JTLS

Version Description Document

April 2012

DEPARTMENT OF DEFENSE
DEPUTY DIRECTOR J7
JOINT AND COALITION WARFIGHTING
116 LAKE VIEW PARKWAY
SUFFOLK, VA 23435-2607

JOINT THEATER LEVEL SIMULATION
(JTLS 4.0.2.0)

[Blank Page]

ABSTRACT

The Joint Theater Level Simulation (JTLS) system is an interactive, multi-sided gaming system that models combined joint and coalition resource air, land, naval, and Non-Govenmental Organization (NGO) environments. This *JTLS Version Description Document (VDD)* describes specific features of the Version 4.0.2.0 delivery of the configuration-managed JTLS software suite.

JTLS 4.0.2.0 is a Maintenance release of the JTLS 4.0 series that includes a newly developed SBDKOR40 Standard Database, as well as major model functionality changes implemented as Engineering Change Proposals (ECPs), 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 JTLS. Corrections, additions, or recommendations for improvement must reference specific sections, pages, and paragraphs with appropriate justification and be forwarded to:

JTLS 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 Version Description Document* (VDD) describes Version 4.0.2.0 of the configuration managed Joint Theater Level Simulation (JTLS) software suite. JTLS 4.0.2.0 is a Maintenance delivery for the JTLS 4.0 series of releases. JTLS 4.0.2.0 includes the entire JTLS suite of software and the SDBKOR40 Standard Database that supports a realistic scenario based on the current Korea Peninsula theater of operations. Database modifications that were accomplished to upgrade the previous JTLS Standard Database to this current version are summarized in this chapter, as well as APPENDIX B. Detailed descriptions of Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS 4.0.2.0 executes on the Red Hat Enterprise Linux Version 5 64-bit operating system **Support for executing the model or its support programs within any 32-bit operating system environment has been discontinued.** The WHIP user workstation interface can be executed from any Java-compatible Web browser available on any operating system.

1.2 INVENTORY OF MATERIALS

This section lists documents and software that are relevant to JTLS. All JTLS documents included in this delivery are provided in PDF format within a documents subdirectory. DoD Military Standards can be accessed through the appropriate military channels.

1.2.1 Obsolete/Outdated Documents

Development of the JTLS Air Tasking Order Generator (ATOG) has been suspended. The *JTLS ATOG User Guide* (JTLS Document 02) has been removed from the JTLS documentation suite for this release.

1.2.2 Unchanged Documents

The model enhancements implemented for JTLS 4.0.0.0 are incorporated within the documentation provided for this release:

- *JTLS Analyst Guide* (JTLS Document 01, Version 4.0.1.0)
- *JTLS ATOT User Guide* (JTLS Document 03, Version 4.0.1.0)
- *JTLS Director Guide* (JTLS Document 07, Version 4.0.0.0)
- *JTLS Executive Overview* (JTLS Document 08, Version 4.0.0.0)
- *JTLS Installation Manual* (JTLS Document 09, Version 4.0.1.0)

- *JTLS PPS User Guide* (JTLS Document 13, Version 4.0.0.0)
- *JTLS Standard Database Description* (JTLS Document 14, Version 4.0.0.0)
- *JTLS Software Maintenance Manual* (JTLS Document 15, Version 4.0.0.0)
- *JTLS Technical Coordinator Guide* (JTLS Document 16, Version 4.0.1.0)
- *JTLS Entity Level Server User Guide* (JTLS Document 19, Version 4.0.0.0)
- *JTLS Federation User Guide* (JTLS Document 20, Version 4.0.0.0)

1.2.3 Updated Documents

- *JTLS Controller Guide* (JTLS Document 04, Version 4.0.2.0)
- *JTLS Data Requirements Manual* (JTLS Document 05, Version 4.0.2.0)
- *JTLS DDS User Guide* (JTLS Document 06, Version 4.0.1.0)
- *JTLS WHIP Training Manual* (JTLS Document 10, Version 4.0.2.0)
- *JTLS Player Guide* (JTLS Document 12, Version 4.0.2.0)
- *JTLS Version Description Document* (JTLS Document 17, Version 4.0.2.0)
- *JTLS C4I Interface Manual* (JTLS Document 21, Version 4.0.2.0)

1.2.4 New Documents

No new volumes are delivered with the JTLS 4.0.2.0 documentation suite.

1.2.5 Delivered Software Components

JTLS 4.0.2.0 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. These software components are included with this release:

- Combat Events Program (CEP)
- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)
- Database Development System (DDS)

- Terrain Modification Utility (TMU)
- Java Symbols Application (JSYMS)
- Lanchester Development Tool (LDT)
- ATO Generator Program (ATOG)
- ATO Translator Program (ATOT)
- ATO Retrieval Program (ATORET)
- Convert Location Program (XCONVERT)
- Count Critical Order Program (CCO)
- Graphical Database Program (GDP)
- 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)

    Air Tasking Order Viewer (ATOV)

- Entity Level Server (ELS)

- JTLS Operational Interface (JOI)
- KML Operational Interface (KOI)
- TBMCS/ICC Interface Program (JTOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- Database Configuration Program (DCP)
- DDS User Interface (DDS)

Instructions for installing JTLS 4.0.2.0 are provided in the *JTLS Installation Manual*. Installing a previous version of JTLS prior to installing JTLS 4.0.2.0 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.

1.2.6 Released Databases

This release includes these sample unclassified databases:

- The scenario developed as the Korea Standard Database and named SDBKOR40 is a large-scale, seven-sided scenario database reflecting the approximate starting positions of units involved in the Korea Peninsula theater of operations. This example scenario was developed using unclassified data sources and is consequently not completely accurate. Discrepancies among actual units and their locations are not detrimental to the intended purpose of this database, which is to provide a recognizable and realistic scenario that demonstrates the simulation capabilities and supports JTLS training.
- The scenario DEMSDBKOR40 is a reduced version of SDBKOR40 that requires fewer resources, loads faster, and is intended for use in demonstration, training, and testing environments that do not require the complete Standard Database.
- The scenario blank40 is the SDBKOR40 database with all force structure data removed, which can be used as a framework for building your customized database.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS 4.0.2.0 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 5 (ES), 64-bit architecture.

  Some JTLS components will not execute under Red Hat Linux Version 4.0. Therefore, model support for this OS version has been discontinued. However, Red Hat Linux 4.0 may be used on client workstations to execute the WHIP.

  Security Enabled (SE) Linux must not be enabled on systems used to execute JTLS 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 is currently not authorized.

- Operating system for client workstations (one of the following):

  Red Hat Linux Enterprise Edition Version 3

  Red Hat Linux Enterprise Edition Version 4

  Red Hat Linux Enterprise Edition Version 5

  CentOS Linux Version 4 or 5

  Windows 2000, XP Professional, Vista, or Windows 7 can be used only if the workstation is an external HTTP client of the simulation network.

- Java Version 1.6.0 is required (Build 25 is recommended) for all platforms and must be used to support all workstations.

- JTLS database tools require use of a certified Oracle database server and the full Oracle Client installation for runtime requirements. Refer to Section 1.6.2 of this chapter for additional installation details.

- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of Red Hat Linux ES 5.0.

- 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 Red Hat Linux ES 5.0 is sufficient.

- The Perl script language is used by the JTLS system and game setup scripts. The version of Perl included with Red Hat Linux ES 5.0 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.

- KDE Desktop support has been added to JTLS 4.0.2.0. Support of the GNOME desktop is continuing, and use of the KDE environment is optional. Details regarding the installation and use of KDE are provided in Section 4.4.3.2 of the *JTLS Installation Manual*.

- SIMSCRIPT II.5 (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS, because all JTLS 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 SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc. The following SIMSCRIPT II.5 versions are recommended for 64-bit Red Hat Linux Version 3.5

- ANSI C Compiler: It is not necessary to use a C compiler to execute JTLS. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS component programs. The C Compiler version delivered with Red Hat Linux ES 5.0 is sufficient.

