JTLS-GO Document 17

JTLS-GO

Version Description Document

August 2020

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.0.0.0)

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 6.0.0.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 6.0.0.0 is a Major release of the JTLS-GO 6.0 series that includes an updated repository of standard data, a demonstration scenario based in the western Pacific, as well as major model functionality improvements implemented as Engineering Change Proposals (ECPs), These ECPs are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

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

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.0.0.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.0.0.0 is a Major delivery for the JTLS-GO 6.0 series of releases.

JTLS-GO 6.0.0.0 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 “wespac60”. Database modifications that were accomplished to upgrade the previous JTLS-GO database format to this current version are summarized in this chapter, as well as APPENDIX B. Detailed descriptions of Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS-GO 6.0.0.0 executes on the Red Hat Enterprise Linux Version 7.8 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 6.0.0.0 is a major redesign of the JTLS-GO system and all of the documentation has been updated for this release.

1.2.3 Updated Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 6.0.0.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 6.0.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.0.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.0.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.0.0.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.0.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.0.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.0.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.0.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.0.0.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.0.0.0)
- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 6.0.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.0.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.0.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.0.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.0.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.0.0.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.0.0.0)

1.2.4 New Documents

The US Government has required JTLS-GO to produce a Department of Defense (DoD) Architecture Framework (DoDAF) document for the entire system. The DoDAF provides a foundational framework for developing and representing architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures across organizational, joint, and multinational boundaries.

It establishes data element definitions, rules, and relationships and a baseline set of products for consistent development of systems, integrated, or federated architectures. Each program or process delivered with JTLS-GO has been included in this new document. The *JTLS-GO DoD Architecture Framework*, JTLS-GO Document 22, Version 6.0.0.0, is delivered with this release for the first time.

1.2.5 Delivered Software Components

JTLS-GO 6.0.0.0 may be delivered either on a CD or as a set of compressed TAR files to be downloaded. Either method includes the complete suite of software executable code and command procedures. The following software components are included with this release:

- Combat Events Program (CEP)

- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)
- JTLS Symbols Application (JSYMS)
- Database Development System (DDS)

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

  Apache Server (APACHE)
  JTLS XML Serial Repository (JXSR)
  Order Management Authority (OMA)
  Synchronized Authentication and Preferences Service (SYNAPSE)
  XML Message Service (XMS)

Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation
- Tactical Electronic Intelligence (TACELINT) Message Service
- KML Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool

Instructions for installing JTLS-GO 6.0.0.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS prior to installing JTLS-GO 6.0.0.0 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 "repository60". Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space. With JTLS-GO 6.0.0.0, it is possible to access and copy records from the repository60 database into your own developed scenarios.
- The scenario “wespac60”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.0.0.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Edition Version 7.8 (ES), 64-bit architecture.

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

RedHat Linux 7.8 - this operating system license must be purchased.

Oracle Linux 7.8 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

CentOS 7.8 - a free version of Linux 7 that has not been approved by DISA for use by U.S. Government agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.0.0.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.7 and Version 7.8.

  CentOS Linux Version 7.7 and Version 7.8.

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

- JTLS-GO 6.0.0.0 no longer relies on the Open Java Development Kit (OpenJDK^{TM}) to be installed at the system level using the Red Hat Package Manager (RPM). Instead, JTLS-GO 6.0.0.0 is delivered with the equivalent AdoptOpenJDK package. Using AdoptOpenJDK provides two benefits:

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

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

- JTLS-GO uses IcedTea to provide the Java Web Start capability that implements the web-enabled JTLS-GO functionality. JTLS-GO supports IcedTea version 1.8.3.

> Note: Red Hat Linux version 7.8 continues to be distributed with IcedTea version 1.7.1. There is an available RPM for IcedTea 1.8.3, and this should be explicitly installed on the JTLS servers and client workstations. If this is not done, then HTTP unsecured communication cannot be used with Apache and Glassfish.

- JTLS-GO database tools require a certified PostgreSQL 11.8 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.0.0.0. There are several alternative methods available for obtaining the PostgreSQL 11.8 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.

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

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

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

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

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

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

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

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

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

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

- KML Operational Interface (KOI)

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

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

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

1.3.2 JTLS-GO Cybersecurity Compliance

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

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

OpenJDK^{TM} has not implement an expiration date. In order to fulfill this DoD Cybersecurity requirement, JTLS-GO has moved to AdoptOpenJDK, 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, AdoptOpenJDK produces an equivalent version using infrastructure, build and test scripts to produce prebuilt binaries of the OpenJDK class libraries. All AdoptOpenJDK binaries and scripts are open source licensed and available for free.

- Within two-weeks of the AdoptOpenJDK release, JTLS-GO provides a bug release version (JTLS-GO 6.0.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specfic Cybersecurity mandate as long as they download and use the bug released versions when distributed.

Contact the U.S. Government Program Manager, Mr. Donald Weter (donald.e.weter.civ@mail.mil) to obtain the completed Cybersecurity paperwork and a current Authority to Operation certificate.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 6.0.0.0 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 have been accomplished. Future plans include expanding the capabilities of the GlobalSim federation.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.0.0.0 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 6.0.0.0 and the previous JTLS-GO 5.1 series database structure.

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

1.4.1 JTLS-GO Using Legacy Default Symbol Set

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

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.0.0.0, 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 “repository60.” APPENDIX B. will continue to provide a summary of the data changes made to the data repository, On of the major improvements in this version of JTLS-GO is the ability to compare and synchronize two different JTLS-GO 6.0.0.0 scenario using a drag and drop interface.

1.5 INSTALLATION

The *JTLS-GO Installation Manual,* a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery, It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 11.8 required to operate JTLS-GO 6.0.0.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.0.0.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2011-10948 Multiple Names For Air Missions

Summary of Model Change Request

Like all JTLS-GO objects, Air Missions have a primary name that is used to identify the object. When a user submits an order, the name of the mission, as specified on the order, is used to identify the mission on future orders, within messages, on the map display within the Web-Hosted Interface Program (WHIP), and held by the After Action Review (AAR) database. For primary Force Side missions, the mission name is typically the mission number assigned as part of the ATO process.

Although the mission number in JTLS-GO is its unique identifier, most communications with the exercise audience and AOC refer to the mission by its call sign, as would be done during real operations. It is difficult for the WHIP user and the response cell to quickly find missions and view a mission's current status when the AOC is asking questions using the mission's call sign, rather than its ATO-assigned mission number.

The user should be able to alter the names shown on the WHIP and to view a mission using its mission number or its call sign.

Design Summary

As with most requested ECPs, R&A took a much more global approach to fulfilling the ECP requirements. The concept of multiple object names was not limited to Air Missions. Table 2.1 lists the various names that can be viewed on the WHIP map. Those names highlighted in Green represent new data added to the model in support of this ECP.

Table 2.1 Implemented Object Multiple Names

OBJECT TYPE	ALLOWABLE NAMES
Land Based Units • Ground Combat Units • Airbase • Squadron • Support Unit • Forward Arm and Refuel Points (FARP)	• Unit Short Name • Unit Identification Code (UIC)

Table 2.1 Implemented Object Multiple Names

OBJECT TYPE	ALLOWABLE NAMES
Naval Units	• Unit Short Name • Unit Identification Code (UIC) • Passive Identification Friend or Foe (PIFF) Code • JU Number • Hull Number • Trigraph • Maritime Mobile Service Identity (MMSI) • International Maritime Organization (IMO)
Air Mission	• Order assigned name • Call Sign • JU Number
Targets	• Configuration Control Facility (CCF) Number • JU Number

There is a very simple rule for allowing the user to see the various names.

