

# Format opisu zadań



- Lista dyrektyw
  - application
  - argument
  - assistant
  - deadline
  - environment
  - error
  - executable
  - grant
  - host
  - include
  - input
  - memory
  - memory-per-slot
  - module
  - monitor
  - mount
  - name
  - native
  - nodes
  - not-after
  - not-before
  - note
  - notify
  - output
  - parameter-sweep
  - persistent
  - postprocess
  - preprocess
  - procs
  - properties
  - queue
  - reservation
  - stage-in-dir
  - stage-in-file
  - stage-out-dir
  - stage-out-file
  - use-reservation
  - use-scratch
  - walltime
  - watch-output
  - variable
- Zmienne systemowe opisu zadania
- Obsługa zdalnych plików i katalogów
  - Składnia

Zlecany plik jest plikiem tekstowym, który może zawierać dyrektywy infrastruktury QCG.

Dyrektywą jest każda linia zaczynająca się od "#QCG".

Jeżeli w pliku nie zdefiniowano dyrektywy "executable" ani "application" uruchamiany jest zlecany plik z opisem zadania.

## Lista dyrektyw

- [application](#) - aplikacja
- [argument](#) - argument aplikacji
- [assistant](#) - skrypt pomocniczy

- **deadline**- wykonaj w okresie czasu
- **environment** - zmienna środowiskowa
- **error** - standardowe wyjście błędów
- **executable** - plik wykonywalny
- **grant** - grant
- **host** - wybór klastra
- **include** - import części zadania z odrębnego pliku
- **input** - standardowe wejście
- **memory** - pamięć
- **memory-per-slot** - pamięć per rdzeń,
- **module**- wymagany moduł
- **monitor** - monitorowanie zadania (patrz. [Monitoring zadań](#))
- **mount** - zamontowanie systemu składowania danych
- **name** - nazwa zadania
- **native**- specyficzne parametry/wymagania zasobowe przekazywane bezpośrednio do systemu kolejkowego
- **nodes** - topologia aplikacji
- **not-after**- nie później niż
- **not-before**- nie wcześniej niż
- **note** - opis zadania
- **notify** - powiadamianie o stanie zadania
- **output** - standardowe wyjście
- **parameter-sweep** - definicja zmienności parametrów
- **persistent** - nieusuwanie katalogu roboczego zadania po jego zakończeniu
- **postprocess** - epilog zadania
- **preprocess** - prolog zadania
- **procs** - liczba procesów
- **properties** - parametry węzłów
- **queue** - wybór kolejki
- **reservation**- użycie rezerwacji z wyprzedzeniem
- **stage-in-dir** - katalog wejściowy
- **stage-in-file** - plik wejściowy
- **stage-out-dir** - katalog wynikowy
- **stage-out-file** - plik wynikowy
- **use-reservation**- użycie rezerwacji z wyprzedzeniem
- **use-scratch** - użycie przestrzeni "scratch" w trakcie obliczeń
- **walltime** - czas wykonania
- **watch-output** - monitorowanie pliku wyjściowego i przesyłanie powiadomień
- **variable** - zmienna w opisie zadania (dotyczy tylko dyrektyw)

## application

- **application** - nazwa aplikacji do uruchomienia. Opcjonalnie możliwe jest podanie wersji aplikacji

```
#QCG application=namd
```

```
#QCG application=gromacs/4.6.3
```

## argument

- **argument** - argument aplikacji w przypadku użycia dyrektywy „executable” lub „application”. Argument może wystąpić wielokrotnie. Każdy argument powinien być przekazany w osobnej dyrektywie. Argumenty do aplikacji przekazywane są w kolejności ich wystąpienia w pliku opisu

```
#QCG argument=arg1
#QCG argument=arg2
```

## assistant

- **assistant** - dyrektywa umożliwiająca uruchomienie równolegle z podstawowym programem skryptu pomocniczego. Skrypt ten zakończony zostanie automatycznie z końcem zadania.



