

Use odysseus console

As many OSGi applications, Odysseus can be controlled via console. This is especially helpful when using a server based product without a gui.

With the latest updates of Eclipse, there are some changes in the console, so some things behave a little different. The most important command is now

odyHelp

which lists all current available commands: e.g.:

```
ExecutorInfo - show internal information about the executor
add -q <query string> [S|E] -r [true|false] [-m <true>|<false>] - add query [with console-output-sink|eclipse-
outputsink] [with|without restructuring the query plan, default true] [with|without metadata set in physical
operators]

addFromFile -f <filename> [S|E] [useProp] [-r <true>|<false>] [-m <true>|<false>] - add query declared in
<filename> for current parser [with console-output-sink] [filepath automatically read from user.files] [with
restructure or not] [with|without metadata set in physical operators]
buffer <buffer placement strategy id> - set the buffer placement strategy
clearmacros - remove all macros
dumpe - dump all physical operators of the current execution plan
dumpp <query id> - dump the physical plan of a query
dumpr - dump physical plan of all registered roots
installFeature - install feature
installableFeatures - lists features that can be installed
installedFeatures - lists current installed bundles
login <login username password> - Login user with name, password and tenantname.
logout <logout> - Logout current user.
lsbuffer - show available buffer placement strategies
lsmacros - show all available macros
lsparser - show available parser
lsqueries - show registered queries
lsscheduler - show available schedulers
lsschedulingstrategies - show available scheduling strategies
lssources - show registered sources
macro <macro name> - show macro content
meta <query id> - dump meta data of the query with QUERYID (only if root is a sink)
parser <parser id> - sets the current parser
qstart <query id> - start execution of query
remove <query id> - remove query
removemacro <macro name> - remove macro
resetUpdateSite - reset location of current update site to default location
restart - restarts odysseus
runFromFile <file name> [useProp] - runs commands contained in a file.
The filename can be specified in an absolute manner or, together with "setPath" command in a relative manner.
If useProp option is set, then the path to the file will be automatically read from system property user.files.
runfile <filename> - Reads a file from the current working or absolute directory and executes each line as if
it comes from the console.
    skips blank and commented (with //) lines.
runmacro <macro name> - execute macro
runscript -f <filename> - run query from <filename> for current parser (Hint set Parser to OdysseusScript
before executing .qry-Files) [with console-output-sink] [filepath automatically read from user.files]
[with restructure or not] [with|without metadata set in physical operators]
schedule - start scheduler
scheduler <scheduler id id_strategy> - set scheduler and it's strategy
sendBugReport - send bug report
setOutputFilename <filename> - Set a filename for result dump. Each result is dumped to this file. Call without
parameter to unset.
setPath <path> - Sets the path from which to read files. E. g. setPath 'C:\Users\name\' and addFromFile 'queries.
txt' uses the file C:\Users\name\queries.txt
setUpdateSite - change location of current update site
startrecord <macro name> - begin macro recording
stoprecord - finish recording of macro
stopschedule - stop scheduler
uninstallFeature - uninstall feature
updateCheck - checks if there are updates for odysseus
updateFeatures - tries to update odysseus
version - checks if there are updates for odysseus
```

