

uHelmScope: Scoping on the IvP Helm

June 2018

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

1 Overview	1
2 The Helm Summary Section of the uHelmScope Output	2
2.1 The Helm Status (Lines 1-8)	2
2.2 The Helm Decision (Lines 9-11)	3
2.3 The Helm Behavior Summary (Lines 12-17)	3
3 The MOOSDB-Scope Section of the uHelmScope Output	3
4 The Behavior-Posts Section of the uHelmScope Output	4
5 Console Key Mapping and Command Line Usage	5
6 Helm-Produced Variables Used by uHelmScope	6
7 Configuration Parameters for uHelmScope	7
8 Publications and Subscriptions for uHelmScope	9
8.1 Variables Published by uHelmScope	9
8.2 Variables Subscribed for by uHelmScope	9

1 Overview

The **uHelmScope** application is a console based tool for monitoring output of the IvP helm, i.e., the **pHelmIvP** process. The helm produces a few key MOOS variables on each iteration that pack in a substantial amount of information about what happened during a particular iteration. The helm scope subscribes for and parses this information, and writes it to a console window for the user to monitor. The user can dynamically pause or alter the output format to suit one's needs, and multiple scopes can be run simultaneously. The helm scope in no way influences the performance of the helm - it is strictly a passive observer.

The example console output shown in Listing 1 is used for explaining the **uHelmScope** fields.

Listing 1.1: Example uHelmScope output.

```
1 (alpha)(PAUSED)===== uHelmScope Report ===== DRIVE (133)
2 Helm Iteration: 85
3 IvP Functions: 1
4 Mode(s):
5 SolveTime: 0.00 (max=0.01)
6 CreateTime: 0.00 (max=0.01)
7 LoopTime: 0.00 (max=0.02)
8 Halted: false (0 warnings)
9 Helm Decision: [speed,0,4,21] [course,0,359,360]
```

```

10   speed = 2
11   course = 114
12 Behaviors Active: ----- (1)
13   waypt_survey [21.29] (pwt=100) (pcs=6) (cpu=0.08) (upd=0/0)
14 Behaviors Running: ----- (0)
15 Behaviors Idle: ----- (1)
16   waypt_return [always]
17 Behaviors Completed: ----- (0)
18
19
20 # MOOSDB-SCOPE ----- (Hit '#' to en/disable)
21 #
22 # VarName          Source          Time  Commtty  VarValue
23 # -----
24 # DEPLOY            pMari..iewer  25.05  alpha    "true"
25 # IVPHELM_STATEVARS pHelmIvP      5.42   alpha    "DEPLOY,MISSION,RETURN"
26 # MISSION           n/a           n/a     alpha    n/a
27 # RETURN            pMari..iewer  25.05  alpha    "false"      8
28
29
30 @ BEHAVIOR-POSTS TO MOOSDB ----- (Hit '@' to en/disable)
31 @
32 @ MOOS Var          Behavior      Iter  Value
33 @ -----
34 @ BHV_STATUS        waypt_return    1     name=waypt_return,p..te=idle,updates=n/a
35 @ -----
36 @ CYCLE_INDEX        waypt_survey    1     0
37 @ VIEW_POINT         waypt_survey    1     x=60,y=-40,active=f..r=red,vertex_size=4
38 @ VIEW_SEGLIST       waypt_survey    1     pts={60,-40:60,-160.._size=4,edge_size=1
39 @ WPT_INDEX          waypt_survey    1     0
40 @ WPT_STAT           waypt_survey    84     vname=alpha,behavio..es=0,dist=30,eta=15

```

There are three groups of information in the [uHelmScope](#) output on each report to the console - the general helm overview (lines 1-17), a [MOOSDB](#) scope for a select subset of MOOS variables (lines 20-27), and a report on the MOOS variables published by the helm on the current iteration (lines 30-40). The output of each group is explained in the next three subsections.

2 The Helm Summary Section of the uHelmScope Output

The first block of output produced by [uHelmScope](#) provides an overview of the helm. This is lines 1-17 in Listing 1, but the number of lines may vary with the mission and state of mission execution. This block is virtually identical to the appcast report generated by the helm itself. So another way of doing [uHelmScope](#) style scoping is with an appcast viewing tool ([uMAC](#), [uMACView](#), and [pMarineViewer](#)). But with these tools, you would only see part of the information found in [uHelmScope](#). The MOOSDB-Scope and Behaviors-Post portion of [uHelmScope](#) is not part of the appcast report posted by the helm.