- C++ Compiler: It is not necessary to use a C++ compiler to execute JTLS. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS HLA component programs. The C++ Compiler version delivered with Red Hat Linux ES 5.0 is sufficient

1.3.2 HLA Compliance

The JTLS 4.0.2.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS in an HLA mode. JTLS has moved to a new Federation Object Model (FOM) located in the $JGAME/data/hla directory. Federation testing of JTLS is not complete, but some initial tests with JCATS 11 have been accomplished. Future plans include expanding the federation capabilities.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) recommended for use with this release is RTI-NG-Pro-v7.0 or Pitch pRTI Evolved 4.3.1. However, this program is not included in the JTLS 4.0.2.0 delivery. Users may obtain a full installation package of the RTI software from either vendor: Raytheon Virtual Technology Corporation (http://www.virtc.com) or 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.

1.3.3 KML Operational Interface (KOI)

The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to 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 C4I Interface Manual describes requirements and procedures for using the KOI capabilities.

1.3.4 JTLS Air Tasking Order Translator (ATOT)

The ATOT executes in two modes:

- A file containing the translated orders is created and the Controller is responsible for submitting an appropriate READ ORDER FILE Order that submits these orders to the model.

- The ATOT connects to the Oracle-based Scenario Database Repository (SDR) and places the translated orders into the appropriate Order Entry Client (OEC) tables. The OEC is responsible for submitting the orders to the model.

The Simscript and Oracle library support required by both ATO-T modes are obtained separately. The Simscript support is currently provided with each JTLS release. The necessary Simscript libraries are released in the ~/bin_support directory for 64-bit Linux. The basic mode creates read.order files to be manually read into the CEP. To run the basic mode, users must obtain, install, and configure the most current Oracle Client. The advanced mode uses the OEC, which is managed with the ATOT forms. To run the advanced mode, users must have access to an Oracle server.

1.4 INSTALLATION CONSIDERATIONS

The procedures for installing JTLS 4.0.2.0 depend on the hardware configuration provided at the installation site. All installation issues are addressed in the *JTLS Installation Manual*.

1.5 DATABASE MODIFICATIONS

No database structure differences exist between JTLS 4.0.2.0 and any previous 4.0 series database, but some errors were discovered in the internally-held database procedures. For this reason, after installing JTLS 4.0.2.0, you must unload and reload any JTLS 4.0 series scenarios that you may have.

To upgrade your previously installed and modified JTLS 4.0.0.0 scenario for JTLS 4.0.2.0, you must unload and reload your scenario. This will recreate the modified stored procedures, database triggers, etc. Failing to do so will cause issues in DDS operations, such as renaming, copying, and deep copying existing records.

If you are upgrading to JTLS 4.0 from JTLS 3.4 or earlier, you must unload and reload your scenario after the modification is completed.

1.5.1 Database Upgrade

Users who currently possess a JTLS scenario (ASCII file set) compatible with a version earlier than Version 4.0.0.0 can use this recommended modification procedure after installing this new version of JTLS:

1. Create a new Oracle account for the scenario.

2. From the new JTLS account, load the scenario ASCII files to the newly created Oracle account.

This process creates the JTLS schema that matches the previous JTLS version. After all data are loaded to the database tables, the process modifies the schema to match the current JTLS version. For JTLS 4.0, this process supports JTLS 3.0 (or higher) series scenarios only. If your scenario version is older than 3.0, the scenario must be upgraded to Version 3.4 first, by using the JTLS 3.4 version upgrade process. Consult Section 1.5.2 of the *JTLS 3.4.0.0 Version Description Document* for details and procedures.

JTLS users must consider that the automatic modification program inserts default values to the new database fields. Also, a new JTLS version may use previously existing data fields in a different way. Consult the appropriate *JTLS Version Description Document* to identify data fields that must be updated. The Database Modify program is the first process to use to update your databases to the new JTLS version. Changing the values in new or changed data fields is equally important.

The JTLS scenario/database modification process requires 10.2.0.5.4 or higher full Oracle Client installation.

1.5.2 Standard Database Changes

Additional Standard Database changes that are applicable to JTLS 4.0.2.0 are described in APPENDIX B.

1.6 INSTALLATION

1.6.1 Installation Instructions

The *JTLS Installation Manual* included in the documents compressed TAR file that is part of this JTLS delivery provides detailed instructions for installing a new version of JTLS.

1.6.2 Oracle Compatibility and Installation

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

the Oracle Client installation. For the 64-bit version of JTLS, a 64-bit Oracle Client installation must be used. The JTLS scenario/database modification process also expects 10.2.0.5.4 or higher full Oracle Client installation. Some sites NFS mount their database server as Oracle Client; other sites prefer a full install of the Oracle Client to a different directory that mounts (simple NFS will suffice) to JTLS. 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 .cshrc file allows connecting to an Oracle database server (10.2.0.5.4 or higher - including 11g XE) running on any Oracle-certified database server platform.

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

Oracle Forms/Reports Developer 6i client/server runtime for DDS Forms and the optional iAS EE 10.1.2.0.2 full stack are no longer supported.

Refer to Chapter 5 of the *JTLS Installation Manual* for additional details pertaining to the Oracle installation.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes model capabilities added to JTLS 4.0.2.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-2009-10337 Remove Unnecessary MTF Message Formats

Summary of Model Change Request

Too many JTLS messages had MTF formatted versions. Messages that will never be transmitted to a Training Audience for electronic processing do not need defined MTF versions. Maintaining these unused MTF message versions was an unnecessary use of resources.

Design Summary

JTLS defines 338 individual numbered messages for Player and/or Controller WHIPs. Each of these can have several, or in some cases very many, submessages. Messages and their submessages are generated frequently and under various circumstances during a game. However, more than one-third of all messages are Controller-only messages. These messages do not need an MTF format version. Also, many of the non-Controller messages do not need MTF formats because are a type that would never be transmitted automatically to a Training Audience. Examples include most messages that simply report a Player order cannot be executed for some reason. Other examples are many of the messages that are responses to Player requests for various tactical and administrative game information.

This ECP removes the MTF definitions for a total of 220 such messages. The design of the WHIP Message Browser is such that MTF versions are not required for every message. If the WHIP user selects MTF as the desired Message Format, and then selects a message for reading that does not have an MTF format definition, the browser defaults to the English version of that message. The messages formatted as English are the master configuration managed set delivered with each JTLS release.

In addition to removing the unused MTF format items, the MTF format was modified for some Player messages that were not deleted. The "1045" message reports to Players the results of certain Controller actions, such as returning a destroyed unit to the game or repairing a target. This message and all its submessages (more than 20) were converted from the AKNLDG to the TACREP MTF format. This required changes to the 1045 MTF data file and also several other minor code modifications.

2.2 JTLS-2012-11154 Restart Unit After Increasing Allowed Move Hours

Summary of Model Change Request

A ground unit stopped moving after reaching its maximum move hours per day limit of 20 hours. The Controller increased the value to 24 hours, but the unit did not resume movement until the

beginning of the next hour. The unit should start moving immediately after the value is increased, if feasible.

Design Summary

A non-Naval unit in JTLS is limited in the number of hours per day it can perform various functions including move, attack, delay, withdraw, attach, or an engineering task. The attribute TUP MAX MOVE HOURS PER DAY holds this limit. When the Controller uses the Set TUP Parameters order to decrease this value sufficiently, all units using this TUP whose accumulated UT HOURS MOVING exceed this limit are placed into DEFEND posture and their current order is suspended until their "rest time" is completed. Similarly, if the maximum move value is increased sufficiently, a unit at rest (UT.RESTING flag is YES) is allowed to continue its suspended order if the accumulated UT HOURS MOVING are less than the limit, but not until the next Hourly Processing Event. The Set TUP Parameter - Thresholds order was enhanced to immediately allow, if criteria are met, a unit to resume its suspended order when the TUP MAX MOVE HOURS PER DAY is increased such that the UT HOURS MOVING is less than the new limit. The design allows the unit to resume movement only under certain conditions, such as not in combat unless attacking, not out of fuel, and not too weak.