- If the Force Side perceives the real name of the object, that Force Side is allowed to view any of the names listed in Table 2.1.
- If the Force Side does not perceive the real name of the object, only the model generated name based on the intelligence collection capability can be viewed.

2.2 JTLS-2013-11707 Improve Simscript Debugging Capability

Summary of Model Change Request

Previous versions of JTLS-GO used the Simscript II.5 programming language. JTLS-GO needed to move the Simscript III programming language to improve the debugging capability for the design team and to ensure that the compiler would continue to function with new releases of the Linux operating system.

Design Summary

There were two reasons the JTLS-GO did not immediate convert to the Simscript III programming language, which was originally released several years ago.

- At the time Simscript III was originally released, JTLS-GO was converting its hexagon terrain to the gridded terrain, and accomplishing both tasks simultaneously was deemed too large of a change to accomplish safely.

- There are several significant changes to the programming language. Specifically, the Simscript II.5 capabilities listed in Table 2.2 have either been deprecated or are new to the programming language, Deprecated capabilities still work, but CACI suggests that plans should be made to remove them from the code-base.

Table 2.2 Simscript III Deprecated And New Capabilities

CAPABILITY	CURRENT STATUS
Events	Events cannot be removed from JTLS-GO. CACI is aware that this deprecated capability cannot be removed unless a suitable alternative is provided.
Single Event Set	CACI indicates that using one event set will drastically speed up JTLS-GO. R&A plans on investigating this claim after JTLS-GO 6.0.0.0 is released. If there is a significant speed increase, JS/J7 project management will need to decide when to move to the new capability.
Bit-packing variables	All bit-packed variables have been removed in JTLS-GO 6.0.0.0.
Variable equivalence	Variable equivalencing, a memory saving technique, has been removed from JTLS-GO 6.0.0.0.
Permanent Entities	Many objects within JTLS-GO are represented as permanent entities, objects that can neither be created or destroyed during game play. Examples of permanent entities include Targetable Weapons, Ship Unit Prototypes, and Sensor Types. In order to get a better understanding of the effort to remove permanent entities from the code-base, JTLS-GO 6.0.0.0 changed the representation of the Targetable Weapon entity. It is now represented as the preferred temporary entity. The task was more difficult than expected, but was successfully completed. The impact on the model is that a new Targetable Weapon can be created during game play, by copying an existing Targetable Weapon and altering some of its characteristics. Although temporary entities can be removed from the game, this capability is not desired and has not been implemented. JS/J7 project management will need to decide when to move the remaining permanent entities to this new capability.

2.3 JTLS-2017-13132 Allow HRUs (UAVs) To Gather Intel From DSAs

Summary of Model Change Request

High Resolution Units (HRUs) perform intelligence gathering missions within JTLS-GO. Under the HRU Tasking Order, an HRU can be directed to Patrol a location or an area to collect essential elements of information (EEI) that are reported to the intelligence player. Directed Search Areas (DSAs) are established for use by reconnaissance air missions to collect intelligence within the specified area or location. Depending on the sensor type carried by the air mission, an intelligence report is also generated.

This ECP requests that HRUs be allowed to use DSAs and generate a Non-Theater DSA Intelligence Report similar to air missions.

Design Summary

JTLS-GO was enhanced to permit a DSA to be assigned to an HRU that is ordered to perform a Patrol task. A DSA collection field was added to the HRU Task order as an optional entry for the Patrol task. The specified DSAs must exist, and the Patrol task must specify a Patrol location that places the HRU within sensor range of the DSA region. No changes were needed to the manner in which the HRU moves to the Patrol location. While patrolling, the HRU determines if any of the assigned DSAs are within sensor coverage range, using whatever sensor targets it owns.

If the HRU sensor target has the following characteristics, a detection method of DETECTION INSTANTANEOUS, a Non- Theater DSA Collection Report is generated for the items detected.

- A detection method of DETECTION INSTANTANEOUS
- A sensor usage of SURFACE SEARCH
- A sensor collection mode that is not equal to ELINT

2.4 JTLS-2017-13139 Better Location Information In IIR Messages

Summary of Model Change Request

This ECP started out with a simple request, but has expanded. The Intelligence Staff at JS/J7 requested that the GEOLOC record in an IIR be allowed to have both a Latitude and Longitude version of the collection location and a Military Grid System version of the location. It was impossible to do within a single message.

After starting that improvement, the JS/J7 Intelligence Staff also requested that the Status Activity (STAACT) record also be expanded.

Design Summary

This IIR message was improved to accomplish each of these tasks.

2.5 JTLS-2019-14351 Allow Viewing Multiple Messages

Summary of Model Change Request

There are many times when a user would like to compare two messages. To accomplish this task, the user must bring up two Message Browsers.It should be possible to double-click on a message in the Message Browser and have it open in another window. Several windows should be able to be open at the same time, to make it easier to compare information in different messages. This option should be configurable by users.

Design Summary

If the user single-clicks on the subject of a message, the message will appear in the message viewing area of the Message Browser. A double-click will display the message in a separate and independent window. From the independent window the user can also e-mail or print the message being displayed.

2.6 JTLS-2019-14352 Develop Maritime Specific DSA Report

Summary of Model Change Request

A maritime-specific intelligence collection report, on detections from non-DSA required sensor assets, should be added to the model.

Design Summary

Two capabilities were identified as being needed for a Maritime Collection Report

- When an air mission is complete, it now produces a non-reported collection message.
- The new AAR has the ability to create an air mission report that lists all naval units detected by an air mission throughout its flight. This is currently a report, but work is currently in progress to plot these detections on a graphical display.

2.7 JTLS-2019-14353 Convoy Data in AAR

Summary of Model Change Request

All data available for convoys should be saved in the AAR, as is currently done for air missions. This should include Supplies carried and all event timing.

Design Summary

All available data for convoys is saved in the AAR and the data is retrievable through the AAR Viewer.

2.8 JTLS-2019-14354 All WHIP Objects Searchable

Summary of Model Change Request

Allow the user to search for any object displayed on the map. These include Directed Search Areas (DSAs), Operations Areas (OPAREAS), Network Arcs, Network Nodes, Battle Damage Assessment (BDA) Areas, etc.

Design Summary

The WHIP map search capability has been expanded. There are now three tabs to the search capability.

- Map object names. A Red X is placed on the objects that match the search.
- Named area searches, such as BDA area and DSA Areas. A Blue X is placed on the areas that match the search.
- Named Network objects such as road nodes and rail arcs. A Green X is placed on the network objects that match the search.

2.9 JTLS-2019-14356 Attacking Mission ROE Help

Summary of Model Change Request

It is easy for the user to forget if an attacking air mission has been given permission to fire. The WHIP needs to provide help for the user.

Design Summary

If an Offensive Air Support mission starts to head toward its attack location and it does not have a permissive ROE against the side that owns its designated target, the model now generates an ROE Warning Alert.

2.10 JTLS-2019-14357 Move from Oracle to PostgreSQL

Summary of Model Change Request

JTLS-GO needs to move completely away from the Oracle Relational Database Management System (RDBMS) to PostgreSQL.

Design Summary

Oracle has been replaced with the PostgreSQL (RDBMS) for all database related functions in JTLS-GO. These include:

- The Database Development System
- The Scenario Data Repository, which includes the After Action Review (AAR) Database.

The user should see no functional changes as a result of moving away from Oracle.

