JTLS-GO

Version Description Document

February 2020

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

JOINT THEATER LEVEL SIMULATION - GLOBAL OPERATIONS
(JTLS-GO 5.1.6.0)

[Blank Page]

ABSTRACT

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

This *JTLS-GO Version Description Document (VDD)* describes the new features of the Version 5.1.6.0 delivery of the configuration-managed JTLS-GO software suite.

JTLS-GO 5.1.6.0 is a Maintenance release of the JTLS-GO 5.1 series that includes an updated wespac51 demonstration database as well as updated repository data held in the repository51 database. There are no major Engineering Change Proposals (ECPs) included with this release, but there are a few minor ECPs that required no new data or data format changes, which are summarized in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

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

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

[Blank Page]

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS-GO Version Description Document* (VDD) describes Version 5.1.6.0 of the configuration managed Joint Theater Level Simulation - Global Operations (JTLS-GO®) software suite. JTLS-GO 5.1.6.0 is a Maintenance release for the JTLS-GO 5.1 series.

JTLS-GO 5.1.6.0 includes the entire JTLS-GO suite of software, a repository of engineering level data, and a realistic demonstration scenario based on the Western Pacific theater of operations, called “wespac51”. No database format modifications have been made for this release, but a static data error for allowable OTH-Gold Ship Types was corrected in Version 5.1.1.0. This does require a user to execute a correction procedure for all of their Version 5.1.0.0 scenarios loaded in Oracle. Information on this procedure can be found on Page 1-11.

Descriptions of minor Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0. Explanations of all Software Trouble Reports (STRs) corrected in this release are provided in Chapter 3.0. Outstanding STRs are provided in Chapter 4.0. Changes made to the JTLS-GO 5.1 engineering data repository are provided in APPENDIX C.

JTLS-GO 5.1.6.0 executes on the Red Hat Enterprise Linux Version 7.6 64-bit operating system. 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 Portable Document Format (PDF) within a documents subdirectory.

1.2.1 Obsolete/Outdated Documents

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

1.2.2 Unchanged Documents

- *JTLS-GO Analyst Guide* (JTLS-GO Document 01, Version 5.1.3.0)
- *JTLS-GO Configuration Management Plan (*JTLS-GO Document 03, Version 5.1.2.0)
- *JTLS-GO Controller Guide* (JTLS-GO Document 04, Version 5.1.3.0)
- *JTLS-GO Data Requirements Manual* (JTLS-GO Document 05, Version 5.1.5.0)
- *JTLS-GO DDS User Guide* (JTLS-GO Document 06, Version 5.1.5.0)

- *JTLS-GO Director Guide* (JTLS-GO Document 07, Version 5.1.2.0)
- *JTLS-GO Executive Overview* (JTLS-GO Document 08, Version 5.1.3.0)
- *JTLS-GO Player Guide* (JTLS-GO Document 12, Version 5.1.3.0)
- *JTLS-GO Repository Description* (JTLS-GO Document 14, Version 5.1.2.0)
- *JTLS-GO Software Maintenance Manual* (JTLS-GO Document 15, Version 5.1.2.0)
- *JTLS-GO Technical Coordinator Guide* (JTLS-GO Document 16, Version 5.1.5.0)
- *JTLS-GO Entity Level Server User Guide* (JTLS-GO Document 19, Version 5.1.2.0)
- *JTLS-GO Federation User Guide* (JTLS-GO Document 20, Version 5.1.2.0)
- *JTLS-GO C4I Interface Manual* (JTLS-GO Document 21, Version 5.1.5.0)
- *JTLS-GO Air Services User Guide (*JTLS-GO Document 24, Version 5.1.2.0)

1.2.3 Updated Documents

- *JTLS-GO Installation Manual* (JTLS-GO Document 09, Version 5.1.6.0)
- *JTLS-GO WHIP Training Manual* (JTLS-GO Document 10, Version 5.1.6.0)
- *JTLS-GO Version Description Document* (JTLS-GO Document 17, Version 5.1.6.0)

1.2.4 New Documents

No new documents are delivered with JTLS-GO 5.1.6.0.

1.2.5 Delivered Software Components

JTLS-GO 5.1.6.0 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)

When operating the TRIPP capability in current JTLS-GO releases, users are not prevented from logging into an actively running TRIPP making a connection to the same Replay JXSR. A TRIPP, as documented, requires its own Replay JXSR to control and perform the replay of the recorded simulation events.

This situation as a consequence can have more than one user concurrently control the playback of the game, as the shared Replay JXSR will honor each of the user's playback requests and will then change what each connected user sees on their TRIPP instance. This issue will be addressed in a future JTLS-GO release by only permitting one login per TRIPP user instance, such as a WHIP login prevents from users logging into the same WHIP instance.

Organizations should develop their own procedures to ensure that only one user logins to a given TRIPP at a time.

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

Instructions for installing JTLS-GO 5.1.6.0 are provided in the *JTLS-GO Installation Manual*. Installing a previous version of JTLS-GO prior to installing JTLS-GO 5.1.6.0 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 “repository51” serves as a repository of engineering level data. Although not useful as a scenario, it does follow all of the database requirements for a scenario, and should be loaded into your Oracle scenario table-space. With JTLS-GO 5.1.6.0, it is possible to access and copy records from the repository51 database into your own developed scenarios.

- The scenario “wespac51”, which is based on the Western Pacific theater of operations and is suitable for training and demonstrations.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS-GO 5.1.6.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers:

- Operating system for the model: Red Hat Linux Enterprise Edition Version 7.6 (ES), 64-bit architecture.

  Security Enabled (SE) Linux must not be enabled on systems used to execute JTLS-GO or its components. Tests reveal that SE Linux-enabled systems cause frequent and random JXSR crashes and errors. Use of SE Linux to support JTLS-GO is currently not authorized.

- JTLS-GO 5.1 has been tested with the following versions of Linux 7:

  Red Hat Linux 7.6 - This operating system license must be purchased, but it has been approved by the Defense Information Systems Agency (DISA) for use by U.S. Government Agencies.