2.3 JTLS-2012-11157 Save Pointers WIth Short Names Versus Long Names

Summary of Model Change Request

The CEP saves ia list of pointers for various game objects while taking a checkpoint. This file is not part of the checkpoint and not read back to the CEP, but is useful for debugging purposes. Instead, support personnel investigating a traceback, such as one generated by a logic errors, can use the file data to correlate pointers in the traceback to object names. The file data consist of pointers and the Long Names of objects, such as unit Long Names and Target names. Since most interface tools use object Short Names, unit Short Name and Target CCF numbers, the debugger must use the file to map a pointer to a Long Name, then map the Long Name to a Short Name to allow the object to be displayed on any interface. Storing only the Short Names and avoiding the mapping process is more efficient.

Design Summary

The file was modified to write Short Names only for Units and Targets. Other objects, such as Air Missions, Formations, or Networks, written to the file that do not use the Short Name and Long Name concept were unchanged.

2.4 JTLS-2012-11158 Default Break Off Order To Mission Exclude

Summary of Model Change Request

The WHIP allows an intercepting CAP mission to receive a Break Off order as a Quick order or via the full order palette. The order palette allows the WHIP user to specify the CAP mission that

should break off its intercept as well as a rule to be observed for future intercepts against the mission. The rule options specify whether the intercepted track can be intercepted again by the CAP, should not be intercepted again by the CAP, or should not be intercepted by any mission on the CAP's Side.

Design Summary

The first option is useful to enforce to no-fly zone. The track that violates the no-fly zone ROEs is therefore intercepted; it is no longer in violation and the CAP has been instructed to Break Off. However, if the track again violates the ROEs, the CAP will intercept again. This rule is the default option for the Break Off order and assigned when the Quick order is used unless the user explicitly overrides the default. Since the Break Off order is typically not used for no-fly zone enforcement, a CAP mission that receives a Quick Break Off order with the Mission Allowed setting will break off, identify the track as remaining a viable intercept, and start the intercept again.

Users can change this behavior by overriding the defaults on the Break Off order palette and setting the default rule to Mission Exclude. Since Mission Exclude is most commonly used, the Break Off order should use this default.

The Break Off order was modified to prevent the track currently being intercepted from being automatically intercepted again by the CAP mission as the default behavior when the order is sent.

2.5 JTLS-2012-11179 Target Quantity Exceeds Hash Table Size

Summary of Model Change Request

The Korea scenario database prepared for the AFAMS JTLS evaluation contains more than 70,000 targets. The existing hash table size in the CEP and SIP can accommodate a maximum of 65,535 targets. The table size must be increased to handle the larger target database and avoid a crash.

Design Summary

The original size of the hash table (an internal array) was limited to 65,656 under the previous 16-bit Simscript compiler. Now that JTLS is compiled exclusively under the 32-bit compiler, the hash table was upgraded to accommodate 100,000 objects. This change was implemented and successfully tested in the CEP and SIP running at the JDIF at Orlando, FL before implementation in JTLS 4.1.

2.6 JTLS-2012-11186 Disallow Turning AAR Flag Off When AARC Is Connected

Summary of Model Change Request

When the AAR Collection Flag was turned off while the AARC was running, the CEP stopped saving AAR data and the information was lost for the AARC.

If the AAR Collection Flag is On, the CEP can send AAR data to the AARC if AARC is running or save these data in the queue for AARC if the AARC client is not connected. The routine that sets the flag was modified to reject the order if the AARC client is connected and the Controller sends the SET MODELLING PARAMETER order to set the AAR collection flag to Off. The routine will generate the rejection message, but will reject the AAR flag only when other parameters were also set in the same order.

2.7 JTLS-2012-11187 Improved Availability Of Unit JU Numbers

Summary of Model Change Request

Online Player Manual (OPM) individual unit data include the unit's assigned JU NUMBER. This JU NUMBER is used for identification when the unit is reported to a Link-16 device. This data item was omitted when a Player received output from a SHOW UNIT GENERAL PARAMETERS order.

Design Summary

To provide consistent unit data, the message output from the SHOW UNIT GENERAL PARAMETERS order was modified to include the JU NUMBER, allowing Players to immediately access the value of the unit's assigned JU NUMBER.

2.8 JTLS-2012-11190 New WHIP Map Missile Type Filters

Summary of Model Change Request

Cruise Missile was the only missile type designated as a filter on the WHIP Map. This feature was incompatible with the model's capability to model Cruise, Ballistic, and Torpedo missiles because it filtered unanimously for all of these types.

Design Summary

A new generic entry for Missile was implemented to allow filtering on each of the Cruise, Ballistic, and Torpedo missile types. This design is similar to Air Mission and Target filters that operate on their various corresponding subtypes.

2.9 JTLS-2012-11207DDS Surface-to-Air Lethality Record Copy

Summary of Model Change Request

No copy functionality for Surface-to-Air Lethality for Aircraft Type by Altitude Zone Record existed. This capability was requested.

Design Summary

A new copy functionality and a Copy Selected Record button were added to the Surface-to-Air Lethality for Aircraft Type by Altitude Zone Record dialog.

3.0 SOFTWARE TROUBLE REPORTS

Software Trouble Reports (STRs) describe discovered and corrected JTLS program code errors. Code corrections implemented for previous JTLS releases have been tested with the model enhancements delivered with JTLS 4.0.2.0 and are included.

Errors that are identified and corrected for JTLS 4.0.2.0 and future Maintenance releases in the JTLS 4.0 series are described in this chapter. Italic text identifies each error description, which is followed by an explanation of the implemented solution.

STRs that remain outstanding from previous JTLS versions are listed and described in Chapter 4.0.

3.1 JTLS-2012-11147 MIssing GDP Flight Path Edit Layer

The WHIP displays a Flight Paths function, but the Graphical Database Program 4.0 did not implement a Flight Paths layer edit function.

Java code was modified to add a new Flight path button and layer edit function.

3.2 JTLS-2012-11148 Automatic Mode3 Hijack Squawk Assignment Possible

During training of international users, it was noted that it was possible for the model to automatically assign a Mode 3 squawk of 7500 to an air mission. This is the real world "I am being hijacked" squawk. Assignment to an air mission should be allowed, but automatic assignment by the model should not be allowed.

A recent ECP allowed specification of blocks of mode 3 squawks to be defined for use by a sides air missions. Players creating missions can either assign a specific Mode 3 squawk on the mission order, or allow the model to assign one from the allowed block of squawks (if such a block exists, its existence is not mandatory). While implementing this ECP, certain universally accepted real world squawks were prohibited from automatic assignment - 7700 meaning "Emergency", and 7600 meaning "Have Lost Communications". The 7500 squawk, meaning "Am Being Hijacked" was not prohibited from automatic assignment. It is now prohibited. Note that each of these three special squawks can still be specifically assigned by Players to individual air missions in support of exercise training objectives.

3.3 JTLS-2012-11149 JINN Did Not Integrate Packaging Functions

After exporting an Organization, JINN users were required to use a separate process to create a package for the JOBE.

Java code was modified to integrate these JINN Organization component packaging function processes:

- Unload a database.
- Execute the Scenario Initialization Program.
- Generate the Online Player Manual.
- Create the JOBE package.

3.4 JTLS-2012-11150 Order Required Railroad Name To Be Entered As A Text

During pre-exercise testing, a tester was unable to enter the name of a Railroad Network when trying to add railcars as a Controller. The field on the Set SMA Parameters order did not allow sufficient characters to accommodate the database name of the Rail Network.

The problem was that the order panel allowed only 15 characters for the name of the Railroad. Railroad names now can be as long as 40 characters. The main problem, however, was that the order should not have asked the user to enter the railroad name as a text. Railroads are JDS objects, and the order should have presented the user a pull-down list of railroads in the game. This has been corrected. Also, the DRM still listed the maximum length or a railroad name as 15 characters. This has also been corrected.