2.1 The Helm Status (Lines 1-8)

The integer value at the end of line 1 indicates the number of [uHelmScope](#) reports written to the console. This can confirm to the user that an action that should result in a new report generation has indeed worked properly. The integer on line 2 is the counter kept by the helm, incremented on each helm iteration. The value on Line 3 represents the the number of IvP functions produced by the active helm behaviors, one per active behavior. The solve-time on line 5 represents the time, in

seconds, needed to solve the IvP problem comprised the n IvP functions. The number that follows in parentheses is the maximum solve-time observed by the scope. The create-time on line 6 is the total time needed by all active behaviors to produce their IvP function output. The loop time on line 7 is simply the sum of lines 5 and 6.

The Boolean on line 8 is true only if the helm is halted on an emergency or critical error condition. Also on line 8 is the number of warnings generated by the helm. This number is reported by the helm and *not* simply the number of warnings observed by the scope. This number coincides with the number of times the helm writes a new message to the variable `BHV_WARNING`.

2.2 The Helm Decision (Lines 9-11)

The helm decision space, i.e., IvP domain, is displayed on line 9. Each decision variable is given by its name, low value, high value, and the number of decision points. So `[speed,0,4,21]` represents values $\{0, 0.25, 0.5, \dots, 3.75, 4.0\}$. The following lines used to display the actual helm decision. Occasionally the helm may be configured with one of its decision variables configured to be *optional*. The helm may not produce a decision on that variable on some iteration if no behaviors are reasoning about that variable. In this case the label "varbalk" may be shown next to the decision variable to indicate no decision.

2.3 The Helm Behavior Summary (Lines 12-17)

Following this is a list of all the active, running, idle and completed behaviors. At any point in time, each instantiated IvP behavior is in one of these four states and each behavior specified in the behavior file should appear in one of these groups. Technically all *active* behaviors are also *running* behaviors but not vice versa. So only the running behaviors that are not active (i.e., the behaviors that could have, but chose not to produce an objective function), are listed in the "Behaviors Running:" group. Immediately following each behavior the time, in seconds, that the behavior has been in the current state is shown in parentheses. For the active behaviors (see line 13) this information is followed by the priority weight of the behavior, the number of pieces in the produced IvP function, and the amount of CPU time required to build the function. If the behavior also is accepting dynamic parameter updates the last piece of information on line 13 shows how many successful updates where made against how many attempts. A failed update attempt also generates a helm warning, counted on line 8. The idle and completed behaviors are listed by default one per line. This can be changed to list them on one long line by hitting the 'b' key interactively.

3 The MOOSDB-Scope Section of the uHelmScope Output

A built-in generic scope function is built into `uHelmScope`, not different in style from `uXMS`. The scope ability in `uHelmScope` provides two advantages: first, it is simply a convenience for the user to monitor a few key variables in the same screen space. Second, `uHelmScope` automatically registers for the variables that the helm reasons over to determine the behavior activity states. It will register for all variables appearing in behavior conditions, runflags, activeflags, inactiveflags, endflags and idleflags. It will also register for variables involved in the helm hierarchical mode definitions. The list of these variables is provided by the helm itself when it publishes `IVPHELM.STATEVARS`.

For example, the output in Listing 1 was derived from scoping on the alpha mission, and launching from the terminal with:

```
$ uHelmScope alpha.moos IVPHELM_STATEVARS
```

In this case the variable `IVPHELM_STATEVARS` itself is added to the scope list, and the *value* of this variable contains the three other variables on the scope list, reported by the helm to be involved in conditions or flags. The `MISSION` variable has not been written to because `MISSION="complete"` is the endflag of the return behavior in the alpha mission. At the point where this snapshot was taken, this behavior had not completed.

The lines comprising the MOOSDB-Scope section of the `uHelmScope` output are all preceded by the `#` character. This is to help discern this block from the others, and as a reminder that the whole block can be toggled off and on by typing the `#` character. The columns in Listing 1 are truncated to a set maximum width for readability. The default is to have truncation turned on. The mode can be toggled by the console user with the `'t'` character, or set in the MOOS configuration block or with a command line switch. A truncated entry in the `VarValue` column has a `'+'` at the end of the line. Truncated entries in other columns will have `".."` embedded in the entry. Line 24 shows an example of both kinds of truncation.

