**Contents**

1. RESA Account
2. Setup
   1. Update ATO Config Files
      1. atoairmap.dat
      2. atobasemap.dat
      3. atoid.dat
      4. atoloadaugment.dat
      5. atotimeshift.dat
      6. atoweaponmap.dat
      7. atocountrycode.dat
   2. Create ato input directory, if needed
   3. Clean directories and files
   4. Create ATO WS 7 window
   5. Create ATO procedure window
   6. Create ATO fix window
   7. Create ATO Permissions window
   8. Notify AWSIM of CEIS account(s)
3. Start
   1. ATO WS
   2. ATO procedure
   3. ATO Permissions
4. Periodically manually process ATO input files
   1. PROCESS
      1. Log into CEIS account on the ALPHA1 server:
      2. When an ato mail message has been received
   2. MORE INFORMATION
      1. Files received with only atoid and update number
      2. Files received that start with ACO
      3. LASTFILE/PERMFILE
      4. Directory structure within the atoinput directory.
5. ATO Missions Process (atomissions.py) further information
   1. ATO Mission Files generated
      1. <mission>.pre
      2. *<*mission*>*.usmtf
      3. *nm.pre*
   2. ATO Output Reports
      1. atoerrors.out
      2. atomissing.out
      3. atostats.out
      4. atoheader.out
6. ATO Workstation commands
   1. ATO ASTAB ORDERS
   2. WS ORDERS
      1. Add aircraft to base
      2. Add weapon to base
      3. Replenish weapon on base
7. resabashrc file
8. **RESA Account**

* Currently the process is started and controlled from the resa account.
* RESA software directory: $RESA\_TRUNK Note: Use gresa to change to source directory
* Game directory: $RESA\_HOME/game Note: Use gwar to change to game directory
* NOTE: There is no longer a separate navy account, nor an ‘ftp’ directory. There is now an ato input directory (see section 2.2)

1. **Setup**

* On Alpha1 (or machine that is to process ATO input files.
* Either directly or with ssh -X resa@alpha1
  1. Update ATO Config Files
* The config files are located in the game directory
* Normally do NOT need changed/updated between events other than timeshift.
* The basic format rules:
* # in column 1 is a comment
* Whitespace between fields
* One line for each action
* No action continuation on multiple lines
* Mapping are in <from ATO> to <to RESA> order
  + 1. atoairmap.dat
* Map aircraft types
* RESA uses ATO aircraft type if no mapping.
  + 1. atobasemap.dat
* Map launching and recovery platforms
  + 1. atoid.dat
* The last line in this file contains the unique suffix character added to missions of the associated atoid
* This file is automatically updated when new ato received.
* This file should be deleted before start of exercise.
  + 1. atoloadaugment.dat
* This file augments a mission load based on aircraft type and mission.
* Aircraft type and mission fields accept wildcard \*.
* Use repeating <quantity><item> for augmentation items
* This file is specifically designed to augment ASW missions with passive and active sonobuoys. This is needed because sonobuoys are not considered weapons and are not included in ATO.
  + 1. atotimeshift.dat
* This file will shift time from ATO by given days hours and minutes.
* Typically this file will contain one line

0 -9 0

* Which converts ato local time (India time) into Zulu Time.
* The exercise always uses Zulu time
  + 1. atoweaponmap.dat
* Maps weapon types
  + 1. atocountrycode.dat
* Country codes after TSKCNTRY keyword in raw ATO data that are not to be processed.
* Usually the file only contains one line for KS (S. Korea) since S. Korea is a federate.

TSKCNTRY KS

* 1. Create ato input directory, if needed

1. Create a directory that is not too complicated to remember. **It must be all lowercase**. This will need typed and accessed (i.e. shorter the path and name, the easier) E.g.
   * **mkdir** **/opt/resa/resa\_build/atoinput**
2. Operator will create sub-directories as needed within this parent directory. Once received, AWSIM provided ATO files will be copied to the directories.
   1. Clean directories and files
3. There are TWO directory structures used. The first is the input data and uses the directory described in 2.2. The second is ato working directory and is where ATO files are actually processed.
4. If the ato input directory is not newly created, delete or move any sub-directories
5. Ato working directory
6. This is the directory that contains the processed ATO data files. It will include the orders that were created and sent to RESA, along with errors.
7. **gwar**
8. **cd ato**
9. delete or move any sub-directories, prepare for new event.
10. atoid
11. This file is created/updated as ATO’s are processed
12. **gwar**
13. delete atoid.dat OR delete all lines within the file
    1. Create ATO WS 7 window
14. Label window
15. Note: Could use a different command WS (e.g. 8). If doing so make appropriate changes in subsequent steps.
16. This workstation will be used as the surrogate for processed orders. Orders created will be submitted beneath this WS identification.
    1. Create ATO procedure window
17. Label the window
18. This window contains the actual ATO process, it will awaken periodically to handle any new posted input data. Results of processed input data are shown here.
    1. Create ATO fix window
19. Label the window
20. Use to look at ato errors and to manually correct/fix issues.
    1. Create ATO Permissions window
21. label the window
22. This window contains a process that will awaken periodically to ensure access is granted (obsolete need) and to indicate the input data is ready to be processed.