2.11 JTLS-2019-14400 GlobalSim Improvements

Summary of Model Change Request

A number of improvements were needed to support the GlobalSim Federation. This federation links the JTLS-GO simulation with the CAE's GESI Tactical Model. These improvements were necessary to fully develop the federation as a working, multi-resolution system for simulating the battle field at both the operational and tactical levels. The specific improvements are designed to facilitate the data mapping between JTLS-GO and GESI.

Design Summary

The improvements made to the federation were designed to properly represent the relfelction of supplies and the consumption of supplies consistently across the two models. For correlation between the simulations, the supplies were identified using the enumerations for Distributed Interactive Simulation (DIS) codes. Each supply category in JTLS-GO can be assigned a DIS enumeration and an appropriate unit of measure recognizable by GESI.

2.12 JTLS-2019-14464 Expand TACELINT Data Fields

Summary of Model Change Request

There are three data fields that the current Tactical Electronic Intelligence (TACELINT) Message Service (TEMS) does not support. These fields are:

- Pulse Duration
- Scan Type
- Scan Rate

Design Summary

In JTLS 5.1.3.0, these new data were added to the model as non-configuration managed data. The data as represented in JTLS 5.1.3.0 was accomplished in an expedited manner. The purpose of this ECP is to formalize this data in the JTLS 6.0.0.0 database structure. Some shortcuts, mandated by the time-frame allowed for implementation in JTLS 5.1.3.0, were corrected to follow the strict programming and database structure guidelines used by the JTLS-GO software.

2.13 JTLS-2019-14476 Move To Adopt OpenJDK

Summary of Model Change Request

Java is required on JTLS-GO servers, which are used to run the model and all of its support software. It is also required on JTLS-GO client machines is used to run:

- The user interface to the model, called the Web Hosted Interface Program (WHIP).
- The user interface to the Database Development System (DDS), called the DDS Client (DDSC), and;
- The user interface to the After Action Review (AAR) capability, called the Total Recall Interactive Post-Processor (TRIPP).

In all previous version of JTLS-GO, each user organization was responsible for ensuring that JTLS-GO client machines had installed the correct version of 64-bit Java. All documentation delivered with JTLS-GO indicated the Java version requirements. This ECP does not alter the requirement for user organizations to properly setup their client machines.

The ECP only addresses how to install Java on JTLS-GO server systems. This ECP removes the requirement for system administrative support to install the proper version of Java on JTLS-GO server systems. This allows JTLS-GO to ensure the most current security release of Java is being used on the JTLS-GO servers.

Design Summary

Oracle, the primary distribution agency for Java, recently changed its Java licensing rules, which ended the free-of-charge business use of Java. The JTLS-GO Design Team had selected the community-led Open Java Development Kit (OpenJDK^{TM}) for alternative access to Java.

In order to use OpenJDK, on-site installation of the Red Hat OpenJDK was required, from either a valid accessible yum repository, or a file server hosting OpenJDK and dependency rpm files. Either installation method requires system administrative support for root access to the Linux server. Because of this system level requirement, R&A ended delivery of Java with the JTLS-GO software within the JTLS-GO Version 5.1 series. The documentation indicated the OpenJDK version that was expected to be used on the JTLS-GO servers.

Because the installation of Java (either JDK or JRE) was no longer under the control of the JTLS-GO installation manager, the JTLS-GO Information Assurance package could not guarantee that the system was using the most current Java security release. To resolve this burden, the Design Team conducted further investigation for a reliable, stable and portable alternative to the Red Hat vendor OpenJDK distribution. The Design Team concluded that the AdoptOpenJDK^{TM} meets JTLS-GO long term needs, and is thus a better viable solution.

AdoptOpenJDK is an open, community-led initiative that provides free, prebuilt binaries of the Java platform from OpenJDK that has a licensing agreement that allows it to be delivered as part of the JTLS-GO application installation package. The AdoptOpenJDK community's goal is to ensure that regularly updated downloads of Java are available for the most modern operating systems and architectures, such as Linux and Windows. It is backed by industry leaders such as Azul, IBM, and Microsoft, It has over 6 million end users from around the world.

The JTLS-GO Version 6.0 series is delivered with the most recent security release of Java 8 using the AdoptOpenJDK Java libraries.

2.14 JTLS-2019-14477 Identify Aircraft Class As UAV

Summary of Model Change Request

There is no easy way to identify whether an aircraft class represents an Unmanned Aerial Vehicle (UAV). This means the Design Team cannot write code-specific logic for UAVs.

Design Summary

The following data currently exist in JTLS-GO to represent various types of UAVs:

- An Aircraft Class has an attribute called AC LOITERING MUNITION FLAG. When set, the Aircraft Class is assumed to be the type of UAV that does not return from its mission. Instead it will either die or it will explode when it has been assigned to attack a target. The capability was built to represent the Israeli and Chinese HARPY UAV.

- An Aircraft Class has an attribute called AC.MAX.CONTROL.DISTANCE, which indicates how far the aircraft can operate from its home squadron. It is assumed that if this attribute has a value greater than zero, that the Aircraft Class represents a UAV.

The issue is that there are many UAVs that are not loitering munitions or have a distance limitation on how far from their home location they can travel. Numerous existing UAVs can operate anywhere in the world via a satellite link and are not limited based on distance or radar horizon from their control or home station.

To provide an easier method to determine which Aircraft Classes are in fact UAVs, the current AC LOITERING MUNITION FLAG was deleted from the database and replaced with a new Aircraft Class attribute called AC CONTROL MECHANISM. This new attribute can be assigned one of the following three values:

- MANNED - The aircraft is piloted by humans.

- UNMANNED - The aircraft is a UAV, and may or may not have a specified AC MAX CONTROL DISTANCE.

- LOITERING_MUNITION - The aircraft is a specialized UAV that will become a kamikaze aircraft as it is assigned to attack a specific target by the player.

2.15 JTLS-2019-14486 Add Tags To Checkpoints

Summary of Model Change Request

Add tags to checkpoints, to help identify restart positions.

Design Summary

Currently, JTLS-GO takes a checkpoint under three different circumstances:

- When the user submits an order to take a checkpoint.
- Periodically and automatically based on a user defined time between checkpoints.
- When the model crashes and the user submits the “Snap” checkpoint command from the debugger.

When a checkpoint is taken, a single line is added to the <scenario_name>.tim file that contains the checkpoint number, the game time of the checkpoint, and the system clock time when the checkpoint was taken. This ECP requests that an additional field be added to the <scenario_name>.tim file that contains an text string identifying the reason for the checkpoint.

Under all three circumstances, the user can enter the desired checkpoint tag string or allow the system to automatically assign the tag.

2.16 JTLS-2019-14487 Torpedo Hits Detectable

Summary of Model Change Request

If a heavy torpedo hits its designated target, in the real-world, the firing unit will always be able to detect whether there was a hit or miss. This does not happen within the model. The WHIP should have an alert message indicating that there was a direct hit.

Design Summary

Once an underwater Targetable Weapon hits either a surface or subsurface naval vessel, a Hit Alert is now generated. This design considers the following issues:

- The Targetable Weapon attribute, TW TYPE IMPACT is used to determine if the weapon is subject to the generation of an alert when it hits its intended target. If the value of the TW TYPE IMPACT attribute is set to SUB-SURFACE BURST, then the weapon hit will cause some noise that can be identified as a weapon hit. A SUB-SURFACE BURST is not only associated with torpedoes, but is also associated with water mines; thus, a ship hitting an underwater mine will generate an Underwater Hit Alert for assets within detection range of the mine explosion.
- All Passive Sonar sensors that are within COLOCATED DISTANCE of the weapon explosion can possibly detect the hit. The model uses the standard sound attenuation loss algorithm, already documented in the Analyst Guide. A new Targetable Weapon database attribute was added to model. This attribute, TW IMPACT NOISE LEVEL, holds the amount of noise made when a successful hit is accomplished. If enough noise reaches the passive sonar, the side that owns the sonar will receive a hit alert.

