JTLS-GO

Version Description Document

August 2024

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.2.7.0)

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.2.7.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.2.7.0 is a Maintenance release of the JTLS-GO 6.2 series that includes a repository of standard data, a demonstration scenario based in the western Pacific, as well as a few 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 Director of Development
Valkyrie Enterprises LLC
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940
United States
jtlsgo@valkyrie.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.2.7.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.2.7.0 is a Maintenance delivery for the JTLS-GO 6.2 series of releases.

JTLS-GO 6.2.7.0 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 "wespac62". Detailed descriptions of some minor Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0, and Chapter 3.0 describes the software errors that have been fixed since the last release of the JTLS-GO 6.2 series.

JTLS-GO 6.2.7.0 executes on the Red Hat Enterprise Linux Server Version 8.7 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 Analyst Guide* (JTLS-GO Document 01, Version 6.2.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 02, Version 6.2.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.2.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.2.5.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.2.5.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.2.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.2.5.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.2.2.0)
- *JTLS-GO JOBE Quick Start Guide* (JTLS-GO Document 11, Version 6.2.2.0)

- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.2.3.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 6.2.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.2.3.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.2.5.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.2.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.2.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.2.5.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.2.0.0)
- *JTLS-GO DDS Training Manual* (JTLS-GO Document 23, Version 6.2.3.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 24, Version 6.2.0.0)

1.2.3 Updated Documents

- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.2.7.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.2.7.0)

1.2.4 New Documents

No new documents are delivered with this version of the software.

1.2.5 Delivered Software Components

JTLS-GO 6.2.7.0 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 188.

- 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

- Keyhole Markup Language (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
- JTLS Version Conversion Program (VCP)

  VCP60 - Converts a JTLS-GO 5.1 database to a JTLS-GO 6.0 formatted database.

  VCP61 - Converts a JTLS-GO 6.0 database to a JTLS-GO 6.1 formatted database.

  VCP62 - Converts a JTLS-GO 6.1 database to a JTLS-GO 6.2 formatted database.

Instructions for installing JTLS-GO 6.2.7.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.2.7.0 is not necessary. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

The basics of installation have not changed significantly, but due to many Cyber-Security improvements, a new Linux RPM package named “xerces-c”, is required to run JTLS-GO 6.2.2.0 and all later versions. The Synapse will not function without this package. Prior to installing JTLS-GO 6.2.7.0, please run the RPM checking script delivered with JTLS-GO to ensure that this package and all other packages are installed as part of your Linux operating system.

Due to repeated problems at exercises, we have changed the method the ICP uses to save WHIP passwords. This improvement resulted in a version change to a scenario’s ICP database files. Please refer to the complete explanation and instructions on how to alter existing scenarios to use the new ICP database structure and format in This chapter summarizes model capabilities added to JTLS-GO 6.2.7.0 as a result of implementing authorized minor Engineering Change Proposals (ECPs). on Page 2-1.

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 “repository62”. 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.

- The scenario “wespac62”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.2.7.0 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 Server (ES) Edition Version 8.7, 64-bit architecture.

  JTLS-GO 6.2 has been tested with the following versions of Linux 8:

  RedHat Linux 8.7 - this operating system license must be purchased.

  Oracle Linux 8.7 - 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.

- 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 6.2.7.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Server Edition Version 7.9, 8.4, and 8.7

  Oracle Linux 8.4 and 8.7

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

- JTLS-GO 6.2.7.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 422 package, which is equivalent to the current version of OpenJDK.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.8.

- JTLS-GO database tools require a certified PostgreSQL 11.19 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 6.2.7.0. There are several alternative methods available for obtaining the PostgreSQL 11.19 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 (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.

- 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 application uses these open source libraries:

  JFreeChart, licensed under a GNU Lesser General Public License (LGPL) by Object Refinery Limited, http://www.object-refinery.com

  JCommon, licensed under LGPL2.1 (GNU Lesser General Public License version 2.1 or later) by Object Refinery Limited, http://www.object-refinery.com

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

- KML Operational Interface (KOI)

  The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to any version of Google Earth™. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using an operational Command, Control, Communication, Computer Information (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 6.2.7.0, using the JODA service, allows connections and data exchange with customer client programs. The customer client programs are linked with a set of JTLS-GO-provided API libraries that permit a TCP/IP connection between the JODA and the client program. These API libraries, called JDSP libraries, are built for Linux and Windows and allow customers to built client applications on either of these operating systems. Below are the development environments under which each of the JDSP libraries are built:

  RedHat Linux 8.7 using gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-15.0.2).