Dyrektywa wprowadzona została w wersji 3.4 i zastąpiła dyrektywę *#QCG monitor*, która zmieniła znaczenie.

Składnia:

```
#QCG assistant=skrypt
#QCG assistant=script->skrypt
```

Parametry:

- script - obligatoryjny, domyślny parametr definiujący skrypt, który ma być uruchomiony. W przypadku korzystania ze skryptu musi być on przegrany dyrektywą #QCG stage-in-file.

```
#QCG stage-in-file=script.sh
```



**WAŻNE:** Skrypt uruchamiany jest jednokrotnie na początku zadania. Jeżeli akcja zdefiniowana wewnątrz skryptu ma być wywoływana cyklicznie użytkownik jest odpowiedzialny za implementację takiej funkcjonalności.

## deadline

- deadline - dyrektywa umożliwiająca zdefiniowanie, że zadanie lub rezerwacja ma być wykonane „w czasie nie dłuższym niż”. Dyrektywa wymusza użycie mechanizmu rezerwacji zasobów z wyprzedzeniem. Nie może być użyta równocześnie z dyrektywą „reservation”. Argumentem dyrektywy jest przedział czasu w formacie *PnYnMnDnHnMnS* (ISO 8601) gdzie:
  - P - obowiązkowy znak rozpoczynający definicję okresu,
  - nY - liczba lat,
  - nM - liczba miesięcy,
  - nD - liczba dni,
  - T - separator czasu (musi być obecny, jeśli zdefiniowane są poniższe wartości)
  - nH - liczba godzin,
  - nM - liczba minut,
  - nS - liczba sekund.

W przypadku nie podania dyrektywy „not\_before” system próbuje dokonać rezerwacji zasobów od aktualnej chwili. Dyrektywa wyklucza równoczesne użycie „not\_after”.

```
#QCG deadline=P3DT12H
```

```
#QCG not-before=2012.07.25
#QCG deadline=P3DT12H
```

## environment

- environment - ustawianie zmiennych środowiskowych. Składnia „nazwa → wartość”. Każda zmienna musi być ustawiana w osobnej linii.

```
#QCG environment=name -> piontek
#QCG environment=location -> poznan
```

## error

- error - lokalizacja gdzie ma być przegrany standardowe wyjście błędów zadania (stderr). Jeśli nie jest to [zdalna lokalizacja](#) zakłada się, że jest ustalana względem katalogu, z którego zlecono zadanie

```
#QCG error=gsiftp://qcg.man.poznan.pl//home/plgrid/plgpiontek/reef/errors/${JOB_ID}.error
#QCG error=error.txt
```

## executable

- executable - lokalizacja pliku do uruchomienia. Lokalizacja może być [lokalizacją zdalną](#) lub ścieżką do pliku. W tym drugim przypadku przyjmuje się, że jest ustalana względem katalogu, z którego zlecone zostało zadanie. Opcjonalnie możliwe jest podanie nazwy pod jaką ma być zapisany plik wykonywalny. W przypadku braku nazwy zapisywany jest pod oryginalną nazwą.

```
#QCG executable=gsiftp://qcg.man.poznan.pl//home/plgrid/plgpiontek/reef/executables/exec1
#QCG executable=executables/exec1
#QCG executable=executables/exec1 -> exec-file
```

## grant

- grant - nazwa grantu, w ramach którego ma być wykonane zadanie.

```
#QCG grant=plgpiontek_grant
```

## host

- host - nazwa maszyny na której może być uruchomione zadanie, lub dokonana rezerwacja zasobów. Może być wiele takich dyrektyw dla alternatywnych maszyn.