The variables included in the scope list can be specified in the `uHelmScope` configuration block of a MOOS file. In the MOOS file, the lines have the form:

```
var = <MOOSVar>, <MOOSVar>, ...
```

An example configuration is given in Listing 5. Variables can also be given on the command line. Duplicates requests, should they occur, are simply ignored. Occasionally a console user may want to suppress the scoping of variables listed in the MOOS file and instead only scope on a couple variables given on the command line. The command line switch `-c` will suppress the variables listed in the MOOS file - unless a variable is also given on the command line. In line 26 of Listing 1, the variable `MISSION` is a *virgin* variable, i.e., it has yet to be written to by any MOOS process and shows `n/a` in the five output columns. By default, virgin variables are displayed, but their display can be toggled by the console user by typing `'v'`.

4 The Behavior-Posts Section of the uHelmScope Output

The Behavior-Posts section is the third group of output in `uHelmScope`. It lists MOOS variables and values posted by the helm. Each variable was posted by a particular helm behavior and the grouping in the output is accordingly by behavior. Unlike the variables in the MOOSDB-Scope section, entries in this section only appear if they were written by the helm. The lines comprising the Behavior-Posts section of the `uHelmScope` output are all preceded by the `'@'` character. This is to help discern this block from the others, and as a reminder that the whole block can be toggled off and on by typing the `'@'` character. As with the output in the MOOSDB-Scope output section, the output may be truncated. A value that has been truncated will contain the `".."` characters around the middle of the string as in lines 34, 37-38, and 40.

5 Console Key Mapping and Command Line Usage

User input is accepted at the console during a **uHelmScope** session, to adjust either the content or format of the reports. It operates in a couple different *refresh* modes. In the *paused* refresh mode, after a report is posted to the console no further output is generated until the user requests it. In the *streaming* refresh mode, new helm summaries are displayed as soon as they are received. The refresh mode is displayed in the report on the very first line as in Listing 1.

The key mappings can be summarized in the console output by typing the 'h' key, which also sets the refresh mode to *paused*. The key mappings shown to the user are shown in Listing 2.

Listing 5.2: Key mapping summary shown after hitting 'h' in a console.

```
1  KeyStroke  Function
2  -----
3  Getting Help:
4    h        Show this Help msg - 'r' to resume
5
6  Modifying the Refresh Mode:
7    Spc      Refresh Mode: Pause (after updating once)
8    r        Refresh Mode: Streaming (throttled)
9    R        Refresh Mode: Streaming (unthrottled)
10
11 Modifying the Content Mode:
12   d        Content Mode: Show normal reporting (default)
13   w        Content Mode: Show behavior warnings
14   l        Content Mode: Show life events
15   m        Content Mode: Show hierarchical mode structure
16
17 Modifying the Content Format or Filtering:
18   b        Toggle Show Idle/Completed Behavior Details
19   '        Toggle truncation of column output
20   v        Toggle display of virgins in MOOSDB-Scope output
21   #        Toggle Show the MOOSDB-Scope Report
22   @        Toggle Show the Behavior-Posts Report
23
24 Hit 'r' to resume outputs, or SPACEBAR for a single update
```

Several of the same preferences for adjusting the content and format of the **uHelmScope** output can be expressed on the command line, with a command line switch. Command line usage is shown in Listing 3, and may be obtained from the command line by invoking:

```
$ uHelmScope --help or -h
```

Listing 5.3: Command line usage of uHelmScope.

```
1  =====
2  Usage: uHelmScope file.moos [OPTIONS] [MOOS Variables]
3  =====
4
5  Options:
6    --alias=<ProcessName>
7        Launch uHelmScope with the given process name rather
8        than uHelmScope.
```

```

9    --clean, -c
10       MOOS variables specified in given .moos file are excluded
11       from the MOOSDB-Scope output block.
12    --example, -e
13       Display example MOOS configuration block.
14    --help, -h
15       Display this help message.
16    --interface, -i
17       Display MOOS publications and subscriptions.
18    --noscope, -x
19       Suppress MOOSDB-Scope output block.
20    --noposts, -p
21       Suppress Behavior-Posts output block.
22    --novirgins, -g
23       Suppress virgin variables in MOOSDB-Scope output block.
24    --streaming, -s
25       Streaming (unpaused) output enabled.
26    --trunc, -t
27       Column truncation of scope output is enabled.
28    --version, -v
29       Display the release version of uHelmScope.
30
31 MOOS Variables
32    MOOS_VAR1 MOOSVAR_2, ..., MOOSVAR_N
33
34 Further Notes:
35    (1) The order of command line arguments is irrelevant.
36    (2) Any MOOS variable used in a behavior run condition or used
37        in hierarchical mode declarations will be automatically
38        subscribed for in the MOOSDB scope.

