 (Air PPLI), J2.3 (Surface PPLI), and J2.5 (Land PPLI) messages used for own force information. New data has been added to these messages as part of this ECP. In addition JTLS-GO 5.1.0.0 now support J2.4 (Sub-Surface PPLI) messages.
- J3.2 (Air Track), J3.3 (Surface Track) messages used for detected track information. Previously, JTLS did not support J3.5 (Land Track) messages. As a result of this ECP, J3.5 messages are now supported. In addition, new Link-16E data was added to the currently supported messages as part of this ECP.
- J13.2 (Air Platform), J13.3 (Surface Platform) messages used to report supply levels for own force assets. In previous versions of JTLS-GO, J13.5 (Land Platform) messages were not supported. As a result of this ECP, J13.5 messages are supported. In addition, new Link-16E data was added to the currently supported messages as part of this ECP.

2.21 JTLS-2018-13691 Add A General Ship Icon

Summary of Model Change Request

When a ship is detected at the limited information levels of Localization or Classification, the WHIP symbol displayed for the Naval Unit provides too much information to the user. A symbol other than the database parameter SUP GRAPHICS SYMBOL should be used until the intelligence collection capability has more detailed information collected on the ship.

Design Summary

Much information about ships may be inferred from their authoritative unit icon. Every ship is derived from a Ship Unit Prototype (SUP). Each SUP has an attribute for the unit symbol. This SUP-based symbol represents the truthful properties of a ship built from that SUP. When a Foreign ship is seen, but is not fully identified, then the true and full properties of that ship should not be exposed by its symbol on the WHIP.

A new global variable has been added to the JTLS-GO database to represent the icon for unidentified ships. This variable shall reference an existing symbol in the scenario database. When a ship is detected at the limited level of Localization or Classification, then the ship is displayed on the WHIP using this symbol. When more detailed Recognition or Identification information is collected, then the symbol change to the symbol specified by the database parameter SUP GRAPHICS SYMBOL.

2.22 JTLS-2018-13693 Add Individual Target JU Numbers

Summary of Model Change Request

The only way to assign a target a Joint Tactical Information Distribution System (JTIDS) Unit Number (JU) number is to give its owning or associated unit a block of JU numbers for targets. It is impossible to assign a specific target a specific JU number. This database capability should exist.

Design Summary

Within JTLS, a unit can be given a target to operate in one of two ways:

- The target can be defined as part of the unit's prototype definition. These targets are called Prototype Owned Targets (POTs). POTS do not exist in the database. They are created by the model when the unit is created during database initialization. Since these targets do not exist in the database, they cannot be assigned a JU number in the database; instead, the unit that will own the targets is given a block of JU numbers. When the targets are created during database initialization, a JU number from the block is assigned to the Link-16 capable targets.

  This process was available in previous versions of JTLS-GO, and has not been changed as a result of this ECP.

- A target can be specifically defined in the database. These targets are records within the database, but due to an oversight, in previous versions of JTLS-GO, the database builder could assign specific JU numbers to the Link-16 capable targets. If a target should have a JU number, the user had to wait for the game to be initialized and via a Controller order set the target's JU number.

This ECP simply adds the ability to assign a non-POT target a JU number as part of the database build procedure.

2.23 JTLS-2018-13704 Order Search Capability

Summary of Model Change Request

Over time, the number of available user orders increases, and it becomes more difficult to remember where within the order menu structure a given order is located. The desire is to build an order search capability. The user should be able to enter the name of an order and have the order panel come up.

Design Summary

Two requirements exist for this ECP:

- Only orders that are allowed to be entered by the WHIP should come up as part of the search results.
- The entry should not require the user to input the full name of an order.

To accomplish these goals, the WHIP Order menu has, as the first selectable menu item, a search order option. When this option is selected, a input box is displayed, in which the user can enter a portion of the desired order name. The WHIP searches the entire menu and lists the orders that match the search string. The user can then select the desired order from the list, and the order panel is displayed

2.24 JTLS-2013-11738 Represent More Than Ten Sides

Summary of Model Change Request

The purpose of this design is to allow JTLS-GO to operate using more than ten sides. This proposal will increase this number to a total of twenty sides. To support this change, the number of available colors for the sides will also increase.

Design Summary

Users are now allowed to build scenario data with up to twenty force sides. Previous versions were limited to no more than ten sides in a scenario. Given that JTLS-GO supports operations on a global scale, this ECP allows for a better representation of conflict between diverse groups located in multiple, critical areas around the world.