NOTE: in the future this window/process may be removed, but for now it will continue to be used.

* 1. Notify AWSIM of CEIS account(s)

1. 1 or more CEIS accounts should be provided to operators in OSAN to receive the input files
2. These are the accounts belonging to the controller(s) responsible for moving files appropriately for processing.
3. See section 4 for operator responsibilities.
4. **Start**

* On Alpha1 (or machine that is to process ATO input files)
* Either directly or with ssh -X resa@alpha1
  1. ATO WS

1. In ATO WS window
2. **gwar**
3. **vsta 7 c as=1 config=cciox** 
   1. ATO procedure
4. In ATO procedure window
5. **gwar**
6. **atomissions -rt /opt/resa/resa\_build/atoinput 7**

* -rt Don’t automatically replenish supplies, Don’t process tlam (cruise missile) shots.
* /opt/resa/resa\_build/atoinput is the input directory created in 2.2
* 7 is the ato workstation.

1. Options:
   * **-r** Do NOT automatically replenish supplies
   * **-t** Do NOT process TLAM (cruise missile) orders
   * **-p** Do NOT create prescript file
   * Options are combined. E.g. -r or -rt or -rpt
2. See section 5 for more information
   1. ATO Permissions
3. In ATO Permissions window
4. **cd /opt/resa/resa\_build/atoinput**
5. **/opt/resa/resa\_build/resang/trunk/atopermissions.py** **/opt/resa/resa\_build/atoinput**

* /opt/resa/resa\_build/resang/trunk/atopermissions.py is the executable
* /opt/resa/resa\_build/atoinput is the input directory

1. **Periodically manually process ATO input files**

* On Alpha1 (or machine that is to process ATO input files)
* Uses a directory structure inherited from AWSIM at Osan
* **REQUIRES** controller manipulation to put files in correct location!
  1. PROCESS
     1. Log into CEIS account on the ALPHA1 server:

1. Open firefox
2. Open correct IP based upon the classification (releasable korea or releasable Japan)
   1. Rel-K

<https://22.135.51.67>/OWA

* 1. Rel-J

https://22.9.80.183/OWA

1. Log into CEIS account
   1. Rel-K prepend “k10k\” to username. E.g. k10k\richard.william.ctr
   2. Rel-J prepend “k10j\” to username. E.g. k10j\richard.william.ctr
      1. When an ato mail message has been received
2. Open message to see name of attachment(s)
3. In a window of ALPHA1 create the necessary sub-directories:
   1. Change directory to the ato input parent directory
   2. If the atoid sub-directory doesn’t exist, create it
   3. Change directory to the atoid sub directory
   4. If the update number sub-directory doesn’t exist, create it
4. Save the mail message attachment(s) into the update number sub-directory
5. Copy a blank file LASTFILE into the update number sub-directory OR create a file by starting an edit session then saving the file. E.g **vi LASTFILE** then do a **wq** 
   1. NOTE, after processing this will be renamed PERMFILE.
6. Monitor the ATO WS window, if orders eventually do not scroll by, use the ATO fix window to investigate cause and take corrective action
7. For CRUISE missiles (if enabled) it may be necessary to move the nm.pre file to the game directory and then start it with “**prescript nm.pre 7 Y**”
   1. MORE INFORMATION
      1. Files received with only atoid and update number

* Contain the mission requirements or updates to missions
* E.g. KWPB.0 , KWPB.1, KJJC.2
  + 1. Files received that start with ACO
* Contain the route information (airspace definitions) or updates
* E.g. ACO.KWPB.0 , ACO.KJCE.2
* Should always have an initial ACO file
  + 1. LASTFILE/PERMFILE

As either one or both of the input files will be received (ACO and ATO), this empty file with the name “LASTFILE” is used to indicate that the sub-directory has all the input files.

