JTLS-GO

Version Description Document

March 2023

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 6.2.0.0)

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

JTLS-GO 6.2.0.0 is a Major release of the JTLS-GO 6.2 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 LLC
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940 USA
jtlsgo@rolands.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 6.2.0.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 6.2.0.0 is a Major delivery for the JTLS-GO 6.2 series of releases.

JTLS-GO 6.2.0.0 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 “wespac62”. 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 6.2.0.0 executes on the Red Hat Enterprise Linux Version 8.7 64-bit operating systems. The Web-Hosted Interface Program (WHIP®) user workstation interface can be executed on any operating system from any Java-compatible Web browser.

1.2 INVENTORY OF MATERIALS

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

1.2.1 Obsolete/Outdated Documents

No documents have been deleted or become outdated as a result of this release.

1.2.2 Unchanged Documents

JTLS-GO 6.2.0.0 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 6.2.0.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 03, Version 6.2.0.0)
- *JTLS-GO Configuration Management Plan* (JTLS-GO Document 03, Version 6.2.0.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 6.2.0.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 6.2.0.0)

- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 6.2.0.0)
- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 6.2.0.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 6.2.0.0)
- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 6.2.0.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 6.2.0.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 6.2.0.0)
- *JTLS-GO Standard Database Description* (JTLS-GO Document 14, Version 6.2.0.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 6.2.0.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 6.2.0.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 6.2.0.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 6.2.0.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 6.2.0.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 6.2.0.0)
- *JTLS-GO DoD Architecture Framework* (JTLS-GO Document 22, Version 6.2.0.0)

1.2.4 New Documents

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

1.2.5 Delivered Software Components

JTLS-GO 6.2.0.0 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 188.

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

    Apache Server (APACHE)

    JTLS XML Serial Repository (JXSR)

    Order Management Authority (OMA)

    Synchronized Authentication and Preferences Service (SYNAPSE)

    XML Message Service (XMS)

    Total Recall Interactive Playback Program (TRIPP)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI) for both OTH-Gold and Link-16 generation

- Tactical Electronic Intelligence (TACELINT) Message Service
- Keyhole Markup Language (KML) Operational Interface (KOI)
- JTLS Transaction Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- JTLS Geographic Information System (GIS) Terrain Building Program
- JTLS Master Integrated Database (MIDB) Tool
- JTLS Version Conversion Program (VCP)

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

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

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

Instructions for installing JTLS-GO 6.2.0.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 6.2.0.0 is not necessary. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS-GO.

The basics of installation have not changed significantly, but due to many Cyber-Security improvements, a new Linux RPM package named "xerxex-c", is required to run JTLS-GO 6.2.0.0. The Synapse will not function without this package. Prior to installing JTLS-GO 6.2.0.0, please run the RPM checking script delivered with JTLS-GO to ensure that this package and all other packages are installed as part of your Linux operating system.

1.2.6 Released Databases

This release includes the following sample unclassified databases:

- The scenario that serves as a repository of engineering level data called “repository62”. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your PostgreSQL scenario table-space.
- The scenario “wespac62”, which is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 6.2.0.0 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 8.7, 64-bit architecture.

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

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

  Oracle Linux 8.7 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

- There are no restrictions on the operating system for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 6.2.0.0 has been tested on the following operating systems:

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

  Oracle Linux 8.4 and 8.7

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

- JTLS-GO 6.2.0.0 is delivered with the Adoptium project Temurin Java Development Kit (JDK) 1.8 Update 362 package, which is equivalent to the current version of OpenJDK.

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

- JTLS-GO database tools require a certified PostgreSQL 11.19 database server and the full PostgreSQL installation. A containerized solution, that fulfills this specification, is provided as part of the JTLS-GO download. It is not necessary to use the delivered containerized solution, but it is the easiest method to meet the requirements of JTLS-GO 6.2.0.0. There are several alternative methods available for obtaining the PostgreSQL 11.19 software. Refer to Chapter 6 of the *JTLS-GO Installation Manual* for additional installation details.

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

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

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

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

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

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

- The JTLS-GO DDS application uses these open source libraries:

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

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

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

- KML Operational Interface (KOI)

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

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

1.3.2 JTLS-GO Cybersecurity Compliance

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

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

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

- Within two-weeks of the Adoptium release, JTLS-GO provides a bug release version (JTLS-GO 6.2.n.0) including a full Version Description Document (VDD) for download to all authorized agencies. All DoD agencies using JTLS-GO will be in full compliance with this specific Cybersecurity mandate as long as they download and use the bug released versions when distributed.