Because the number of Alert Types are growing, this design also redesigned the Alert Selection and Filter Panel.

2.17 JTLS-2019-14508 Build Geo-Spatial Service

Summary of Model Change Request

A Geo-Spatial Service (GSS) would provide the Combat Events Program (CEP) with a method to utilize parallel processing for some tasks, speeding up the CEP. Because of the model programming language, SIMSCRIPT III®, the CEP Is not capable of performing parallel processing. It is not uncommon during large exercises to have game speed limited by Central Processing Unit (CPU) capability, with the CEP running on a single thread.

If discrete, well-contained tasks that require a lot of CPU capacity could be identified and offered the potential for parallel processing, the CEP could give these tasks to a program that can utilize the additional CPUs available on the hardware. An examination of the CEP code has identified a small number of such tasks that exist within the CEP, all of which are centered around distance calculations. This ECP entails the development of an optional GSS service to accomplish the parallel processing of these tasks.

Design Summary

This ECP developed a service which can receive tasks and data from the CEP, accomplish the tasks with parallel processing, and return the results to the CEP. An examination of the CEP code has identified three tasks that require a minimum amount of data to be passed and which might benefit from parallel processing:

- **Determining Terrain Grids Within Range Of Location.** The algorithm to perform this starts with the top-level terrain layer, goes through each of its grids to determine if the grid is within range, discarding those that are not. Those grids that are in range are checked to determine if they are covered by a finer resolution terrain layer, which if true is then processed in the same manner. Instead of processing the grids within a terrain layer sequentially, this task offers the opportunity to perform the processing in parallel.

- **Determine Objects Within Range Of Location.** Terrain grids track the objects that are within their geographic region, so gathering grids is the first phase of gathering objects that are within range of a specific location. Once the grids are gathered, the CEP goes through each grid and checks the objects within that grid to determine if they are in range of the location. Some minor code changes would allow all the objects in the grids to be gathered, and then to process the objects to determine if they are within range, which would permit the parallel processing of this task.

- **Determine DSAs Within Range Of Location.** Air missions within JTLS-GO can carry sensors that are capable of searching for objects within DSAs. Either the air mission is provided a specific list of DSAs to search or the model will choose one that is within range every time the mission moves. In both cases every movement event of the air mission requires the model to prioritize and calculate a list of DSAs that are within range. Satellites also search DSAs with the model prioritizing and selecting ones that are within range for every satellite movement.

These three tasks all met the criteria required to pass them off to the GSS:

- They are well contained tasks that require a limited set of data to accomplish.
- Can be done either sequentially or in parallel.

Since the GSS is an optional service if the service is not being used, then the tasks are still done sequentially by the CEP.

2.18 JTLS-2019-14529 Allow Selection Of MGRS Format

Summary of Model Change Request

The user can select whether fields should be displayed using a Latitude and Longitude format or a Military Grid System (MGRS) system. In JTLS-GO messages, the format that should be used for Latitude and Longitude is very flexible and robust. There is no equivalent for the format that should be used for MGRS formatted fields. Some fields require MGRS fields to the nearest meter and some need the format to be provided to the nearest 10 meters. It is not possible to specify this in the message format file.

The purpose of this ECP is to address the need to specify the precision of Military Grid Reference System (MGRS) coordinates in messages.

Design Summary

A new format specification for MGRS locations was added, using a method similar to the format specification currently implemented for latitude and longitude locations. The specification format is defined completely within the *JTLS-GO Software Maintenance Manual*.

2.19 JTLS-2019-14570 Data Repository Synchronization

Summary of Model Change Request

JTLS-GO has added the concept of a data repository, in which users can drag and drop data from one database to another. The last step in this implementation plan was to build a capability to compare existing data in the data repository to the data in a user database. The user can then decide whether to use or not use newly updated repository data. This ECP covers this last step called Repository Synchronization.

Design Summary

In previous versions of JTLS-GO, the user can currently accomplish the following tasks:

- Move Order of Battle (OOB) data from the repository to the scenario database.
- Move all other non-Order of Battle data from the repository to the scenario database.
- “Repair” of cycle data that could not be moved during the transfer of the above data.

This ECP added a fourth option to the “Repository Tool” bar, named “Synchronize Data”. When selected, the user selects the name of a table for which the data should be synchronized. The DDSC brings up a dual-line table that lists each record in the repository and each record in the scenario database. The comparison of these two lists result in three situations:

- A data record exists in the repository, but does not exist in the scenario destination database.
- A data record exists in the scenario destination database, but not in the repository.
- The data record exists in both databases, but the individual pieces of data do not match.

The synchronization process allows the user to decide on a record-by-record basis and an attribute-by-attribute basis, which data should be moved from the repository to the scenario database.

2.20 JTLS-2019-14608 Add Unit Comment Field

Summary of Model Change Request

The database-building community has indicated that they could use a comment field on each unit record, which could be used for a variety of purposes. For example, the comment might include information such as “NOT APPROVED FOR USE IN EXERCISE ALPHA”. These comments must be searchable and filterable.

Design Summary

A unit attribute was added to each unit table, including the High Resolution Unit (HRU) table. The database builder can view, change, and clear the comment as desired. The new field is sortable, searchable, and filterable.

During the design phase, the user community also requested that a comment field be added to the following tables:

- Combat System Table

- Supply Category Table
- Targetable Weapon Table

The new data fields are in the initialization database, read into the model, assigned to the unit, and saved as part of the checkpoint procedure. The data held in the new fields cannot be accessed or used in any way by the model. There is no ability for the Controller or the Player to change the contents of the comment attribute during game play.

2.21 JTLS-2020-14624 Save WHIP Layout

Summary of Model Change Request

The WHIP is a composite program made up of several components. These components allow users to access information about the game, as well as interact with it. Commonly used components are the Map, IMT (Information Management Tool), SITREP (Situation Report), and Command Hierarchy.

Players often arrange and format the components of the WHIP to satisfy their needs, according to their tastes and the role they are performing. This process of arranging the components of the WHIP may be tedious, especially if it must be done repeatedly. Users should be able to save the arrangement of WHIP components as a Layout and restore that Layout at will.

Design Summary

Layouts are a collection of WHIP components, with each component having a specific size and location. Some components may have additional state information saved. Some of the other implemented characteristics of this capability include:

- Each WHIP is able to save and restore a single Layout.
- Layouts are saved and restored manually by the user with the Layout tool.
- Layouts are neither saved nor restored automatically by the WHIP. Because of this, WHIPs continue to use the default of components, consisting of a Map, Command Hierarchy, and SITREP, when launched.

To support saved WHIP layouts within the game directory structure, a redesign of the location of saved orders and order groups was also implemented. The new directory structure provides for a more consistent and easily understood directory structure.

2.22 JTLS-2020-14625 Quickly Change Mission Speed/Altitude

Summary of Model Change Request

Air Operations Players require fast access to alter altitude and speed for an Air Mission without using the Change Air Mission Parameter Order.

Design Summary

The design entails the creation of two new quick orders:

- Quick Mission Speed Order - used to change air mission speed.
- Quick Mission Altitude Order - used to change air mission altitude

These orders were placed in the quick order menu for both the Player Menu and the Air Player Menu. The Controller does not have access to these orders, because managing air missions is not part of their assigned tasks.

2.23 JTLS-2020-14626 Different Coordinate Formats In IMT

Summary of Model Change Request

Currently, users can change several WHIP wide preferences, such as the format that should be used to display location information. The selected preferences apply to all WHIP components.

