

JTLS-2018-13797 Expand Allowable Length Of Classification

Ellen F Roland, Zafer Aktan, John Ruck

1.0 Summary of Model Change Request

Many of the classifications that are needed to support coalition exercises are not as easy as “Secret” or “Top Secret”. Many of the needed classification markings must include the countries and/or organization to which the data can be released. The current classification text string can only be 20 characters and no spaces are allowed. At least 40 characters are required for Game Classification. The database parameter needs to be expanded to accommodate this requirement.

2.0 Design Summary

The Design team questioned several users and based on that input the decision is to increase the maximum length of the full-name classification from the current value of 20 characters to 50 characters. In addition, the classification will be allowed to contain spaces.

3.0 Detailed Design

3.1 Data Table Changes

Since the design calls for increasing the size and flexibility of the database parameter, there is no need to worry about altering the existing value of classification as part of the database update procedure. The `jtls_create_tables.sql` command file, which creates the JTLS relational database tables, will be changed to allow the user to specify up to 50 characters and input spaces in the classification string as needed. All other character restrictions, such as the use of the slash (/) character, ampersand (&) character, and double quote (“) will remain in effect.

3.2 Program Changes

The following code changes are needed to support this change:

- The Database Development System (DDS) needs the following changes:
 - a. The data download capability will output the classification string in the global definition file surrounded by the ampersand (&) character.
 - b. The data upload capability will expect the classification to be surrounded by the ampersand (&) character.

- The Scenario Initialization Program (SIP), which is used to verify data consistency, setup a scenario for execution, and generate the Online Players Manual (OPM), needs the following changes:
 - a. The SIP needs to read the classification using the proper formatted read.
 - b. The SIP OPM capability needs to change the Header and Footer format to account for the larger possible classification.
 - c. The interface file to the JTLS Transaction Operational Interface (JTOI) needs to hold the classification surrounded by the ampersand (&) character.
 - d. The interface file to the JTLS Reformat Spreadsheet Program (RSP) needs to hold the classification surrounded by the ampersand (&) character.
- The Combat Events Program (CEP) needs the following changes:
 - a. The CEP needs to read the classification using the proper formatted read.
 - b. The CEP needs to write out the classification surrounded by the ampersand (&) character.
 - c. The interface file to the JTLS Transaction Operational Interface (JTOI) needs to hold the classification surrounded by the ampersand (&) character.
 - d. The interface file to the JTLS Reformat Spreadsheet Program (RSP) needs to hold the classification surrounded by the ampersand (&) character.
 - e. The headers and footers on the CEP generated Lanchester Battle Summary Report must be altered to handle the larger classification.
 - f. Two messages contained the classification, the NATO ENESITREP and the NATO OWNSITREP. These messages need to be altered to properly hand a game classification that contained spaces.
- The Interface Control Program (ICP) needs to be changed to properly read in the classification from the database global values file. All of the web services that read the configuration files created by the ICP do not require any changes.
- The Web Hosted Interface Program (WHIP) needs to change the font associated with the classification so it can be properly displayed on the WHIP Map Component.

3.3 Other Changes Implemented To Support Design

As noted in [Section 3.2](#), the interface files for the JTOI and the RSP need to be changed because the classification can now have spaces. There is one other interface file that contains the Game Classification, namely the Strategic Operational Interface (SOI). The SOI was used to interface JTLS with the Joint Deployment Logistics Model (JDLM). This interface has not existed for years.

Instead of changing the interface file, which could never be tested anyway, the Design Team decided to remove the SOI Interface file from JTLS. Neither the SIP nor the CEP will write out a SOI Interface File as a result of implementing this ECP.

4.0 Data Changes

No data based format changes are required to implement this design. The data table definition has changed to increase the number of allowable characters from 20 to 50. In addition, the definition was changed to allow the user to enter spaces in the data field.

5.0 Order Changes

Because the SOI Interface File was removed from the model, the WRITE INITIALZIATION FILE Order was changed, to remove the ability for the user request a regeneration of the SOI Initialization File.

No other order changes are required as a result if this design. Currently the user cannot simply submit an order to change the game classification. The procedure to change Game Classification during execution is documented in the Technical Controller Guide. This procedure has not changed as a result of this ECP.

6.0 JODA Changes

No JODA Data System parameter, structure, or protocol changes are required to implement this design.

7.0 Test Plan

7.1 Test The Entire Process

Purpose: The purpose of this test is to insure that the entire JTLS system can be executed when a Game Classification with over 20 characters and spaces can be brought up and properly executed.