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

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

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

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

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

1.3.3 JTLS-GO High Level Architecture Compliance

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

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 6.2.0.0 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

Several database structure differences exist between JTLS-GO 6.2.0.0 and the previous JTLS-GO 6.1 series database structure.

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

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 6.2.0.0 formatted data from your PostgreSQL database server into the JTLS-GO 6.2.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in the *JTLS-GO DDS User Guide*, Appendix B.11. This procedure will reorganize the structure of the <scenario_name>.gs and databases symbol.scf file.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 6.2.0.0, 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. No symbols changes have been made to JTLS-GO 6.2.0.0. The symbols available in JTLS-GO 6.2.0.0 are identical to the symbols available in the JTLS-GO 6.1 software series. This means that the user should not need to make any symbol changes when converting from a JTLS-GO 6.1 series scenario to a scenario usable by JTLS-GO 6.2.0.0. If an organization has made their own symbol changes, 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 “repository62”. The DDS comparison and synchronization function can be used to determine if any of the changes delivered are of use to a JTLS-GO user organization.

1.5 INSTALLATION

The *JTLS-GO Installation Manual,* a Portable Document Format (pdf) file available for direct download, is part of this JTLS-GO delivery, It provides detailed instructions for installing the new version of JTLS-GO and the installation of PostgreSQL 11.19 required to operate JTLS-GO 6.2.0.0.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS-GO 6.2.0.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2014-0086 Allow DDS Users To Change Unit Type

Summary of Model Change Request

Database Development System (DDS) users should be able to easily change a Unit's type or a Target's category. Currently, in order to change a ground Unit into a support Unit, the Unit must be deleted from the Ground Unit table first, and then recreated in the Support Unit table. Similarly, changing a sensor Target into a communications Target requires deleting the Target from the Sensor Target table, and then recreating it in the Communication Target table.

When recreating a Unit or Target, the user must reenter most of the same parameters (name, long name, location, etc.), making the whole process tedious. Ideally, the user should be able to simply select an option to initiate the change, allowing the DDS to handle the process.

Design Summary

In the JTLS-GO database, Units are organized into six separate tables, based on type, and Targets are organized into twenty-one separate tables, based on category. This design makes no effort to modify the database, but adds three options to automate the process of moving a Unit or Target record from one table to a different table:

- Users can start the process from the individual object tables (for example, the Airbase Unit table or the Communication Site Target table);
- Users can start the process from the "All Unit" and "All Target" tables;
- Users can start the process from the individual Unit or Target's context sensitive menu.

2.2 JTLS-2015-12489 Too Easy To Delete All DSAs

Summary of Model Change Request

The current capability to delete Directed Search Areas (DSAs) in JTLS-GO has limitations. It only allows the Player to delete either one DSA or **all** of the DSAs in the game. The Player should have the option to delete multiple specific DSAs at once.

Design Summary

The Manage DSA order has a button which allows the Player to delete all of the existing DSAs. If the DSAs were deleted during a critical juncture in an exercise, this could be disastrous for

intelligence collection. This ECP allows the user to delete multiple DSAs using a new DSA Information Management Tool (IMT) screen.

2.3 JTLS-2016-12595 IMT Objects By Specified Geographic Region

Summary of Model Change Request

It would be valuable to display Information Management Tool (IMT) data for objects in a specific geographic region. An IMT menu that allows the user to select a rectangular area on the Web Hosted Interface Program (WHIP) map window was added. When the box is drawn, the appropriate records are displayed in the IMT screen.

Design Summary

IMT screens are currently started from the WHIP's IMT menu. Once the desired IMT is selected, the IMT screen is displayed and lists all units, targets, or other objects that meet the IMT filtering criteria in a table. The current IMT filters do not allow users to filter the displayed table objects by geographic region. This ECP adds such a capability, and adopts the same method of filtering table rows by geographic region in DDS tables.

This ECP adds two new buttons to the Navigator Tools found on the left-hand margin of the WHIP Map window. One button activates the drawing of a geographic region as a polygon over the Map by a series of user-selected points. The second button opens a window that lists the user-created and shared geographic regions. From this window, users are able to share and remove the regions that they have created and own.