```

The command line invocation also accepts any number of MOOS variables to be included in the MOOSDB-Scope portion of the **uHelmScope** output. Any argument on the command line that does not end in `.moos`, and is not one of the switches listed above, is interpreted to be a requested MOOS variable for inclusion in the scope list. Thus the order of the switches and MOOS variables do not matter. These variables are added to the list of variables that may have been specified in the **uHelmScope** configuration block of the MOOS file. Scoping on *only* the variables given on the command line can be accomplished using the `-c` switch. To support the simultaneous running of more than one **uHelmScope** connected to the same **MOOSDB**, **uHelmScope** generates a random number N between 0 and 10,000 and registers with the **MOOSDB** as **uHelmScope.N**.

6 Helm-Produced Variables Used by uHelmScope

There are six variables published by the helm to which **uHelmScope** subscribes. These provide critical information for generating **uHelmScope** reports.

The first two variables, **IVPHELM.STATE** and **IVPHELM.SUMMARY** are published on each iteration of the helm. The former is published regardless of the helm state. This variable serves as the helm heartbeat. The latter is only published when the helm is in the **DRIVE** state. The below examples give a feel for the content:

```

IVPHELM_STATE      = "DRIVE"
IVPHELM_SUMMARY    = "iter=66,ofnum=1,warnings=0,utc_time=1209755370.74,solve_time=0.00,
                      create_time=0.02,loop_time=0.02,var=speed:3.0,var=course:108.0,
                      halted=false,running_bhvs=none,
                      active_bhvs=waypt_survey$6.8$100.00$1236$0.01$0/0,
                      modes=MODE@ACTIVE:SURVEYING,idle_bhvs=waypt_return$55.3$n/a,
                      completed_bhvs=none"

```

The `IVPHELM_SUMMARY` variable contains all the dynamic information included in the general helm overview (top) section of the `uHelmScope` output. It is a comma-separated list of `var=val` pairs. The helm publishes this in a journal style, omitting certain content if they are unchanged between iterations. When `uHelmScope` launches, it publishes to the variable `IVPHELM_REJOURNAL` which the helm interprets as a request to send a full-content message on the next iteration, before resuming journaling.

The `IVPHELM_LIFE_EVENT` is posted only when a behavior is spawned or dies. For missions without dynamic behavior spawning, this variable will only be posted upon startup for each initial static behavior. Note that the helm only publishes life events as they occur, so if the helm scope is launched after the helm, earlier events may not be reflected in the life event report. The below example gives a feel for the content of this variable:

```

IVHELM_LIFE_EVENT = "time=814.09, iter=3217, bname=bng-line-bng-line132--104,
                    btype=BHV_BearingLine, event=spawn,
                    seed=name=bng-line132--104#bearing_point=132,-104"

```

The `IVPHELM_DOMAIN`, `IVPHELM_STATEVARS`, and `IVPHELM_MODESET` variables are typically only produced once, upon startup.

```

IVPHELM_DOMAIN      = "speed,0,4,21:course,0,359,360"
IVPHELM_STATEVARS   = "RETURN,DEPLOY"
IVPHELM_MODESET     = "---,ACTIVE#---,INACTIVE#ACTIVE,SURVEYING#ACTIVE,RETURNING"

```

The `IVP_DOMAIN` variable also contributes to this section of output by providing the IvP domain used by the helm. The `IVPHELM_STATEVARS` variable affects the MOOSDB-Scope section of the `uHelmScope` output by identifying which MOOS variables are used by behaviors in conditions, runflags, endflags and idleflags.