A related problem discovered during the same testing involved convoys transporting units. The Supply Run IMT Screen has a field designed to display the "Transported Unit". This field was always empty. This problem was traced to an error in the code that updates the JODA when the Supply Run entity is created. This error was also corrected.

3.5 JTLS-2012-11151 JINN Import Failed To Read Vocabulary Files

The JINN, which shares the code from JOBE, displayed Java exceptions while loading the Command Hierarchy to update the database because the JINN failed to initially read the database vocabulary files.

After the JOBE program implemented the OPM function, its initial read code was changed. The JINN copies and stores the dynamic_voc.xml in the ~/JOBE/update directory while creating a package for the JOBE. When the JINN imports the __dynamic.xml file returned from the remote client, the program reads the dynamic_voc.xml and static_voc.xml files.

3.6 JTLS-2012-11152 Data Requirements Manual Corrections

Several JTLS Data Requirements Manual (DRM) errors were discovered during a recent exercise and pre-release testing.

The DRM errors included missing variables, variables no longer used, and variable description changes. These variable descriptions were modified:

- FS.FS.RELATIONSHIP

- GS.ORGANIZATION.TYPE
- GS.CATEGORY
- GS.INDEX
- GC.NAME
- AC.AVG.MALFUNCTION.TIME (removed)
- AC.AVG.MALFUNCTION.REPAIR.TIME
- AC.AVG.DAMAGE.REPAIR.TIME
- AC.AVG.COMBAT.REPAIR.TIME (removed)
- ST.ELINT.NOTATION
- ST.ELINT.PULSE.CODE
- ST.ELINT.FREQ.VALUE
- JT.ELINT.NOTATION
- JT.ELINT.PULSE.CODE
- JT.ELINT.FREQ.VALUE
- PSS.TYPE
- PSS.PROTOTYPE
- PSS.NUMBER
- REFERENCE.DETECTION.DISTANCE
- MIN.END.OVERLAP
- MIN.HALF.WIDTH

3.7 JTLS-2012-11153 Submarines Enter Map On Surface

A surfaced submarine remained on the surface while entering the playbox from off the game board and was more vulnerable to detection. Submarines moving from off-map to on-map should automatically submerge if the depth is sufficient.

The off-board region surrounding the playbox does not have defined terrain. Therefore, oceans have zero depth and submarines are forced to the surface when they are outside the playbox. The existing movement logic attempts to retain the submarine's current depth, which could be the Player's intended depth in other situations.

New logic automatically submerges the submarine when it moves from off-map to on-map, provided the hex depth can accommodate the vessel's SUP OPERATIONAL DEPTH. This logic was also added to the Magic Move process to accommodate similar movement. The logic was further modified to submerge the submarine operating alone or part of a formation entering the map. Existing logic that automatically surfaces a submarine moving from on-map to off-map was not changed.

3.8 JTLS-2012-11155 Submarine Depth State Check Error

In at least two instances, CEP legacy code determined the depth status of a submarine by assuming that if the hex depth is greater than the SUP SUBMERGED DEPTH, the submarine must be fully submerged and therefore detectable only by sub-surface (sonar) sensors. That assumption was incorrect because the submarine could likely be on the surface or snorkeling, regardless of the hex depth.

Two routines used an obsolete method to determining a submarine's depth status based solely on the hex depth. One routine dealt with processing intelligence detections within a single hex, while another dealt with detection of objects within a Directed Search Area. Both routines were updated to use newer logic employed elsewhere in the CEP to determine whether the submarine is surfaced, snorkeling/periscoping, or submerged. The newer logic considers a surfaced or snorkeling/periscoping submarine to not be submerged for purposes of detection.

3.9 JTLS-2012-11156 Map Find Marked Undetected Foreign Object Locations

Wildcard searches on the WHIP Map Find function marked locations of undetected foreign objects. The results indicated the location of all objects matching the wildcard search string launched from a Player WHIP as displayed for the Controller WHIP.

Player and Controller WHIP searches were unfiltered and displayed all matching objects. Code was corrected to filter the results of Player WHIP searches to include and mark only Side-owned and detected object locations.

3.10 JTLS-2012-11159 AIr Mission Success Flag Not Saved On Checkpoint

Every Air Mission has a success flag attribute for the purpose of reporting a success code to TBMCS. This attribute was not saved during checkpoints. If the CEP was restarted from a checkpoint, all Air Missions had a success flag value of No (not successful), regardless of the setting prior to the checkpoint.

The success flag attribute was added to the data saved during the checkpoint procedure. Insert/Extract missions performing an HRU Insert also erroneously set the mission success flag to No when a successful insert was performed. This error was also corrected.

3.11 JTLS-2012-11160 Invalid Modify Weather Front Order Error Message

When any order panel parameter(s) except probabilities parameters were submitted, a Controller order Modify Weather Front error message stated that the order was rejected because all probabilities are zero.

The Modify Weather Front CEP routine was modified to recognize when no changes to any probability parameters are entered, which prevents generating inadvertent error messages. No changes to error messages were required.

3.12 JTLS-2012-11161 New Target Search List Not Processed For OAS Attack

The Change Mission Parameter (CMP) order allows the target search list to be replaced with a new priority list for certain mission types, including an OAS mission. If the OAS mission was converted to an Air-Ground Attack mission by an Assign Target order before the CMP order was sent, the CEP did not process the new target list. This error caused an empty CMP compliance message because no mission parameter was changed.

The underlying CEP subroutine that processes the Change Mission Parameter (CMP) order was modified to recognize an AIR ATTACK mission type when processing a new target type search list. This acceptance applies only when an original ORBITING OAS mission has been converted to an AIR ATTACK mission type within the model, because the CMP order panel design does not allow Players to enter a new target search list for an AIR ATTACK mission.

3.13 JTLS-2012-11162 BE Facilities OPM Page Displayed Slowly

The Online Player Manual (OPM) contains numerous HTML pages that describe data in the database. For BE Facility data, all BE Facilities were listed on a single page. The Web browser slowly loaded and scanned the page due to the large number of facilities listed in the HTML file.

BE Facility data were written to separate HTML files for each facility.

3.14 JTLS-2012-11163 Excessive Flight Path Nodes Caused Recursive Calls

During Steadfast Juncture 2011 at JWC, the database contained a robust flight path network comprised of more than 1500 nodes. The model read and filed these nodes and read the flight paths between each node. Each flight path had a starting and ending node that would then need to be found based on its name. Due to the large number of nodes, this was a time consuming process. After the exercise, the nodes were placed into a hash table for faster look-up and retrieval. As part of this implementation, the number of nodes, and thus the hash table size, were allowed to increase, since the Controller can create additional nodes.

A code error in the dimensioning of the hash tables created the array too small for a large number of nodes. The code properly recognized this condition and would re-invoke code to dimension the hash table and load it. However, the number of nodes was the same, the array was too small, and the process would repeat until the calling stack was full and the program would prematurely exit. This issue existed in the SIP and the CEP.

Two routines were changed to address the problem. The error in the dimensioning of the hash table was corrected to avoid the endless recursive calls. A comment was also added to the recursive calling section to better explain the concept of the recursion.

3.15 JTLS-2012-11164 No Victim Type For AAR Reporting; Logic Error

A Player can fire artillery or missiles at a location or air drop weapons on a location instead of specifying an object to attack. A Controller can order a weapon damage assessment against a location. In all cases, any potential victims within the blast radius will be assessed for damage. The AAR recognizes that the victim of the attack may be a location instead of an object. However, the code that called the AAR code always expected a victim and attempted to determine the type of victim when assessment was performed against a location. This caused a logic error, which is self-correcting protective code to handle an unexpected situation.

The code was modified to attempt to resolve the type of victim only if a victim is specified.