2.25 JTLS-2018-13772 Allow Various Country Codes For C4I Systems

Summary of Model Change Request

JTLS-GO currently assumes that all C4I country codes are two characters long. Some systems require three-character country codes.

Furthermore, the country codes for NATO systems are different than those used by US systems. The JTLS Operational Interfaces need to support these differences.

Design Summary

This ECP considers several situations:

- Some C4I systems require 3 character country codes, while some systems require 2-character country codes. When one system is communicating with a different system, it is a C4I system issue and not a JTLS-GO issue to bridge the communication barrier. On the other hand, if two independent JTLS Operational Interfaces (JOIs) are feeding two different systems, it is the responsibility of JTLS-GO to output the proper code recognized by the system.

- Different countries have different C4I systems with different requirements with respect to the allowable country codes. It is the responsibility of the delivered JOIs to output the proper country codes for the different country requirements.

- Some exercises use fictitious countries and the fictitious country code requirements for the various systems to which JTLS can link are different.

- Not only do the various JOI processes feed real-world C4I systems, but several messages directly generated within the model are passed to C4I systems. This means that a method to alter their country codes is also needed.

When building a JTLS database, each of the represented database Political Countries (POC) has a single assigned Country Code. It is up to the database builder to ensure that all POCs use a consistent Country Code standard. A new database parameter has been added, which identifies the Country Code standard that was used when building the database.

The JOI obtains this information from the database. As part of the setup procedure, the JOI is told what Country Code standard should be output to the C4I system. The JOI then accomplishes the translation between the Country Code standard sent to the JODA from the model and the Country Code standard expected by the C4I system being supported by the JOI.

2.26 JTLS-2018-13796 Specific Selection For Range Rings

Summary of Model Change Request

If an aircraft, ship, or ground unit has different sensors, combat systems, or weapons on board, then JTLS-GO currently only displays range rings for the one with the largest range.

Design Summary

In previous versions of JTLS-GO, a user had the ability to display “on-demand” range rings for objects by using the context sensitive menu (right-clicking on the object). There were no specific options for these on-demand range rings; the WHIP always displays the ring with the largest range for that object. This ECP improved on the existing range rings by allowing the user to select the specific type of object for which a range ring will be displayed.

For example, if an air mission has 3 on-board sensors called A, B, and C. Sensor A has a range of 10 KM, sensor B has a range of 50 KM, and sensor C has a range of 25 KM. Using the pervious JTLS-GO capability, the on-demand range rings always display a 50 KM range ring for sensors on this air mission. This ECP allow the user to select which sensor (A, B, or C) should be displayed when a range ring is requested.

This range ring capability works not only for sensors, but Combat System ranges, and weapon firing ranges.

2.27 JTLS-2018-13797 Expand Allowable Length Of Classification

Summary of Model Change Request

Many of the classifications that are needed to support coalition exercises are not as easy as “Secret” or “Top Secret”. Many of the needed classification markings must include the countries and/or organization to which the data can be released. The current classification text string can only be 20 characters long, with no spaces allowed.

At least 40 characters are required for Game Classification. The database parameter needs to be expanded to accommodate this requirement.

Design Summary

The maximum length of the full-name classification has been increased from the previous value of 20 characters to 50 characters. In addition, the classification is allowed to contain spaces.

3.0 SOFTWARE TROUBLE REPORTS

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

Chapter 4.0 describes STRs that remain outstanding from previous versions; however, because of the model enhancements included with JTLS-GO 5.1.0.0, no previously-identified STRs are considered valid. Errors identified for JTLS-GO 5.1.0.0 and corrected for future Maintenance releases in the JTLS-GO 5.1 series will be documented in this chapter.

3.1 JTLS-2018-13977 DDSC Supply Category Preference Issue

Setting UOM conversion preferences in the DDS preference tool for Supply Categories, after modifying (adding/updating/deleting) them, often led to incorrect Supply Category UOM conversion.

The Supply Category index was used to track the Supply Category UOM preference. When Supply Categories are added, renamed, or deleted in the DDS, the indices of that Supply Category, as well as subsequent ones, would be affected. The Supply Category name is now used to track UOM preferences.