When opening a geographic region-capable IMT screen or DDS table, users are able to select the desired Region or Regions from the available defined regions to use for filtering the IMT screen or DDS table by the enclosed objects.

2.4 JTLS-2016-12634 Allow Modifications to Directed Search Areas

Summary of Model Change Request

The order used to manage DSAs in JTLS-GO is cumbersome and incomplete. The current capability only allows Players to create and delete DSAs. There is no functionality to modify an existing DSA.

Because DSAs hold history data concerning what has previously been detected within the DSA, it is important to not unnecessarily delete a DSA and lose this historical collection information.

Design Summary

This design overhauls the structure of the Manage DSA order to streamline the current usage (Create, Delete), and implement the capability to modify DSAs.

A better, more logical layout for the Manage DSA order was implemented to enable Players to efficiently modify DSAs. The ECP accomplishes two tasks:

1. Streamline the Manage DSA order panel.

2. Implement the capability to modify the properties of existing DSAs.

The end result is an order which allows the Player to easily modify the properties of DSAs. Existing capabilities (creating and deleting DSAs) continue to be part of the same order panel.

2.5 JTLS-2019-14545 Airspace Representation

Summary of Model Change Request

Enable the use of airspaces defined in an Airspace Control Order (ACO) to be referenced and used by Players when building JTLS-GO air missions.

Design Summary

This ECP allows JTLS-GO to translate a real-world Airspace Control Order (ACO), and has added an order to create Airspace Control Mean (ACM) areas within the model. The ACO has a unique identifier for each ACM, which is known as an ACM Identifier (ACMID). ACMIDs are used by air missions, air defense sites, naval units, and land combat units.

Various JTLS-GO orders will now accept tasks that refer to ACMIDs, instead of using latitude and longitude information.

2.6 JTLS-2020-15043 New DDS Logistics Hierarchy Report

Summary of Model Change Request

Develop a Logistics Hierarchy Report that can be submitted from the DDS to list the various Logistics Hierarchies, and the truck and tanker assets at each Support Unit within the hierarchy. The format of the report should make it easier to review and approve database-assigned assets.

Design Summary

The DDS currently provides a Unit Vehicle Report, which provides a breakdown of Cargo, Tanker, and Heavy Equipment Transport (HET) assets for individual sides, all sides, or a faction. The report does not provide a breakdown of assets by specific Combat System or a logistics hierarchy of assets. This ECP modifies the current Unit Vehicle Report so that specific assets will be listed, and the user may select to view assets for a specific logistics hierarchy.

Although the ECP does not specifically mention supplies, it was the feeling of the design team that a list of supplies, including supplies at individual units and a complete rolled-up list, was also desired for a logistics hierarchy. To accomplish that, this ECP implemented a new Logistics Roll-Up Report, that lists the hierarchy and supplies available at each unit within the hierarchy.

2.7 JTLS-2020-15046 Include Aircraft In Tactical Intel Reports

Summary of Model Change Request

Aggregate Resolution Units (ARUs) and High Resolution Units (HRUs) provide Tactical Intelligence Reports. These reports represent the information Units and HRUs gather on a continual basis, and include Convoys, Units, and HRUs coming into and out of a Unit's or HRU's area of operations. This ECP requests that aircraft information be added to Tactical Intel Reports.

Design Summary

Currently, JTLS-GO provides Tactical Intelligence information for Units, Targets, HRUs, and Convoys.

Tactical Intelligence reports include detection information picked up by HRU or ARU Lookouts. By default, both HRUs and ARUs are given a visual sensor from the INTELLIGENCE INFORMATION PROTOTYPE, which is the IIP VISUAL SENSOR. This is the default sensor the model uses when processing HRU and ARU lookout detections. If the Unit or HRU has a stronger or longer range visual sensor than the one listed by the IIP, then the stronger sensor is selected.

This ECP renames IIP VISUAL SENSOR to IIP GROUND VISUAL SENSOR and adds an IIP AIR VISUAL SENSOR. As implied, the IIP GROUND VISUAL SENSOR is used to process ground lookout detections, and the IIP AIR VISUAL SENSOR is used to process air lookout detections. Any air mission that takes off, lands, or flies through a unit's detection range is reported as part of its Tactical Intelligence Information report.

2.8 JTLS-2021-15210 Imaging Of Chemical Or Nuclear Impacts

Summary of Model Change Request

JTLS-GO should have an easy way to automatically display contamination areas in the WHIP Map when chemical and nuclear events occur in the game. This can be a switch in the Contamination filters to automatically show areas of impact as they occur.

