JTLS-GO

Version Description Document

June 2025

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.4.0.0)

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.4.0.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.4.0.0 is a Major release of the JTLS-GO 6.4 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as major model functionality improvements implemented as Engineering Change Proposals (ECPs), summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) will be described in Chapter 3 in future releases - because this is the first release of the JTLS-60 6.4 series, there have been no STRs to correct. Known, 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 6.4.0.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.4.0.0 is a Major delivery for the JTLS-GO 6.4 series of releases.

JTLS-GO 6.4.0.0 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 “wespac64”. 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 the Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS-GO 6.4.0.0 executes on the Red Hat Enterprise Linux Version 9.4 and Oracle Linux 9.4 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any 64-bit 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 6.4.0.0 does not include a major redesign of the JTLS-GO system, but does include several extensive engineering change proposals. All of the documentation has been updated to reflect these functional capability changes.

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.4.0.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 6.4.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.4.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.4.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.4.0.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.4.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.4.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.4.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.4.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.4.0.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.4.0.0)
- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 6.4.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.4.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.4.0.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.4.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.4.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.4.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.4.0.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.4.0.0)

1.2.4 New Documents

No new documents are required for this version of the software.

1.2.5 Delivered Software Components

JTLS-GO 6.4.0.0 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)

    The current JODA build number is 229.

- 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.

    VCP63 - Converts a JTLS-GO 6.2 database to a JTLS-GO 6.3 formatted database.

    VCP64 - Converts a JTLS-GO 6.3 database to a JTLS-GO 6.4 formatted database.

Instructions for installing JTLS-GO 6.4.0.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.4.0.0 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 “repository64”. 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 “wespac64”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.4.0.0 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 Server (ES) Edition Version 9.4, 64-bit architecture.

  JTLS-GO 6.4 has been tested with the following versions of Linux 9:

  RedHat Linux 9.6 - this operating system license must be purchased.

  Oracle Linux 9.6 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by Defense Information System Agency (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 be a 64-bit architecture with a Java-enabled web browser. JTLS-GO 6.4.0.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 9.6

  Oracle Linux 9.6

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

- JTLS-GO 6.4.0.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 452 package. Both the ICP and DCP have the option for an organization to increase the maximum memory heap for the WHIP and DDSC. For large scenarios and databases, an organization should consider increasing the maximum heap size.

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.4.

- JTLS-GO database tools require a certified PostgreSQL 15.13 database server and the full PostgreSQL installation. PostgreSQL 15.13 that has been compiled under Linux 9.6 is bundled with the JTLS-GO 6.4 release tar files. It is not necessary to use the delivered solution, but it is the easiest method to meet the requirements of JTLS-GO 6.4.0.0. There are several alternative methods available for obtaining the PostgreSQL 15.13 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.

- 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.

1.3.2 JTLS-GO Cybersecurity Compliance

Because of recent incidents of intrusions into software systems, the United States Department of Defense (DoD) has implemented a strong and strictly enforced Cybersecurity program. JTLS-GO,

as software that executes on DoD systems, must comply to the mandates of the program, along with all of the third party software used by JTLS-GO, such as PostgreSQL and Java.

One of the DoD requirements is that the software must implement a methodology that ensures that the end user keep the software up-to-date and all security patches are properly installed. In previous versions of JTLS-GO, Java 8, as delivered by Oracle, fulfilled this mandate by implementing an expiration date for its software. The concept of an expiration date has been removed from the DoD requirement, but the concept of always using the latest version of third-party software remains a strong component of DoD Cybersecurity requirements.

The following procedure has been established and approved by the JS/J7 Cybersecurity branch to meet the software update requirement:

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

- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a bug release version (JTLS-GO 6.4.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.

Please contact the U.S. Government Program Manager, Mr. Douglas Failor (douglas.l.failor.civ@mail.mil) to obtain the completed Cybersecurity paperwork. It is expected that a current Gate completion certificate will be available with four to five weeks of this initial release. Due to time and funding considerations, the JTLS-GO 6.4.0.0 project has not obtained a Checkpoint Gate certificate.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.4.0.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO currently belongs to one federation known as GlobalSim. GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming that helps commanders and all levels of staff prepare for a range of operational scenarios.