  Oracle Linux 7.6 - This operating system is free to download, use, and distribute, and is provided in a variety of installation and deployment methods. It has been approved by DISA for use by U.S. Government Agencies.

  CentOS Linux 7.6 - A free version of Linux 7 that has **not** been approved by DISA for use by U.S. Government Agencies.

- There are no restrictions on the operating system used for client workstations, except that the operating system must have a Java-enabled web browser. JTLS-GO 5.1.6.0 has been tested on the following operating systems:

  Red Hat Linux Enterprise Edition Version 7.6.

  CentOS Linux Version 7.6.

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

- The JTLS-GO 5.1 series no longer uses Oracle Java, and has moved to the latest version of OpenJDK 8, which is OpenJDK 8 Version 242. We no longer deliver the Java Runtime Environment (JRE) within the JTLS-GO delivered software TAR files. Each user organization must obtain the latest version of the OpenJDK Red Hat Package Manager (RPM) and install the RPM on the servers used by JTLS-GO.

- JTLS-GO uses IcedTea to provide the OpenJDK web start capability that implements the web-enabled JTLS-GO functionality. The current version of JTLS-GO supports IcedTea version 1.8.3.

  IcedTea version 1.7.1 does not support use of unsecured HTTP to access web-enabled JTLS-GO components for the DDS Client application. Users must configure their Glassfish instance to use HTTPS, by enabling the "Use SSL" column within the DCP when running with IcedTea version 1.7.1. The use of SSL is the current default setting in the DCP.

  Red Hat Linux version 7.7 continues to distribute with IcedTea version 1.7.1. There are available RPM packages for a later version of IcedTea, so users who wish to use unsecure HTTP with JTLS-GO must explicitly install IcedTea 1.8.3 on the JTLS-GO servers and client workstations.

- JTLS-GO database tools require use of a certified Oracle database server and the full Oracle Client installation for runtime requirements. Additional installation details can be found in Section 1.5.2 of this chapter.

- 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 service programs. The version of TCP/IP included with the supported versions of Red Hat Linux ES is sufficient.

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

- SIMSCRIPT II.5 (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS-GO code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS-GO, because all JTLS-GO software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only by a U.S. Government Agency that can obtain source code and plans 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 needed only by a U.S. Government Agency that can obtain source code and plans 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 needed only by U.S. Government Agency that can obtain source code and plans 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 LICEN.SE 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 the COP, C2PC, or other C4I systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

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

- JTLS-GO 5.1 implements SSH Tunneling between Apache and the services, and among the services. Rigorous testing should be done prior to use in any exercise, and particular attention should be paid to network performance under load.

1.3.2 JTLS-GO Information Assurance Compliance

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

One of the DoD IA requirements is that the software must implement a methodology that ensures that the end user keeps the software up-to-date and properly installs all security patches. 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.

Information available at the time of this release indicates that OpenJDK will not implement an expiration date. In order to fulfill the DoD IA requirement, Maintenance releases of JTLS-GO will be released as new versions of OpenJDK are released. In a future version of JTLS-GO, all delivered Java programs will check that the latest version of Java is being used. Each local installation will be able to override the warning generated when the program detects that the latest version is not properly installed.

JTLS-GO has completed the IA program mandates and the JTLS-GO 5.1 series of releases has been granted an Authority To Operate (ATO) on DoD systems. Contact the U.S. Government Program Manager, Mr. Don Weter (donald.e.weter.civ@mail.mil), for additional information.

1.3.3 JTLS-GO High Level Architecture Compliance

The JTLS-GO 5.1.6.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS-GO in an HLA mode. JTLS-GO uses the Federation Object Model (FOM), located in the $JGAME/data/hla directory. Federation testing of JTLS-GO is not complete, but initial tests with CAE's GESI wargaming system have been accomplished. Future plans include expanding the capabilities of the JTLS-GO-GESI federation, called "GlobalSim".

GlobalSim is a comprehensive constructive simulation solution for joint training and wargaming, that helps commanders and all levels of staff prepare for a range of operational scenarios. The solution combines JTLS-GO with CAE's GESI constructive tactical entity-level simulation system as a federation. CAE's GESI constructive simulation system is designed to run complex and comprehensive exercises from the company level up to division level. The CAE GESI system is used to represent a virtual battlefield, including weapons, vehicles, aircraft, and ground forces.

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.

All JTLS-GO software needed to run GlobalSim is included in this delivery.

The HLA RTI (Run Time Infrastructure) executive program recommended for use with this release is Pitch pRTI Evolved 4.4.2.0. However, this program is not included in the JTLS-GO 5.1.6.0 delivery. Users may obtain a full installation package of the RTI executive program from Pitch Corporation (www.pitchtechnologies.com). 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 the JTLS-GO 5.1 series and the previous JTLS-GO 5.0 series database structure.

To upgrade your JTLS-GO 5.0 scenario to JTLS-GO 5.1 compatibility, see instructions listed in Chapter 3.1 of the *JTLS-GO DDS User Guide*.

1.4.1 JTLS-GO Using Legacy Default Symbol Set

If a user organization is still using the pre-JTLS-GO 5.0.0.0 legacy default symbol set, prior to unloading your JTLS-GO 5.1.0.0 formatted data from your Oracle database server into the JTLS-GO 5.1.0.0 scenario American Standard Code for Information Interchange (ASCII) text files, you must execute the JSYMS program using the procedure outlined in Appendix B.11 of the *JTLS-GO DDS User Guide*. This procedure will reorganize the structure of the .gs and .scf symbols-related files.

1.4.2 JTLS-GO Using New Default Symbol Set

You should not make any modifications to the Default Symbol Set delivered with JTLS-GO 5.1.6.0, but end user organizations are free to use the Default Symbol Set in their scenarios and alter the scenario symbol set to meet specific organizational needs.

