

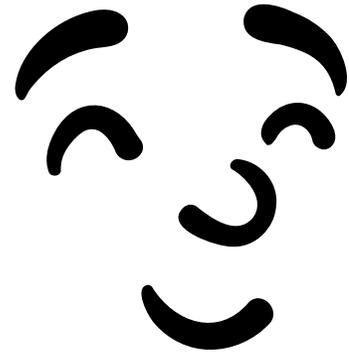
Customization

**Submit Requirements, Transforms,
Hooks, Plugins**

**European HTCondor Workshop 2025
Prague**

**Todd Tannenbaum
Center for High Throughput Computing
University of Wisconsin-Madison**

Legend

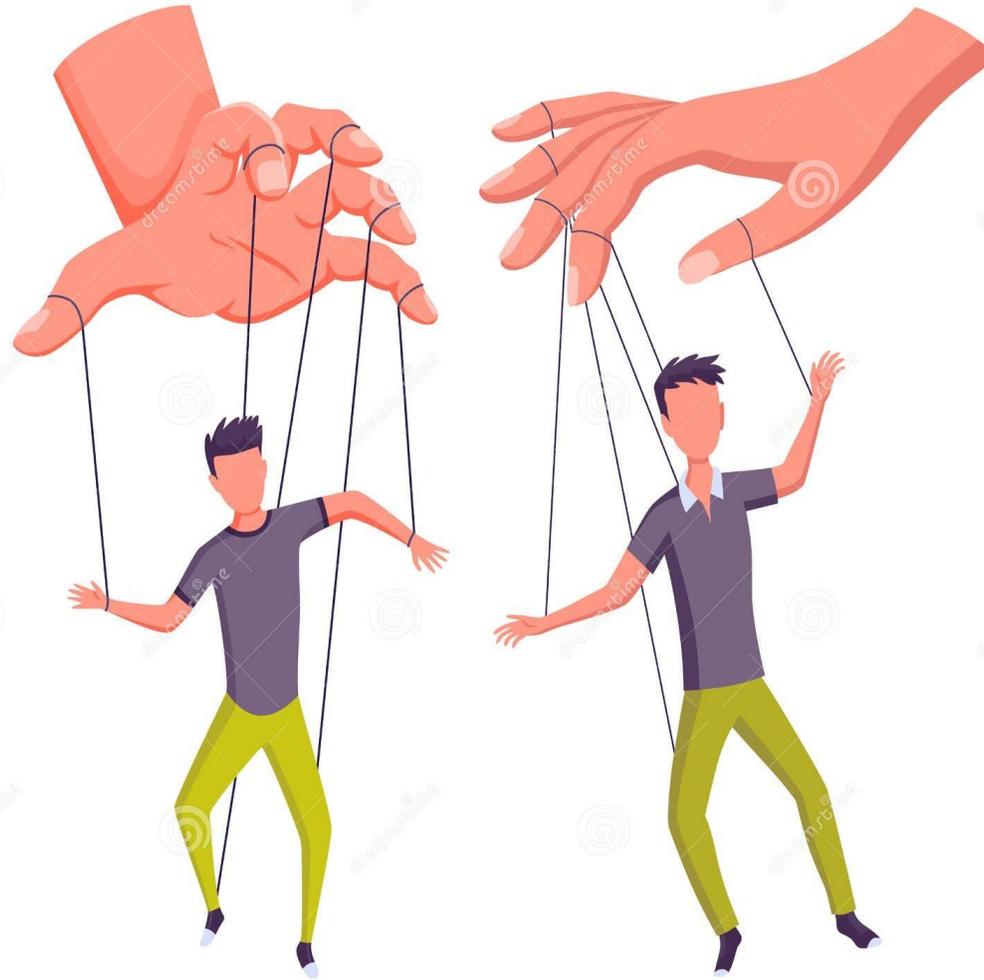


= “Voila”



= “What’s the
Bad News?”

What if you could control your users?