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

Combining JTLS-GO and GESI brings together operational and tactical level constructive simulations to prepare commanders and staff to make timely, informed and intelligent decisions

across the full spectrum of operations, including conventional combat, disaster relief, and operations other than war.

From the JTLS-GO perspective, all software needed to run GlobalSim is included in this delivery. JTLS-GO uses the Federation Object Model (FOM) located in the $JGAME/data/hla directory. Federation testing of JTLS-GO with CAE's GESI model has been accomplished. The reader should note that the JTLS-GO Development Team, to date, has not been able to test this federation. If there is interest in running this federation, please contact the JTLS-GO Help desk at jtlsgo@valkyrie.com.

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 6.4.0.0 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 6.4.0.0 and the previous JTLS-GO 6.3 series database structure. APPENDIX B. Version 6.4.0.0 DATABASE CHANGES has a summary of all database changes.

To upgrade your JTLS 6.3 scenario to JTLS-GO 6.4 compatibility, see instructions listed in the *JTLS-GO DDS User Guide*, Chapter 3.1.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 6.4.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.4.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario_name>.gs and databases symbol.scf file.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.4.0.0, 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. Some new symbols have been created to meet end-user requirements. No previously existing symbols were deleted nor were any of the preexisting symbol names changed.

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

1.4.3 Standard Repository Changes

R&A has continued to improve and expand the unclassified data repository, which has been renamed to “repository64”. 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. Specifically, significant effort has been applied to ensuring that all important Targetable Weapons have a unique Supply Category from the weapon should be drawn. This results in the model managing a detailed weapon count of all used weapons.

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 15.13 required to operate JTLS-GO 6.4.0.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.4.0.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

Several ECPs originally planned for release in JTLS-GO 6.4.0.0 were released under the JTLS-GO 6.3 series of releases, for operational reasons. Please see the Version Description Documents (VDDs) for the JTLS-GO 6.3 series of releases for information concerning the following ECPs:

- JTLS-2024-16930 - Change Data For All COMAO Missions
- JTLS-2024-16879 - DDSC Privileges
- JTLS-2023-16449 - Run JTLS-GO From User Account
- JTLS-2022-15976 - Encrypt All Passwords Used To Start A WHIP Or DDSC
- JTLS-0293 - Redesigns Automatic Supply Calculation Tool

The remainder of this section describes the ECPs delivered for the first time with JTLS-GO 6.4.0.0.

2.1 JTLS-2013-11469 Improve Damage Combat Systems Order

Summary of Model Change Request

In the JTLS-GO 6.3 series, an ECP was implemented to provide more control over magically-created damage using the Damage Combat System External Event and the Controller Change Unit capability. The Damage Combat System External Event can exist in the JTLS-GO initialization database, but because the database structure cannot be changed after the initial release of a version series, the improvements made did not apply to the database version of the Damage Combat System External Event.

For this reason, this ECP remained open until the delivery of this initial JTLS-GO 6.4 series.

Design Summary

The database structure was altered to match the deterministic options for the Damage Combat System order delivered in the JTLS-GO 6.3 series. Two new fields were added to the database table used to define Damage Combat System External Events:

- Combat System Damage options - Table 2.1 summarizes the available options. During database conversion, any existing Combat System Damage External Events are given a Combat System Damage option of “Use Probabilities”.