Design Summary

In order to see chemical and nuclear impact sites on the WHIP Map, users are currently required to turn on the chemical and nuclear events from the Contamination filters by checking their corresponding checkboxes in the filter table. From the Map alone, a user cannot tell whether such an event occurred without navigating to the Contamination filter to see if any exist. If one does exist, the user is then required to select it "on" to see it displayed on the map.

The design adds a means to automatically show the impact zones of these events. This ECP took a broad look at the current filter set and implemented a similar mechanism for other filters that would also benefit from the capability.

2.9 JTLS-2021-15214 Include Maximum Arc Size In GIS Tool

Summary of Model Change Request

The JTLS-GO Geographic Information System (GIS) Tool provides a capability to “trim” a network, by combining smaller arcs, to reduce the overall number of arcs and nodes in the network. A capability that does the opposite is desired: If an arc is too long, the arc will be divided to create multiple arcs.

This is needed because objects can only get onto a network at nodes. If nodes are too widely separated by long arcs, the object will not be close enough to a node, and therefore will not be able to get on the network. Dividing long arcs into smaller arcs will create additional nodes that will give the object a better chance to enter the network.

Design Summary

A Road, River or Rail network is currently trimmed by specifying the Minimum Arc Length and Maximum Deviation. These are entered in the Enter Node Reduction Constraints dialog. Sea Lane and Air Corridor networks are currently not trimmed because their nodes tend to be far apart (50 KM or more).

For Road, River and Rail networks, the new desired capability requires a new Maximum Arc Length field to be added to the dialog. As a result of adding the new field, the dialog has been renamed the Merge/Divide Arc Constraints dialog. The trimming process is expanded to take into account the constraint.

As stated previously, Sea Lane and Air Corridor arcs tend to be long, so combining arcs to reduce the number of nodes and arcs in these networks is not desirable. However, there is a need to address the inability of objects to access either the sea or air network because the arcs are too long. For these networks, a new Divide Arc Constraint dialog has been created. The dialog contains only one field, called “Maximum Arc Length”.

2.10 JTLS-2021-15458 Link 16 J-Message Filters

Summary of Model Change Request

The Link 16 Message Service (L16MS) filtering should be expanded to have an additional filter setting. Operators need to configure L16MS to send only specific J-Messages. These settings will allow track filtering by generic J-Message type.

Design Summary

A new J-Message filter panel has been added to the L16MS WHIP module for filtering by J-Message type. This panel contains a list of supported Link 16 message types.

This filter group operates as a generic filter (such as the Over-The-Horizon Gold Message Service) that may only be overridden by track-specific filters in the track summary table. There are several generic filters, including filtering by side, object class (air, ground, naval, etc.), aircraft class, and more. Those filters are also overridden only by track-specific filters in the track summary table.

2.11 JTLS-2021-15459 Display OTH-Gold Track Time-Late Values

Summary of Model Change Request

In JTLS-GO, the Over-The-Horizon Gold (OTH-Gold) message system operates as a difference-based communication system. Updates for tracks are only processed when something has changed with that track.

One of the most important pieces of data that may be updated is a track's time of last detection. If a track has not been detected within a certain time margin, it is considered to be "time-late" and may be dropped from the Common Operational Picture (COP). The ability to monitor and perform maintenance on time-late tracks is therefore important to exercise staff. The purpose of this ECP is to provide new ways for operators to monitor track time-late data.

Design Summary

The new time-late implementation adds a column to each track summary table. This new column displays each track's time since its last update, or time-late value. If a track is considered to be time-late, this table cell is painted with a gray background. Users are able to sort the table by the time-late column.

2.12 JTLS-2021-15460 Link 16 Air PPLIs Report Self As Tracks

Summary of Model Change Request

JTLS-GO represents the setting of a Special Interest Flag for Link 16 tracks. The Special Interest Indicator exists in Link 16 only on J3.2 messages, or air tracks. This flag is not available on the J2.2 message, known as a self-reporting mission or PPLI.

In JTLS-GO, PPLIs only generate J2.2 messages for themselves, and not a J3.2 message. There is no way for PPLIs to be marked with a Special Interest Indicator by an operator.