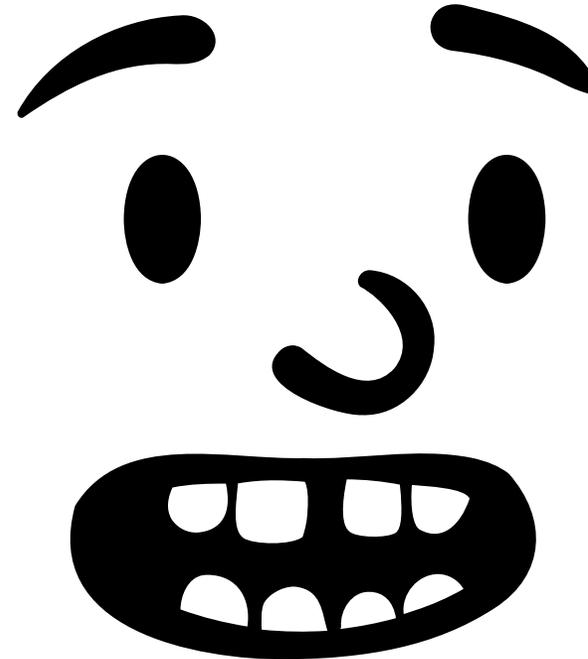
Jobs must request more than 1GB

Jobs must request more than 1GB

```
#!/bin/bash
FILE="$2"
# Extract the value of RequestMemory from the file
MEMORY=$(grep -E '^RequestMemory[[:space:]]*=[[:space:]]*[0-9]+' "$FILE" | awk -F=
'{print $2}' | tr -d '[:space:]')
# Check if MEMORY is a number and less than 1000
if [[ -n "$MEMORY" && "$MEMORY" =~ ^[0-9]+$ ]]; then
    if [ "$MEMORY" -lt 1000 ]; then
        echo "Error: RequestMemory must be at least 1000. Found: $MEMORY"
        exit 2
    fi
else
    echo "Error: RequestMemory line not found or invalid format."
    exit 3
fi
# If all checks pass, submit the file
exec condor_submit "$FILE"
```

Jobs must request more than 1GB

```
#!/bin/bash
FILE="$2"
# Extract the value of RequestMemory from the file
MEMORY=$(grep -E '^RequestMemory[[:space:]]*=[[:space:]]*[0-9]+'
"$FILE" | awk -F= '{print $2}' | tr -d '[:space:]')
# Check if MEMORY is a number and less than 1000
if [[ -n "$MEMORY" && "$MEMORY" =~ ^[0-9]+$ ]]; then
  if [ "$MEMORY" -lt 1000 ]; then
    echo "Error: RequestMemory must be at least 1000. Found:
$MEMORY"
    exit 2
  fi
else
  echo "Error: RequestMemory line not found or invalid format."
  exit 3
fi
# If all checks pass, submit the file
exec condor_submit "$FILE"
```



Submit Requirements

› SUBMIT_REQUIREMENT_NAMES

- A list of unique names, where each is used in the formation of a configuration variable name that will represent an expression evaluated to decide whether or not to reject a job submission. Each name in the list will be used in the name of configuration variable SUBMIT_REQUIREMENT_<Name>.

› SUBMIT_REQUIREMENT_<Name>

- A boolean ClassAd expression evaluated in the context of a in the process of being placed (submitted). When False, it causes the *condor_schedd* to reject the submission.

Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =  
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory =  
RequestMemory > 1000
```

Jobs must request more than 1GB

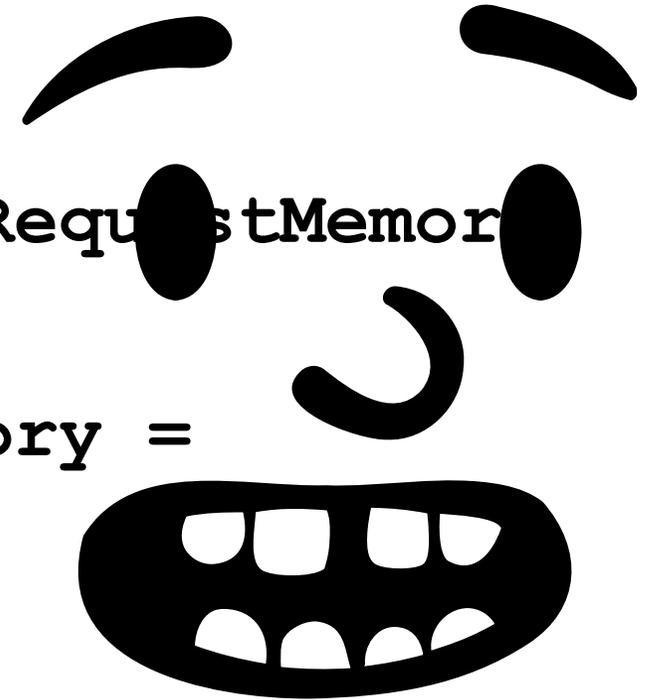
```
SUBMIT_REQUIREMENT_NAMES =  
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory =  
RequestMemory > 1000
```

Why bother with
SUBMIT_REQUIREMENT_NAMES?

Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =  
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory  
  
SUBMIT_REQUIREMENT_MinimalRequestMemory =  
RequestMemory > 1000
```



Why bother with SUBMIT_REQUIREMENT_NAMES?

Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =
```

```
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory =
```

```
RequestMemory > 1000
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory_REASON =
```

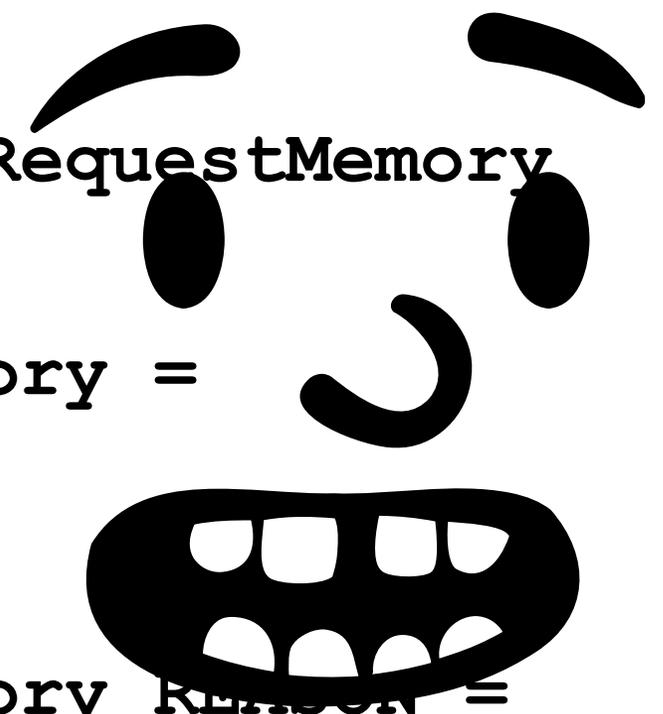
```
"RequestMemory must be more than 1GB"
```

Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =  
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory  
  
SUBMIT_REQUIREMENT_MinimalRequestMemory =  
RequestMemory > 1000  
  
SUBMIT_REQUIREMENT_MinimalRequestMemory_REASON =  
strcat( "The job only requested ", RequestMemory,  
" Megabytes. RequestMemory must be more than 1GB" )
```

Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =  
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory  
  
SUBMIT_REQUIREMENT_MinimalRequestMemory =  
RequestMemory > 1000  
  
SUBMIT_REQUIREMENT_MinimalRequestMemory_Reason =  
strcat( "The job only requested ", RequestMemory,  
" Megabytes. RequestMemory must be more than 1GB" )
```



Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =
```

```
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory =
```

```
RequestMemory > 1000
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory_REASON =
```

```
strcat( "The job only requested ", RequestMemory,
```

```
" Megabytes. RequestMemory must be more than 1GB" )
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory_IS_WARNING =
```

```
TRUE
```

Jobs must request more than 1GB

```
SUBMIT_REQUIREMENT_NAMES =
```

```
$(SUBMIT_REQUIREMENT_NAMES) MinimalRequestMemory
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory =
```

```
RequestMemory > 1000
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory
```

```
strcat( "The job only requested ", RequestMemory,
```

```
" Megabytes. RequestMemory must be more than 1GB" )
```

```
SUBMIT_REQUIREMENT_MinimalRequestMemory_IS_WARNING =
```

```
TRUE
```

Real World Example

We had a problem with users either filling up /tmp with their job logs or writing them to the shared filesystem, causing slowdowns there, so we decided to force them to put the logs in their home directories (which have quotas).

Instead of breaking their submissions right away, we started out with a warning, gave them a few weeks, then turned the warning into an error.

Real World Example

```
#Only warn users violating this requirement until February 7th, 2023
```

```
WARN_ULOGS_IN_HOME_REQ = ifThenElse(time() > int(absTime("2023-02-07T00:00:01-06:00")), FALSE, TRUE)
```

```
SUBMIT_REQUIREMENT_ULogsInHomeDir_IS_WARNING = $(WARN_ULOGS_IN_HOME_REQ)
```

```
#User can submit jobs if logs are undefined or the logs are being written somewhere under home
```

```
SUBMIT_REQUIREMENT_ULogsInHomeDir = ifThenElse(isUndefined(UserLog), TRUE, (substr(UserLog, 0, 6) == "/home/")) && ifThenElse(isUndefined(DAGManNodesLog), TRUE, (substr(DAGManNodesLog, 0, 6) == "/home/"))
```

```
SUBMIT_REQUIREMENT_ULogsInHomeDir_REASON = join(" ", "Job event logs produce by 'log =' in the submit file and the DAGMan nodes.log must be written inside your home directory", ifThenElse($(WARN_ULOGS_IN_HOME_REQ), "beginning", "as of"), "February 7th, 2023.")
```

**Want to change attributes about the jobs
your users submit?**

Use Job Transforms!