For example, changing the location preference to Military Grid System (MGRS) results in all location data being displayed using the MGRS format. This includes locations in the Message Browser, in the SITREP component, and in each of the IMT displays.

Users should be able to change the location format for individual IMT displays, in order to show locations in an Naval IMT display using the Latitude/Longitude format, but to show locations on a ground IMT display using MGRS.

Design Summary

Although the ECP only mentioned the ability to change specific IMT location preferences, a more global approach was implemented. For any specific IMT, the user can set:

- Location Preference
- Unit of Measure (UOM) Preference for the general UOMs, such as distance, speed, and area
- Specific Supply Category UOMs.

2.24 JTLS-2020-14627 IMT Column Size Adjustment

Summary of Model Change Request

Players frequently have multiple IMTs open with multiple rows displayed that are cumbersome to manage, because the columns cannot be adjusted to optimum widths quickly and easily. No capability exists to automatically resize a column to fit its contents. A column width can be adjusted only by selecting the edge with the mouse and dragging left or right. The width can be increased almost limitlessly, but cannot be decreased below a fixed minimum width.

Individual column widths in the IMT should be fully and quickly adjustable.

Design Summary

Column size is completely user configurable. If the contents of the IMT cell is truncated, then what is known as the ellipsis (three dots ...) is shown at the end of the visible contents.

Because the DDS also uses table displays that are very similar to the WHIP IMTs, the design team decided to expand this ECP to implement the same enhancements in the DDS application.

2.25 JTLS-2020-14628 Add Concept Of COMAO ID

Summary of Model Change Request

The NATO Air Tasking Order (ATO) generated by the Integrated Command and Control (ICC) system can contain information concerning the Combined Air Operations Identifier (COMAO ID) for a mission. Until recently, this ATO field was misinterpreted and equivalenced to what JTLS-GO calls an Air Mission Package. This is not true.

NATO ATOs use the concept of an Air Mission Package, but in addition, they group missions into COMAO groups. A COMAO consists of multiple air missions and perhaps multiple air mission packages. For the JTLS-GO Air Response cell to properly and efficiently interface with the exercise audience and the AOC, they need quick access to both a mission's Air Mission Package and its COMAO group.

This ECP requests two things:

- The JTLS-GO ATO-T should pull the COMAO ID from the ATO message generated by ICC and send that information to the model.
- Once an air mission has been given a COMAO ID is should be readily viewable in the following places:

  a. The IMT display for a mission.

  b. The SITREP component for a mission.

c. The order group viewer for a group of Air Mission orders.

d. The ATOT Summary mission panel.

Design Summary

The COMAO ID has absolutely no impact on a mission's flight profile or tasking. It is simply an organizational name used to group missions from a planning point of view. Thus, the implementation of the requirement is fairly straight forward and consists of the following steps:

1. Add the COMAO ID data to Air Mission Orders.
2. Save the information on the mission structure within the model.
3. Output this information to the JODA.
4. Alter the WHIP components to get this information from the JODA and display it as needed by the user.
5. Alter the NATO ATOT Parser to get this information from the ATO message.
6. Have the ATOT generate orders with the parsed COMAO ID in each of its generated orders.

Each of these steps were implemented and are functional within this version of JTLS-GO

2.26 JTLS-2020-14629 Improve Order Route Display

Summary of Model Change Request

In large exercises, the ATO-T is used to translate the exercise audience's daily ATO into JTLS-GO orders. Each exercise audience has its own methodology for creating ATOs. The ATO-T is constantly being refined to understand each of the possible options that can be used by the Air Operations Center (AOC) when building an ATO.

For this reason, it is very important that once the ATO has been translated, the JTLS-GO air cell operators need to quickly determine whether the translation resulted in orders that properly represent the desires of the exercise audience. There are two methods available to the Air Cell to accomplish this task:

- The orders can be sent to test game and the execution of the orders can be monitored prior to entering the orders into the production game. JTLS-2020-14630 Light Air Game is also part of JTLS-GO 6.0 and was implemented to help speed up this process.
- The air cell can review the orders prior to sending them to the model. This ECP addresses needed WHIP improvements to support this review methodology.

The desire is for a JTLS-GO operator to be able to access the ATO-T generated orders and graphically view the planned routes for the Air Missions that will be created when the order is sent to the game. Users should be able to display full Air Mission Route on the WHIP Map, by just opening the parsed order panel from the ATO-T generated Order groups within the Order Group Editor.

Design Summary

The order graphics display options were significantly expanded. A new "connect" option is available, which results in connecting the currently available disjointed graphical information. When the formerly disjointed information is connected, a user can now quickly determine whether the generated order seems logical and meets the desires of the exercise audience.

2.27 JTLS-2020-14630 Light Air Game

Summary of Model Change Request

When running a large exercise, normal standard operating procedures call for running the translated ATO into a shadow game and to test the translated orders for consistency. Testing the ATO-T orders is not a mandatory requirement, but is a part of the ATO-T verification best operating procedures discussed within JTLS-GO documents. For large exercises, the test run can take hours and there is not enough time to complete the test before the orders must be entered into the production game.

The purpose of this ECP is to create an ATO-Lite capability in which only air events execute. The concept is that the ATO-translated orders can be tested without spending processing time accomplishing other event tasks.

Design Summary

There are two basic concepts behind the implementation of this ECP.

- The implementation should be simple. In other words, the determination of whether an algorithm should or should not be executed when in ATO-Lite mode had to be centrally located and not embedded in a complicated decision structure. At the top of each event, a check was added to determine if the event should or should not be executed when operating within the ATO-Lite mode.

- Since running in ATO-Lite mode will ruin the scenario, great effort needed to be allocated towards ensuring that the ATO-Lite decision is not accidentally activated within the exercise production game. The ATO-Lite mode can only be initiated from the JODA console and must be verified on the CEP execution window. Only Technical Control personnel have access to these capabilities and would be responsible for properly deciding whether the ATO-Lite execution model is or is not appropriate.

2.28 JTLS-2020-14789 Update MIDB Tool For JDPI

Summary of Model Change Request

The Modernized Integrated Database Base (MIDB) is a real-world US database that contains the following information:

- Enemy targets of interest to the US
- Established Joint Desired Points of Impact (JDPI) identified
- Enemy order of battle information.

The desire is to create an automated tool to help take available MIDB data and enter it into a JTLS-GO scenario database.

Design Summary

The MIDB Tool delivered with JTLS-GO 5.1.5 automatically entered enemy targets into a scenario database. JTLS-GO Version 6.0 addresses the second issue. Given a spreadsheet of JDPI data from MIDB, the MIDB Tool allows for the automatic entry of the data into a scenario database. A future version of JTLS will address moving the enemy order of battle information into a scenario database.

3.0 SOFTWARE TROUBLE REPORTS

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

Chapter 4.0 describes STRs that remain outstanding from previous versions; however, because of the model enhancements included with JTLS-GO 6.0.0.0, no previously-identified STRs are considered valid. Errors identified for JTLS-GO 6.0.0.0 and corrected for future Maintenance releases in the JTLS-GO 6.0 series will be documented in this chapter.

4.0 REMAINING ERRORS

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

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

4.1 DDSC – TMU Line Mode Changes Multiple Grids

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

4.2 DDSC – Multiple Types In DDS History Table

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

4.3 WHIP - Pipeline Not Shown On IMT

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

4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

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

APPENDIX A. ABBREVIATIONS AND ACRONYMS

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

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

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

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

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

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

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

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

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

APPENDIX B. VERSION 6.0.0.0 DATABASE CHANGES

The following changes were made to the JTLS-GO 6.0 database:

B.1 DATABASE FORMAT CHANGES

- The format of each data file changed as a result of Chapter 2.10 JTLS-2019-14357 Move from Oracle to PostgreSQL, but the data held in the database did not need to change for the movement from Oracle to PostgreSQL.