4.0 REMAINING ERRORS

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

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

4.1 DDSC – TMU Line Mode Changes Multiple Grids

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

4.2 DDSC – Multiple Types In DDS History Table

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

4.3 WHIP - Pipeline Not Shown On IMT

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

4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

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

4.5 WSM - Many Messages Cause Lockup

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

APPENDIX A. ABBREVIATIONS AND ACRONYMS

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

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

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

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

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

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

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

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

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

APPENDIX B. Version 5.1.0.0 DATABASE CHANGES

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

B.1 ADDED DATA TABLES/FIELDS

1. Added 10 colors to the Force Side Color Lookup table - BROWN, CORAL, LAVENDER, LIME, MAROON, MINT, NAVY, OLIVE, PINK, and TEAL.
2. Added the values ELECTRO-OPTICAL, GPS, INFRARED, RADAR, and REMOTE-LASER to the tw_guidance_type_lu lookup table.
3. Added the values GUIDANCE_ARM_JAMMER, GUIDANCE_EO_JAMMER, GUIDANCE_GPD_JAMMER, GUIDANCE_IR_JAMMER, GUIDANCE_LASER_JAMMER, and GUIDANCE_RADAR_JAMMER to the jammer_type_lu lookup table.
4. Created the hup_control_type_lu lookup table and populated with the values AUTONOMOUS, MANNED, and REMOTE.
5. Created the sc_special_capability lookup table and added the values BARRIER, BRIDGING, CASUALTIES, CLASS.III.AIR, CLASS.III.GROUND, CLASS.III.NAVY, CLASS.IV, LEAFLET, PERSONNEL, and REMAINS to it.
6. Created the cs_special_capability_lu lookup table and added the values AIRCRAFT, AMPHIB, CARGO, CREW, HET, PERSONNEL, and TANKER to it.
7. Added SB.NOISE and SB.OPERATIONAL.DEPTH to the small_boat table.
8. Added HUP.CONTROL.TYPE, HUP.CONTROL.DISTANCE, HUP.CRITICAL.COMBAT.SYSTEM, HUP.LOITERING.MUNITION.FLAG, and HUP.TRAFFIC.CONTROL.FLAG to the high resolution prototype table.
9. Added AIRCRAFT.TYPE to the airbase, farp, ground_unit, naval_unit and support_unit tables.
10. Added the max_sorties_per_day field to the airbase, farp, ground_unit, naval_unit and support_unit tables.
11. Added AC.MAX.CONTROL.DISTANCE and AC.LOITERING.MUNITION.FLAG to the aircraft class table.
12. Added ST.COVERAGE.SECTOR to the sensor_type table and AD.COVERAGE.SECTOR to the air_defense_class table.
13. Added ICON.GEN.SHIP.SYMBOL to the global_values table.

14. Created the link16_surface_platform_lu, link16_surface_type_lu, link16_subsurface_platform_lu, link16_subsurface_type_lu, nato_c4i_ship_type_lu, and oth_gold_category_lu lookup tables.

15. Created the ship_real_world_data lookup table and added the following variables to it:

- SRW.NAME
- SRW.LINK16.SUBSURFACE.PLATFORM
- SRW.LINK16.SUBSURFACE.TYPE
- SRW.LINK16.SURFACE.PLATFORM
- SRW.LINK16.SURFACE.TYPE
- SRW.NATO.C4I.TYPE
- SRW.OTH.GOLD.CATEGORY
- SRW.OTH.GOLD.TYPE

16. Added JU.NUMBER to the sam_aaa_target, sensor_site_target, facility_target, ssm_target, and communication_target lookup tables.

17. Created the link16_structure lookup table and added LINK16_NETWORK_NAME and LINK16_FEED_NAME to it.

18. Added FAT.LINK16.FEED.NAME, FAT.LINK16.NETWORK.NAME, SAM.LINK16.FEED.NAME, SAM.LINK16.NETWORK.NAME, SMT.LINK16.FEED.NAME, SMT.LINK16.NETWORK.NAME, SS.LINK16.FEED.NAME, and SS.LINK16.NETWORK.NAME.