7 Configuration Parameters for uHelmScope

Configuration for `uHelmScope` amounts to specifying a set of parameters affecting the terminal output format. An example configuration is shown in Listing 5, with all values set to the defaults. Launching `uHelmScope` with a MOOS file that does not contain a `uHelmScope` configuration block is perfectly reasonable.

Listing 7.4: Configuration Parameters for uHelmScope.

<code>behaviors_concise:</code>	If true, the idle and completed behaviors are reported all on one line rather than separate lines. Legal values: true, false. The default is true.
<code>display_bhv_posts:</code>	If true, the behavior-posts section of the report is shown. This can also be toggled at run time with the '@' key. Legal values: true, false. The default is true. Section 4.
<code>display_moos_scope:</code>	If true, the MOOS variable scope section of the report is shown. This can also be toggled at run time with the '#' key. Legal values: true, false. The default is true. Section 3.
<code>display_virgins:</code>	If true, variables in the MOOS scope section of the report is shown will be shown even if they have never been written to. This can also be toggled at run time with the 'g' key. Legal values: true, false. The default is true. Section 3.
<code>paused:</code>	If true, <code>uHelmScope</code> launches in the paused mode. Legal values: true, false. Default value is true.
<code>truncated_output:</code>	If true, output in the MOOS-scope or behavior-scope section of the report is truncated. This can also be toggled at run time with the '' key. Legal values: true, false. The default is false.
<code>var:</code>	A comma-separated list of variables to scope on in the MOOS-Scope block. Multiple lines may be provided. Section 3.

An example configuration file may be obtained from the command line with:

```
$ uHelmScope --example or -e
```

This will show the output shown in Listing 5 below.

Listing 7.5: Example configuration of the uHelmScope application.

```

1  =====
2  uHelmScope Example MOOS Configuration
3  =====
4
5  ProcessConfig = uHelmScope
6  {
7      AppTick    = 1      // MOOSApp default is 4
8      CommsTick  = 1      // MOOSApp default is 4
9
10     paused     = false   // default
11
12     display_moos_scope = true    // default
13     display_bhv_posts  = true    // default
14     display_virgins    = true    // default
15     truncated_output   = false   // default
16     behaviors_concise  = true    // default
17
18     var   = NAV_X, NAV_Y, NAV_SPEED, NAV_DEPTH // MOOS vars are
19     var   = DESIRED_HEADING, DESIRED_SPEED     // case sensitive
20 }
```


Each of the parameters can also be set on the command line, or interactively at the console, with one of the switches or keyboard mappings listed in Section 5. A parameter setting in the MOOS configuration block will take precedence over a command line switch.

8 Publications and Subscriptions for uHelmScope

The interface for **uHelmScope**, in terms of publications and subscriptions, is described below. This same information may also be obtained from the terminal with:

```
$ uHelmScope --interface or -i
```

8.1 Variables Published by uHelmScope

- **APPCAST**: Contains an appcast report identical to the terminal output. Appcasts are posted only after an appcast request is received from an appcast viewing utility.
- **IVPHELM_REJOURNAL**: A request to the helm to rejournal its summary output. Section 6.

8.2 Variables Subscribed for by uHelmScope

- **APPCAST_REQ**: A request to generate and post a new appcast report, with reporting criteria, and expiration.
- **<USER-DEFINED>**: Variables identified for scoping by the user in the **uHelmScope** will be subscribed for. See Section 3.
- **<HELM-DEFINED>**: As described in Section 3, the variables scoped by **uHelmScope** include any variables involved in the preconditions, runflags, idleflags, activeflags, inactiveflags, and endflags for any of the behaviors involved in the current helm configuration.
- **IVPHELM_LIFE_EVENT**: A description of a helm life event, the birth or death of a behavior and the manner in which it was spawned. See Section 6.
- **IVPHELM_SUMMARY**: A comprehensive summary of the helm status including behavior status summaries and most recent helm decision. See Section 6.
- **IVPHELM_STATEVARS**: A helm-produced list of MOOS variables involved in the logic of determining behavior activation. Any variable involved in mode conditions or behavior conditions. See Section 6.
- **IVPHELM_DOMAIN**: The specification of the IvP Domain presently used by the helm. See Section 6.
- **IVPHELM_MODESET**: A description of the helm's hierarchical mode specification. See Section 6.
- **IVPHELM_STATE**: A short description of the helm state: either **STANDBY**, **PARK**, **DRIVE**, or **DISABLED**. See Section 6.