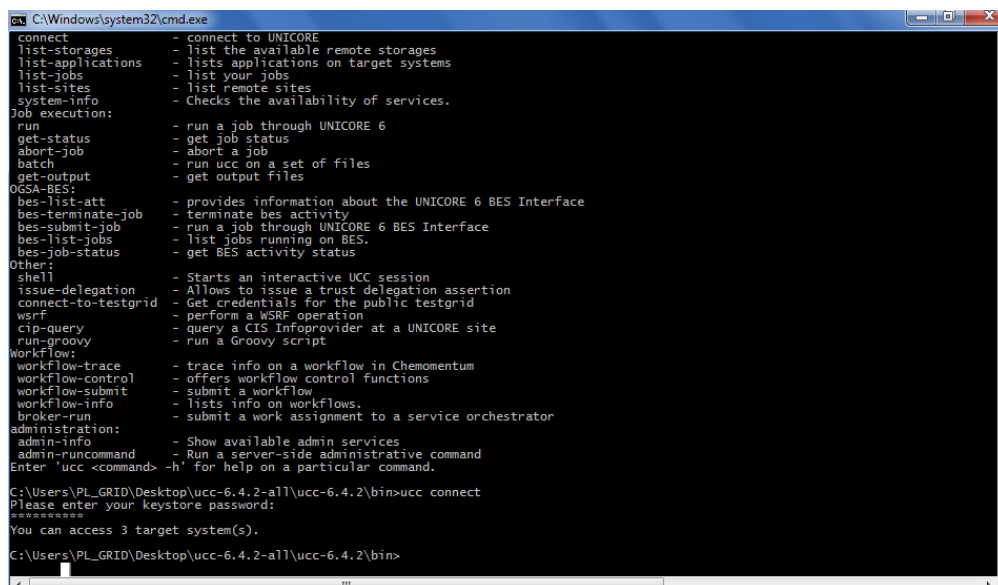


UNICORE Commandline Client

- Instalacja i konfiguracja na maszynie użytkownika
- UNICORE Commandline Client na serwerach dostępowych
- Polecenia UNICORE Commandline Client
- Co dalej?

UNICORE Commandline Client jest tekstowym klientem systemu UNICORE.



```
C:\Windows\system32\cmd.exe
connect          - connect to UNICORE
list-storages   - list the available remote storages
list-applications - lists applications on target systems
list-jobs       - list your jobs
list-sites      - list remote sites
system-info     - Checks the availability of services.
Job execution:
run             - run a job through UNICORE 6
get-status     - get job status
abort-job      - abort a job
batch          - run ucc on a set of files
get-output     - get output files
OGSA-BES:
bes-list-att   - provides information about the UNICORE 6 BES Interface
bes-terminate-job - terminate bes activity
bes-submit-job - run a job through UNICORE 6 BES Interface
bes-list-jobs  - list jobs running on BES.
bes-job-status - get BES activity status
Other:
shell          - Starts an interactive UCC session
issue-delegation - Allows to issue a trust delegation assertion
connect-to-testgrid - Get credentials for the public testgrid
wsrf           - perform a wSRF operation
cis-query      - query a CIS InfoProvider at a UNICORE site
run-groovy     - run a Groovy script
Workflow:
workflow-trace - trace info on a workflow in ChemoMentum
workflow-control - offers workflow control functions
workflow-submit - submit a workflow
workflow-info   - lists info on workflows.
broker-run     - submit a work assignment to a service orchestrator
administration:
admin-info     - Show available admin services
admin-runcommand - Run a server-side administrative command
Enter 'ucc <command> -h' For help on a particular command.
C:\Users\PL_GRID\Desktop\ucc-6.4.2-all\ucc-6.4.2\bin>ucc connect
Please enter your keystore password:
*****
You can access 3 target system(s).
C:\Users\PL_GRID\Desktop\ucc-6.4.2-all\ucc-6.4.2\bin>
```

Instalacja i konfiguracja na maszynie użytkownika

Klient UCC jest uruchamiany za pośrednictwem platformy Java, która musi być zainstalowana w systemie. Platforma dostępna jest pod adresem <https://java.com> dla maszyn z systemem operacyjnym Windows, Mac i Linux. Dla użytkowników Linuxa wygodniejszą alternatywą jest zainstalowanie platformy Java za pomocą systemowych narzędzi do instalacji oprogramowania.