Table 2.1 Combat System Damage Options

OPTION	EXPLANATION
Use Probabilities	This is the algorithm that is in effect in all previous versions of JTLS-GO. When the Combat System is damaged, the model accesses the following data to determine whether the system should go into maintenance or whether it is considered a catastrophic kill: • CS COMBAT ARMS EQUIVALENT - This database parameter, which holds a Combat Arms Type (CAT), is accessed for the one Combat System that represents Crew. The data parameter is needed to determine how long it will take to repair each of the wounded crewman. • CSP CAT PROB SYSTEM RECOVERED - This database parameter is accessed by accessing the damaged unit's Combat System Prototype (CSP) and the CAT used by Combat Systems. For each damaged system, the model draws a uniformly distributed random variate and compares it to this probability. If the determination is that the system can be recovered, then the Combat System is placed in maintenance. If the Combat System cannot be recovered, a Catastrophic Kill object is placed on the ground and is visible to collection sensors.
Always Catastrophic	None of the described above data is accessed. All damaged Combat Systems are assumed to be killed and will become Catastrophic Kill objects. The unit's Killed In Action (KIA) column will be updated with these systems.
Always Repairable	None of the described above data is accessed. All damaged Combat Systems are assumed to be damaged, but repairable. They will show up in the unit's maintenance column and will come out of maintenance based on the database specified repair times.

- Crew Damage Options - Table 2.2 summarizes the available options. During database conversion, any existing Combat system Damage External Events are given a Crew Damage option of "Use Probabilities":

Table 2.2 Crew Options

OPTION	EXPLANATION
Use Probabilities	This is the algorithm that is in effect in all previous versions of JTLS-GO. When the Combat System is damaged, the model accesses the following data to determine what to do with the crew: • CS CREW COUNT - A Combat System that is considered "Manned" always has its full complement of its crew count.

Table 2.2 Crew Options

OPTION	EXPLANATION
Use Probabilities (Con't)	• CS PROB KILL - This is the Probability that each represented crew is killed when the Combat System is killed. If the CS CREW COUNT is 5, then 5 uniformly distributed random variates are drawn and compared to this probability of kill. If the crewman is considered killed, it is moved to the Killed In Action (KIA) column of the unit's Combat System data. • CS PROB WIA - After the kills are determined, the model looks at all remaining crew to determine if they are wounded. Assume in the example, that one crewman was killed, leaving four crewman that still need to be handled. In this situation, the model will draw four uniformly distributed random variates between 0.0 and 1.0 and compare it to the Probability that the crewman is wounded. Any crewman that is not wounded, will automatically be placed back in the available column and can be used by other Combat Systems as needed. • CS COMBAT ARMS EQUIVALENT - This database parameter, which holds a Combat Arms Type (CAT), is accessed for the one Combat System that represents Crew. The data parameter is needed to determine how long it will take to repair each of the wounded crewman. • CAT REPAIR TIME - This database parameter is access for the CAT used for the Crew Combat Systems. For each wounded crewman, an exponentially distributed random variable with this mean is drawn. This determines the amount of time needed to repair the wounded crewman. • FC WIA MAX TREAT UNIT TIME - This database parameter is used to determine which of the wounded crewman are so badly wounded that they need to be evacuated. Each of the repair times computed above are compared against this database parameter for the damaged unit's Faction. If the repair time is greater than this parameter, then the crewman is moved to the Wounded In Action (WIA) column and must be evacuated. If the random repair time is less than or equal to this database parameter, the crewman enters unit maintenance and eventually will be returned to action.
Always Kill	None of the data described above are accessed. All crewman are assumed to be killed and will show up in the unit's KIA column for Combat System Crew.
Wounded All Evacuated	None of the data described above are accessed. All crewman are assumed to be wounded so severely that they need to be evacuated. They will show up in the unit's WIA column for Combat System Crew.
Wounded None Evacuated	None of the data described above are accessed as part of the damage algorithm. All crewman are assumed to be wounded and will go into maintenance at the unit. They will show up in the maintenance column and eventually be returned to action after a random repair time based on the database parameter CAT REPAIR TIME.
Not Affected	None of the data described above are accessed. All crewman will be placed back in the unit's Available column where they could immediately be assigned to an available Combat Systems waiting for crew; otherwise they will stay in the Available column of the Unit's Combat System array.

2.2 JTLS-2021-15207 Implement JTLS-GO Map Server

Summary of Model Change Request

The original name of this ECP was “Zoom In With GeoTIFF”. This design calls for delivering a map server with JTLS-GO and implementing the capability of the WHIP and DDSC to access map tiles from the map server, allowing the user to zoom into any desired detail map level. For this reason, the ECP has been renamed “Implement JTLS-GO Map Server”. This will fulfill the need to move away from the older CADRG maps that are currently displayed on the WHIP and DDSC.