PPLIs should be able to maintain a Link 16 Special Interest Indicator. In addition, the L16MS must be able to indicate to the user that the Link 16 Special Interest Indicator is active for a track.

Design Summary

Operators are now allowed to set Special Interest Indicators on PPLI missions by allowing PPLIs to generate J2.2 and J3.2 messages for themselves simultaneously. This setting is available as a

"Self-Tracking Air PPLIs" check box on the Message Configuration panel. This check box is deselected by default to match current behavior.

For a PPLI to send both a J2.2 and J3.2 message for itself, the Combat Events Program (CEP) must fill all of the required JU numbers to produce those messages. The L16MS performs this check when processing Air Missions to determine if the object is eligible to be sent as both types of messages. This functionality applies only to Air objects, and not Ground or Naval objects.

This ECP was implemented on the direction of the North Atlantic Treaty Organization (NATO). In discussions with US Air Force agencies, the concept of publishing both J2.2 and J2.3 for self-reporters is not consistent with current operations. US organizations should strongly consider the consequences of using this new flag during exercise operations.

2.13 JTLS-2022-15858 Convoy Conditionally Enter Enemy Unit Radius

Summary of Model Change Request

This ECP enhances the CEP to enable Convoys to enter or pass through the radius of an enemy or suspect unit under certain conditions.

The model currently halts a Convoy if its next move would place it inside the radius of an enemy or suspect unit. This criterion will be enhanced with an assessment based on the level of risk the Convoy is willing to accept as determined by a new Supply Logistics Prototype (SLP) database parameter.

Design Summary

The Convoy movement algorithm considers the threat posed by nearby enemy/suspect units in the next location before the Convoy moves to its new location. The criteria for assessing the threat from each unit is identical and considers the following:

- The unit's relationship with the side of the convoy must be either Enemy or Suspect.
- The unit's posture must be either Attack, Defend, or Delay. When in any other posture, the unit is not considered a threat. The posture of Hasty Defense was considered, and the Design felt that while setting up a Defensive posture, the unit would be too busy to consider a convoy within its footprint.
- The unit's Rules of Engagement (ROE) Ground Combat Range must be greater than or equal to the distance between the unit's center of mass and the Convoy location.
- The unit's radius covers the Convoy.

If all of the above criteria are met, the unit is considered a threat and the Convoy is susceptible to losses. The Convoy's pending move is postponed and a Ground Wait task is applied. The

responsible Player then must decide how to modify the Convoy's route to avoid the threat. If no nearby units meet all of the criteria, the Convoy is allowed to complete its next move.

After the next move is completed, the same criteria are again considered because the situation may have changed. For example, an enemy unit may have moved closer to the Convoy in the interim and although the Convoy was safe prior the move, it is no longer safe. Again, if all of the criteria are met, damage is assessed against the Convoy using the closest unit. This portion of the existing logic has not changed. The Convoy has moved into danger, and the assessment will be accomplished.

This design only considers the pre-move algorithm and improves the decision-making rules used by the Convoy to determine if it should start a move.

The ECP made the fourth criterion less restrictive. Instead of simply preventing the Convoy from entering the radius, the logic now conditionally allows the Convoy to enter if the calculated Probability of Kill (Pk) is less than the new SLP database parameter SLP IGNORE ENEMY PROB. This allows the scenario designer to manage the likelihood that a Convoy might traverse an enemy unit's radius according to the level of acceptable risk represented by SLP IGNORE ENEMY PROB.

2.14 JTLS-2022-15872 Allow Artillery Order On Unit Order Menu

Summary of Model Change Request

The WHIP provides a context sensitive menu, where a list of object-specific orders are available to open from the orders sub-menu.

The Fire Artillery order used the unit's prototype caliber attribute to determine if the unit was capable of firing artillery munitions. The caliber attribute was removed several years ago, and the Fire Artillery order could no longer be accessed from the context-sensitive object menu. This ECP allows the order to use the unit's *artillery_capability* array attribute to determine whether the Fire Artillery order should be included in the context sensitive menu for a selected unit.

Design Summary

The game now looks at the unit's artillery capability array to determine whether to enable the listing of the Fire Artillery order in the unit's context sensitive menu. Once the user selects the Fire Artillery order from the object's order list, the order will open with the unit field pre-filled with the selected unit's name.