[Lista zasobów infrastruktury PL-Grid dostępnych z usług QCG.](#)

```
#QCG host=reef.man.poznan.pl
#QCG host=zeus.cyfronet.pl
```

## include

- include - dyrektywa umożliwia włączenie do opisu zadania zawartości innego pliku. Linia z dyrektywą w finalnym opisie zastępowana jest zawartością pliku na który wskazuje. Dyrektywa umożliwia podawanie ścieżki względnej i bezwzględnej a także rekurencyjne osadzanie kolejnych dyrektyw *include* w plikach.

```
#QCG include=includes/application.qcg
#QCG include=/home/piontek/includes/notifications.qcg
```

## input

- input - lokalizacja skąd ma być wzięte standardowe wejście dla aplikacji (stdin). Jeśli nie jest to [zdalna lokalizacja](#) zakłada się, że jest ustalana względem katalogu, z którego zlecono zadanie.

```
#QCG input=gsiftp://qcg.man.poznan.pl//home/plgrid/plgpiontek/reef/inputs/experiment.input
#QCG input=input.txt
```

## memory

- memory - dyrektywa definiująca deklarowane maksymalne zapotrzebowanie aplikacji na pamięć operacyjną. Podana wartość jest w MB.

```
#QCG memory=1024
```

## memory-per-slot

- memory-per-slot - ilość pamięci per rdzeń (slot) wyrażona w MB. Całkowite maksymalne zapotrzebowanie na pamięć liczone jest jako iloczyn liczby żądanych rdzeni (slotów) i wartości dyrektywy "memory-per-slot"

```
#QCG memory-per-slot=512
```

## module

- module - dyrektywa umożliwiająca zdefiniowanie wymaganego przez zadanie modułu (nazwy programu/biblioteki oraz wersji). Możliwe jest wielokrotne podanie dyrektywy. Przy wyborze klastra do uruchomienia zadania system weźmie pod uwagę tylko te klastry na których dostępne są wszystkie wymagane moduły.

```
#QCG module=nwchem/6.0
#QCG module=namd
```

## monitor

- monitor - dyrektywa umożliwiająca monitorowanie stanu i postępu wykonania zadania w infrastrukturze QCG z wykorzystaniem usługi [QCG-monitoring](#).



Dyrektywa #QCG monitor od wersji QCG 3.4 zmieniła znaczenie. Poprzednia funkcjonalność dostępna jest poprzez dyrektywę [#QCG assistant](#).

Składnia:

```
#QCG monitor[=skrypt]
#QCG monitor=script->skrypt[,delay->N]
```

Dla aplikacji posiadających jeden predefiniowany schemat monitorowania dyrektywa może być bezparametrowa:

```
#QCG monitor
```

W przypadku, gdy chcemy zdefiniować konkretny schemat, lub użyć własnego skryptu monitorującego konieczne jest przekazanie parametru. Domyślnym parametrem jest parametr "*script*".



W przypadku korzystania z własnego skryptu monitorującego należy go zdefiniować jako plik wejściowy aplikacji dyrektywą [#QCG stage-in-file](#).

```
#QCG monitor=gaussian
#QCG monitor=skrypt_monitorujcy.sh
#QCG stage-in-file=skrypt_monitorujcy.sh
```

Częstotliwość uruchamiania skryptu monitorującego definiuje parametr *delay*, którego wartości jest okres czasu w sekundach co jaki będzie uruchamiany skrypt. Powiadomienie wysyłane jest tylko w przypadku wykrycia zmiany od poprzedniego wysłania.

```
#QCG monitor=gaussian,delay->60
#QCG monitor=script->gaussian,delay->60
```

## mount

- mount - dyrektywa umożliwiająca automatyczne zamontowanie dla zadania określonej usługi składowania danych.

Aktualnie jedynym wspieranym systemem jest *veilFS*.

```
#QCG mount=veilfs[->katalog]
```

Opcjonalny parametr *katalog* pozwala zdefiniować nazwę katalogu w którym ma być zamontowany system. Domyślną wartością jest "mount".

```
#QCG mount=veilfs->veilFS_dir
```

## name

- name - dyrektywa określająca nazwę zadania. Nazwa zadania pojawi się jako końcówka identyfikatora zadania.

```
#QCG name=nobel-experiment
```

## native

- native - dyrektywa umożliwiająca przekazanie do systemu kolejkowego specyficznych dla niego wymagań zasobowych. Wymagania takie są specyficzne do systemu kolejkowego i powinny być używane z dyrektywami „host” i w przemyślany przez użytkownika sposób.