Design Summary

This improvement adds the ability to connect the WHIP and DDSC to a map server to display background maps. The Web Map Server (WMS), Open Street Map (OSM), and Tile Map Server (TMS) protocols are supported. WHIP and DDSC users are able to configure their own map server connections through their respective client application. Technical Control is able to pre-configure map server connections for use by WHIP users through the ICP. WHIP users are able to save their configured connects and visible map layers. Instructions on using the new map server capability are documented in Chapter 8 of the *JTLS-GO Tech Coordinator Guide*.

2.3 JJTLS-2021-15303 Represent UAV Swarms

Summary of Model Change Request

This original name of this ECP was “Valkyrie Drone Can Launch Smaller Drones”. The reason to launch the smaller drones is to create Drone swarms. The ability of an Air Mission to launch other air mission needs to be implemented to properly represent Drone Swarms, but there is no reason to implement one Air Mission launching other Air Missions unless swarms are implemented and properly represented in JTLS-GO. For this reason, the ECP has been renamed “Represent UAV Swarms”.

Design Summary

There are two major improvements associated with this ECP:

2.3.1 Ability For Mobility Missions To Carry Other Air Missions

The implementation of mobility missions to carry other Air Missions was accomplished, to operate in a manner similar to a Mobility Mission carrying High Resolution Units (HRUs). When creating a Mobility Mission, the WHIP user indicates that the Mobility Mission should carry other Air Missions. There are four options for specifying a Mission Insert Extract List of tasks:

- Mission Pickup indicates the mission will be picked up from the ground and placed on-board the Mobility Mission

- Mission Drop-off indicates the mission will be offloaded from the Mobility Mission and placed on the ground associated with the same airbase or runway that the Mobility Mission is currently using.

- Mission Extract indicates the mission will be caught and placed on board the Mobility Mission while both missions are flying.

- Mission Insert indicates the mission will be offloaded from the Mobility Mission and placed in a flying posture.

Not all aircraft are allowed to be placed on a Mobility Mission. A new database parameter was added to the database that indicates the weight of each Aircraft in an Aircraft Class. If the weight is set to zero, this indicates that the Aircraft Class cannot be placed on Mobility missions. The Mobility Mission can only carry the number of aircraft allowed based on the Mobility Mission's number of aircraft and the AC DRY CARRY CAPACITY of the Mobility Mission's aircraft. There are a few additional rules:

- The weight of the fuel and weapons being carried by the mission being carried is not considered by the algorithm. The model currently only considers the AC WEIGHT and ensures that the AC DRY CARRY CAPACITY is not exceeded.

- When placing a mission on a Mobility Mission, the entire mission being onboarded must fit. The model does not split a mission during the onboarding process.

Before an Air Mission can be onboarded, it must indicate that it is ready to do so. There are two new Air Mission Tasks developed to support this concept:

- Await Pickup - A mission that is waiting to be picked up by a Mobility Mission must be executing an "Await Pickup" task.

- Await Extract - A mission that is waiting to be extracted while airborne must be executing an "Await Extraction" task.

These two tasks can be assigned to the following types of Air Missions as part of their multiple tasks assignment utilities.

- Offensive Air Support (OAS)

- Defensive Air Support (DCA)

- Reconnaissance (RECCE)

- Suppression of Enemy Air Defense (SEAD)

- Electronic Combat (EC)

- Airborne Early Warning (AWACS)

The awaiting tasks can also be assigned as part of the Manage Mission Task Order.

2.3.2 Swarms

A new type of timing option can be selected for Air Mission Packages. This new option is called “Swarms”. Although the Air Mission Package structure is being used by both Attack Packages and Swarms, the two types of Air Mission Packages are quite different. No changes were made to the existing Attack Package capabilities.

The following Swarm logic has been implemented:

- Only the mission types listed as eligible for being lifted by a Mobility Mission can belong to a Swarm package.

- Missions can only join Swarms when they are airborne. When they join a Swarm, they assume a posture of “Swarming” and can be filtered on the map display. The IMT also lists the Swarm to which a mission belongs.