Step 1: Create a database with a Game Classification that has over 20 characters and spaces. As an example, "UNCLASSIFIED FOR OFFICIAL USE ONLY (FOUO)" is a reasonable test classification.

Step 2: Unload the database.

Step 3: Reload the database.

Expected Results: No bad files should be generated as a result of the load procedure. The classification should look as it did when the database was unloaded.

Step 4: Run the SIP Verification procedure.

Step 5: Run the SIP Setup procedure.

Step 6: Run the SIP OPM procedure.

Step 7: Bring up the ICP and set up a game with the basic web services and WHIPs. Do a save of the ICP data.

Step 8: Bring up the CEP.

Step 9: Bring up the Web Services Manager (WSM)

Step 10: Using the WSM bring up the defined web services.

Step 11: Start several WHIPs

Expected Results: Each procedure should progress normally. The proper Game Classification should show in each Web Service Status tab in the WSM. The WHIP Map Component should show the full Game Classification.

Step 12: Bring up at least one message in the WHIP's Message Browser.

Expected Results: The proper Game Classification should show in the Message.

Step 13: Conduct a Stop Checkpoint

Step 14: Restart the CEP from the Stop Checkpoint

Expected Results: The checkpoint and restart should progress normally. The Game Classification should properly show on the each Web Service Status tab, on the WHIP Map Component, and on messages displayed within the Message Browser.

7.2 Test The RSP Capability

Purpose: The purpose of this test is to insure that the RSP can properly execute with the newly formatted interface file.

Step 1: Bring up the RSP for a scenario that has a Game Classification over 20 characters and spaces.

Step 2: Create a simple spreadsheet to use for the RSP execution. An easy spreadsheet may be to set a short list of units to a new weighted strength, This spreadsheet would only need the name of a unit and the newly desired weighted strength.

Step 3: Run the RSP and submit the orders to the executing scenario.

Expected Results: The program should execute normally and the generated orders should be accepted by the model and properly executed.

7.3 Test The JTOI Capability

Purpose: The purpose of this test is to insure that the JTOI can properly execute with the newly formatted interface file.

Step 1: Configure a scenario that has a Game Classification over 20 characters and spaces to run the JTOI.

Step 2: Bring up the JTOI.

Expected Results: The JTOI should connect to TBMCS or ICC as expected.

7.4 Test The Lanchester Battle Report Capability

Purpose: The purpose of this test is to insure that the CEP generates a Lanchester Battle Report that has properly formatted headers and footers with the newly extended Game Classification.

Step 1: Bring a game up with a Game Classification that has over 20 characters and spaces.

Step 2: Change the ROE for some Ground Units and start a Lanchester Battle.

Step 3: Have the Controller turn on the Lanchester Battle Report Flag

Step 4: Execute the model long enough to get a few Lanchester Battle Reports.

Step 5: View the Lanchester Battle Reports in a terminal window,

Expected Results: The Lanchester Battle Reports should have a proper header and footer that displays the full Game Classification.

7.5 Test The NATO Report Capability

Purpose: The purpose of this test is to insure that the extended Game Classification with spaces does not cause an issue with the NATO ENSITREP and NATO OWNSITREP Report.

Step 1: Bring a game up with a Game Classification that has over 20 characters and spaces.

Step 2: Submit a NATO REPORT Order to periodically generate ENESITREP.

Step 3: Submit a NATO REPORT Order to periodically generate OWNSITREP.

Expected Results: The reports should be generated and the contents of the reports should be readable within the Message Browser.

7.6 Test The Message Delivery Program (MDP) Capability

Purpose: The purpose of this test is to insure that the MDP works properly with the newly extended Game Classification database parameter.

Step 1: Bring a game up with a Game Classification that has over 20 characters and spaces.

Step 2: Using the MDP GUI, set up to output one type of message to the file system. Use a header and footer file that includes Game Classification

Step 3: Have the model generate the select message type.

Step 4: Using a terminal window view the contents of the message that the MDP placed on the file system.

Expected Results: The MDP should place the selected message on the file system as specified in the GUI. The contents of the message should look as expected and the Game Classification properly lists in the message header and footer.

7.7 Test Changing Game Classification Capability

Purpose: The purpose of this test is to insure that the process to change Game Classification during Game execution still works as documented.

Step 1: Bring a game up with a Game Classification that has over 20 characters and spaces.

Step 2: Using the documentation in the Technical Controller's Guide change the game classification.

Expected Results: After following the steps in the Technical Controller's Guide, all JTLS processes should show the newly specified Game Classification.