2.15 JTLS-2022-15968 Limit Web Access To Needed Files

Summary of Model Change Request

The WHIP currently has access to all of the files in the $JTLSHOME/game/data directory, because a link to that directory is placed under a scenario's webroot directory. Although all files

within this directory are unclassified, only those files needed by the system should be accessible from a scenario's webroot directory.

Design Summary

During the scenario setup procedure, the entire $JGAME/<scenario_name> directory structure is created. One of the directories within this structure is called “webroot”. Any files that need to be accessed by the WHIP or the WHIP modules via Apache must exist in “webroot”. However, all configuration managed scenario independent data is maintained in a directory called $JGAME/data. Everything within the $JGAME/data directory is unclassified; therefore, a link was placed under the $JGAME/<scenario_name>/webroot directory to the $JGAME/data directory. This ensured that all needed configuration managed files were available to the WHIP. the Cyber-Security Team did not approve of the fact that unneeded files were generally available to the WHIP.

This ECP evaluated all the files within the $JGAME/data directory. Instead of creating a single all-inclusive link to the full directory, links to only the subdirectories and files needed by the WHIP and WHIP modules are now created. No changes to code were required to implement this design, because the directory location and names for the needed files were not change as a result of this evaluation.

As part of this evaluation, the location of the generic jtls_ca certificate was moved:

- From the **$JGAME/data/certificate** directory,
- To the **$JTLSHOME/lib/security/certificate** directory.

Follow the instructions in Chapter 6 of the *JTLS-GO Installation Manual* to apply the new jtls_ca certificates to your PostgreSQL server.

If you are using the same PostgreSQL server for both JTLS-GO 6.1 and JTLS-G 6.2.0.0, the new certificates need to be copied from the JTLS-GO 6.2.0.0 **$JTLSHOME/lib/security/certificate** directory to the JTLS-GO 6.1 **$JTLSHOME/game/data/certificate** directory.

2.16 JTLS-2022-15975 Stop Arbitrary File Upload and Delete

Summary of Model Change Request

This ECP has resulted in a major change in the manner in which Public Key Infrastructure is handled within JTLS-GO. There were two issues to which the Cyber-Security Team objected. These two issues were:

- It was possible for a user to upload any non-scanned picture file when creating a new slide.
- It was possible to spoofed the system and have an Insider Threat look like a WHIP and upload or delete files from the JTLS-GO.

Design Summary

These two issues were addressed as follows;

- The ability for the WHIP user to upload a picture when creating a slide was removed from the WHIP. The user can still specify a picture to include on a slide, but the picture must exist on the server. It is up to network system personnel to properly vet and scan any pictures needed for slide creation before putting the picture files on the server.

- The second issue was much more difficult. JTLS-2022-15992 Prevent Unauthorized Command Injections, Section 2.18 already addresses the issue of checking to ensure that files placed on the system meet all system requirements and expectations. What JTLS-2022015992 does not address is how to stop an Insider Threat from spoofing the system into thinking they are a valid WHIP.

  JTLS-GO already has the capability of limiting user access to Apache based on a Public Key Infrastructure physical chip-based card. Unfortunate it is difficult for small organizations to setup and manage physical card-based PKI capability. JTLS-GO 6.2 no longer supports physical card-based PKI; instead, it supports a software certificate file based system. No extra hardware is required to implement the software certificate PKI capability. Any user organization can setup PKI and ensure only authorized users have access to the JTLS-GO Apache and thus the JTLS-GO servers.

As before, users do not need to implement PKI limitations on their systems. It is a security management decision and is configurable for scenario. If PKI is used, several new steps are required as part of the Interface Control Program (ICP).

2.17 JTLS-2022-15976 Encrypt Passwords To Start WHIP/DDSC

Summary of Model Change Request

WHIP passwords have been stored in plain text and WHIP users have no ability to change their own passwords. This is poor security practice. This ECP rectifies these two issues by only storing hashed versions of user passwords, and by providing users the ability to change their own password.

Design Summary

This ECP introduces a new WHIP database file for storing all WHIP-related configuration data. It also introduces a new tool for verifying and hashing WHIP passwords. Passwords stored in the WHIP database file are hashed and stored along with the algorithm used to generate the password hash.

The Interface Configuration Program (ICP) now:

- Allows Tech Control to set a common default password for all WHIPs;

- Allows Tech Control to set a unique initial password for any WHIP;
- Allows Tech Control to reset a forgotten or compromised user password back to the default password, or to a unique password, and;
- Reads, writes, and monitors the WHIP database file in a manner similar to the existing JTLS 6.1 master ICP database file.