1.4.3 Standard Repository Changes

The JTLS-GO 5.1 series of JTLS-GO is the first series in which R&A is delivering an unclassified data repository called “repository51”. In future Major releases of JTLS-GO, APPENDIX B will provide a summary of the data structure changes made to the data repository. No data structure changes have been made is this Maintenance release; therefore, APPENDIX B is empty. Refer to Appendix B in the *JTLS-GO 5.1.0.0 Version Description Document*, included with this release, for data structure changes made for the JTLS-GO 5.1 series.

1.5 INSTALLATION

1.5.1 Installation Instructions

The *JTLS-GO Installation Manual,* a PDF file available for direct download, is part of this JTLS-GO delivery. It provides detailed instructions for installing a new version of JTLS-GO.

1.5.2 Oracle Compatibility and Installation

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

should be applied to the Oracle Client installation. A 64-bit Oracle Client installation must be used.

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

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

Oracle offers free Express Editions (XE) of the Oracle relational database management system. Compared to the 11gR2 XE version, the newer 18c XE has a larger footprint and a much more complex database architecture. For test environments and scenario building purposes, or for collecting AAR data for a short period of time, the installation and setup of the 11gR2 XE version is much simpler.

The DDS application utilizes the Oracle GlassFish J2EE server, which, like the JTLS-GO WHIP Apache server, is delivered with JTLS-GO and requires no separate installation.

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

1.5.3 Special Installation Instructions

This section describes special instructions that should be followed because of errors corrected in this version and previous bug releases for the JTLS-GO 5.1 series.

1.5.3.1 OTH-Gold Ship Types

JTLS-GO 5.1.0.0 was delivered with an old list of OTH-Gold ship types. This problem was corrected in JTLS-GO 5.1.1.0. To properly implement this solution, users must execute the following additional procedures for each of their JTLS-GO Version 5.1.0.0 scenarios loaded in Oracle.

Users who have already executed this procedure for their scenarios after installing JTLS-GO Version 5.1.1.0, 5.1.2.0, 5.1.3.0, 5.1.4.0, or 5.1.5.0 do not need to execute the provided procedure again.

New OTH-Gold ship types were added to JTLS-GO, due to STR JTLS-2019-14238.

1. Execute the following command:

   cd $JTLSHOME/script/dds/version5.1/scripts/

2. Execute the following command:

   sqlplus yourScenario/OraclePassword @update_oth_gold_types.sql

3. Verify the related execution listing file under the $JDATA/scenario/ directory for errors.

4. Unload your scenario using the JTLS-GO Menu, Options 1 -> 1 -> 4

5. Verify the .srw ascii file for your scenario.

To fix the problem renaming Force Sides, due to STR JTLS-2019-14267:

6. Reload the database.

1.5.3.2 Generation Of Combat System Summary Files

STR JTLS-2019-14518 Move Combat System Summary Files, delivered as part of JTLS-GO 5.1.4.0, solved a problem overwriting the summary Combat System Character Separated Value (.csv) files during Batch Runs. This STR was solved by moving the location of the .csv files from the game/<scenario_name>/location directory to a sub-directory under each checkpoint.

For every active game scenario, one of the following procedure options must be executed to establish the directory structure needed by the STR solution. Users who have already executed this procedure for their scenarios after installing JTLS-GO 5.1.4.0, do not need to execute one of the selected options again:

- Option 1: Rerun the Setup Procedure for each active scenario.
- Option 2: Hand-create a cbtsys_summary sub-directory in the game/<scenario_name> directory. This can be accomplished using the following steps from a command terminal for each existing game that has already been set up and prepared for execution:

  a. Enter the command: game - this puts the terminal in the $JTLSHOME/game directory.

  b. Enter the command: cd <scenario_name> (for example, cd wespac51) - this puts the terminal in the game/<scenario_name> directory.

c. Enter the command: **mkdir cbtsys_summary**. This creates the necessary new directory.

1.5.3.3 Aircraft Type Foreign Key Definitions

STR JTLS-2019-14541 Referenced Aircraft Class Can Be Deleted, delivered as part of JTLS-GO 5.1.5.0, solved a problem that allowed the deletion of aircraft types that were referenced by units. The STR was solved by removing a rule in the database that allowed the user to set the aircraft type-related foreign keys of the unit tables to NULL.

After loading JTLS-GO 5.1.5.0, the user must unload and then reload their JTLS-GO 5.1 scenarios to have the new foreign key definitions in their database schemas.

Users that have accomplished this procedure after installing JTLS-GO 5.1.5.0, do not need to re-execute this procedure after installing this version of JTLS-GO.

2.0 ENGINEERING CHANGE PROPOSALS

The following model capabilities were added to JTLS-GO 5.1.6.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

These ECPs were built on an emergency basis to support a large upcoming NATO exercise. The documentation for these ECPs will be completed as part of the JTLS-GO 6.0 series, due to be released in the Summer of 2020. These ECP are technically not closed until all documentation requirements have been fulfilled. These ECPs will also be included in the JTLS-GO 6.0 Version Description Document.

2.1 JTLS-2019-14624 Save WHIP Layout

Summary of Model Change Request

Users spend considerable time arranging their desired WHIP components on their available workspace. Every time they restart their WHIP, they must repeat the process of rearranging these WHIP components.

Design Summary

The WHIP now has a Save Layout function. Users can arrange their WHIP workspace as desired and select the Save Layout option. The layout details are saved on the server, so that no matter where the specific WHIP is started, the layout details can be recalled and all WHIP components contained in the save will be restored to the proper position and settings.

2.2 JTLS-2019-14625 Quickly Change Mission Speed/Altitude

Summary of Model Change Request

Users needed a quick method to change the speed and altitude at which an air mission is flying. Although speed and altitude can be changed from the SET MISSION PARAMETER Order, the order panel is now so large that it is difficult to quickly issue needed speed and altitude changes.

Design Summary

Two new Quick Orders were developed, to allow a user to change the speed of an air mission and to allow the user to change the altitude of an air mission. These Quick Orders can be added to any menu file desired by each end user.