Przykład dla PBSa oznaczający dane zadanie jako wznowialne.

*(-r y/n Declares whether the job is rerunnable.)*

```
#QCG native=-r n
```

## nodes

- nodes - dyrektywa pozwalająca zdefiniować na ilu węzłach i rdzeniach w ramach węzła ma być uruchomione zadanie. Opcjonalnie można podać ile procesów ma być uruchomione na węzłach. Domyślnie, jeżeli nie zostanie podane inaczej, liczna procesów równa jest liczbie rdzeni przydzielonych w ramach węzła. Składnia liczna\_węzłów:liczba\_rdzeni\_na\_węźle:liczba\_procesów

```
#QCG nodes=10:5:1  
#QCG nodes=12:12
```

## not-after

- not-after - dyrektywa umożliwiająca zdefiniowanie, że zadanie lub rezerwacja ma być wykonane „nie później niż”. Argumentem dyrektywy jest data z opcjonalnym czasem. Dyrektywa wymusza użycie mechanizmu rezerwacji zasobów z wyprzedzeniem. Nie może być użyta równocześnie z dyrektywą „reservation”, wymaga podania dyrektywy „walltime”.

```
#QCG not-after=2012.08.25  
#QCG walltime=PT2H
```

```
#QCG not-after=2012.08.25 13:20  
#QCG walltime=PT2H
```

## not-before

- not-before - dyrektywa umożliwiająca zdefiniowanie, że zadanie lub rezerwacja ma być wykonane „nie wcześniej niż”. Argumentem dyrektywy jest data z opcjonalnym czasem. Dyrektywa wymusza użycie mechanizmu rezerwacji zasobów z wyprzedzeniem. Nie może być użyta równocześnie z dyrektywą „reservation”. W przypadku braku dyrektywy przyjmuje się, że zadanie może być uruchomione „od razu”. Dyrektywa wymaga dyrektywy „walltime”.

```
#QCG not-before=2012.07.25  
#QCG walltime=PT2H
```

```
#QCG not-before=2012.07.25 15:20  
#QCG walltime=PT2H
```

## note

- note - krótka informacja o zadaniu.

```
#QCG note=moje pierwsze zadanie QCG
```

## notify

- notify - dyrektywa umożliwiająca podanie protokołu i adresu docelowego dla powiadomień o stanie zadania. Możliwe jest wysyłanie powiadomień mailem i xmpp (dawniej Jabber).

```
#QCG notify=mailto:piontek@man.poznan.pl
#QCG notify=xmpp:tomasz.piontek@gmail.com
```

W przypadku chęci otrzymywania powiadomień protokołem xmpp konieczne jest dodanie do kontaktów nadawcy: [qcg-notification@plgrid.pl](mailto:qcg-notification@plgrid.pl)

## output

- output - lokalizacja gdzie ma być przegrane standardowe wyjście zadania (stdout). Jeśli nie jest to lokalizacja zdalna (gsiftp/irods) zakłada się, że jest ustalana względem katalogu, z którego zlecono zadanie.

```
#QCG output=gsiftp://qcg.man.poznan.pl//home/plgrid/plgpiontek/reef/outputs/${JOB_ID}.output
#QCG output=output.txt
```

## parameter-sweep

- parameter-sweep -dyrektywa umożliwiająca zdefiniowanie przestrzeni parametrów dla zadania. Dyrektywa może być użyta wielokrotnie definiując wielowymiarową przestrzeń parametrów. Dla każdego zestawu parametrów uruchamiane jest osobne zadanie. Dyrektywa składa się z nazwy zmiennej, która będzie przyjmować kolejne wartości oraz z definicji zmienności parametru.

```
#QCG parameter-sweep=VAR->PARAMETER_SPACE
```