B.2 ADDED DATA FIELDS

- Added HRU.COMMENT and UT.COMMENT to the unit tables. JTLS-2019-14608
- Added CS COMMENT to the combat_systems table. JTLS-2019-14608
- Added SC.COMMENT to the supply_category table. JTLS-2019-14608
- Added TW COMMENT to the targetable_weapon table. JTLS-2019-14608
- Added TW.IMPACT.NOISE.LEVEL to the targetable_weapon table. JTLS-2019-14497
- Added AC.CONTROL.MECHANISM to the aircraft_class table. JTLS-2019-14477
- Added ST.PULSE.DURATION. ST.SCAN.TYPE, and ST.SCAN.RATE to the sensor_type table. JTLS-2019-14464
- Added JT.PULSE.DURATION. JT.SCAN.TYPE, and JT.SCAN.RATE to the jammer_type table. JTLS-2019-14464
- Added AC CONTROL MECHANISM to the aircraft_class table. JTLS-2019-14477
- Added IIP.MARITIME.INTEL.REPORT.PERIOD to the intelligence_information_prototype table. This new attribute is currently not used in the model. It was added to the database to support a new training gap that was identified late in the development cycle of JTLS-GO Version 6.0. Specifically there is a desire that surface tracks are updated periodically based on a database parameter. They are currently updated hourly and this periodic update cannot be altered. Expect the database parameter to be implemented and used in a future JTLS-GO 6.0 version.
- Added UT TRIGRAPH UT IMO, and UT MMSI to the naval_unit table. JTLS-2011-10948

B.3 DELETED DATA FIELDS

- Deleted ACP POST ATTACK SEARCH TIME, ACP POST DETECT SEARCH TIME, TW FIRING NOISE TIME from the database. JTLS-2019-14497

- Deleted AC LOITERING MUNITION FLAG from the database. JTLS-2019-14477

APPENDIX C VERSION 6.0.0.0 REPOSITORY CHANGES

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

C.1 Added New Force Sides:

- ANZUS
- GCC (Gulf Cooperation Council)
- IRAN
- NATO
- RUSSIA
- SOUTH.CHINA.SEA.GROUP

C.2 Added New Factions

- ANZUS_MILITARY
- AUSTRALIA_MILITARY
- CANADA_MILITARY
- INDONESIA_MILITARY
- NATO_MILITARY
- NEW.ZEALAND_MIL
- SCGS_MILITARY
- SINGAPORE_MILITARY
- UK_MILITARY

C.3 Added New Air Control Prototypes:

- GCC_ACP
- IRAN_ACP
- RUSSIA_ACP

C.4 Added New Combat System Prototypes:

- GCC_CSP
- IRANIAN_CSP
- RUSSIAN_CSP

C.5 Removed Combat System Prototypes

- NEW_NAVY_CSP
- OTHER_SF_CSP
- SOF_CSP
- US_ARMY_AF_CSP
- US_MAR_NAV_CSP

C.6 Renamed Navy Combat Systems

- IFV-LA-TC renamed to NVL.GUN-20-27MM-GO
- IFV-HA-TC renamed to NVL.GUN-20-27MM-GT
- ARTYSP-MHV-T renamed to NVL.GUN-3.9IN-100-FGT
- ARTYSP-MHV-O renamed to NVL.GUN-3.9IN-100-SGT
- ARTYTOW-MHV renamed to NVL.GUN-3.9IN-100-VSG
- IFV-ATSMLATC renamed to NVL.GUN-30-35MM-GO
- IFV-ATSMHATC renamed to NVL.GUN-30-35MM-GT
- IFV-ATSHXATC renamed to NVL.GUN-37-40MM-FGO
- MTRSPLT-LAO renamed to NVL.GUN-37-40MM-FGT
- IFV-ATLHXATC renamed to NVL.GUN-37-40MM-SGO
- IFV-ATSHHATC renamed to NVL.GUN-37-40MM-SGT
- ATG100-125MC renamed to NVL.GUN-3IN-76-FGO
- ARTYSP-LT-O renamed to NVL.GUN-3IN-76-FGT

- ARTYTOW-LT renamed to NVL.GUN-3IN-76-SGO
- ARTYSP-LT-T renamed to NVL.GUN-3IN-76-SGT
- ARTYTOW-VLT renamed to NVL.GUN-3IN-76-VSG
- ARTYSP-MLT renamed to NVL.GUN-4.5IN-120-FGT
- ARTYTOW-MLT renamed to NVL.GUN-4.5IN-120-SGT
- MTRSPHVY-LAO renamed to NVL.GUN-57MM-FGO
- MTRSP120-LAT renamed to NVL.GUN-57MM-FGT
- ARTYSP-HV-O renamed to NVL.GUN-5IN-130-FGT
- ARTYTOW-HVY renamed to NVL.GUN-5IN-130-SGT
- MRL-SR-TOWED renamed to NVL.MRL-107MM
- MRL-MR-VEH renamed to NVL.MRL-BM11
- MRL-SR-VEH renamed to NVL.MRL-BM14
- o MRL-LR-VEH renamed to NVL.MRL-BM21
- MTRDISM50-60 renamed to 60MM_MORTAR
- TANK100-LFLS renamed to 100MM-TNK.TRT
- TANK105-NFMS renamed to 115MM-TNK.TRT
- TANK76-NFLS renamed to 76MM-TNK.TRT
- TANK100-NFLS renamed to 85MM-TNK.TRT

C.7 Removed Navy Combat Systems

- ATG73-106NMC

C.8 Added New Bridge Types

- WOODEN_ROAD_BR

C.9 Added New Minefield Types

- MK54_HAMMERHEAD

C.10 Removed Equipment Shelter Types

- VEH.CO.FIGH.POS
- VEH.PLT.FIG.POS

C.11 Added Aircraft Target Types

- B1B
- B2A, B52, TU160, TU22M
- H6A, H6D, H6M
- H5, HZ5, IL28
- FT5,JJ5, MIG15
- F5, F7M, J6, J7
- F4, MIG23, MIG27, SU24
- F22, F35, FC31, J20, SU57
- MQ4C, RQ4, U2S
- PC6, PC7, T34C
- MC130E, MC130H, MC130W
- HC130H, HC130J, V22
- A400M, C130H, C130J, C160
- AN72, C20, C295, C37
- AH64E, AH1Z, TIGER
- AH1S, A129, MI28, T129
- CH47D, CH47J, CH53E
- A109,OH58,MD500
- BO105, SA319, SA342
- AH60L, MH53M, MH60K

- AH6G, MH47G, MH6
- HH60M, MH60R, UH60M
- AS332, EC225, EH101
- S70B, S76, SA365
- MI14, MI8, HH60J
- CH5, HARFANG, MQ1C
- RQ2A.PIONEER, RQ7B,SHADOW
- AS16, AS17, AS4
- AS4BSAP,KITCHEN
- ARABEEL50.APHE
- AGM129C, C611
- AGM84AB, C701R, AS10
- C801S, RBS15
- AGM84G, C704, GABRIEL
- AS16B.KICKBACK, AS6B. KINGFISH
- AGM45, AGM88D, ALARM
- AGM86D, AS18, DH10
- AGM123, AGM130, AGM65
- NSM.AL, NSM.SL
- DF4, DF5, DF41