In JTLS-GO 6.0, this ECP will be expanded to also make the SET MISSION PARAMETER Order more user-friendly, by introducing tabs to reduce the size of the order panel.

2.3 JTLS-2019-14626 Different Coordinate Formats In IMT

Summary of Model Change Request

Users need to view locations in both Latitude and Longitude format and Military Grid Reference System (MGRS) format simultaneously on the same WHIP. Currently, the user can select how locations should be displayed from the WHIP Preferences menu. When a location preference is selected, it applies to all WHIP component locations.

Design Summary

A preferences menu option is now included on Information Management Tool (IMT) windows, allowing the user to select the preferences desired for that specific IMT display. In this release of JTLS-GO, only the IMT location preference option has been implemented.

In JTLS-GO 6.0, all of the WHIP-wide preferences will be selectable and changeable for each IMT display. For example, the user will eventually be allowed to bring up two supply IMT screens and display supplies using the default Tons and Gallons in one IMT display and the Unit Of Measure “Each” in a different IMT component.

2.4 JTLS-2019-14627 IMT Column Size Adjustment

Summary of Model Change Request

Users could not reduce the size of IMT columns to meet their requirements. The IMT definition files contain a preferred size for each column, based on the information that is likely to be displayed in that column. A user has no ability to reduce that column size, if their data does not require the column to be that large.

Design Summary

The default sizes for IMT columns have not been changed, but the user can now reduce the column width to meet their scenario data requirements.

2.5 JTLS-2019-14628 Add Concept Of COMAO ID

Summary of Model Change Request

NATO has a concept of a Combined Air Operations (COMAO) package of missions. Each COMAO has its own ID, included in the Air Tasking Order. A given COMAO can consist of one or more JTLS-GO Air Mission Packages and one or more independent JTLS-GO Air Missions. The COMAO is simply a grouping of missions for the purpose of a combined air operation. There is no functional aspect to a COMAO group, and it is strictly used for identification purposes.

NATO response cell personnel were having difficulty communicating properly with the exercise audience, because they did not have access to a mission’s COMAO ID.

Design Summary

An Air Mission can now be given a COMAO ID as part of its order. This new optional field is included on the ATO Tab of each Air Mission Order panel. This COMAO ID is saved on the mission and passed to the JTLS Object Data Authority (JODA) for display on a WHIP's Air Mission IMT component. The IMT can be sorted on this new field and missions can be retrieved based on the COMAO to which they belong.

The NATO Air Tasking Order (ATO) Parser and ATO Translator (ATOT) were both changed to take the COMAO ID data available in an Interactive Command and Control (ICC) system-generated ATO and automatically fill the new COMAO ID field available on all Air Mission Orders. Other than reporting the COMAO ID for display, there were no modeling function changes implemented.

This ECP resulted in a change to the format of the Air Mission checkpoint ASCII file. This means that any existing JTLS-GO 5.1 series checkpoints will not be readable after installing JTLS-GO 5.1.6.0. If this is a major problem for an organization, please contact the JTLS-GO Help Desk for a procedure to update the Air Mission checkpoint ASCII file.

This ECP should not impact a non-NATO JTLS-GO user in any way.

2.6 JTLS-2019-14630 Light Air Game

Summary of Model Change Request

One of the standard ATOT operating procedures is to run the translated ATO in a test game prior to submitting the ATO into the production game, By watching the test game, the air cell can determine if there are any logic errors in the translation or problems that need to be addressed prior to the start of the ATO period.

If the exercise is large, it becomes almost impossible to follow this best practice approach to the translation process, because the game cannot run fast enough to get through or sometime even to the start of the ATO period.

The purpose of this ECP is to be able to put the test game into a fast running mode that skips many of the more time-consuming JTLS-GO algorithms and simply view the ATO progress from a movement perspective.

Design Summary

It is possible to place the model into what the Design Team has called ATO Lite mode. In this mode, only important air related event are executed. All non-air events are skipped. Entering this mode has drastic consequences and should only be used for the purpose of testing an ATO in a test game. Once the model enters this mode, the represented conflict situation has been compromised.

Due to the seriousness of entering this mode on a production game, several steps have been implemented to ensure that the ATO Lite mode cannot be inadvertently started:

- The ATO Lite mode can only be started from a JODA console, which is only available to Technical Control on a JTLS-GO server. ATO Lite cannot be requested from an Order Panel.
- Once a request to enter ATO Lite mode has been made, the CEP pauses and asks Technical Control whether ATO Lite mode is truly desired. This requires positive confirmation from Technical Control.

Initial performance testing of the ATO Lite mode has shown significant improvement in attainable game speeds.

2.7 JTLS-2020-14691 APP6/2525 Symbol Names In GlobalSim

Summary of Model Change Request

The GESI federate for the GlobalSim federation required that the names of the APP6/2525 symbols be published on HLA. The model already provided these data on the JODA, so a simple change was required to also supply these names on the EODA for entity-level objects.

Design Summary

The JDS protocol was modified to include a text attribute on the ELS objects for the EODA. Additional changes to the ELS code were required to publish the symbol names in these new fields.

2.8 JTLS-2020-14701 Two JOIs Feeding One COP

Summary of Model Change Request

A user wants to run two simultaneous JTLS-GO games and have each game feed the same Over The Horizon Gold (OTH-Gold) Common Operational Picture (COP). The problem with doing so is that each game's JTLS Operational Interface (JOI) would use the same track numbers, creating duplicate tracks on the COP.

Design Summary

The ICP and the OTH-Gold interfaces were changed to allow the user to specify the track range that should be used by the OTH-Gold JOI. For example, JOI 1 can now be told to use track block 00001 through 50000, while JOI 2 can be told to use track block 50001 through 99999.