- Dyrektywa dopuszcza następujące sposoby definiowania zmienności parametru:
- *list* - wartości podane są jako lista oddzielona przecinkami

```
#QCG parameter-sweep=var->list:mon,tue,wed,thu,fri,sat,sun
#QCG parameter-sweep=var->list:1,2,3,4,5,6
```

- *for* - wartość wyznaczane są od wartości początkowej (start) do końcowej (end) z opcjonalnie definiowalnym krokiem (step) (domyślnie 1). Liczba miejsc po przecinku wyznaczana jest jako największa liczba cyfr po przecinku dla parametrów *start*, *end*, *step*.

```
#QCG parameter-sweep=var->for:start..end
#QCG parameter-sweep=var->for:start..end..step

#QCG parameter-sweep=var->for:0..10
#QCG parameter-sweep=var->for:0..10..2
#QCG parameter-sweep=var->for:1.0..2.0..0.5
```

- *pattern* - kolejne wartości przyjmują nazwy plików zgodnych z podanym wzorcem wyszukiwanych względem katalogu, z którego zlecono zadanie.

```
#QCG parameter-sweep=var->pattern->*.input
```

- *file* - kolejnymi wartościami są kolejne linie pobierane z pliku

```
#QCG parameter-sweep=var->file:file_with_values.txt
```

## persistent

- persistent - dyrektywa określająca że po zakończeniu zadania system ma pozostawić katalog roboczy, w którym wykonywane było zadanie.



Dyrektywa ta powinna być używana tylko na etapie testowania. Przy normalnym trybie wykonywania aplikacji dyrektywa ta nie powinna być ustawiona, gdyż powoduje, że katalog roboczy zadania nie jest usuwany i zajmuje miejsce na zasobach.

Zadania zlecone jako "persistent" mogą być po zakończeniu "wyczyszczone" poleceniem [qcg-clean](#).

```
#QCG persistent
```

## postprocess

- `postprocess` - dyrektywa umożliwiająca wykonanie polecenia/skryptu po zakończeniu się właściwego zadania. Wartością dyrektywy może być polecenie lub ścieżka do pliku, który ma być wykonany.

```
#QCG postprocess=tar cvf wynik.tar *
```

```
#QCG postprocess=postprocess-script.sh
```

## preprocess

- `preprocess` - dyrektywa umożliwiająca wykonanie polecenia/skryptu przed uruchomieniem właściwego zadania. Wartością dyrektywy może być polecenie lub ścieżka do pliku, który ma być wykonany.

```
#QCG preprocess=mkdir outputs
```

```
#QCG preprocess=preprocess-script.sh
```

## procs

- `procs` - liczba rdzeni obliczeniowych, na których ma być wykonane zadanie, lub które mają zostać zarezerwowane. (Stosowane dla zadań MPI.)

```
#QCG procs=32
```

## properties

- `properties` - dyrektywa definiująca właściwości węzłów na których ma być uruchomione zadanie.

```
#QCG properties=mpi,ib,lustre
```

## queue

- `queue` - wybrana kolejka systemu kolejkowego.

```
#QCG queue=plgrid
```

## reservation

- `reservation` - dyrektywa umożliwiająca podanie lokalnego identyfikatora rezerwacji w ramach której ma być wykonane zadanie. Użycie tej dyrektywy wymaga jednoznacznego podania hosta na którym jest założona podana rezerwacja. Podana rezerwacja może być lokalną rezerwacją założoną bezpośrednio w systemie kolejkowym (wymaga podania typu "local"), bądź rezerwacją założoną przy użyciu polecenia `qcg-reserve` (typ `qcg`, domyślny).

```
#QCG host=reef.man.poznan.pl
#QCG reservation=R1366398248299__4039
```

```
#QCG host=reef.man.poznan.pl
#QCG reservation=piątek.0 local
```