19. Created the SP_SUT child table and added SP.NAME, SUT.NAME, and SP.SUT.PACKET.SIZE to it.

20. Created the TUP_CS, SUP_CS, and HUP_CS child tables.

21. Created the SUT_SLP_SLP child table and added the following variables to it:

- SUT_NAME
- RECOVER_SLP_NAME
- DUMPING_SLP_NAME
- SUT_SLP_SLP_DUMP_USBL_FRACTION

22. Created the combat_arms_type table, mapped the old Combat Arms Type (CAT) subcategories to new CAT names, recreated the Foreign Key (FK) constraints, and added the following variables to it:

- CAT_NAME
- CAT_RADIUS
- CAT_REPAIR_TIME
- CAT_DESTROY_TIME
- CAT_CAT_CODE
- CAT_STORAGE_FLOOR_AREA
- CAT_ICON_SYMBOL
- CAT_SHELTER_PRIORITY

23. Created the supply_category table and added the following variables to it:

- SC_NAME (increased the SC.NAME field length from 15 to 25 characters)
- SC_CONSUMPTION_TYPE
- SC_SHIPMENT_TYPE
- SC_SUPPLY_TYPE_EQUIVALENT
- SC_RIC, SC_NIC
- SC_PREFERRED_UOM
- SC_UOM_TYPE
- SC_SPECIAL_CAPABILITY

24. Created the combat_system table and added the following variables to it:

- CS_NAME (Increased the CS_NAME field length from 15 to 25 characters)
- CS_ELS_CARRY_FLAG
- CS_ELS_LIFT_FLAG
- CS_ELS_PUBLISH_FLAG

- CS_ELS_PROTECTED_FLAG
- CS_SPECIAL_CAPABILITY
- CS_SPECIAL_SUBCATEGORY
- CS_WEIGHT_EACH
- CS_RESUPPLY_CATEGORY
- CS_EFFECTIVE_RANGE
- CS_ATTRITION_TYPE
- CS_NO_FUEL_EFFICIENCY
- CS_SUPPLY_CATEGORY_TO_FIGHT
- CS_AMMUNITION_SUPPLY_USAGE
- CS_CREW_COUNT
- CS_KILL_PROBABILITY
- CS_WIA_PROBABILITY
- CS_COMBAT_ARMS_EQUIVALENT
- CS_HIGHRES_EXPLICIT_TW
- CS_HIGHRES_EXPLICIT_ROUND_TIME
- CS_HIGHRES_AMBUSH_ROUND_TIME
- CS_HIGHRES_DISENGAGE_RND_TIME
- CS_HIGHRES_AMBUSH_MODIFIER
- CS_NONCOMBAT_FUEL_USAGE
- CS_COMBAT_FUEL_USAGE
- CS_FUEL_PER_KM
- CS_STORAGE_FLOOR_AREA
- CS_HIGHRES_SPEED

- CS_AGILITY_TYPE
- CS_DIS_CODE
- CS_RIC
- CS_NIC
- CS_SCORE

25. Created the CAT_TT table and added CAT.NAME, TT.NAME, and CAT.TT.PERCENT.VISIBLE to it.

26. Created the CSP_CAT table and added the following variables to it:

- CSP_NAME
- CAT_NAME
- CSP_CAT_MEAN_TIME_BTW_REPAIR
- CSP_CAT_PROB_SYSTEM_RECOVERED
- CSP_CAT_PROB_NONCOMBAT_FAILURE
- CSP_CAT_MEAN_TIME_BTW_FAILURE
- CSP_CAT_KILLER_FWL_MODIFIER
- CSP_CAT_VICTIM_FWL_MODIFIER

27. Added the CSP field to the FACTION_COUNTRY table.

28. Created the SP_DF_CS table and added the following variables to it:

- SP_NAME
- DF_NAME
- CS_NAME
- DAILY_KILL_RATE
- DAILY_RECOVERY_RATE
- DAILY_OCCURRENCE_RATE

29. Created the cs_tgtwpn table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- CS.NAME
- TW.NAME

30. Created the EEV_DAMAGE_COMBAT_SYSTEM table the following variables to it:

- EE_DCS_EVENT_INDEX
- EE_DCS_EVENT_TIME
- EE_DCS_DAMAGED_UNIT
- EE_DCS_COMBAT_SYSTEM
- EE_DCS_SYSTEMS_KILLED
- EE_DCS_REASON

31. Created the AC_LOAD_SC table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- AL_NAME
- SC_NAME
- AL_SC_LOAD

32. Created the SUP_SC table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- SUP_NAME
- SC_NAME
- SUP_SC_BRING_TO_THEATER
- SUP_SC_REORDER_LEVEL
- SUP_SC_STOCK_OBJECTIVE
- SUP_SC_BASIC_LOAD
- SUP_SC_NORM_CON_PER_DAY