When a user logs into a WHIP for the first time, the user must enter the default password. The Synapse is responsible for ensuring that the entered password, once hashed, matches the hashed version of the password created by Tech Control. After this successful first login, users are prompted to reset their password. Even Tech Control cannot view or see the new selected User password. It is unique to each user, and should be treated as any other security password that should not be revealed to others.

The Synapse makes calls to the password hashing and validation tool to validate user credentials for login requests and update passwords for reset requests.

The new password hashing and verification tool is Java based. It implements a consolidated set of library code for hashing and validating passwords. It also serves as a wrapper for the ICP WHIP database, so that native ICP code can be called to read, modify, and write the WHIP database.

2.18 JTLS-2022-15992 Prevent Unauthorized Command Injections

Summary of Model Change Request

During Game play, the WHIP sends files through Apache to the Synapse, and the Synapse places files on the server. These files include saved game orders, slides, saved views, and WHIP configuration files. Only authorized files that follow a specific format should be allowed on the system. This will prevent the possibility that an insider threat will place a file on the server that contains system commands capable of doing harm to the JTLS-GO server file system.

Design Summary

The WHIP is specifically designed to not have any information on the client machine. This design was implemented primarily for the following reasons:

- JTLS-GO wanted to give each user the freedom to log on from any client machine and still have their WHIP/TRIPP fully configured according to the user's specification. This may be necessary because of a client machine's hardware failure, or communication interruption from a remote client site. This flexibility is useful and expected by the entire JTLS-GO community.

- By saving all exercise-related data on the JTLS-GO servers, System Administrators can conduct appropriate backup and save procedures to ensure exercise data, orders, and information are saved during the exercise and can be recalled at a later date for a follow-on exercise.

Figure 2.1 shows how the WHIP sends data files to the JTLS-GO Server. All WHIP communication goes through the Apache Server. When Apache receives a request to "PUT" a file onto the server, the request is sent to the Synapse, and the Synapse places the data file in the appropriate directory structure location.

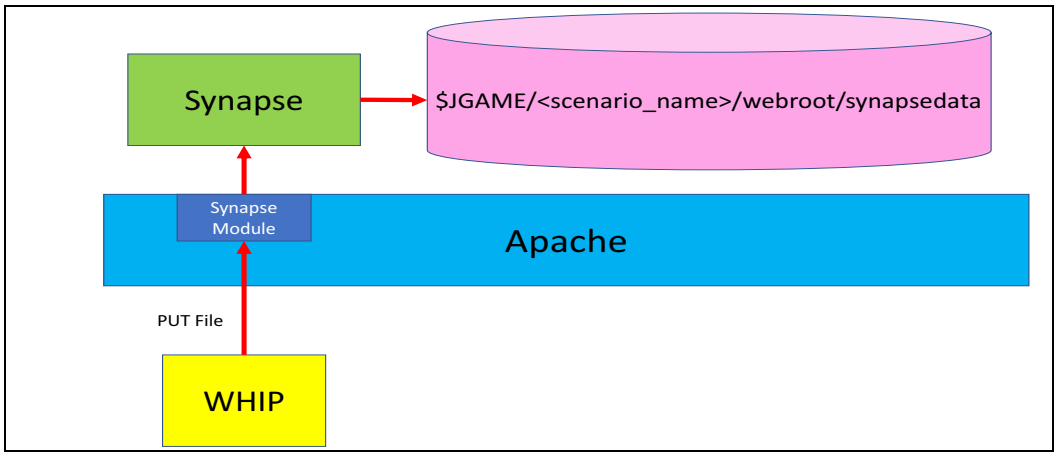

Figure 2.1 Basic WHIP File Placement Diagram

This ECP keeps the basic file placement paradigm in effect but requires the Synapse to check all files placed on the system for a proper format to ensure that no errant system commands or other undesirable contents are placed on the JTLS-GO server. The majority of files placed by the WHIP are Extended Markup Language (XML) Files, and the Synapse has been given an XML schema to check that each file properly follows the prescribed format.