## stage-in-dir

- `stage-in-dir` - dyrektywa kopiowania katalogu wejściowego. Funkcjonalność i składnia analogiczna jak dla dyrektywy „`stage-in-file`” tyle, że kopiowany jest katalog.

```
#QCG stage-in-dir=gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/inputs -> inputs
#QCG stage-in-dir=gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/inputs
#QCG stage-in-dir=input_dir -> inputs
#QCG stage-in-dir=input_dir
```



Skopiowanie całego katalogu w którym uruchomiony został klient odbywa się poprzez podanie "." (kropki) jako nazwy katalogu źródłowego i ewentualnie docelowego

```
#QCG stage-in-dir=. -> .
#QCG stage-in-dir=.
#QCG stage-in-dir=. -> input
```

## stage-in-file

- stage-in-file - dyrektywa kopiowania pliku wejściowego. Składnia „lokalizacja\_źródłowa → nazwa\_docelowa\_pliku”. Lokalizacja źródłowa może być [lokalizacją zdalną](#) lub ścieżką do pliku. W tym drugim przypadku zakłada się, że jest ustalana względem katalogu, z którego zlecono zadanie.

W przypadku braku drugiego parametru plik kopiowany jest pod nazwę źródłową.

```
#QCG stage-in-file=gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/inputs/input.txt -> input.txt
#QCG stage-in-file=gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/inputs/input.txt
#QCG stage-in-file=input_file.txt -> input.txt
#QCG stage-in-file=input_file.txt
```

## stage-out-dir

- stage-out-dir - dyrektywa kopiowania katalogu wyjściowego. Funkcjonalność i składnia analogiczna jak dla dyrektywy „stage-out-file” tyle, że kopiowany jest katalog.

```
#QCG stage-out-dir=results -> gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/results/${JOB_ID}
#QCG stage-out-dir=results -> gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/results/${JOB_ID}
#QCG stage-out-dir=results -> result
#QCG stage-out-dir=results
```



Skopiowanie całego katalogu w którym wykonane było zadanie odbywa się poprzez podanie "." (kropki) jako nazwy katalogu źródłowego.

```
#QCG stage-out-dir=. -> .
#QCG stage-out-dir=.
#QCG stage-out-dir=. -> output
```

## stage-out-file

- stage-out-file - dyrektywa kopiowania pliku wyjściowego. Składnia „nazwa\_pliku\_źródłowego → lokalizacja docelowa pliku”. Lokalizacja docelowa może być [lokalizacją zdalną](#) lub ścieżką do pliku. W tym drugim przypadku zakłada się, że jest ustalana względem katalogu, z którego zlecono zadanie. W przypadku braku drugiego parametru plik przegrywany jest pod nazwę źródłową.

```
#QCG stage-out-file=results.txt -> gsiftp://qcg.man.poznan.pl/home/plgrid/plgpiontek/reef/results/result.1
#QCG stage-out-file=results.txt
#QCG stage-out-file=result.txt -> ${JOB_ID}.result
#QCG stage-out-file=result.txt
```

## use-reservation

- use-reservation - bezparametrowa dyrektywa umożliwiająca zdefiniowanie, że dane zadanie ma być uruchomione z wykorzystaniem mechanizmu rezerwacji zasobów z wyprzedzeniem. Dyrektywa wymaga podania czasu wykonania zadania (dyrektywa #QCG walltime). Stosowanie tej dyrektywy pozwala „wymusić” rezerwację, gdy nie były użyte dyrektywy „not-after”, „not-before”, „deadline”. Dyrektywa wyklucza użycie równocześnie z dyrektywą „reservation”.

```
#QCG use-reservation
#QCG walltime=PT2H
```