3.0 SOFTWARE TROUBLE REPORTS

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

Chapter 4.0 describes STRs that remain outstanding from previous versions; however, because of the model enhancements included with JTLS-GO 5.1.0.0, no STRs identified in prior JTLS-GO series are considered valid. Uncorrected errors identified for JTLS-GO 5.1.6.0 are documented in this chapter and should be corrected in future maintenance releases.

3.1 JTLS-2020-14668 DDS TMU Depth Missing Reset Button

The Elevation panel of the TMU allows the user to input a value, and to reset the value using a "No Change" button. However, the Depth panel did not have a "No Change" button.

There was a "No Change" button capability in the code, but it did not display on the Depth panel. The panel size was not set wide enough, so the button did not display. The code was modified to expand the preferred panel size to display the "No Change" button in the Depth panel.

3.2 JTLS-2020-14669 DDS Symbol Filter Panel Echelon Level

The DDS Symbol Size filter panel has the column "Echelon Level". The "All" checkbox does not work properly - when checked, the Echelon Level checkbox can turn off filters, but cannot turn on filters.

The DDS filter panel code was corrected.

3.3 JTLS-2020-14670 Change Air Mission Parameter Altitude

An Air Mission follows along a route. The Air Mission has a route point at the end of each grid through which the mission will pass. When changing an Air Mission's Altitude using the CHANGE.MISSION.PARAMETER order, the Altitude would update in the model. However, after a short amount of time, the Mission's Altitude would revert to the original altitude, before the CHANGE.MISSION.PARAMETER order was sent.

The altitude of all current route points held by the Air Mission are now updated as a result of a CHANGE.MISSION.PARAMETER order. The newly assigned altitude is now maintained through the remainder of the mission's current move task.

3.4 JTLS-2020-14671 DDSC Small Island Cannot Change To Ocean

Small Island grids cannot be changed to Ocean grids, because the elevation check always indicates the elevation is greater than 0, even though the TMU automatically sets elevation to zero.

The code was modified so that the elevation check is not done by the TMU when changing the terrain type of a Small Island grid.

3.5 JTLS-2020-14672 Crash Picking Up Unit From Moving Formation

An air mission picked up a portion of a unit from a formation. When the mission went back to get a second load, the ship on which the unit was located had moved. The pickup task was canceled but the unloading task was not. When the mission got to the offload location, the model crashed because the mission was not carrying anything.

When the load task needs to be canceled, the mission's offload task is also canceled. This is not a permanent solution, but will simply stop the crash from occurring. A more robust solution will be developed for JTLS-GO 6.0.

3.6 JTLS-2020-14673 Default Value For Show In Confirmation Messages

When sending an order with a "Confirmation Message" field, such as "SET DEBUG REPORT FLAGS", the user has the option to show "CHANGES_ONLY" or show "ALL_DATA". This is a mandatory field and requires extra work on the part of the user when using these order panels.

When sending an order which uses the "Confirmation Message" field, the default value is now "ALL_DATA, resulting in the user not being required to select an option for this mandatory field. The user still has the option to select "CHANGES_ONLY" when desired.

3.7 JTLS-2020-14674 Determining UTM Northing From MGRS

An MGRS Coordinate consists of three parts. The first is a grid zone designator which is made up of a zone number (1-60) and a latitude band letter (C-X omitting I and O). The second is the 100,000 meter square identification, which is made up of two letters (also omitting I and O). The last part consists of digits representing easting in meters and northing in meters.

When using the grid zone designator to determine UTM Northing, there are two conditions that can never be met if the latitude band letter is "L". This two conditions were properly implemented in the algorithm.

These conditions have been corrected. If the zone number is odd, we check to see if the second letter is less than "C" OR greater than "M". If the zone number is even, we check to see if the second letter is less than "H" OR greater than "S".

3.8 JTLS-2020-14675 Allow Asterisk In OPM File Links

It is not uncommon for a database to contain objects, such as Tactical Unit Prototype (TUP) names, with asterisks in their names. This is permitted within the DDS, but because an asterisk is also a Linux wildcard character, it can cause issues when using the name in scripts.

This happens when generating the Online Player Manuals (OPMs), where each TUP has a file that corresponds to the TUP name. After OPM generation, a script attempts to establish links between the Player directory structure and the Controller data files. Those file names that contain an asterisk generate an error and a link is not established.

The script that creates the OPM links was modified to explicitly quote the file name variable, which prevents the asterisk from being considered a wild card.

In addition, the script was checking if the file already existed as a link or as a read only file before attempting to create the link. This was modified to check if the file already exists.

3.9 JTLS-2020-14676 Permit Batch Run Without AAR Database

When running in batch mode, the CEP will wait for the AAR to connect before setting game speed during each iteration. This requires the AAR database to be run in batch mode, even if it is not otherwise required.

It was also determined, when running in batch mode, that having additional variables, such as the iteration number and whether it is the final iteration, passed to the user collection script would be useful.

It was also determined that batch iterations should be called iterations, rather than runs, to avoid confusion with JTLS-GO runs that may or may not be the result of batch iterations.

The issue of running the CEP without the AAR was partially addressed in JTLS-GO 5.1.5.0. This STR completes the implementation of that issue.

The question of whether or not the batch was being run in accordance with the AAR was removed from the setup. Instead, the setup relies on whether the AAR Client is configured for the scenario or not to make the determination. This is the same method that the CEP uses to determine if it should collect AAR data.

The setup was also modified so that, after every iteration, the desired additional data items are passed to the user's collection script. Wording on the setup was modified to refer to the user's desired number of iterations versus the number of runs. The data collection scripts were modified to correct the help and ensure the user can specify a desired output file.

3.10 JTLS-2020-14677 Incorrect Mission Type Conversion

The ATO Translator (ATOT) was mapping TBMCS mission type OCA as an OAS mission instead of a DCA mission.

The mapping of external software mission types to JTLS-GO mission types is controlled using the ato_constants.xml file. Three instances of incorrect mapping of the OCA mission type were modified to correctly map as JTLS-GO DCA missions.

Additionally, we identified a formatting error in the stylesheet used to check the format of the ato_constants.xml file, which has been corrected.