  Windows 10 using Visual Studio 2017 version 15.9.60 and Visual C++ 00369.60000.00001-AA807.

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.

JTLS-GO has moved to Adoptium, a full OpenJDK Java environment with licensing alternations allowing an application to deliver the software. The following procedure has been established and approved by the JS/J7 Cybersecurity branch:

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

- Within two-weeks of the Adoptium release, JTLS-GO provides a bug release version (JTLS-GO 6.2.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 bug released versions when distributed.

Contact the U.S. Government Program Manager, Ms. Jessica Camacho by email at jessica.l.camacho.civ@mail.mil to obtain the completed Cybersecurity paperwork and a current Gate completion certificate.

As a result of new security requirements built into JTLS-GO 6.2.0.0 as part of ECP JTLS-2022-15976 "Encrypt Passwords To Start WHIP/DDSC", users must delete **all** pre-JTLS-GO 6.2 scenarios from their $JGAME directory. This will require you to set up your scenarios from a fresh state:

1. Convert your scenarios to JTLS-GO 6.2.0.0 using the Version Conversion Program (see Chapter 13 of the *JTLS-GO DDS User Guide* for instructions).

2. Perform Option 3, "Setup System For A Specific Scenario", for each scenario.

3. Perform Option 5, "Run Interface Configuration Program", for each scenario.

This is **not** a requirement if you have already installed JTLS-GO 6.2.0.0 and are upgrading to JTLS-GO 6.2.n.0 maintenance release.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.2.7.0 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 6.2.0.0 with CAE's GESI model has not been accomplished. CAE should be contacted concerning the continued support of GlobalSim.

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 6.2.7.0 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

Several database structure differences exist between JTLS-GO 6.2 series and the previous JTLS-GO 6.1 series database structure.

To upgrade your JTLS 6.1 scenario to JTLS-GO 6.2 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1, followed by the instructions in Chapter 13 of the *JTLS-GO DDS User Guide.*

> Users should download and re-load their scenarios into PostgreSQL, due to changes made in STR JTLS-2023-16290 PSQL Statement Fails For SVP Warning Correction, included in JTLS-GO 6.2.2.0, which was released in July 2023. This will recreate the related check constraints in the database schema. This procedure only needs to be execute once for every scenario. If the procedure was followed after the release of JTLS-GO 6.2.2.0, it does not need to be followed again,

1.4.1 JTLS-GO Symbol Set

Over the past several years, the Database Team has added and updated the Default Symbol Set used for the delivered scenarios. An organization is not required to use the JTLS-GO Default Symbol Set, but If there is a desire to do so, the following information is provided to help determine how to use the JTLS-GO Default Symbol Set improvements and changes that have been made.

With the release of JTLS-GO 6.2.5.0, STR JTLS-2024-16648 Bad Symbol ID Code From JSYMS resulted in several changes to the Default Symbol Set. There are two possible situations in which a user organization can currently stand with respect to the use of the JTLS-GO Default Symbol Set. The method that should be used to update a scenario's Symbol Set to the latest Default Symbol Set is included for each situation. Please see the *JTLS-GO 6.2.5.0 Version Description Document* for further information.

1.4.1.1 User Organization Using Pre-JTLS-GO 5.0.0.0 Legacy Default Symbol Set

To change a scenario's Symbol Set under this situation, prior to unloading your JTLS-GO 6.2.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.2.0.0 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.

1.4.1.2 User Organization Using Post-JTLS-GO 6.0 Symbol Set

The symbol 2525 ID Codes, which are used by C4I systems to identify the type of object, were expanded and refined as part of this JTLS-GO 6.2.5.0 release. If you have scenarios based on the JTLS-GO repository symbols, you are encouraged to update the symbol sets for your organization's scenarios. This can be accomplished by executing the following steps after the installation of JTLS-GO 6.2.5.0:

1. Conduct a download of the scenario.
2. Go to the $JDATA/scenario/<scenario_name>/symbols directory by entering the following command:

   cd $JDATA/scenario/<scenario_name>/symbols

3. Compare your symbol.scf file to the JTLS 6.2.5.0 version under the repository62 scenario by enter the following command:

   diff symbol.scf $JDATA/scenario/repository62/symbols/symbol.scf_jtls60

4. If the names of symbols in the scenario symbol file are identical to the repository symbol file or if your scenario symbols are a subset of the repository62 version, update your symbol file by entering the following command:

   cp $JDATA/scenario/repository62/symbols/symbol.scf ./symbol.scf

   Your symbol set will now be updated. For safety and to ensure Step 4 is done correctly, do a load of the scenario. Check the log files to ensure there was a clean load

5. If your symbol file has additional symbols not found in the current repository62 version, you can still update your symbol file, but will need to perform a manual integration of the symbol.scf file.

   Bring up the jsyms program for your scenario, by selecting Option 1 "Prepare or Alter a Scenario Database", and then Option 5 "Configure Symbols", from the JTLS Menu.

6. Perform a save in the jsyms program.

- If the scenario is loaded on the DDS, select "Yes" to fully update the symbols in the scenario.
- If your scenario is not loaded into the DDS, select "No" to simply update the ASCII version of the scenario's usable graphics symbol located in the <scenario_Name>.gs file. The next time you perform a load, the most current updated symbols will be loaded from the <scenario_name>.gs file.

7. Exit the jsyms program.

1.4.2 Standard Repository Changes

The JTLS-GO Database Team has continued to improve and expand the unclassified JTLS-GO data repository, named to “repository62”. 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.

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 11.19 required to operate JTLS-GO 6.2.7.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.2.7.0 as a result of implementing authorized minor Engineering Change Proposals (ECPs).

2.1 JTLS-2024-16752 Satellite Report Added to Query-Only WHIP Menus

Summary of Model Change Request

The Manage Satellite order contained an option to generate a report, containing the dates, times, and locations for when a single satellite enters and exits a specific area on the Map. This order was only accessible from a Controller WHIP.

At an exercise, the Space response cell requested these details for satellites in the game. The Space cell was given access to a query-only Controller WHIP during the exercise, which did not allow the cell to use the Manage Satellite order.

Design Summary

Changes were made to the Manage Satellite order to create a quick order option, which only generates the satellite report, and does not permit any parametric changes to existing satellites.

Additional changes were made to the query-only menus for the WHIP.

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.

3.1 JTLS-2024-16750 Custom Naval Menu Missing Essential Order

A menu for organizing orders in the WHIP did not contain an essential order to manage Air Control Mean objects (ACM).

The customized menu file was modified to include the Manage ACM order and the associated utility order details.

3.2 JTLS-2024-16753 Unable To Airlift Same Unit More Than Once

A ground unit was airlifted to a new location using the Mobility Mission order. Later, the Player attempted to airlift that same unit to another location. The mission reached the location of the unit, then rejected the Load Unit task. An Airlift/Airdrop Problem Report was generated, which said an airlift of this unit was already in progress to a different destination. However, no such mission existed in the game.

When the unit was picked up by first mobility mission, a flag on the unit was set to "Yes". This flag indicated that the unit was part of an ongoing lift. When the entire unit was delivered to the destination, this flag was never set to "No". As a result, each subsequent airlift mission rejected the task to load the unit because it conflicted with an ongoing lift.

The ongoing lift flag was set to "No" in the code which finalized the delivery of the unit to the new location.

3.3 JTLS-2024-16755 ATO Parser Overwrites Mobility Stops

The Air Tasking Order (ATO) Parser reads the “ROUTE” record and correlates the data with any following “AIRMOVE” records. When a named stop, such as an ICAO code, is repeated in the route, but both are independent stops, the Parser assumes they are the same, and overwrites the first stop with the time from the later stop of the same name.

Mobility air missions sometimes visit the same ICAO stop more than once. The comparison between ROUTE and AIRMOVE records was insufficient for determining that items, like ICAOs, would be different and independent.

The Parser logic was changed to make the comparison more accurate.

3.4 JTLS-2024-16756 Default ELS Templates With Non-Random Layouts

The initial default templates for the Entity Level Server (ELS) were built using the Scenario Initialization Program (SIP) during database development. As the model was running, new default templates were built as needed by the ELS.

In both cases, the templates were designed to exhibit a random layout of the combat systems within the unit radius. The generated templates actually showed a spiral layout, which was not random.