## use-scratch

- dyrektywa wymuszająca użycie dla zadania przestrzeni dyskowej typu "scratch". W przypadku tej dyrektywy jeśli zadanie nie jest domyślnie uruchamiane na systemie "lustre" to zostanie ono automatycznie przeniesione i uruchomione w przestrzeni "scratch" o ile takowa istnieje.

```
#QCG use-scratch
```

## walltime

- walltime - dyrektywa definiująca deklarowany przez użytkownika czas wykonania zadania lub długość rezerwacji w formacie PnYnMnDnHnMnS (format ISO 8601) gdzie:
  - P - obowiązkowy znak rozpoczynający definicję okresu,
  - nY - liczba lat,
  - nM - liczba miesięcy,
  - nD - liczba dni,
  - T - separator czasu (musi być obecny, jeśli zdefiniowane są poniższe wartości)
  - nH - liczba godzin,
  - nM - liczba minut,
  - nS - liczba sekund.

```
#QCG walltime=P3DT12H
```

## watch-output

- watch-output - dyrektywa umożliwiająca monitorowanie pliku wyjściowego aplikacji i przysyłanie powiadomień z informacją.



Od wersji QCG 3.4 składnia dyrektywy #QCG watch-output uległa zmianie.

Składnia:

```
#QCG watch-output=consumer->konsument,[pattern->wyrażenie | script->skrypt ,delay->N]
#QCG watch-output=konsument
```

Dyrektywa ma następujące parametry:

- *consumer* - obligatoryjny, domyślny parametr określający odbiorcę powiadomienia. (wspieranymi protokołami są *mail* i *xmpp* - patrz dyrektywa [notify](#))
- *pattern* lub *script* - parametry definiujące albo poszukiwane w pliku wyjściowym wyrażenie regularne lub własny skrypt monitorujący (skrypt powłoki bash). W obu przypadkach wynik (output) przesyłany jest na wskazany adres. W przypadku korzystania ze skryptu monitorującego należy w zadaniu zdefiniować dyrektywę (#QCG stage-in-file) kopiującą plik.
- *delay* - okres pomiędzy kolejnymi uruchomieniami procedury monitoringu (w sekundach). Wiadomość wysyłana jest tylko w przypadku stwierdzenia zajścia zmiany od poprzedniego wysłania.

```
#QCG watch-output=mailto:piontek@man.poznan.pl,pattern->ENERGY,dalay->60
```

```
#QCG watch-output=consumer->mailto:piontek@man.poznan.pl,script->skrypt_monitorujacy.sh,dalay->60
#QCG stage-in-file=skrypt_monitorujacy.sh
```

## variable

- variable - dyrektywa pozwalająca zdefiniować zmienne w obrębie opisu zadania.



Zmienne podmienione zostaną TYLKO w obrębie dyrektyw QCG.

Proszę nie mylić dyrektywy *variable* ze zmiennymi "systemowymi" QCG podmienianymi przez system.

```
#QCG variable=PROCESSES->10
#QCG procs=${PROCESSES}
```

## Zmienne systemowe opisu zadania

W opisie zadania możliwe jest użycie następujących zmiennych "systemowych":

- **HOSTNAME** - nazwa hosta (klastra) na jakim uruchomione zostało zadanie,
- **HOME** - katalog domowy użytkownika,
- **TASK\_DIR** - link gridftp:// do katalog roboczego zadania,
- **JOB\_ID** - identyfikator eksperymentu,
- **TASK\_ID** - identyfikator zadania,
- **USER\_DN** - DN (Distinguished Name) użytkownika (identyfikator użytkownika widoczny w certyfikacie, w formacie PEM, np. /C=PL /O=GRID/O=PSNC/CN=Tomasz Piontek)
- **PROCESS\_GROUP** - identyfikator grupy procesów.
- **NOTE** - wartość dyrektywy "note".

```
#QCG output={JOB_ID}.output
#QCG stage-out-file=file.output->file.${NOTE}
```

## Obsługa zdalnych plików i katalogów

Pliki wejściowe jak i wyjściowe dla zadań zleczanych w systemie QCG mogą znajdować się na zdalnych zasobach. Broker infrastruktury QCG zadba o ich skopiowanie na odpowiedni zasób na którym będzie wykonywane nasze zadanie, a następnie skopiuje pliki wyjściowe do wskazanych lokalizacji.