Klient UCC pobrany może zostać ze strony projektu <https://sourceforge.net/projects/unicore/files/Clients/Commandline%20Client/>. W dalszej części podręcznika zakładamy użycie wersji z rozszerzeniem zip, która to może być użyta na każdym systemie operacyjnym. Wersja z rozszerzeniem tar.gz jest przeznaczona dla użytkowników Linuxa, pozostałe wersje - .deb i .rpm - również, aczkolwiek wymagają uprawnień administratora do instalacji. W zamian integrują się z systemem operacyjnym (np. poprzez dodanie polecenia ucc do ścieżki wykonywalnej).

Ściągnięta paczka może zostać rozpakowana w dowolnej lokalizacji na komputerze użytkownika systemu.



Wersja UCC

Opisana konfiguracja dotyczy klienta w wersji **7.3.0**, generalnie powinna być jednak poprawna również dla nieco wcześniejszych wersji 7.1.0 i 7.0.4.



Poniżej podano wersje komend dla systemu **Linux** lub **Mac**. W przypadku pracy na **Windows** należy zmienić separator katalogu ('/' na '\'), używać skryptu `ucc.bat` zamiast `ucc` oraz wprowadzić faktyczny katalog domowy użytkownika zamiast znaku tyldy (~).



Poniższa instrukcja działa przy pierwszym uruchomieniu, tzn. zakłada że nie istnieje wcześniej utworzona konfiguracja użytkownika. Jeśli taka istnieje zostanie ona nadpisana, jednak program UCC utworzy jej kopię zapasową.

Aby skonfigurować dostęp do Gridu najlepiej skorzystać z profilu konfiguracji klienta, który m.in. doinstaluje certyfikaty zaufanych CA oraz określi rejestry. W tym celu należy:

- Włączyć konsolę (terminal tekstowy) i przejść do katalogu w którym rozpakowano ściągnięty program UCC.
- Wykonać polecenie:

```
bin/ucc download-config -b -e -u https://unicore.grid.icm.edu.pl:8443/ccm/plgrid-profile-icm.xml 06:21:
```

42:3C:28:28:29:93:99:F9:BB:23:AE:B9:A7:DE:04:F1:87:FA

Klient utworzy domyślny katalog konfiguracyjny w `~/ucc/` oraz prekonfiguruje klienta do pracy z infrastrukturą PL Grid.

- W pliku `~/ucc/preferences` należy ustawić poprawną ścieżkę do paczki ze swoim certyfikatem i kluczem, oraz poprawnie ustawić format.
 - Dla pliku w formacie PKCS12:
`credential.format = pkcs12`
`credential.path = CIEKA_DO_PLIKU_PKCS12`
 - Dla pary plików w formacie PEM:
`credential.format = pem`
`credential.path = CIEKA_DO_PLIKU_PEM_Z_CERTYFIKATEM`
`credential.keyPath = CIEKA_DO_PLIKU_PEM_Z_KLUCZEM_PRYWATNYM`
 - Dla pliku w formacie JKS:
`credential.format = jks`
`credential.path = CIEKA_DO_PLIKU_JKS`
- Dla bezpieczeństwa NIE należy wpisywać hasła do klucza w pliku oraz koniecznie należy usunąć domyślne hasło (linia rozpoczynająca się od `credential.password =`).



Podstawowe informacje o użytkowaniu UCC zaprezentowane są poniżej. Można również wykorzystać szkolenia zdalne UNICORE, na które zapisać się można za pośrednictwem Portalu.

UNICORE Commandline Client na serwerach dostępowych

Użytkownik może wykorzystać gotową instalację tekstowego klienta systemu UNICORE (UCC w wersji 7.x.y) dostępnego w ICM, WCSS, Cyfronet, PCSS, TASK. W tym celu należy jednak, oprócz aplikacji o globalny dostęp do systemu UNICORE, wystąpić o dostęp do wybranych maszyn UI (*User Interface*).