3.11 JTLS-2020-14678 SVPR Error/Warning List Resolved Issues

The SVPR Error/Warning List in the DDS did not clear resolved issues after running the SVP from the jtlsmenu.

The script called from the javamenu now clears all resolved issues. If the resolved issues appear again as a result of the SVP run, the issue will be displayed in black indicating that the previously selected solution did not properly fix the problem.

3.12 JTLS-2020-14680 Closest Network Node Not Always Best

Numerous convoys did not find a road route to its destination, because the closest road node was on the other side of a river and the convoy could not get to the node. A node that was further away, but on the correct side of the river, was not selected.

The code was changed. The model now selects the closest node to which the object can freely move. If the object cannot move to the node, the node is not considered as the possible starting node. If the object cannot move from the node to its designated destination location, the node is not considered as a possible ending node.

3.13 JTLS-2020-14681 Checkpoint Not Writing Entire Symbol Name

During a checkpoint, the High Resolution Unit Prototype (HUP) supply category data was only writing out the first 15 characters for the supply category name, even though the database permits a supply category name of 25 characters. This caused the checkpoint to fail during a restart.

Because of this issue, we searched for other instances of writing a name out as 15 characters when it should be 25 or more characters.

In the SIP and the CEP, the write statement for writing an RTM Record was changed to permit 30 characters for the Graphics Symbol name, because the database permits 30 characters.

In the CEP, the write statement for writing terrain data was changed to permit 20 characters for Road Type name, because the database permits 20 characters.

3.14 JTLS-2020-14682 Incorrect HIP ELS Air Mission Update Order

The HIP receives BaseEntity updates through the High Level Architecture Run Time Infrastructure (RTI). BaseEntities are published by the GESI federate for the objects owned by the GESI external simulation. The order being sent by the HIP to the CEP for reflecting the BaseEntity aircraft updates was rejected because of a duplicate field keyword in the order stream.

The HIP's routine for processing BaseEntity updates was in fact writing the same keyword twice in the order stream to the model. This caused the order to be rejected whenever the GESI federate owned an air mission and published BaseEntities for the aircraft in the mission. This has been fixed.

3.15 JTLS-2020-14683 HIP Not Converting Altitude To Feet

The HIP uses the spatial (X,Y,Z) attribute from a BaseEntity update from GESI to compute the position and altitude for an aircraft in an air mission. Although the latitude and longitude are properly computed, the altitude is being sent to the ELS in meters, rather than feet, in the associated order stream.

The HIP routine for processing BaseEntity updates was modified. The computed altitude is now converted to feet prior to sending the order to the ELS.

3.16 JTLS-2020-14684 SVPR Needed Reset Function For Resolved Items

The SVPR automatic correction allows fixing a large group of the same errors or warnings. When the group fix capability is used, the DDSC marks each of the errors as "corrected", and renders the error description as green to indicate that it has been fixed. However, some errors require two fixes. In order to accomplish the second fix for the large group of error, the user must uncheck each error individually. This is time consuming and reduces the usefulness of the group error fix capability.

A "Reset Resolved" checkbox was added to the View menu and code was modified to enable it.

3.17 JTLS-2020-14685 Unmanned HUP With Crewed Combat Systems

The ELS had an issue with an Unmanned Land Vehicle, which is represented by an HRU that had a Combat System that needed Crew to operate. JTLS-GO ignores crew requirements for HRU Combat Systems, but having crewed Combat Systems on what should be unmanned vehicles does not make sense.

SVP Error 253 was added, which will be generated if an UNMANNED HUP has at least one Combat System for which the crew count is greater than 0.

Warning 1275 was also added, which will be generated for a MANNED HUP if the total number of personnel for the HUP's assigned Combat Systems (including personnel or crew) is less than the total number of crew needed.

3.18 JTLS-2020-14686 Invalid OTH Gold Type

The SIP alterdata function to generated TBMCS initialization spreadsheets entered simdebug.

The Ship Unit Prototype (SUP) accessed by the routine did not have a REAL WORLD IDENTIFIER set specified. It defaulted to UNASSIGNED, which does not have an entry for the OTH GOLD TYPE.

The routine was modified to check for the UNASSIGNED value and, if found, assign a OTH GOLD TYPE of UNK.

3.19 JTLS-2020-14687 No Data In Political Country Polygon Child Table

When building databases, users may not put any data points for the political country polygons. This data resides in the <scenario_name>.poc_pt file. Without political country polygons, detection IIRs cannot properly report the country in which the contact was made. It will report the country code of the object being detected, which is fine if in its home country, but if the detection is of an invading object, the intelligence report will say the detection was made in the detected object's home country, which is obviously false and misleading.

SVP Warning 1156 is now generated if a political country has no specified polygon points.

3.20 JTLS-2020-14688 Crash Creating Supply Cache

If the user attempted to create a supply cache, the model crashed.

This crash was entered into the code as a result of a recent change to ensure that a replenishment task resulted in the moving unit intercepting the receiving unit. The new code ignored the fact that the replenishment unit may not be giving the supplies to a receiving unit. The supplies may simply be designated for a supply cache as a specific location. The code was corrected to again consider this situation.

The recent change also attempted to have the replenishment unit follow a moving receiving unit as the supply offload was being processed. This code did not consider all of the possible movement situations and caused numerous crashes. This code was removed because it could not be thoroughly tested before release.

3.21 JTLS-2020-14689 Crash Rejoining HRU Located In Shelter

If an HRU was ever in an Equipment Shelter and the user had it rejoin its parent, the model would crash.

The records that indicated the shelters in which the HRU was or had been located were not being properly cleared. When the model attempted to remove the HRU from the game, the uncleared set caused the model to crashed. Two changes were made:

- Before rejoining the parent, the model now ensures that the HRU vacates its current shelter.
- When object leaves a shelter, the model now removes the tags owned by the HRU or Unit indicating it is currently in the shelter.