3.16 JTLS-2012-11165 OPM Icon Not Displayed With WHIP Browser

The Online Player Manual (OPM) HTML pages reference to an icon that resides within the $JGAME/data directory structure. This icon was not present on the system. Also, the reference to the icon was accurate when the HTML pages were opened from the disk, but was incorrect when the pages were opened from the WHIP and requested from the Apache server. This discrepancy pertains to the virtual directory structure used to allow the Apache server access to the data in $JGAME/data. Since this discrepancy cannot be resolved, correcting the WHIP access path to the OPM pages is preferable to correcting the disk access path.

A copy of the WHIP icon was created to use for the OPM pages. A copy of this icon is moved into the directory structure for the OPM pages and the individual pages reference the copy. The icon can be properly accessed whether the pages are viewed from the disk or from the Apache server.

3.17 JTLS-2012-11166 Erroneous JSYMS Upgrade Message

The JSYMS Save button must be selected to upgrade this file to update Organization types. An "Upgrade is done" message was displayed each time the Save button was selected.

A Save button integer variable was added to display the upgrade message only the first time.

3.18 JTLS-2012-11167 National Boundary Exceeded Packet Size

A 1Km hex terrain was developed and data that included National Boundaries was moved to the scenario from a scenario based on larger hex sizes. After reading the National Boundary, the CEP maps it to hex edges and sends the JODA a GRAPHIC object that contains the hex edge mapping. The smaller hexes resulted in a National Boundary mapping with more than 4,000 points. When this mapping was sent as a GRAPHIC object, the packet exceeded the JDSP packet size limit of 32,000 bytes. The JODA terminated and re-established the connection to the CEP, starting a new download that included the excess GRAPHIC packet, which caused the socket to be terminated.

A trial-and-error process determined that a JODA GRAPHIC object could hold 4,000 points and not exceed the packet size of 32,000 bytes. Code that was a remnant of the GIAC existed to separate an excessively large National Boundary into multiple GRAPHIC objects. The trigger point for this code was 32,000 points. This trigger point was modified to 4,000 points.

3.19 JTLS-2012-11168 ADA Site Attempted Firing Without Idle Launcher

A game crash occurred when a SAM/AAA site attempted to fire at an enemy Air Mission when no IDLE launcher was available at the site. This crash previously occurred but could not be replicated by means of a push to identify the specific circumstances. and therefore were unable to identify the exact circumstances and fix it. Duplicate representations of the target in the set of available-to-fire SAM/AAA sites is the likely cause of this crash.

Examination of the code revealed a possibility that would permit a SAM/AAA target to occur twice in the available-to-fire set twice. Every SAM/AAA target puts down hex effect markers in every hex that it can cover. These markers represent the first cut at determining which targets can fire on an Air Mission in a specific hex. A moving SAM/AAA target goes through all the new covered hexes and grabs one of the existing markers, removes it from its old hex, and files it in the new hex. As long as the number of covered hexes does not change, which it will not if the range of the SAM/AAA target does not change, this process works.

When a SAM/AAA target's range changed, the code checked whether the number of covered hexes also changed. If it did the existing markers were picked up and destroyed and a new set of markers were created and deployed. If there was no change in the number of covered hexes, nothing was done. However, the code that checked the number of covered hexes divided the range of the target by the average hex size and rounded the result, whereas the code that moves the marker truncated the same result. This could result in excessive hex markers existing that would not be picked up and moved when a SAM/AAA target moved, potentially resulting in two such markers in a single hex. Such a situation would cause the SAM/AAA target to be filed twice in the available to fire set and could result in the crash.

The code to check whether the new number of covered hexes is the same as the old number of covered hexes was modified to use the truncation function, making it consistent with the code that deploys the markers. As a secondary measure to prevent other similar circumstances

crashing the model, the code to find an Idle launcher was modified to check to ensure that a launcher is found, and if not produces a logic error and assigns a previous fired launcher.

3.20 JTLS-2012-11169 Same Warning Number Used For Different Messages

The routine that checks and generates error or warnings messages while matching the JTLS database against the initial TBMCS database assigned the same warning message number for three different messages.

If the JTOI is used to update the TBMCS, the JTOI generates a text file that contains initial TBMCS data items during the initialization process. The Alter program then checks the file content against the JTLS database and generates any errors or warnings. The errors/warnings list is used to modify the JTLS database to match the TBMCS database. The routine was modified to assign different warning numbers to different messages.

3.21 JTLS-2012-11170 SSM Subcategory Compared To Targetable Weapon

The code that changed the range of a Targetable Weapon checked all SSM targets to determine whether they used the Targetable Weapon. This check was erroneous because it checked the SSM Target's Subcategory to determine whether it was the same as the Targetable Weapon. The code should have been checking the set of weapons that the SSM Subcategory could use to determine whether any of them were the Targetable Weapon.

The code was modified to properly check the set of SSM weapons that could be fired to determine whether any of them were the Targetable Weapon that was changed. The purpose of this comparison is to determine whether any of the target ranges, which for SSMs are limited by the maximum range of the Targetable Weapons they can fire, need to be adjusted. The logic of assigning a new range was also modified to reflect the fact the SSM may have multiple Targetable Weapons that it can fire instead of using the single Targetable Weapon as a limiting value.

3.22 JTLS-2012-11171 WHIP Login Screen Exit Failed To Clear Java Process

Exiting the WHIP Login screen by closing the application window does not clear the Java process that is cleared when the Cancel button is used. This behavior can potentially leave many similar processes executing in the background to affect the performance of other executing programs.

This behavior was corrected by invoking a system exit instead, thus completely clearing the Java process. This correction also allows for consistency and expected behavior within the Login frame.

3.23 JTLS-2012-11172 JOBE Copy Issue

If a unit or target was copied more than once, the resulting new units or targets had the same Long Names and UICs if the object was a unit. This error violates unique key constraints implemented for the database.

Code was corrected to prevent multiple object copies from being assigned the same Long Names or UICs.

3.24 JTLS-2012-11173 JOBE IMT Issue

When a user selects a IMT row screen row that represents a map object, the JOBE SITREP window automatically displays the object's data. When a secondary column was used to sort the IMT, the SITREP displayed data pertaining to a different map object.

The code selected the incorrect "affected" object from an unsorted list when the related Java event was activated. This error was corrected.

3.25 JTLS-2012-11174 JOBE C2 Window Menu Issue

The C2 Command Hierarchy window menu was not displayed after the window was closed and redisplayed.

A Java exception was thrown when the code attempted to display the pop-up menu for a re-opened Command Hierarchy window. This error was corrected.

3.26 JTLS-2012-11175 Force Side Renaming Failed To Update Area Of Operations

The DDS Force Side renaming function failed to update the Area Of Operations table (the "ao_owning_fs" field).

The related Oracle package for Force Side renaming was modified to include the Area Of Operations table and update the ao_owning_fs field accordingly.

3.27 JTLS-2012-11176 JINN Verifies Target's Owned and Associated Unit

When a Target may have an Owned unit and Associated unit that are different. If each unit is selected by two different organizations, the Owned target and Associated target, which are the same, will be exported to two different Organizations' dynamic.xml files. If the attributes of either of target change in the JOBE program, JINN upload errors may occur.

During an Organization's export, the JINN checks each target's Owned and Associated units. If the target has Owned and Associated units that are not the same unit, the Associated target is not allowed to export, whether or not the Owned unit is selected.

3.28 JTLS-2012-11177 Template With No Combat Systems Crashed ELS

The SIP created templates for the ELS and created separate templates when more than one Combat System was present in the template. The SIP default created three sub-templates. When the SIP processed a template with two systems, one sub-template contained zero Combat Systems. The ELS crashed while processing unit updates that used the template.

The SIP processing rules were modified to create templates that match the number of Combat Systems.

3.29 JTLS-2012-11178 Assess Weapon Damage Air Strike Crash

The CEP crashed when the Controller sent an Assess Weapon Damage order that set the Area of Impact option to Aircraft Load.

The routine that reports the results of damage called a routine that saves observed damage results as part of the mission data for the Air Mission that delivered the weapons. This routine should not have been called. No such Air Mission existed because the Assess Weapon Damage code does not create a mission, but generates an External Event. This error existed in code that creates the associated Controller Damage Report message. These errors were corrected.