- Each Swarm has a single “Commander” mission. This “Commander” mission must follow its AC Control Mechanism Rules. If for example, the Aircraft Class specifies the UAV cannot go more than 50 KM from its home squadron, then the entire Swarm must stay within the 50 KM Control Mechanism range.

- All other missions in a Swarm are consider a “Follower” and the model does not access the mission’s aircraft AC Control Mechanism. The “Commander” mission is assumed to be controlling all of the “Follower” missions. Since all missions in a Swarm package are assumed to be at the same location, communication is considered to exist.

- No jamming of communication between Air Missions in a Swarm is currently considered by the model.

- If the “Commander” mission is completely killed, all missions in the Swarm package are killed.

- A Recce missions that belong to the package will collect intelligence information according to their sensors that are turned on.

Once missions join a Swarm package, the Swarm package moves all missions that belong to the Swarm. Unlike the Attack Package logic, in which missions that belong to the package move individually and according to Attack Package rules, a Swarm package take full control over the movement of missions that belong to the package. The following rules apply to the Swarm package movement:

- OAS missions take priority in directing the Swarm package. The Swarm package looks at the Fire Weapon tasks assigned to each OAS mission and selects the next Fire Weapon task that is closest to the current Swarm location. Once the Swarm is in position, the OAS mission is allowed to fire and damage is assessed. The Swarm then picks the next target.

- Recce missions take second priority - Once all OAS Fire Weapon tasks are executed or there are no OAS missions in the Swarm, the Swarm will select a Recce mission can follow its directed search path tasks. There is no priority if more than one Recce mission exists in a Swarm.

As a Swarm moves, the following logic is followed:

- Mission Fuel is consumed, but no “Is Fuel Needed” checks are done. If a Swarm mission runs out of fuel, it is removed from the Swarm and destroyed.

- Mission Detections are made. If one mission in the Swarm is detected, all missions in the Swarm are detected. Swarms are not a displayable Map Object, but the missions with a posture of “Swarming” are detected and can be used to display the current location of Swarm.

- Enemy Air Missions can intercept the detected Air Missions that belong to a Swarm. Currently, an artificial limitation has been placed to only allow one mission from each side to intercept a Swarm. Because Swarms are assumed to contain tens or even hundreds of Unmanned Aerial Vehicles, it is possible for a single Swarm to draw all interceptors when the ACP INTERCEPTOR SEND PERCENT was applied to Swarms if this limitation was not implemented.

- When an air-to-air weapon is fired on a mission that belongs to a Swarm, the weapon may hit any mission in the Swarm. Each Swarm holds a current count of the total number of aircraft that belong to the Swarm. A random uniformly distributed integer between 1 and the number of aircraft N is drawn, and this random integer determines which mission gets hit and possibly killed.

- DCA missions that belong to the Swarm will not leave the Swarm, but will fire back if within range.

- Air Defense is checked and it too determines which missions in the Swarm it will fire on. The damage algorithm works similarly to the Air-to-Air algorithm. Although the SAM Site fires on an individual mission, the damage is assessed against the Swarm as a whole.

The user can order an Air Mission to leave a Swarm. Once the Swarm’s set of Air Missions is empty, the Swarm is removed from the game.

2.4 JTLS-2024-16802 Satellite Destruction Representation

Summary of Model Change Request

Satellites may be damaged in the model through weapon fire. However, there is no way to tell when this happens, because the satellite status is not published on the JODA. The ECP proposes to implement a satellite status attribute to clarify the state when damaged has occurred.

Design Summary

When a satellite is killed, the model turns off all onboard emitters and the satellite will be transformed into a debris field. The satellite will continue to orbit, but will hold a range ring to indicate the size of the debris field. The status attribute holds the values of Alive, Maneuvering, Holding, and Destroyed. These status attributes have the following meaning:

- Alive - When a satellite is operating normally, it has a state of Alive.
- Maneuvering - If the satellite is executing a burn to follow a new orbital path, it goes through a black-out phase and has a state of Maneuvering.
- Holding - If the satellite has run out of route points, the status is changed to Holding.
- Destroyed - When the satellite is killed through weapon fire, the state is changed to Destroyed.