3.22 JTLS-2020-14690 Infinite Loop Attempting To Intercept Moving Unit

Moving Unit 1 was trying to intercept moving Unit 2, and the model ended up in an infinite loop.

The issue was tracked down to the fact that the ground move gridded terrain optimization algorithm did not put the destination location in the unit's planned route. Unit 1 was very close to Unit 2. Unit 1 asked for a new route to Unit 2 and the algorithm always gave a route that had Unit 1's location but did not have Unit 2's location. Unit 1 kept moving to its current location and could never make it to the desired location. The move to the current location has a time of zero and this repeated over and over again.

The gridded terrain algorithm was corrected to also put the destination grid in the route. This allowed Unit 1 to move to Unit 2's location and the remainder of the logic was executed as needed.

3.23 JTLS-2020-14692 HRU On Board Unit Destroyed ELS Crash

The ELS crashed while attempting to destroy an HRU. The HRU was on board a ship, and it was being destroyed as part of the mission to rejoin its parent.

The code to process a delete of an HRU was not correctly processing the HRU. The object was out of the game because it had already rejoined an aggregate unit but it continued to be on board a ship. It was not properly separated from its association with the ship before the delete was attempted. Code changes were made to fix this crash.

3.24 JTLS-2020-14693 ELS Command Objects Inconsistent Names

When a unit or an HRU is ordered to attach to another unit, the base name of the resulting command object had the same name as the original. This caused some confusion when inspecting the involved objects on the ELS Order Distribution Authority (EODA).

During an attachment, the original command object is added to the command hierarchy of the new parent unit. Code changes were made to rename each attaching command object such that the name reflects the new aggregate parent unit. This applied to both aggregate unit attachments and to rejoining HRUs.

3.25 JTLS-2020-14694 Checkpoint Restart ELS Crash

The ELS crashed while restarting from a checkpoint when there were destroyed entities. The dead entities were linked to a burn event but were not properly restored during the restart.

If the entity was destroyed, the ELS was modified to allow a burn event to continue when the entity was not present. This prevented the crash during a restart from a checkpoint.

3.26 JTLS-2020-14695 Customized Templates Named Incorrectly In ELS

The ELS created customized templates for units which are involved in an attach or HRU rejoin. These templates contained extra template locations as positions for the additional combat systems. When the receiving unit was a Naval unit, the customized template was saved to disk with the wrong name.

The base of the name of the customized template used the name of the unit prototype. The ELS code incorrectly assumed that the prototype was a TUP, and never a SUP as for Naval units. Changes were made to utilize the correct prototype name.

3.27 JTLS-2020-14696 Target-JDPI Link Made Multiple Times ELS Crash

A Controller submitted an order to associated a target with a JDPI. This same order was submitted more than once. The ELS crashed when the order was sent a second time.

The ELS crashed because the target was already associated with the specified JDPI. A check was added to the code to prevent the ELS from attempting to associated a target with a JDPI where a link already existed.

3.28 JTLS-2020-14697 Crash SEAD Changing Altitude At Route End Point

If a SEAD mission reaches the endpoint of its orbit location, and at that precise time decides to fire on a threat SAM site, the model will crash if the SEAD mission decided it needs to change altitude to fire its available weapon.

This very specific situation was not accounted for in the logic. The SEAD mission had no next route point in its route set. In this case, two route points were added. The first route point to have the mission drop to the proper altitude and the second route point to have it climb back to its assigned orbit altitude.

3.29 JTLS-2020-14698 Run-Time Debugging ELS Code Changes

Code changes in the ELS were needed to allow the user to debug specific parts of the code by a specified object.

The code changes utilized a global variable called OBJECT DEBUG NAME which facilitated the debugging of the ELS when a specific object was involved. The code changes did not adversely affect the normal operation of this program.

3.30 JTLS-2020-14699 Crash When Naval Unit UAV Returns Home

When a UAV owned by a Naval Unit returns home, and its owning ship is in an ocean grid, the model crashed attempting to find a landing location for the UAV. It could not find a place because the model asked whether the UAV's home squadron was Naval Qualified. The Naval Qualified Flag only exists on Squadrons.

The landing allowed logic was changed. If the aircraft's home, is not a squadron, then it is allowed to land anywhere.

4.0 REMAINING ERRORS

Every effort has been made to correct known model errors. All reproducible errors that resulted in Combat Events Program (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 5.1.0.0 represents a major release of new functionality, remaining outstanding errors from the JTLS-GO 4.1 series and earlier have been considered to be obsolete and no longer relevant to JTLS-GO and have been removed from consideration for correction at this time. In future Maintenance releases, outstanding errors related to JTLS-GO will be listed in this chapter, with information provided regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

4.1 DDSC – TMU Line Mode Changes Multiple Grids

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

4.2 DDSC – Multiple Types In DDS History Table

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

4.3 WHIP - Pipeline Not Shown On IMT

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

4.4 DDSC/WHIP/JOBE - CADRG Map Zoom

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

4.5 WSM - Many Messages Cause Lockup

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

APPENDIX A ABBREVIATIONS AND ACRONYMS

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

AAA	Anti-Aircraft Artillery
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
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
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
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	Japan Self-Defense Forces
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
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-GO 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
XML	Extensible Markup Language
XMS	XML Message Service

APPENDIX B Version 5.1.6.0 DATABASE CHANGES

No database structure changes were made for JTLS-GO 5.1.6.0.