3.30 JTLS-2012-11180 BE Units Arrive After Game Start

Basic Encyclopedia (BE) Facilities are collections of Objects (Units and/or Targets) that represent important enemy assets that can be the focus of intelligence collection and/or attacks associated with JPDIs. Before exercise execution, BE Facilities are typically defined in the JTLS database during the scenario build process. As pre-established entities, the Units belonging to BE Facilities should be present in the game at time zero. However, the Internal Look 2012 database included several BE Units that had TPFDD arrival times later than time zero. This situation caused a CEP crash.

It does not make any sense for a BE Facility to exist in the game from the beginning without its owned Units also being present. A Warning 631 message was added to the Scenario Verification Program to flag any Units belonging to BE Facilities that have TPFDD arrival times later than game time zero.

Also, the model was changed to only report on units and targets associated with a BE facility if they are on the game board. For example, it is possible for the Controller to Magic Move a target and leave it off the game board for a period of time. These targets should not be subject to detection solely because they are associated with a BE facility.

3.31 JTLS-2012-11181 Incorrect MISREP TGTPOS Record

The final Mission Report (MISREP) lists the position of the mission's primary target in a record labeled Target Position or TGTPOS. This record was incorrectly filled when the mission was a JTLS Multi-Target Attack mission.

Code was modified to access the target position from the first item listed in the Multi-Target Attack mission's Firing Array. After a proper target is obtained, the remainder of the code fills the record properly.

3.32 JTLS-2012-11188 JINN Scenario Deletion Option

The JINN deletes the working scenario from the $JGAME directory at end of the Organization Package process. A user reported running the game while using an incomplete working scenario and simultaneously running a packaging process. Running both processes concurrently has occurred during database building and testing. Therefore, the JINN should not delete a scenario from the $JGAME directory without notifying the user.

A Delete Scenario in JGAME Directory option was added to the end of the Package script.

3.33 JTLS-2012-11189 Detach By Same TUP As Parent Yielded Illogical Units

One of the DETACH order options is to Detach a unit by Tactical Unit Prototype (TUP). This created a problem when a Player sent an order that specified the same TUP for the Child unit to be created as that defining the Parent unit.

There were two problems with this. The first case is when the order specifies detachment by TUP, Prorated. "Prorated" means that the TUP quantities are adjusted to reflect the current Parent unit strength. In this case, the Child unit would take all supplies and Combat Systems that the Parent owned. The Parent then indicated zero values for all SC and CS attributes, including the TOE values. The model therefore showed the Parent at 100% strength. Since the TOE values were all zero, the unit had its full allotment of zero of everything and was shown at "full" strength.

The second case was that the order specified detachment by TUP, Non-Prorated. In this case, the Detach order was invariably rejected. Detach orders are rejected if the Parent cannot provide what the Child needs. Detaching by TUP, Non-Prorated, means that the Parent must give the Child the FULL TOE quantities based on the specified TUP. The Parent would only be capable of doing this if it were at 100% strength or greater. This would occur only if the Parent artificially received extra Combat Systems without associated TOE adjustments.

Detaching by TUP using the same TUP as the Parent Unit is illogical and can lead to very unrealistic situations. The model rejects such a Detach order. The user can still detach by Fraction, in which case the Child unit receives the same TUP as the Parent, but each unit has its TOE values adjusted so that the resulting two units are defined realistically.

3.34 JTLS-2012-11191 SVP WARNING 213 Written To ERROR Section

The JTLS Menu Verify an Existing Scenario Database option executed on the SDBKOR40 database generated an SVP Error/Warning output listing that showed all WARNING 213s written to the ERROR section, not the WARNING section. This caused confusion when the listing was reviewed. This code error may also have allowed WARNING 213 to be written if WARNING 213 was deselected in the Option 2 SVP Warning choices.

The SIP code in the cheperand.sim routine that created WARNING 213 set the warning or error flag for "Is this a warning message?" as .NO before calling the finerr.sim routine. Code was

modified to set this flag to .YES. WARNING 213 correctly appears in the WARNINGs section of SVP output listing.

3.35 JTLS-2012-11192 Misleading Ground Route Report Estimate Message

An JTLS user reported an apparent inconsistency in the message response to a GROUND ROUTE REPORT query. The message includes an estimated travel time along the route and also an estimated arrival time at the end of the route. These values did not correlate on the message.

The user observed a estimated time of arrival that was significantly later than the game time that would result if the specified estimated travel time were added to the current game time. The user also observed that when the unit was ordered to move along the route, the travel time was significantly greater than the estimate.

Investigation revealed that the reported values correlated. The key was that user requested an estimate for a route that would take several days to traverse. What The code is designed to compute both items for the message. First, it computes an estimate of the travel time along the route as if the unit were to continuously move. This is the "estimated travel time". The code then considers that each type of unit has an allowed maximum number of hours to move during any 24-hour period. This is a Tactical Unit Prototype (TUP) attribute. This required rest time is used to compute the "estimated time of arrival". The message reported both items, but did not make this distinction clear to the reader. The message was reworded.

As for the fact that the actual travel time took significantly longer that the estimate, this also made sense. Actual travel time, especially over a several-day route, depends upon the tactical situations encountered along the route. It also depends upon the availability of fuel to keep the unit moving. This is especially true if the unit is moved a significant distance from its support unit. Relevant wording was also added to the revised estimated message.

The CEP routine that processes the estimate order and writes the messages was changed. However, the change only reordered the data sent to the Message Browser, to match the sequence they are used on the submessage. The estimate computation methods were not changed.

3.36 JTLS-2012-11193 Implement JINN Setup/Package Options Interface

When a user generates an Organization package, the JINN unloads the database, runs a script to set up the scenario, then generates a online manual. Finally, the JINN runs a packager script.

Generating online manual cannot proceed unless a copy of the scenario exists in the /game directory. For the database building and testing purposes, the latest scenario is used. No flexibility was allowed for users to select package options, such as using an existing game scenario or online manual.

A Setup menu button and Setup Scenario Options interface were added was added to the Organization component. The Setup Scenario Options verifies whether the $JGAME/ and $JGAME//onlineman directories exist. The component provides Unload/Setup, Generate Online Manual, and Generate a Package options. These options combine seven different selections.

3.37 JTLS-2012-11194 JTLS Menu Lists Merged Long Scenario Names

When a user attempts to start the Count Critical Orders (CCO) program using the text-based JTLS Menu, a list of scenario names is provided. This list is displayed as a series of rows and columns. In some cases, the name of a scenario appeared to be combined with the next name. Names in the columns were not separated.

When various JTLS programs were started, the lack of separation between the names occurred only when one of the scenario names contained than 10 characters. The script used to start each of the affected programs printed the scenario names with a fixed width of 10 characters. This width was increased to allow a maximum of 15 characters for scenario names.

This error was not limited to the CCO program and affected the startup of several JTLS programs. All scripts used to start these affected programs were modified to correct this issue.

3.38 JTLS-2012-11196 DDS SAM/AAA Target MGRS Lat/long Conversion Error

Entering a valid MGRS address (15SUD0370414710) to a SAM/AAA Target location generated a Java exception.

The Ground unit performed the conversion correctly. Comparing the two stacks revealed BasicTableCellEditor was employed for the SAM-AAA-TARGET editor, which caused the editor to pass the straight MGRS string to the LatLon routine in its raw format. LatLon must receive the location value in Lat/Long form, not MGRS. The "basic" verses "location" instance found in the SAM-AAA-TARGET GUI XML file was changed to "location", which resolved the issue.

3.39 JTLS-2012-11197 Message Used As Broadcast and Directed Type

From the CEP, Message 1100 is used to reflect a state of "Cannot Comply with Controller Order". This message was intended to be sent as a Directed message to a specific Controller. In some cases, it was incorrectly sent as a Broadcast message and delivered to many Players. When a message is sent in Broadcast format, it is difficult to filter it for a customized message view or for delivery to a real-world device.