The satellite state is visible as an attribute in the SITREP panel and on the IMT.

2.5 JTLS-2024-16869 SAM Fire Control Sensors Air Search Capability

Summary of Model Change Request

Some Surface-to-Air Missile (SAM) Sites use the assigned Fire Control Sensor for its Air Search Capability. This needs to be represented.

Design Summary

A new database parameter has been added to the Sensor Type table. This new parameter labeled “Fire Control Range” indicates if the Sensor Type can be used as a Fire Control radar for a SAM/AAA site. If a SAM/AAA Site’s Fire Control radar also has an Air Search Range specified in the database, then the SAM/AAA Target will also accomplish target acquisition.

During database conversion, all Fire Control Sensors are given a Fire Control Range equal to the maximum range of the Air Defense Classes that uses the Fire Control Sensor. An organization’s database team should determine if they can remove Acquisition Sensors owned by units that own SAM/AAA sites with an acquisition capability.

2.6 JTLS-2025-17240 WHIP/DDSC Decimal Location Preference Option

Summary of Model Change Request

The WHIP and DDSC provide preference options to view locations in Latitude/Longitude and Military Grid Reference System (MGRS) formats, but not in Decimal Lat/Lon. Decimal Lat/Lon is a preferred location option when working with Air Tasking Orders.

Design Summary

The ability to set the location format in Decimal Lat/Lon was added to both the WHIP and DDSC Preferences for Location.

2.7 JTLS-2025-17243 Represent Container Units

Summary of Model Change Request

There is a need to represent units in the command hierarchy that have absolutely no capabilities. Within the Joint Live Virtual Construction (JLVC) federation these unit are known as “Grouping Folders”. Within the NATO federation of models, these are known as Container Units.

Design Summary

A new type of Aggregate Unit was created in the JTLS-GO 6.4 database structure. These units are known as “Container” units. A new database table was created to hold these units. Table 2.3 outlines the data associated with “Container” units. This table is color-coded with the following meaning:

- Green cells exist in the Container unit definition is a manner similar to all other types of JTLS-GO Aggregate Level units.
- Yellow cells indicate data that is unique to Container units.
- Red cells indicate data that is defined for all other Aggregate Units, but **not** for Container Units.

Table 2.3 Container Units Database Information

ATTRIBUTE	EXPLANATION
Short Name	
Long Name	
Unit Identification Code (UIC)	
Faction	

Table 2.3 Container Units Database Information

ATTRIBUTE	EXPLANATION
Prototype	Container units can only report down their Command Hierarchy chain. They cannot fight nor consume, so no prototype is required.
Effectiveness	Container units can only report. They have no access to any timed tasks such as repairs or requisition of supplies. There is no need to represent the Effectiveness of units.
Night Effectiveness	The same is true for Night Effectiveness.
Attack, Protect, Screen, Cover	Container units cannot participate in Lanchestrian attrition. There is no need to hold data for these parameters.
Orientation	Container units cannot detect or participate in battles. There is no use for this data.
Supply Adjust Interval and First Adjust Supply Time	Container units do not own supplies or consume supplies. There is nothing to requisition; therefore, the units to not go through the supply adjustment procedure.
TPFDD Time	Although Container Units can be created during game play, all database Container Units are assume to arrive at time zero.
Location (Latitude / Longitude)	
Graphics Symbol	Note all other unit types obtain their Graphics Symbol from their Prototype. Because Container units do not have a Prototype, the Graphics Symbol is specified as part of the Unit information.
Command Level	Note all other unit types, obtain their Command Level from their Prototype. Because Container units do not have a Prototype, the Command Level is specified as part of the Unit information.
Arrival Serials	All Container Units simply arrive in the game. They own nothing; therefore, there is no reason for them to arrive through a Port and no need to specify the number of Arrival Serials that should be used.
Service	
Federation Models	Container Units are not supported to belong to other JTLS-GO linked models.
Political Country	
JU Number	
Link-16 Network	
Link-16 Network Feed	
Link-16 Start and End Track	Because Container Units cannot own targets, it is impossible for them to detect anything. They can report themselves to the Link-16 network, but they cannot detect and report any tracks.