NOTE: This file will be renamed to PERMFILE as part of the automated processing.

* + 1. Directory structure within the atoinput directory.

<*atoid*> {4 letters, although can be more}

<*atoid update number*> {0-9, 0 for the initial}

<*atoid.atoid update number*>

ACO. <*atoid.atoid update number*>

LASTFILE

Example:

KWPB

0

KWBP.0

ACO.KWPB.0

LASTFILE

1

KWPB.1

LASTFILE

Example full path:

/opt/resa/resa\_build/atoinput/KWPB/0/ACO.KWPB.0

/opt/resa/resa\_build/atoinput/KWPB/0/KWPB.O

/opt/resa/resa\_build/atoinput/KWPB/0/LASTFILE

/opt/resa/resa\_build/atoinput/KWPB/1/KWPB.1

/opt/resa/resa\_build/atoinput/KWPB/1/LASTFILE

1. ATO Missions Process (atomissions.py) further information

* This procedure controls the entire ATO process.
* This procedure is designed to run un-interrupted for the duration of the exercise.
* The process moves input files from the input directory structure to a mirror directory structure beneath /opt/resa/resa\_build/game/ato where the data is processed and output files are created.

* 1. ATO Mission Files generated
     + Located in the ATO working directory under game/ato/*<atoid*>/*<atoid update>*
     1. <mission>.pre
* This file contains the RESA orders for the single specified mission. The USMTF mission definition is contained within comments below the RESA order.
* This file may be modified to correct any problems with the mission and then re-entered with cntrl-f
* Missions flown by AWSIM do not have a <mission>.pre file.
* AWSIM flies all missions with type: XINT, SCAS< SACTK, SAEW, AEW, EW, WW
  + 1. *<*mission*>*.usmtf
* This file contains the USMTF mission definition which is also appended to <mission>.pre file.
  + 1. *nm.pre*
* This file contains airspace definition orders, RESA orders for all missions and replenish orders if requested.
* Since every airspace definition and mission are preceded by a delete, this file may be re-entered before 6AM. Re-entry of all missions is only needed if there is a problem affecting many missions.

* 1. ATO Output Reports
* Located in ATO working directory e.g. $RESA\_HOME/game/ato/KSA2/1
  + 1. atoerrors.out

Summary of insufficient or no resources for ATO.

Input is atostats.out and error message appended to input records.

* + 1. atomissing.out

List of missing missions determined by comparing missions in BB with missions in ATO

* + 1. atostats.out

Summary of resources needed for ATO by platform

* + 1. atoheader.out

Contains atoid and update number for output to printer

1. ATO Workstation commands
   1. ATO ASTAB ORDERS
      * *SHOW AIR ATO 1 BLUE 4*

This displays the pending missions

* + - SHOW AIR ATO 1 *BLUE* 3

This is sub view for CM launches

* + - SHOW AIR MISSION 1 BLUE 4

This displays aircraft mission history

* + - SHOW AIR EVENTS 1 BLUE 4

This displays aircraft after resource allocation and in flight

* 1. WS ORDERS
     1. Add aircraft to base
* If an aircraft type is missing or insufficient aircraft, use ENTER AIR to add available aircraft to the base.
* See $RESA\_HOME/db/kr15/kr15a.txt (force file) for ENTER AIR orders at game initialization. (Change kr15 to match current scenario.)
* ENTER AIR <2 letter side number prefix> <starting side number sequence> <side number count> <ac type> <launching platform> <time flown> <up time> <squadron>
* Example: ENTER AIR GW 100 20 FA18C NNGW 0 0 GW100

NOTE: the 2 letter side number prefix uniquely identifies launching platform. The side numbers must be unique!

* + 1. Add weapon to base
* FOR <force> ADD [|CHAFF | COUNTERMEASURE | CRUISE | TORPEDOES | WEAPONS] ITEM <qty> <weapon>

NOTE: The type of item is optional. This is only needed to resolve different types of items with the same name.

* + 1. Replenish weapon on base
* FOR <force> REPLENISH TO <qty> <weapon>

1. resabashrc file

* The file $RESA\_HOME/resabashrc is called by .bashrc
* This file contains environment variables, aliases for RESA programs, Linux command short-cuts and station definitions.
* Commonly used command shortcuts are:
  + gwar – go to game directory
  + gresa – go to resa source code directory
  + ldir – list directories in current directory