Pliki wejściowe zadania definiowane są następującymi dyrektywami:

- executable,
- input,
- stage-in-dir,
- stage-in-file,

natomiast pliki wyjściowe dyrektywami:

- error,
- output,
- stage-out-dir,
- stage-in-file.

W obu przypadkach, jako lokalizację takich plików można podać zdalny zasób. Obsługiwane są następujące protokoły przesyłania danych:

- GridFTP,
- FTP,
- HTTP/HTTPS,
- iRods (obecnie autoryzacja następuje przy użyciu nazwy użytkownika/hasła, trwają prace nad autoryzacją z użyciem certyfikatu).

Tak jak w przypadku plików lokalnych, definiując zdalną lokalizację plików możemy używać [zmiennych opisu zadania](#).



W chwili obecnej obsługa protokołu **HTTP/HTTPS nie wspiera katalogów oraz plików wyjściowych**.



Przy kopiowaniu plików z użycie protokołu **HTTPS nie następuje autoryzacja** zdalnego serwera, tzn. transmisja danych następuje szyfrowanym kanałem danych, natomiast QCG-Broker nie sprawdza i nie waliduje certyfikatu jakim posługuje się zdalny serwer.

## Składnia

- gridftp

```
gsiftp://{host}[:{port}]/{path}
gridftp://{host}[:{port}]/{path}
```

gdzie:

- {host} - adres maszyny na której znajduje się serwer GridFTP,
- {port} - opcjonalny numer portu (domyślnie 2811),
- {path} - ścieżka do pliku; jeśli pierwszym znakiem ścieżki nie będzie "/" (slash), tzn. adres maszyny od ścieżki będzie oddzielał tylko jeden znak "/", wówczas ścieżka jest traktowana jako względna (względem katalogu domowego)

Przykłady

```
gsiftp://qcg.man.poznan.pl/plik.txt
gsiftp://qcg.man.poznan.pl:2811//home/plgrid/plgkopta/reef/plik.txt
```

- ftp

```
ftp://[{user}[:{pwd}]@]{host}[:{port}]/{path}
```

gdzie:

- {user} - opcjonalna nazwa użytkownika,
- {pwd} - opcjonalne hasło użytkownika,
- {host} - adres maszyny na której znajduje się serwer FTP,
- {port} - opcjonalny numer portu (domyślnie 22),
- {path} - ścieżka do pliku

Przykłady

```
ftp://ftp.kernel.org/pub/linux/kernel/v3.x/patch-3.9.gz
ftp://pkopta:haslo@serwer.ftp.man.poznan.pl:22/plik.txt
```

- http/https

```
http://{host}[:{port}]/{path}
https://{host}[:{port}]/{path}
```

gdzie:

- {host} - adres maszyny na której znajduje się serwer http/https
- {port} - opcjonalny numer portu (domyślnie 80 dla http, oraz 443 dla https),
- {path} - ścieżka do pliku

Przykłady

```
https://www.kernel.org/pub/linux/kernel/v3.x/patch-3.9.gz
```

- irods

```
irods://{user}:{pwd}@{host}[:{port}]/{zone}/{path}
```

gdzie:

- {user} - nazwa użytkownika w systemie iRods,
- {pwd} - hasło do konta w systemie iRods,
- {host} - adres maszyny na której znajduje się serwer iRods,

- {port} - opcjonalny numer portu (domyślnie 1247)
- {zone}- parametr zone w systemie iRods,
- {path} - ścieżka bezwzględna do pliku

#### Przykłady

```
irods://pkopta:haslo@elder12/testZone/home/pkopta/plik.txt
```