A thorough search of the CEP code was accomplished to identify all cases where Message 1100 was improperly used as a Broadcast message. The code was modified such that all instances are converted to Directed messages. In most cases, the message is directed to the interface program that originally sent the order. If this information is not available, the message is sent to the primary Controller.

3.40 JTLS-2012-11198 DDS Exception While Adding Target Pickups

This two-part error involved an uninitialized variable used for comparison when tables were empty or non-selected row adds were performed, which was apparent in several of the cell editors. The related error resulted from an incomplete conversion from using a file designed for variables defined with doubles to use with integers.

First-time entries to the External Events Target Pickup table depended upon the former value of a cell being available to determine the type of value it contained. This was resolved by initializing the appropriate comparison variable accordingly. Completing the conversion of the file designed to handle integers instead of doubles resolved the second issue.

3.41 JTLS-2012-11199 PSYOP Operation Against Naval Unit Crash

A psychological broadcast jammer mission was sent to target a naval unit. This resulted in a game crash when personnel were adversely affected. The same mission against a ground unit had no adverse affect on the model.

A code error for the updating of the personnel combat system was discovered. The code looked for the Supply Category Equipment Item .CATEGORY.PERSONNEL within the unit's list of Equipment Items. This was not found. This value is actually a defined constant meaning 7, which would indicate Supply Category 7, Ground Fuel, was being updated. The naval unit did not have an entry for this Supply Category, which resulted in the crash. The code was modified to use the variable CATEGORY.PERSONNEL (leading dot omitted) which indicates the proper Supply Category (Number 310).

3.42 JTLS-2012-11200 Rail Movement Incorrectly Identified As Infeasible

When two different rail networks pass through a specific sending hex and receiving hex and one of those networks is infeasible because a loading does not exist, the model did not check the other rail network for feasibility. The model reported the rail movement of the supplies could not be accomplished.

The model was modified to examine all rail networks that pass through the sending and receiving hex, not only the first such network encountered.

3.43 JTLS-2012-11201 DDS Exception Involving Lists In Add Mode

A Java exception appeared when a user included a table column in the Add Record option of the XML. The column must also have the criteria of "not equal" validation and uses a list of items from the database (referenced as "type=reference" in the editor tag) to populate its field. When all of these conditions were met, the DDS threw an exception with an uninitialized variable.

The DDS failed to call its configure routines, which initializes base level variables with values necessary to continue editing a list returned by the database. Providing the appropriate values for these routines to function correctly resolved the issue.

3.44 JTLS-2012-11202 Reformat Spreadsheet Program (RSP) Write Error

When the RSP processed a Controller Change Target order, the completed order file failed the Order Verification Tool (OVT) validity check and the read order file could not be submitted to the model.

When the RSP processed any order containing a text field allowed to contain embedded blanks, the text was incorrectly written to the order file with ampersand delimiters (e.g., &text text text&). The OVT expects the text field to be delimited by double quotes (e.g., "text text text"). The offending RSP write statement was corrected to use double quotes, not ampersands, for this situation.

3.45 JTLS-2012-11203 Air-to-Air Combat Kills Displayed On C4I

Aircraft killed by Air-to-Air combat were not removed from a Link-16 C4I display.

The routines that release the Link-16 tracks held by an Air-to-Air killed aircraft and the Link-16 tracks held by others on the dead aircraft were not being released. The code to release the track numbers and the Link-26 Source numbers was centralized into the Destroy Mission routine. In the new code, it does not matter whether the kill was due to a Controller kill, Air-to-Air kill, Surface-to-Air kill, lack of fuel kill, or a runway crash landing; the tracks are not properly adjusted and the missions are removed from the C4I display.

3.46 JTLS-2012-11206 DDS Naval Unit Copy Failed When ICAO Defined

The DDS Naval Unit record copy operation failed when the ICAO field was defined for the selected record. Only the copy option is valid for units. Therefore, the existing Deep Copy option for unit screens was misleading.

The database trigger designed to verify whether the ship types CARRIER, HELO_CAPABLE, AMPHIB_HELO_CAPABLE, SUPPLY_HELO_CAPABLE, have a valid ICAO value for the specific record was failing. The database trigger (trig_nv_icao_check) was modified. The unit screens were modified to display only the Copy Record option. The Deep Copy icons were removed from the screen definitions.

3.47 JTLS-2012-11208 Sinking Ship Deployed Lifeboat Crash

A JTLS 4.0.1.1 naval vessel was heavily damaged and began sinking. Afterward, the CEP crashed when the first lifeboat was about to be deployed.

Based on the trace output, the model crashed because of a reference to an unreserved array in the subroutine that deploys a single lifeboat. The array pointer that indicates whether the lifeboat's supplies have been reported to the JDS was assigned to the HRU. However, a typographical error in the statement omitted the full name of the HRU attribute that holds that pointer. Execution of this statement would cause a crash when any sinking ship attempted to deploy a lifeboat for the first time. The statement was corrected.

3.48 JTLS-2012-11209 CEP Crash While Splitting AWACS Mission With Tracking

A model crash occurred during an exercise when a user attempted to split a two-aircraft AWACS mission into two separate missions. The original mission was being used in conjunction with a Link-16 system and the tracking attributes were properly set for this function in the model. The model crashed immediately after the user sent the order to split the mission. Detailed crash information was provided for tracing the source of the issue.

When the Air Mission with Link-16 capabilities was split, the parent mission was retained and a new mission was created. Since the parent mission had Link-16 tracking capabilities, the model attempted to reassign the appropriate tracking information from the parent mission to the new mission. This reassignment of tracking data was not appropriately processed, causing the model to use a single data field to hold the Link-16 data for both missions. The model crashed when one mission moved away from the other. Code was modified to ensure the tracking information for the parent contains data only for the parent's subset of aircraft from the original mission when the mission is split. The new tracking data are assigned to the new mission. Proper tracking for both Air Missions can continue without further model issues.

3.49 JTLS-2012-11210 XMS Did Not Fulfill Time-Based Requests

The restarted XMS relies on the message cache files in each checkpoint to fulfill requests for past messages from the WHIP. When the XMS filled a request from the Message Browser with time criteria specified after a restart, the service searched only for messages that matched criteria of the first checkpoint and did not continue to consider successive checkpoints. This issue manifested only when checkpoints were taken and the XMS was restarted.

The XMS code was corrected to continue searching all checkpoint-cached messages for messages that match the time-based request criteria.

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.

Correction of the remaining STRs, however, must be postponed to a later version due to time and resource constraints. These problems may be corrected prior to the next release of JTLS. If an immediate need arises for code corrections to remedy any of these outstanding STRs, such as an exercise planned to occur before the next release, contact the JTLS Development Team Leader. Refer to the Abstract of this document for the current address.

Code errors described in this chapter should be noted specifically because they affect the basic functionality of JTLS. Information is provided regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

4.1 JTLS-0942 Air Transport Cannot Combine Wet And Dry Supplies

When both wet and dry supply categories are included in the same Transport Instructions List for an Air Transport mission, they will not be transported at the same time. The first supply category shipment type will be loaded, but the second will not. If both are included in the same Supply List, the wet category is preferred. The aircraft go through the motions as if loading and delivering the denied category, including MISREP confirmation. No pickup or delivery is made, although an empty storage area may be created. There is no documentation to support this situation, and the user is not notified of the problem.

4.2 JTLS-0956 MPP Messages For Canceled Missions In Error

If an airbase is magic moved with several squadrons on active missions that need to be canceled or with squadrons in the middle of a self lift, the subsequent message generated for the situation has several errors. The changes required are too risky during the exercise. The problem will not cause a crash, but will cause the MPP to incorrectly display the message contents.