The computations to lay out combat systems in the templates were utilizing random numbers to distribute the entities. Unfortunately, the random number streams in the program were never initialized with values for the seeds. As a result, the computed locations were never randomized.

The code to generate the templates was reorganized and a single central routine was created to utilize truly random numbers to position the entities.

3.5 JTLS-2024-16757 Crash Replacing Satellite Route Points

The model crashed when the Controller attempted to replace the route for a satellite. The specific order used the manual option, and they chose to replace the route points with a new set of locations.

When the model processed the Manage Satellite order to replace the orbit locations, it first removed the existing route, so the time of the next route point was undefined. No time was required on the order when adding or replacing route points. As a result, the replacement route points were not added to the route due to having unknown times to leave each location.

The code was modified to use the current time for the first new route point. This resolved the crash and produced the expected satellite motion.

3.6 JTLS-2024-16758 Random Number Seed Mismatch

JTLS-GO has 100 different Random Number Seeds to be used for the various random processes that exist in JTLS-GO. Each of these 100 Random Number Seeds have names assigned, and the DDS, the SIP, the CEP, and the WHIP must refer to them in the same manner. This was not true.

- The DDS did not have the Load Time Random Number Seed, and had a Random Number Seed named RNS SPARE FOUR, which should not have existed.
- The SIP did not recognize the Load Time Random Number Seed
- The Static Vocabulary file was missing 15 of the available seeds, and of those that were there, several of the names of the seeds were incorrect.

All of these errors were corrected.

3.7 JTLS-2024-16759 ATO Spreadsheet Errors

The sample ATO spreadsheet crashes the ATO Parser.

The sample spreadsheet uses the abbreviation “INT” as a type for Offensive Air Support (OAS) missions. In a recent exercise, we were informed that “INT” is used for various intelligence missions. At that time the ato_constants.xml file was updated to reflect the change.

The spreadsheet was still using “INT” for OAS mission representation, so the fields were not what the ATO Parser was expecting. The spreadsheet was updated to use “OAS” instead of “INT”.

The ato_constants.xml file was also reviewed, and other “INT” type missions were corrected to create RECCE missions instead of OAS missions.

3.8 JTLS-2024-16760 Embarked Formation Magic Move Spreadsheet

It is necessary to create the spreadsheet definition in the order definition file when defining an Information Management Tool (IMT) spreadsheet. Once the spreadsheet is defined, the IMT definition file needs to be changed to indicate which IMT columns link to the spreadsheet fields.

A change was made to the Magic Move order. The Combat Events Program (CEP) automatically cancels all of a unit's tasks when the unit is magic moved onto a formation. The Magic Move order allows the user to decide whether to cancel existing tasks. In fact, the user had no choice, because the model automatically canceled all unit tasks as a result of the magic move.

For this reason, the Magic Move order was changed to remove the “Cancel Task” option field when the user Magic Moved a unit onto a formation. This change caused an error in the IMT tables that could be used to create the Magic Move To Formation spreadsheets. These IMT definition files incorrectly indicated the "Cancel Task" field was needed for these orders.

Each of the IMT table definition files that allowed the user to select the Embark Formation Magic Move order was corrected. The link from the IMT to the order “Cancel Task” field was removed.

3.9 JTLS-2024-16766 Write Neutral Networks Only Once in OPM

Neutral networks, such as Air Routes and Sea Lanes, should only be written once for all Players.

Neutral networks are the same for all Players. This data is normally available and known to all Force Sides in the scenario. The same applies to terrain objects, such as roads, rivers, railroads and pipelines.

The SIP now only writes these networks once when generating the Online Player Manual (OPM).

3.10 JTLS-2024-16772 National Asset Pass Help Inaccurate

The field help description for "Communications Intelligence" on the National Asset Pass order was inaccurate.

The field help description was corrected.

3.11 JTLS-2024-16774 Changing Satellite Payload Unneeded Code

There was a section of code that was executed for no reason when the Controller entered a Change Satellite Payload order.

The code was removed.

3.12 JTLS-2024-16779 WHIP Country Code Preference Changes

Changing the Country Code WHIP preference did not convert the country code displayed in the Message Browser messages.

The WHIP was not reading the default country code on start, which the WHIP to incorrectly show a default setting of "null" in the user's country code preference.

The WHIP was changed to read in the default country code on start.