<https://htcondor.readthedocs.io/en/latest/admin-manual/ap-policy-configuration.html#schedd-job-transforms>

Quantize RequestMemory

```
JOB_TRANSFORM_NAMES = QuantizeMemory
```

```
JOB_TRANSFORM_QuantizeMemory @=end
```

```
# Quantize RequestMemory into 4GB buckets
```

```
SET RequestMemory = quantize(RequestMemory, {4096})
```

```
@end
```

Route/Transform commands

- Commands start with a keyword to modify the job
SET, DEFAULT, EVALSET
RENAME, COPY, DELETE
- Can have **REQUIREMENTS**
- Like syntax for submit/config files, transforms can macros and conditionals **if / else / endif**

Real World Example

Users would submit jobs with PeriodicRelease expression that did not have any upper bound.

Here's a short job transform to prevent people from PeriodicReleasing their jobs endlessly. It saves the user's original PeriodicRelease, then replaces it with one that's only true if there were fewer than 4 holds, and marks PeriodicRelease as a protected attribute so the user can't change it back. (We make an exception for jobs that went on hold due to exceeding their execution time limit (codes 46 and 47)).

Real World Example

```
PROTECTED_JOB_ATTRS = $(PROTECTED_JOB_ATTRS) PeriodicRelease

PERIODIC_RELEASE_LIMIT = 4

JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES) ReleaseLimit

# Absolute limit on the number of times that PeriodicRelease can trigger
JOB_TRANSFORM_ReleaseLimit @=jt
  if ! defined MY.ReqPeriodicRelease
    COPY PeriodicRelease ReqPeriodicRelease
    SET PeriodicRelease = (NumHolds?:0) <= $(PERIODIC_RELEASE_LIMIT) &&
      (HoldReasonCode != 46 && HoldReasonCode != 47) && ReqPeriodicRelease
  endif
@jt
```

Real World Example

```
PROTECTED_JOB_ATTRS = $(PROTECTED_JOB_ATTRS) PeriodicRelease

PERIODIC_RELEASE_LIMIT = 4

JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES) ReleaseLimit

# Absolute limit on the number of times that PeriodicRelease can trigger
JOB_TRANSFORM_ReleaseLimit @=jt
  if ! defined MY.ReqPeriodicRelease
    COPY PeriodicRelease ReqPeriodicRelease
    SET PeriodicRelease = (NumHolds?:0) <= $(PERIODIC_RELEASE_LIMIT) &&
      (HoldReasonCode != 46 && HoldReasonCode != 47) && ReqPeriodicRelease
  endif
@jt
```

Note: Or just `SYSTEM_MAX_RELEASES=4`

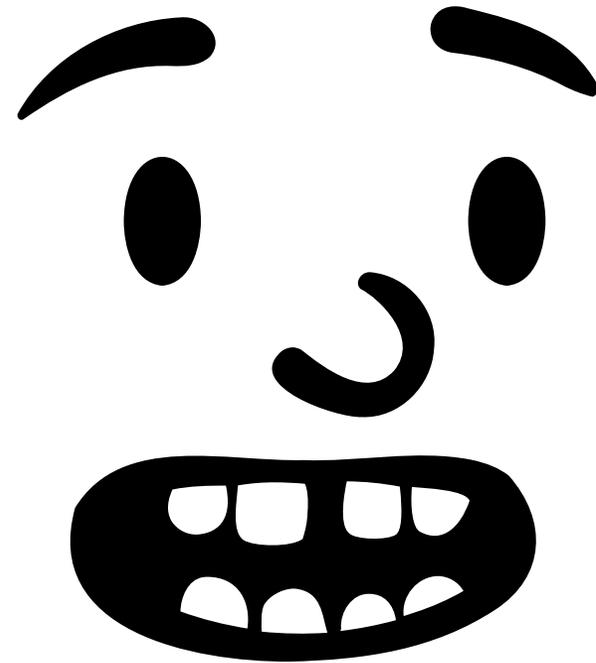
What if I want to run a script before/after a job?

What if I want to run a script before/after a job?

USER_JOB_WRAPPER = myscript.sh

What if I want to run a script before/after a job?

› **USER_JOB_WRAPPER =
myscript.sh**



Hooks

- › HTCondor can perform a call-out to a script provided by the administrator.
- › Prepare Job Hooks
 - Allow to be modified before activation execution (at the starter or shadow).
 - StdIn: the job classad
 - StdOut: any job attributes to insert or modify
 - **HOOK_PREPARE_JOB_BEFORE_TRANSFER**
 - **HOOK_PREPARE_JOB** (after file transfer)
- › Exit/Vacate Job Hooks
 - **HOOK_JOB_EXIT**
- › Monitor Hooks
- › Job Router Hooks (specific to the HTCondor CE)

Plugins

› Be careful!

Thank You!

*Please add your institution
to our world map of HTCondor Users at:
<https://htcondor.org/user-map>
and click "Add Your Institution" on upper right*

PATH PARTNERSHIP to ADVANCE
**THROUGHPUT
COMPUTING**

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATH Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.