This design only discusses the files that are placed on the server by the WHIP or TRIPP. Files created by services running the on various JTLS-GO servers are not considered as part of this ECP since these files pose no security hazard to the integrity of the JTLS-GO system

2.19 JTLS-2022-16013 HRU Interim Reports

Summary of Model Change Request

An HRU collects intelligence data and reports this information periodically based on a database parameter. If a piece of collected data meets the assigned Essential Elements of Information (EEI), the HRU generates a report immediately. If no EEI information is collected, the information collected by the HRU is not available to the Player until the scheduled report time. A trusted Player should be allowed to request an interim HRU report to determine what has been collected and is waiting to be reported.

Design Summary

A consolidated order allows the Player to request interim reports for HRU and for Air Missions, and Units.

- For Air Missions, the Player can request an interim mission report or a current reconnaissance report for detections held by the mission.
- For units, the Player can request a tactical intelligence report.
- For HRUs, the Player can request a current report on intelligence collection made by the HRU.

Each of these reports may be generated from a single order. The mission report order has been expanded to include options for units and HRU.

2.20 JTLS-2022-16016 Create Automated RIC Update Capability

Summary of Model Change Request

The JTLS-GO database contains Reportable Identification Codes (RICs) used by NATO for their Logistics Functional Area Services (LOGFAS). RICs change every year and NATO publishes a change list. JTLS-GO needs a procedure to process the NATO RIC change list and update a JTLS-GO scenario database accordingly.

Design Summary

This ECP adds a new Scenario Initialization File (SIP) Alter Data Function, that reads in a current scenario database and the NATO-provided RIC change list. The new function produces two files - a Standard Query Language (SQL) file that can be executed within PostgreSQL, and a text Summary file that lists all of the database changes that will be made when the SQL file is executed. The procedure must be run on each active scenario being maintained by the NATO organization.

2.21 JTLS-2023-16042 Import Terrain Layers Without Restrictions

Summary of Model Change Request

If a user organization misplaces or accidentally deletes the JTLS-GO Geographic Information System (GIS) Tool terrain project, it is impossible to recover the needed GIS Tool project and later the terrain layers. The purpose of this ECP is to allow the GIS tool to recreate the full GIS Tool terrain project.

Design Summary

The JTLS 6.1 GIS Tool had the ability to import the current JTLS-GO terrain grid data, The JTLS-GO 6.2 version of the GIS Tool also has the ability to import the JTLS-GO terrain layer definition. If the GIS project is misplaced, the terrain layers and terrain grids of an existing scenario can be imported to recreate the needed project data files.

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 6.2.0.0, no STRs have been reported. All STRs uncovered for JTLS-GO 6.1 have also been corrected in the baseline JTLS-GO 6.2.0.0 version.

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 6.2.0.0 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 6.2.0.0 DATABASE CHANGES

The following changes were made to the JTLS-GO 6.2 database:

- Added IIP GROUND VISUAL SENSOR to the database, JTLS-2020-15046 Include Aircraft In Tactical Intel Reports, Section 2.5. The Version Conversion Program (VCP) used to convert a 6.1 scenario into the required 6.2 format, set this new database parameter to the old IIP VISUAL SENSOR.

- Added IIP AIR VISUAL SENSOR to the database, JTLS-2020-15046 Include Aircraft In Tactical Intel Reports, Section 2.5. The VCP sets this new database parameter to the old IIP VISUAL SENSOR. If a user organization desires to represent different air and ground visual sensor capabilities, these two database parameters should be reviewed for each represented Intelligence Information Prototype (IIP) in the scenario.

- Added SLP IGNORE ENEMY PROB to the database, JTLS-2022-15858 Convoy Conditionally Enter Enemy Unit Radius, Section 2.13. The VCP sets this new database parameter to zero which indicates that a convoy should never take a chance on entering an enemy or suspect unit radius,

- Added LOGFAS RIC VERSION to the database, JTLS-2022-16016 Create Automated RIC Update Capability, Section 2.20. The VCP sets this new database parameter to 19, which was the version that was used by JTLS-GO is recent exercises. The setting of this parameter was simply selected based on our latest information, but will be automatically updated when a RIC change sheet is processed by the RIC Alter Database Tool. The JTLS-GO Development Team suggests that if LOGFAS is to be used, that a new RIC Change File be processed so the LOGFAS RIC VERSION number matches the RIC codes assigned to supplies and tracked combat systems.

APPENDIX C. VERSION 6.2.0.0 REPOSITORY CHANGES

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

Although R&A will continue to support JTLS-GO 6.1 for a minimum of one year, from here forward, all repository work will only be accomplished within JTLS-GO 6.2 only.