APPENDIX C Version 5.1.6.0 REPOSITORY CHANGES

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

C.1 Added new PH:

- 3K60BAL_PH
- AGM183A_PH
- TLAM-VA(MST)_PH

C.2 Added new PK:

- 3K60BAL_PK
- AGM183A_PK
- TLAM-VA(MST)_PK

C.3 Added new SKL:

- 3K60BAL_SKL
- AGM183A_SKL
- TLAM-VA(MST)_SKL

C.4 Added new AK:

- AGM183A.PEN_AK

C.5 Added new ADA Class:

- SA-23A.(4)BTRY
- SA-23A.(6)BTRY
- SA-23B.(4)BTRY
- SA-23B.(6)BTRY

C.6 Added new SAL:

- SA-23A_SAL
- SA-23B_SAL

C.7 Added new SSM:

- 3K60.BAL(TEL)
- P70.LNCHR(2)

C.8 Added new Supply Categories:

- CL.V.AS-AGM183A
- CL.V.SA-LR-3.SA-23A
- CL.V.SA-LR-3.SA-23B
- CL.V.SS-RDR-MR-3K60BAL
- CL.V.TLAM-VA(MST)

C.9 Added new TW:

- 3K60.BAL(KH35)
- AGM183A.ARRW
- SA-23A.9M83ME
- SA-23B.9M82ME
- TLAM-VA(MST)

C.10 Renamed TW:

- MALD to ADM160B(MARL)
- MALD-J to AADM160C(MALD-J)

C.11 Changed TW Speed:

- HARPY.BOMB - Speed to 5,280 km per Day
- HARPY.UAV - Speed to 5,280 km per Day

C.12 Changed TW MID PHASE PROBABILITY OF DETECTION for:

- SSN33(ZIRCON) to 0
- DF-ZF-HGV to 0

C.13 Added new UOMs:

- 3K60BAL(KH35)
- AGM183A.ARRW
- SA-23A
- SA-23B
- TLAM-VA(MST)

C.14 Deleted UOM:

- MACH

C.15 Added New Sensors:

- ALQ218_AEE
- BILL.BOARDB_LAA
- GRILLSCREEN_LAF
- HIGHSCREENB_LAA

C.16 Changed Aircraft Speeds:

- Changed A/C HARPY.UAV - MAX Speed to 5,280 km per Day
- Changed A/C HARPY.UAV - Speed to 4,880 km per Day

C.17 Changed Aircraft Tanker Loads:

- A330.MRTT_TANKER load to 35,555 gals
- KC10_TANKER load to 50,370 gals
- KC130_TANKER load to 12,600 gals
- KC135R_TANKER load to 29,600 gals
- KC46A_TANKER load to 30,814 gals
- KC767_TANKER load to 23,700 gals

C.18 Created New A/C Loads:

- 12A158CX16A160B
- 12A158CX16A160C

C.19 Created New Combat Systems for Unmanned Vehicles

- M2.12.7MM-AUTO
- MK19.AGL.40MM-AUTO

C.20 Deleted Aircraft Load Assignments

- LIFTING_LOAD for all Fighter Aircraft

C.21 Removed Sensors from A/C Loads:

- Removed APQ122_AGT from RC135V.DEF
- Removed RVT.JOINT-C_AEC from RC135V.DEF
- Removed RQ4.SAR_AGK from RQ4.BLK30.GLOBAL.HAWK.DEF
- Removed RQ4.SAR_AGK from RQ4.BLK40.GLOBAL.HAWK.DEF

C.22 Updated SUP Torpedo Loads:

- L.ANGELES_NV_US
- L.ANGELES_VI_US
- L.ANGELES_VL_US
- OHIO_CM_US
- OHIO_NM_US
- VIRGINIA_US

C.23 Updated Values

- Intel Information Prototype

C.24 Added Air-to-Air Lethality Data:

- F22.RAPTOR - Added Air to Air Lethality Data to AIM120P3_AAL - for LDM-AS

- F22.RAPTOR - Added Air to Air Lethality Data to AIM120P3_AAL - for LDM-PS
- F22.RAPTOR - Added Air to Air Lethality Data to AIM120P3_AAL - for LDM-NJ
- F15C.EAGLE - Added Air to Air Lethality Data to AIM120P3_AAL - for LDM-AS
- F15C.EAGLE - Added Air to Air Lethality Data to AIM120P3_AAL - for LDM-PS
- F15C.EAGLE - Added Air to Air Lethality Data to AIM120P3_AAL - for LDM-NJ

C.25 Added Aircraft Class Attain Firing Position Data for:

- F22.RAPTOR - Added Aircraft Class Attain Firing Position Data for LDM-AS
- F22.RAPTOR - Added Aircraft Class Attain Firing Position Data for LDM-PS
- F22.RAPTOR - Added Aircraft Class Attain Firing Position Data for LDM-NJ
- F15C.EAGLE - Added Aircraft Class Attain Firing Position Data for LDM-AS
- F15C.EAGLE - Added Aircraft Class Attain Firing Position Data for LDM-PS
- F15C.EAGLE - Added Aircraft Class Attain Firing Position Data for LDM-NJ

C.26 AIRCRAFT TARGET CLASS Changes:

C.26.1 Created a 5GEN.FIGHTERS - AIRCRAFT TARGET CLASS for the following aircraft:

- F22
- F35A
- F35B
- F35C
- J-20
- SU-57

C.26.2 Renamed FIGHTER1 to 4GEN.FIGHTERS for the following aircraft:

- F-14
- F-15
- F-15E

- F-16
- F-2
- FA-18
- FA-18E
- FA-18F
- FA-50
- F-CK-1 CHING KUO
- J-10
- J-11
- J-15
- J-16
- JAS-39 GRIPPEN
- JF-17
- MIG-29
- MIG-31
- MIRAGE 2000
- RAFALE
- SEA HARRIER FA 2
- SU-27
- SU-30
- SU-33
- SU-35
- TEJAS
- TORNADO ADV

- TYPHOON
- YAK-130

C.26.3 Renamed FIGHTER2 to 3GEN.FIGHTERS for the following aircraft:

- F-4
- J-8
- KFIR
- MIG-23
- MIG-25
- MIG-27
- MIRAGE F1

C.26.4 Renamed FIGHTER3 to 2GEN.FIGHTERS for the following aircraft:

- ETENDARD IV
- F-5
- J-6
- J-7
- MIG-21
- MIRAGE 5

C.26.5 Renamed FIGHTER4 to 1GEN.FIGHTERS for the following aircraft:

- J-5
- MIG-15