Table 2.3 Container Units Database Information

ATTRIBUTE	EXPLANATION
Link-16 Start and End Target JU numbers.	Because Container Units cannot own targets, there is no need to assign Target JU numbers to their owned targets as is done with other types of units.
Aircraft Type	Container Units cannot own Combat Systems; therefore, it is impossible for them to own aircraft.
Max Sorties Per Day	Because Container Units cannot own aircraft, there is no need to data concerning the maximum number of sorties per day that the unit can fly.
C3 Quality Rating	
Highest C3 Quality Rating	Because Container Units contain no Combat Systems and no Supplies, they cannot be damaged; therefore their C3 Quality Rating cannot be changed. The model automatically assigns the Highest C3 Quality Rating to the Container Unit's database C3 Quality Rating
C4I Name	Container Units were not given a C4I Name.
Comment	Container Units were not given a database comment field.
Higher HQ	
Initial Support Unit	Container units have no supplies; therefore, there is no need for specifying an Initial Support unit.
Support Unit	Container units have no supplies; therefore, there is no need for specifying a Support unit.
Arriving Port	Container Units own nothing to process through a port.
Parent	Container units cannot move; therefore, there is no need to specify a Parent Unit.

All Container units are automatically known to all sides in the game. If users make their Top Unit on each side a Container unit, there is no longer a need to submit Unit Reports for the Top Unit for each side. By making the Top Unit on each side a Container unit, it will be possible to easily select intelligence sharing with coalition partners following detected Command Hierarchy relationships.

Container units are restricted from most orders. They can be specified in all queries and orders that have a Subordinate option. In general, the orders do not apply to the Container units, but all the command hierarchy to be followed so real Aggregate Units can be receive the order. For example, Container units can be specified in the Rules Of Engagement Order if the Subordinate field is set to Yes. Any Container units within the Command Hierarchy specified are skipped, but the Command Hierarchy chain is followed and the order is applied to all of its subordinates.

No Container units are added as part of the database conversion process.

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. Because this is the first release of JTLS-GO Version 6.4.0.0, no STRs have been reported.

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 6.4.0.0 represents a major release of new functionality, all outstanding errors have been reviewed. If the error could not be reproduced, it was considered obsolete and no longer relevant to JTLS-GO. These errors have been removed from consideration for correction at this time.

In future maintenance releases, newly uncovered outstanding errors related to JTLS-GO will be listed in this chapter, along with information regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

APPENDIX A. ABBREVIATIONS AND ACRONYMS

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

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

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

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

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

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

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

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

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

APPENDIX B. Version 6.4.0.0 DATABASE CHANGES

The following changes were made to the JTLS-GO 6.4 database:

- AC WEIGHT was added as a result of JJTLS-2021-15303 Represent UAV Swarms, Section 2.3.
- EEV DCS DAMAGE METHOD was added as a result of JTLS-2013-11469 Improve Damage Combat Systems Order, Section 2.1.
- EEV DCS CREW DAMAGE was added as a result of JTLS-2013-11469 Improve Damage Combat Systems Order, Section 2.1.
- ST FIRE CONTROL RANGE was added as a result of JTLS-2024-16869 SAM Fire Control Sensors Air Search Capability, Section 2.5.
- The new Container Unit table was added as a result of JTLS-2025-17243 Represent Container Units, Section 2.7 and adds the following database parameters:

  CT SHORT NAME

  CT LONG NAME

  CT UIC

  CT FACTION NAME

  CT HIGHER HQ

  CT SYMBOL

  CT COMMAND LEVEL

  CT UT SERVICE

  CT POLITICAL COUNTRY

  CT LINK16 NETWORK NAME

  CT LINK16 FEED NAME

  CT C3 QUALITY RATING

APPENDIX C. Version 6.4.0.0 REPOSITORY CHANGES

Significant changes were made to the JTLS-GO 6.4 repository - in particular, related to the pairing of important Targetable Weapons with unique Supply Categories.