Accomplishments to Date

* (Feb 10-11) Local dev machine setup.
  + Install MySQL, build a3\_wasteland schema, install extDB (as .dll install), and configure existing mission to use extDB as the persistence framework.
  + Test this configuration and validate that current build is working correctly
* (Feb 12) Establish gitHub repository for g4-gamers missions (repo name: g4-gamers-Wasteland)
* (Feb 13) Incorporate territory persistence prototype (iniDBi only) and extend to support extDB persistence
  + Mission Files modified:
    - \territory\server\monitorTerritories.sqf
    - \territory\server\\territoryPayroll.sqf
    - \territory\client\updateConnectingClients.sqf
    - \A3Wasteland\_settings\main\_config.sqf (add A3W\_territorySaving, A3W\_territoryLogging options)
    - \server\default\_config.sqf (add A3W\_territorySaving, A3W\_territoryLogging options)
  + Mission Files added:
    - \persistence\server\world\tLoad.sqf
    - \persistence\server\world\default\getTerritories.sqf
    - \persistence\server\world\default\saveTerritory.sqf
    - \persistence\server\world\extDB\saveTerritory.sqf \*\*\* START HERE \*\*\*
    - \persistence\server\world\extDB\getTerritories.sqf
    - \persistence\server\world\tSaveInit.sqf
  + extDB modifications:
    - add Territory persistence, logging tables to a3\_wasteland\_db schema
      * update extdbModel,mwb to include new tables, relationships & indicies.
      * generate new schema creation .sql as a3wasteland\_db\_v2.04.sql
      * create schema 2.03->2.04 update sql as a3wasteland\_db\_v2.04.sql
    - update \extDB\db\_custom\a3wasteland.ini to include SQL for the following handlers:
      * newTerritory
      * getServerTerritoriesCaptureStatus
      * newTerritoryCaptureStatus
      * updateServerTerritoriesCaptureStatus
* (Feb 15) Initial testing of work so far, debugging tLoad & getTerritories scripts so that getTerritories will load existing territory data recs from the TerritoryCaptureStatus table, and create new records for territories not existing for the ServerID+MapID combo.
  + Tested and working though extDB call to exec the newTerritoryCaptureStatus sql in getTerritories returns \_markerID as a single element array, instead of the desired integer value. Need to look at the persistence\server\setup\extDB\async\_database.sqf to see whether return val needs to be modified or if I’m calling it incorrectly.
* (Feb 16) Tasks:
  + Figure out \_markerID return
  + Start work on \peristence\server\world\extDB\saveTerritory.sqf to have it correctly fetch and save the correct data elements for a territory
    - Use \p\s\e\saveVehicle as model
      * This uses two extDB sql calls:
        + 1st checks if a vehicleID exists in the db for the vehicle and creates one if it doesn’t exist
        + 2nd uses *updateServerVehicle* sql taking vehicleID and a set of \_updateValues encoded with extDB\_pairsToSQL (p/setup/extDB/fn\_pairsToSQL)

extDB\_pairsToSQL takes [[\_veh,\_flying] call fn\_getVehicleProperties,0] as inputs

fn\_getVehicleProperites pulls data from the vehicle obj as stores as an array of [STRING key, value] pairs

2nd param (0|1) switches return value format:

0 : format for INSERT/UPDATE

1: format for ON DUPLICATE KEY UPDATE

[0,1]: format to return both as array

Returns STRING(s) containing

0: \_setValues: comma separated key=value set

1: \_refValues: comma separated key=VALUES(key) set

* + - * + Here, recs will exists, we’ll be UPDATE’ing so will call extDB\_pairsToSQL with an array of [STRING key, value] pairs and return option 0, i.e., [ [[key,value]],0] call extDB\_pairsToSQL
    - saveTerritories: returns nil after call to save with:
      * \_updateValues = format["Occupiers=%1,SideHolder=%2,TimeHeld=%3",\_currentTerritoryOccupiersUIDs,\_currentTerritoryOwner,\_currentTerritoryChrono];
      * [format ["updateTerritoryCaptureStatus:%1:", \_markerID] + \_updateValues] call extDB\_Database\_async;
  + Update territoryPayroll to use revised form of currentTerritoryDetails

Notes:

* tLoad returns array of the form:

// 0 = Marker ID

// 1 = Name of capture marker

// 2 = List of players in that area [uids]

// 3 = List of players in that area [player objects] (set to null array)

// 4 = Team owning the point currently

// 5 = Time in seconds during which the area has been held

// 6 = Time in seconds during which the area has been contested (set to 0)

* this is returned to monitor\_territories which assigns it to the global currentTerritoryDetails, which *currently* expects the form:

// 0 = Name of capture marker

// 1 = List of players in that area [uids]

// 2 = List of players in that area [player objects] (set to null array)

// 3 = Team owning the point currently

// 4 = Time in seconds during which the area has been held

// 5 = Time in seconds during which the area has been contested (set to 0)

* + e.g., default: [\_markerName, [], [], sideUnknown, 0, 0]
* need to update all gets and sets of currentTerritoryDetails, config\_territory\_markers & ~~A3W\_currentTerritoryOwners~~ to include the MarkerID for use in subsequent db calls

MySQL install and setup:

http://www.mysql.com/why-mysql/windows/

Create the database:

extDB pack with config and stuff:  
<https://github.com/A3Wasteland/Release_Files/raw/master/A3W_extDB_pack.zip>

* Contains a3wasteland\_db\_v2.03.sql for creating the database

1. Extract everything from this ZIP to your Arma 3 install dir

2. Run the a3wasteland db SQL script with your MySQL tool of choice

3. Open extdb-conf.ini and put your MySQL connection infos in the [A3W] ection

4. Try to start your server, and hope it doesn't blow in your face

MySQL Workbench file:  
<https://github.com/A3Wasteland/Release_Files/raw/master/extDB/a3wasteland_db.mwb>

Extdb Setup: Windows

<https://github.com/Torndeco/extdb/releases>

Download the latest Windows.rar for extDB Archive has normal / debug / test versions inside

1. Copy either normal version to /path/to/arma3 i.e(windows/30/\* -> /path/to/arma3)
2. Edit your arma3 launch parameters add @extDB to your mod line.
3. Edit extdb-conf.ini and edit your settings etc...

extDB will only kill Server if config file is missing (from v26+).

**Note:**  
The Debug Version is extra logging, otherwise just use the normal version, this is just for performance reasons. This build is for extDB <https://github.com/Torndeco/extdb>

To setup extDB you need to use the Windows pre-compiled build and edit the extdb-conf.ini Database 2 field with your Database name and login SQL user details to match pretty much your Arma2MySQL one. Make sure the following files are in the root directory of your ARMA 3 server files (where the exe is):

extdb-conf.ini sqlite.db tbb.dll tbbmalloc.dll

You can drag the extdb.dll to that folder as well or you can copy over @extdb over as a regular mod. If you choose to use @extdb as a regular mod your commandline will be: -mod=@life\_server;@extdb

If you are not using @extdb as a mod and choose to drag extdb.dll into the main arma 3 server directory then you only need @life\_server in the commandline.