- SUP_SC_PER_AIR_OPS_USG
- SUP_SC_PER_GUNFIRE_USG
- SUP_SC_PER_KM_USAGE

33. Created the TUP_SC table, updated the table from old SC Names to the new SC Names, refined the new TUP_CS records for AMPHIB-type Combat Systems by querying the legacy TUP boat type field and added the following variables to it:

- TUP_NAME
- SC_NAME
- TUP_SC_BRING_TO_THEATER
- TUP_SC_REORDER_LEVEL
- TUP_SC_STOCK_OBJECTIVE
- TUP_SC_BASIC_LOAD
- TUP_SC_NORM_CONS_PER_DAY
- TUP_SC_PERIOD_USG_ATTACK
- TUP_SC_PERIOD_USG_DEFEND
- TUP_SC_PERIOD_USG_DELAY
- TUP_SC_PERIOD_USG_WDRAW
- TUP_SC_PER_KM_USAGE

34. Created the HUP_SC table, updated the table from old SC Names to the new SC Names, refined the new HUP_CS records for AMPHIB-type Combat Systems by querying the legacy hup organic boat type field, and added the following variables to it:

- HUP_NAME
- SC_NAME
- HUP_SC_STARTING_ON_HAND

35. Created the LOGIN_SUPPLIES table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- LOGIN_INDEX
- SC_NAME
- LOGIN_SC_AMOUNT

36. Created the TRANS_CAPACITY table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- TC_NAME
- SC_NAME
- TC_SC_CAPACITY

37. Created the TARGET_SUPPLY_LIST table, updated the table from old SC Names to the new SC Names, table and added the following variables to it:

- TSL_TARGET
- TSL_SC_NAME
- TSL_DUMP_AMOUNT

38. Created the SSA_SC_INIT_SUPPLY table, updated the table from old SC Names to the new SC Names, table and added the following variables to it:

- SSA_NAME
- SC_NAME
- SSA_SC_INIT_SUPPLY

39. Created the UNIT_SUPPLY_CATEGORY table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- UNIT_ID
- SC_NAME
- ROUTINE_SUPPORT

40. Created the EEV_DAMAGE_SUPPLIES table, updated the table from old SC Names to the new SC Names, and added the following variables to it:

- EE_DSC_EVENT_INDEX

- EE_DSC_EVENT_TIME
- EE_DSC_DAMAGED_UNIT
- EE_DSC_SUPPLY_CATEGORY
- EE_DSC_AMOUNT_KILLED
- EE_DSC_REASON

B.2 DELETED DATA TABLES/FIELDS

1. Deleted the ship_basic_type_lu lookup table.
2. Deleted the CALIBER, CS_TT, CSP_CS, DF_SP_CS, SC_SLP_SLP, SLP_SC, SP_SC, SPECIAL_CS, SPECIAL_SC, TW_CALIBER child tables.
3. Removed SUP.SPECIFIC.CLASS.TYPE from the ship_unit_proto lookup table.
4. Removed NV.BASIC.TYPE from the naval_unit lookup table.
5. Removed SLP.CARGO.TRUCK, SLP.TANKER.TRUCK, SLP.HET and SLP.OTHER.TRUCK from the sustainment_log_proto table.
6. Removed TUP.CALIBER, TUP.CSP.NAME and TUP.AB.NAME from the tact_unit_proto table.
7. Removed SUP.CALIBER and SUP.CSP.NAME from the ship_unit_proto table.
8. Removed CSP.POLITICAL.COUNTRY from the combat_system_proto table.

B.3 RENAMED DATA TABLES/FIELDS

1. Renamed the specific_class_type_lu lookup table to oth_gold_type_lu.
2. All JDPI-related database object names were changed from DMPI to JDPI.
3. The air_ops_shutdown_required field from the ACP_WC table was renamed acp_wc_takeoff_capable, to match the model.

B.4 MOVED DATA TABLES/FIELDS

1. Moved the SLP_SC table's reserved fraction field to the SUPPLY_TYPE table and added the new SUT.RESERVED.FRACTION values using the legacy SLP_SC records.
2. Created the ccp_sut_base_density, ccp_sut_up_density, ccp_sut_tt_density child tables from the legacy Supply Density data.

3. Created the following child tables from the old Combat System child tables, to refer to the new Combat Arms Type table:

- est_cat
- ccp_cat_base_density
- ccp_cat_up_density
- ccp_cat_tt_density
- sp_cat_tt
- sp_cat
- csp_cat_mft
- flp_csp_cat
- flp_cat_cat
- ephemeris
- fwl_coefficient