4.3 JTLS-0961 Group Ground Move Delayed To Lead Unit

There is a problem when a group ground move is sent. The directive is delayed to the lead unit. When the lead unit learns about the move, it immediately tells the units in the follow-on group. This could lead to directives being received out of order. Assume the user sends a directive at 0100 and the CEP determines the lead unit should receive the message at 0200. The lead unit cannot receive any other directives until after 0200. The CEP ensures that directive receipt is in the same order as the user sent the directives. This is not true for the follow-on units. If the user sent an order at 0115 directly to one of the follow-on units, the follow-on unit could receive the 0115 directive prior to the order sent at 0100. If this error is causing problems for upcoming exercises, the Configuration Manager should be contacted for a code fix to solve this problem.

4.4 JTLS-0968 Inconsistency Between Regular Run And Pusher

There is a major inconsistency between a regular run and a run created using pusher. When an order with ASAP is sent, the READ KEYWORD routine sets the data parameter to TIME.V. When pusher reads in the order, TIME.V is much earlier than it was when the order arrived in the first place. For orbiting missions and alert missions, this alters when they will go off alert by a great deal. This must be fixed and made consistent. It appears that both TIME.V and order receipt time must be saved to the ci1 file to accomplish this task.

4.5 JTLS-0971 Ship Continuous Tracking Not Working

The new unidentified object design indicates that ships which are continuously tracked will not have unidentified objects created. A continuously tracked Naval unit and all of its targets are creating unidentified objects. They should not be doing this.

4.6 JTLS-0973 Periodic Report Air Supplies And Fuel Not Correct

The arrays which hold air supply usage are not being maintained correctly given the new MISSION RESOURCE ALLOCATION event.

4.7 JTLS-0974 Submarine Detection By Ground Sensors

A moving submarine does not get full credit for coverage time by sonars on board other ships or submarines. It gets full coverage time for airborne sensors but not ground based sensors.

4.8 JTLS-0981 Formation With No Posture

The model crashed when a formation reached a Destination Two hex and the formation no longer had a posture. Therefore it did not know what to do. A Destination Two hex indicates that the Formation should conduct its assigned Amphibious Operation, drop off its Sealifted supplies or clear mines from a minefield. The posture of the formation is used to tell the formation which of these three tasks should be accomplished.

When the formation reached its Destination Two hex, the posture of the formation was zero; therefore, the formation did not know which of the three tasks should be accomplished.

4.9 JTLS-1384 Area, Target, And Unit Report Documentation

Some users have indicated that the documentation of Area Report, Unit Report, and Target Report similarities and differences are incomplete or inaccurate. A review of this documentation is needed.

4.10 JTLS-2005-1457 Target Auto Assign Errors In Orbiting OAS

An Orbiting OAS mission was created with Auto Assigned allowed and search target category as SSM. An SSM was magic moved to the area of the orbit. Perception of the SSM was given to the

mission's side using the controller Target Report order. The mission saw the target and immediately attacked it. When the OAS mission returned, the Mission Report said it was assigned to attack target UI011816U but the target could not be found. It appears that mission was previously holding the target as unidentified and it was not found after it was assigned a specific target number.

4.11 JTLS-2011-10810 SE Linux Causes JSXR Crashes

Frequent and random JXSR Web Service crashes occurred while executing JTLS 3.4.3 in an exercise network configuration supported by a Security Enhanced Linux-enabled platform. The system configuration included 7 JXSRs and 125 WHIPs. Java 1.6.0.16 was used to support workstations. Accumulation of numerous core.jxsr files in the $/game directory and logic_error files in the $/debug directory was observed. Server CPU resources and system memory were adequate and not adversely affected.

This issue did not occur after SELinux was disabled on the system servers. Until this issue is resolved, executing JTLS or its components on SELinux-enabled systems is not authorized.

APPENDIX A. ABBREVIATIONS AND ACRONYMS

Terms are included in this Appendix to define their usage in JTLS 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
ATOG	Air Tasking Order Generator
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
GDP	Graphical Database Program
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
JCW	Joint and Coalition Warfighting (formerly JWFC: Joint Warfighting Center)
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
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)
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	Materiel 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
NCSA	National Center for Supercomputing Applications (University of Illinois)
NEO	Noncombatant Evacuation Operations
NFS	Network File Server
NGO	Non-Governmental Organization
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 4.0.2.0 STANDARD DATABASE CHANGES

Major Changes include:

- New Tws were added.
- New Supply Categories were added.
- New SUPs were added.

B.1 New Targetable Weapons (TW) Added

- TW - TLAM_E
- TW - PATRIOT.2 GEM-C
- TW - PATRIOT.2GEM-T
- TW - PATRIOT.3MSE

B.2 New Supply Categories Added

- Supply Category - CL.V.TLAM-C
- Supply Category - CL.V.TLAM-D
- Supply Category - CL.V.TLAM-E
- Supply Category - CL.V.SA-SR-3MSE
- Supply Category - CL.V.SA-LR-GEMC
- Supply Category - CL.V.SA-LR-GEMT

B.3 Surface to Air Lethality (SAL) Added

- SAL - PATRIOTGEMC_SAL
- SAL - PATRIOTGEMT_SAL
- SAL PATRIOT3MSE_SAL

B.4 Calibers Added

- New Caliber - 3FGM40SGS

B.5 Ship Unit Prototypes (SUPs) Added

- SUP - GERALD.FORD_US (Carrier)
- SUP - KSS-2_KS (Submarine)
- SUP - KDX-3_KS (Destroyer - DDG)
- SUP - DOKDO.LPH_KS (Amphib Transport Dock - LPH)
- SUP - SAMBONGHO.PS_KS (Patrol Ship)
- SUP - INCHEON.FF_KS (FFX Frigate)
- SUP - YUN.YOUNGHA.PKGA_KS (PKG-A Missile Patrol Boat)
- SUP - GUMDOKSURI.PKGB_KS (PKG-B Missile Patrol Boat)
- SUP - SHI.LANG_CH (Carrier)
- SUP - YUAN_CH (Submarine)

B.6 SUP Changes (CG Ships)

Ships changed from SUP TICONDERO_VL_US to TICON_SM3_US:

- CG61_US
- CG67_US
- CG70_US
- CG72_US
- CG73_US

B.7 SUP Changes (DDG Ships)

Ships changed from SUP BURKE.1_US to BURKE.1.SM3_US:

- DDG52_US
- DDG53_US
- DDG54_US
- DDG55_US

- DDG58 _US
- DDG59_US
- DDG60_US
- DDG61_US
- DDG62_US
- DDG63_US
- DDG65_US
- DDG67_US
- DDG68_US
- DDG69_US
- DDG70_US
- DDG71_US
- DDG72_US
- DDG73_US
- DDG76_US
- DDG77_US

B.8 SUP Changes (DDG Ships)

Ships changed from KONGO_JA to KONGO_SM3_JA (SUP):

- DDG173_JA
- DDG174_JA

B.9 New Ships Added To Scenario

- CV83.SHI.LANG_CH (SHI.LANG_CH - SUP)
- SSG330_CH (YUAN_CH - SUP)
- SSG331_CH (YUAN_CH - SUP)

- SSG332_CH (YUAN_CH - SUP)
- SS72_KS (KSS-2_KS - SUP)
- SS73_KS (KSS-2_KS - SUP)
- SS75_KS (KSS-2_KS - SUP)
- DDG991_KS (KDX-3_KS - SUP)
- DDG992_KS (KDX-3_KS - SUP)
- DDG993_KS (KDX-3_KS - SUP)
- LPH6111.DOKDO_KS (DOKDO.LPH_KS - SUP)
- PS5001_KS (SAMBONGHO.PS_KS - SUP)
- FFG811_KS (INCHEON.FFX_KS - SUP)
- PKG711_KS (YUN.YOUNGHA_KS)
- PKG712_KS (YUN.YOUNGHA_KS)
- PKG713_KS (YUN.YOUNGHA_KS)
- PKG715_KS (YUN.YOUNGHA_KS)
- PKG716_KS (YUN.YOUNGHA_KS)
- PKG717_KS (YUN.YOUNGHA_KS)
- PKG718_KS (YUN.YOUNGHA_KS)
- PKG719_KS (YUN.YOUNGHA_KS)