C.12 Added Small Boat Types

- AAV_BMD-3.AT4
- AAV_BMD-3.AT5
- ARISGATOR

- ARISGATOR-M2
- ARISGATOR-M240
- ARISGATOR-MK19
- FLEET-USV
- HARBOR-USV
- K-61-AMPHIB
- LARC-V
- LVTP-5.M2
- LVTP-5.Mk19
- ORAC.XLUUV
- PTS
- PTS-2
- PTS-3
- PTS-4.12.7MM
- PTS-M

C.13 Added New Target Type Groups

- BRIDGE_TGG
- INTRDCT_PT_TTG
- NVL.AUX_TTG
- NVL.CARGO_TTG
- NVL.CARRIER_TTG
- NVL.CRUISER_TTG
- NVL.DDG_TTG
- NVL.FRIGATE_TTG

- NVL.INTEL_TTG
- NVL.LANDING_TTG
- NVL.LG.AMPH_TTG
- NVL.MINE_TTG
- NVL.PC-PB_TTG
- NVL.SERVICE_TTG
- NVL.SUB.DH_TTG
- NVL.SUB_TTG
- NVL.TANKER_TTG
- RUNWAY_TTG

C.14 Renamed Target Type Groups

- BOAT_RUBBER.BOAT_TTG renamed to BOAT_RUBBER_TTG

C.15 Added Surface-to-Surface Missile Types

- SK60.BAL(TEL)
- SS-14.UVLS(8)
- CJ10.VLS(6)
- CJ10.VLS(8)
- CM401.LNCHR
- DF-ZF.HGV.TEL
- DF11(CSS7)TEL
- DF11A(CSS7M2TEL)
- DF15(CSS6)TEL
- DF15A(CSS6M2TEL
- DF16(CSS11)TEL

- DF21(CSS5M1TEL
- DF21A(CSS5M2TEL
- DF21C(CSS5M4TEL
- DF21D(CSS5M5TEL
- DF26D(CSS18M3TEL
- DF26B(TEL1)
- DF3(CSS2)TEL
- DF31(CSS9)TEL
- DF31A(CSS9A)TEL
- DF4(CSS3)TEL
- DF41(CSSX10)TEL
- DF5(CSS4.M1)TEL
- DF5A(CSS4M2)TEL
- DH10(TEL)
- HYUNMOO2AB(TEL)
- HYUNMOO2C(TEL)
- HYUNMOO3AB(TEL)
- JL1.VLS(SUB)
- JL2.VLS(SUB)
- JY12(TEL2)
- KN-08(TEL)
- KN-11(TEL)
- KN-11.VLS(1)
- M11(TEL1)

- M7(CSS8)TEL
- P70.LNCHR(2)
- SSN-262.VLS(SUB)
- SYLVER.A70.8VLS
- UKSK.8VLS
- YJ12(TEL2)
- YJ18(NVL.LNCHR4
- YJ18A.SHIP.4VLS
- YJ18B.VLS(SUB)
- YJ62(NVL.LNCHR4
- YJ62(TEL3)
- YJ82.LNCHR(SUB)
- YJ83(NVL.LNCHR4

C.16 Created New Air Defense Classes:

- FL3000N.18L
- IRON.DOME.LAUNCHER
- PANTZYR-SA(SP)
- SA-25(MP)TM

C.17 Added New Airbases:

- Australia Airbase Units
- Canadian Airbase Units
- Indonesia Airbase Units
- New Zealand Airbase Units
- Singapore Airbase Units

- UK Airbase Units
- VANDENBERG.AFB_US

C.18 Created New Aircraft Classes:

- A50U.BERIEV
- AN22.COCK
- AP3C.ORION
- CH148.CYCLONE
- CN235.MPA
- DHC5A.BUFFALO
- EMB314.S.TUCANO
- F15SG
- FC31.GYRFALCON
- FOKKER.F50.MPA
- HAWK.MK109
- HAWK.MK127
- HAWK.MK209
- KC30A.MRTT
- L100-30.HERCULES
- MH139.GREY.WOLF
- MIG31K
- SU35S.FLANKER-E
- WING.LOONG.UAV
- Y20
- Z18

- Z18F.ASW

C.19 Renamed Aircraft Classes:

- A50.MAINSTAY to A50M.MAINSTAY
- TU22M.BACKFIREB to TU22M2
- TU22M.BACKFIREC to TU22M3

C.20 Aircraft Classes:

- Updated Aircraft Class - Control Mechanism Column to UNMANNED for all UAVs
- Verified Aircraft Class types CONTROL MECHANISM - UNMANNED PROBABILITY CREW SURVIVAL are set to 0

C.21 Created New AKL Tables:

- KH47M2.KINZHAL.HE_AKL
- SSN30A_AKL

C.22 Created New Combat Systems:

- BMD1KSH-VEH-LA
- BMD-4M.100MM-IFV.HA
- BTR-MDM.12.7MM-APC.HA.OW
- CWS.JAVELIN2_AT-HAW-LR-TA
- CWS.M134.7.62MM-CREW.WPN
- CWS.M2.12.7MM-CREW.WPN
- CWS.M240.7.62MM-CREW.WPN
- CWS.M249.5.56MM-CREW.WPN
- CWS.MK19.40MM-CREW.WPN
- PCL181.155M-ART.(SP)MHVO
- T80BVM.125MM-TANK.LFES

- NOTE: CWS stands for Containerized Weapon System

C.23 Added New Ground Units:

- 21SPACE.WING_US
- 30SPACE.WING_US
- 45SPACE.WING_US
- 460SPACE.WING_US
- 50SPACE.WING_US
- SCHRIEVER.AFB_US
- US.SPACE.OPS.CMD_US
- Australia Ground Units
- Canadian Ground Units
- Indonesia Ground Units
- New Zealand Ground Units
- Singapore Ground Units
- UK Ground Units

C.24 Created New Jammers:

- KG300G.RJ
- L175.KHIBINY_M.RDR.JAM

C.25 Created New Minefield Type:

- MK54.HAMMERHEAD

C.26 Added New Naval Units:

- CHAGAN.LAKE_CN (SUP - FUYU.(TYPE901)_CN)
- HAGURO_JP (SUP - MAYA.DDG_JP)
- HULUN.LAKE_CN (SUP - FUYU.(TYPE901)_CN)

- KANSAS.CITY_US (SUP - INDE.LCS.NSM_US)
- LHD1(TYPE75)_CN (SUP - TYPE75(YUSHEN).LHD_CN)
- LHD2(TYPE75)_CN (SUP - TYPE75(YUSHEN).LHD_CN)
- MAYA_JP (SUP - MAYA.DDG_JP)
- OAKLAND_US (SUP - INDE.LCS.NSM_US)
- VERMONT_US (SUP - VIRGINIA_US)
- ZIBO_CN (TUP - LUYANG.III.(TYPE052DL)_CN)
- Australia Naval Units
- Canadian Naval Units
- Indonesia Naval Units
- New Zealand Naval Units
- Singapore Naval Units
- UK Naval Units

C.27 Created New PH Tables:

- F21.533TP_PH
- KH47M2.KINZHAL.HE_PH
- MN.MK54_PH
- SSN30A_PH
- TYPE12.SSM_PH

C.28 Created New PK Tables:

- F21.533TP_PK
- KH47M2.KINZHAL.HE_PK
- MN.MK54_PK
- SSN30A_PK

- TYPE12.SSM_PK

C.29 Added New Runway Target:

- KVBG.RWY12-30 (VANDENBEG.AFB)

C.30 Created New SAL Tables:

- SA-25_SAL
- TAMIR_SAL

C.31 Created New Sensors:

- APS154.AAS_AGR
- DB2000_VGN
- FLAT.FACE.B.(P-19)_LAA
- IRBIS-E.AESA_AAD
- IRBIS-E.AESA_AGK
- JY26.METER.WAVE_LAA
- JY27A.METER.WAVE_LAA
- KH-SHARPEYE_VGN
- KNIFE.REST.A.(P-8)_LAA
- METER.WAVE(STEALTH)_SAA
- OLS35.IRST_AAD
- POZITIV-ME(5P-26)_SAA
- POZITIV-ME(5P-26)_SGN
- SCANTER4100_SAA
- SCANTER4100_SGN
- SCANTER6002_SAA
- SCANTER6002_SGN

- SHMEL.II_AEE
- SPOON.REST.D.(P-18)_LAA
- SPY-6_SAA
- SPY-6_SGN
- THALES.NS-100_SAA
- THALES.NS-100_SGN
- TRS-4D_SAA
- TRS-4D_SGN
- TYPE382.SEA.EAGLE_SAA
- TYPE382.SEA.EAGLE_SGN
- TYPE760_VGN

C.32 Created New SKL Tables:

- F21.533.TP_SKL
- KH47M2.KINZHAL_SKL
- MN.MK54_SKL
- SSN30A_SKL
- TYPE12.SSM_SKL

C.33 Added New Squadron Units:

- 44.RS_US
- Australia Squadron Units
- Canadian Squadron Units
- Indonesia Squadron Units
- New Zealand Squadron Units
- Singapore Squadron Units

- UK Squadron Units

C.34 Created New SSM:

- TYPE07-VL-ASROC.VLS(8)
- TYPE12.SSM(TEL6)

C.35 Added New Support Units:

- Australia Support Units
- Canadian Support Units
- Indonesia Support Units
- New Zealand Support Units
- Singapore Support Units
- UK Support Units

C.36 Added TW to SSM:

- MDCN to SSM - TT533SI.WIRE
- SSN33(3M22.ZIRCON) to SSM - BASTION-P(TEL)
- SSN33(3M22.ZIRCON) to SSM - UKSK.8VLS

C.37 Renamed SSM:

- 3S-14.8VLS to UKSK-M.8VLS
- UKSK to UKSK.3S14.8VLS

C.38 Created New SUPs:

- ALEXANDRIT.MHS_RU
- ANADOLU.LHD_TR
- ARCHER_SG
- BEDOK_SG
- BUYAN-M_RU

- CAPE_AU
- CHALLENGER_SG
- DYUGON.LCU_RU
- F125.(BADEN.WURT)_DE
- FUCHI.II.(TYPE903A)_CN
- FUYU.(TYPE 901)_CN
- GRACHONOK.PB_RU
- INDEPENDENCE(LMV)_SG
- INVINCIBLE_SG
- IVAN.GREN.LST_RU
- JIANGKAI.II(54A+)_CN
- KARAKURT_RU
- KINGSTON_CA
- LUYANG.III.(TYPE052DL)_CN
- MAYA.DDG_JP
- RAPTOR.PB_RU
- RIVER.BATCH2_UK
- SHANDONG_CN (Aircraft Carrier)
- SIRIUS_AU
- SUFFREN.SSN_FR
- SYCAMORE.MATV_AU
- TYPE.75(YUSHEN).LHD_CN
- YASEN_RU

C.39 Created New Supply Categories:

- CL.V.AS.KH47M2.KINZHAL
- CL.V.AS-RDR-LR-TYPE12
- CL.V.MINE-MK54.HAMMERHEAD
- CL.V.SA-LR-TAMIR
- CL.V.SA-SR-SA-25
- CL.V.SS_RDR-MR-SSN30A
- CL.V.SS_RDR-MR-SSN30A(T)
- CL.V.SS-PRSM
- CL.V.SS-RDR-LR-SSM2
- CL.V.SS-RDR-LR-TYPE12
- CL.V.TORP-324N-GRX4.AL
- CL.V.TORP-324N-GRX4.SL
- CL.V.TORP533SN-F21

C.40 Updated the Supply Category Table Special Capability Column for:

- CL.IV.BARRIER (Special Capability - Barrier)
- CL.IX.BRIDGING.REPAIR (Special Capability - Bridging)
- NON.COMBATANTS (Special Capability - Personnel)

C.41 Renamed Supply Categories:

- CL.V.SA-FL3000N to CL.V.SA-SR-FL3000N
- CL.V.SS-MDCN to CL.V.SS-LR-MDCN

C.42 Created New TUPs:

- AIRBASE.NO.RUNWAY_US
- Australia TUPs

- Canadian TUPs
- Indonesia TUPs
- New Zealand TUPs
- Singapore TUPs
- UK TUPs

C.43 Updated TUPs:

- Added Supply Category - CL.IX.BRIDGING.REPAIR to All TUPs checked as Bridge Capable

C.44 Created New TWs:

- F21.533STP
- KH47M2.KINZHAL.AL
- MN.MK54.HAMMERHEAD
- PRSM.PRECISION.STRIKE.MSL
- SA-22 (57E6)-SA
- SA-25.9K333.VERBA
- SSN-30A(3M14)
- SSN-30A(3M14T)
- TAMIR.(IRON.DOME)
- TYPE12.AL.ANTISHIP
- TYPE12.SSM

C.45 Renamed TWs:

- DF-ZF.HGV to DF-ZF.HGV(DF17)
- GRX4.AL324TP to GRX4.(TYPE97)AL324TP
- GRX4.SL324TP to GRX4.(TYPE97)SL324TP
- SSM2 to SSM2.TYPE17

- SSN-26(3M55)YAKHONT to SSN-26(3M55)STROBILE

C.46 Updated TW Table Column:

- IMPACT NOISE LEVEL column for all Torpedoes

C.47 Created New UOM's:

- F21.533STP
- GRX4.324TP.SL
- KH47M2.KINZHAL
- MACH
- MN.MK54.HAMMERHEAD
- PRSM
- SA-25.9K333.VERBA
- SSN30A(3M14)
- SSN30A(3M14T)
- TAMIR.(IRON.DOME)
- TYPE12.AL
- TYPE12.SSM

C.48 Renamed UOM's:

- SSM2 to SSM2.TYPE17
- GRX4.324TP to GRX4.324TP.AL

C.49 Created New Weapon Loads:

- 1KH47M2.KINZHAL
- 2A84CX2A244
- 2BLUE.ARROW7X2LT2
- 31AGM183A.ARRW

- 4AGM183A.ARRW
- 4KH47M2.KINZHAL
- A50U.DEF
- EMB314.DEF
- F15SG.DEF
- FOKKER.F50-MPA.DEF
- SU35S.DEF
- TRANSFER.NO.WEAPONS
- WING.LOONG.UAV.DEF

C.50 Renamed Weapon Loads:

- TU22M-B.DEF to TU22M2-B.DEF
- TU22M-C.DEF to TU22M3-C.DEF

C.51 Added Hypersonic Weapons to:

- (4) SSN33(3M22.ZIRCON) on SUP - GORSHKOV_RU
- (8) SSN33(3M22.ZIRCON) on SUP - KIROV_RU
- 1KH47M2.KINZHAL on MIG31K
- 4KH47M2.KINZHAL on TU22M3M
- 4AGM183A.ARRW on B52H.STRATOFORT
- 31AGM183A.ARRW on B1B.LANCER

C.52 General Improvements

- Continued to add Units to Improve Chinese and Russian OB: Added Ground, Naval, Support, Squadron, HRUs and Airbase Units