3.13 JTLS-2024-16786 Manual Pair Rejection Stops Intercepting Mission

While heading to its orbit location, an aircraft mission carrying air-to-air weapons was ordered to intercept an enemy aircraft. The Manual Pair order was rejected, with a message stating the mission "does not carry any weapons that can kill such a track".

The interceptor mission then stopped moving at its current location. The current Move had been postponed resulting in the task's sequence number changing from 0 to 5.

When an interceptor aircraft carries air-to-air weapons, and none of the weapons have positive PH/PK against a specific aircraft target category, a Manual Pair order is properly rejected. However, the logic unnecessarily postponed the mission's currently executing task.

New code was added to restart the postponed task when the mission was not assigned to the intercept.

3.14 JTLS-2024-16794 Generated Templates Not Saved By ELS

Templates were used by the ELS to disaggregate units. If a specific template file did not exist, the ELS generated one at the start of execution. The ELS crashed when restarting from a checkpoint because a generated template was not found.

When the ELS created templates during runtime, those templates were not saved to disk. Changes were made to ensure that generated templates are saved for access during restarts.

3.15 JTLS-2024-16804 HRU Patrol Via Route Rejected

A Player submitted an HRU Task order to Patrol along a specified route through an ocean terrain grid. The HRU Task order did not pass the checker and could not be submitted to the game. The HRU Task Patrol order was accepted only if a Polygon or OPAREA was specified.

The HRU Task order template incorrectly referenced the Ground Route utility to hold the specified route points. The Order Management Authority recognized that the points were in water terrain, and rejected the order.

The HRU Task template was corrected to use the HRU Route utility, which permits movement through both water and land terrain.

Note that the HRU must still possess a small boat to move through water.

3.16 JTLS-2024-16805 Windows Coordinate Converter Not Copying

When running the Coordinate Converter in the WHIP or DDSC on Windows, the "Copy to Clipboard" option failed to copy the coordinate selected.

This was due to differences in how Windows and Linux handle mouse events (such as button clicks or releases) for the popup option.

The issue was corrected by universally handling the right-click option on a selected coordinate. Windows captures the event on a mouse press, while Linux systems captures on a mouse release. Both of these events were implemented.

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.

The following list of issues is known and have not been fixed in time to make it into this release of JTLS-GO 6.2.7.0.

4.1 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.

4.2 MHE Targets Loading Air Mission Can Cause a Crash

MHE targets should be avoided for loading and unloading air missions. It is suggested that the database be set to “Do Not Use” for Air Missions.

4.3 The JTLS-GO Strategic Lift Missions Are Not Working Properly

Strategic Lift Missions, used to move TPFDD assets into the Theater and report the results to a real-world TPFDD processing system, has not been updated to work within JTLS-GO 6.2.

4.4 Tactical Ground Formation Attacks Do Not Work

The ability to send a Tactical Ground Formation on an Attack mission has been temporarily disabled due to reliability issues.

4.5 ATOT Spreadsheet Lacks Detailed Field Checking

The ATOT Spreadsheet Parser has been found to have numerous issues within the Spreadsheet format that are not caught and cause the spreadsheet parser to crash. Fixing the uncovered issues are being worked and should be fixed prior to the next maintenance release of the JTLS-GO 6.2 series.

4.6 Moving Combat System Supplies Can Reduce Unit Strength To Zero

If a user does a mandatory transfer of Combat System supplies from one unit to another, the providing unit can be emptied out and exist without any Combat Systems or personnel. This situation needs to be thoroughly and properly handled.

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
PKI	Public Key Infrastructure
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 6.2.0.0 DATABASE CHANGES

Refer to the JTLS-GO 6.2.0.0 Version Description Document (VDD) for the list of database changes between the JTLS-GO 6.1 series and the JTLS-GO 6.2 series.

APPENDIX C. VERSION 6.2.0.0 REPOSITORY CHANGES

The JTLS-GO Database Team is continually adding and vetting unclassified data to expand and maintain the JTLS-GO Data Repository. Over the last year as part of the JTLS-GO 6.2 development effort, this entire process has been conducted in both the JTLS-GO 6.1 and JTLS-GO 6.2 versions of the repository. No specific, unique, additions were made to the JTLS-GO 6.2 repository. The repository delivered with this initial version of JTLS-GO 6.2 contains the same data as the JTLS-GO 6.1 version of the repository, except the format has been altered to meet the requirements of JTLS-GO 6.2.