The screenshot shows the PL-Grid Portal interface. The top navigation bar includes 'Strona główna', 'Podręcznik użytkownika', 'Rejestri PL-Grid', 'Moje konto', 'Zespoły i grupy', and 'Narzędzia'. The main content area is divided into several sections:

- Certyfikaty SimpleCA:** Information about online certificate generation.
- Zarządzanie certyfikatami:** A table showing certificate details for 'Autologin', 'DN', and 'Serial'. The 'Serial' column shows '53 8a 08 d2 cb 08 53 7e'.
- Eksport certyfikatu do LDAP:** A form for exporting certificates to LDAP, including fields for PKCS12 file name and private key password.
- Usługi:** A table listing global services with columns for 'Nazwa', 'Status', and 'Strona z informacjami'. The 'Globalny dostęp UNICORE' row has a red circle around the 'Aplikuj o usługę' link.
- Aplikuj o usługę:** A table listing specific services with columns for 'Nazwa', 'Zasieg', 'Usługa nadrzędna', 'Akcje', and 'Strona z informacjami'. The 'Akcje' column contains 'Aplikuj o usługę' links for various services like 'Dostęp do MySQL - Cyfronet', 'Dostęp do HPCPU', etc.
- Zarządzanie bazami danych:** A section with a message: 'Portal jest nieaktywny, aktualnie zalogowany użytkownik nie posiada żadnej aktywnej usługi bazodanowej.'

Instalacja i konfiguracja

Aby móc korzystać z zainstalowanego klienta UCC użytkownik powinien:

- aplikować o dostęp i zalogować się do UI wybranego ośrodka (klient UCC zainstalowany jest we wszystkich ośrodkach: ICM, WCSS, Cyfronet, PCSS, TASK) zgodnie z opisem dostępnym w rozdziale [dostęp do User Interface](#);
- wpisać polecenie `ucc`, które spowoduje przygotowanie plików konfiguracyjnych użytkownika - zostanie utworzony katalog `~/ucc` z domyślną konfiguracją.
- klucz i certyfikat powinny być dostępne automatycznie, o ile użytkownik ma je wygenerowane.

Przykładowe użycie programu pokazane jest w szkoleniach zdalnych UNICORE, na które zapisać się można za pośrednictwem Portalu.

Polecenia UNICORE Commandline Client



Poniżej prezentujemy podstawy użycia klienta UCC. Bardziej dogłębne informacje można znaleźć w szkoleniach zdalnych UNICORE (dostęp przez portal PL-Grid) oraz w oficjalnym podręczniku UCC <http://www.unicore.eu/documentation/manuals/unicore/files/ucc/ucc-manual.html>

Połączenie z Gridem

Aby połączyć się z Gridem, należy wykonać polecenie `ucc connect`. Pomoc można uzyskać za pomocą polecenia `ucc` z opcją `-h`:

```
$ ucc connect
$ ucc -h
```

Jeżeli hasło użytkownika nie jest zapisane w pliku `preferences`, wówczas będzie on pytany o nie przy każdym wywołaniu UCC. Aby tego uniknąć wykorzystać można tryb interaktywny wpisując: `ucc shell`. Tryb ten nie wymaga pisania `ucc` przed poleceniami. Oferuje ponadto rozwijanie poleceń.

Zasoby

Aby sprawdzić listę dostępnych systemów docelowych, magazynów (stoarge) bądź aplikacji, użytkownik wykonać powinien następujące polecenia:

```
$ ucc list-sites
$ ucc list-storages
$ ucc list-applications
```

Opis zadania

UNICORE Commandline Client wykorzystuje opis zadań w formacie JSON pozwalając na wskazanie uruchamianego programu, argumentów, ustawień środowiska oraz przesyłanych plików. Przykład opisu zadania przedstawiony został poniżej. Może on zostać skopiowany do pliku `date.u`.

```
# simple job: run Date
{
  ApplicationName: Date,
  ApplicationVersion: 1.0,
}
```

Uruchamianie zadania

Aby uruchomić zadanie, użytkownik powinien wpisać polecenie:

```
$ ucc run date.u -v
```

Standardowym wyjściem może być np. plik `1bc1bb08-7737-4fb1-854e-5d89ba18d7f0.stdout`.

Podstawowe opcje:

- `-v` pozwala wyświetlić szczegółowe wyniki.
- `-b` zwraca skrócone nazwy plików (bez hasha).
- `-a` pozwala na uruchomienie zadania w sposób asynchroniczny: pliki wejściowe są wysyłane na zdalny system, gdzie zadanie jest zatwierdzane do uruchomienia, jednak wyniki mogą zostać pobrane w późniejszym terminie za pomocą polecenia `get-output`. Aby uzyskać status zadania, użyć należy komendy `get-status`. Jako argument podać można plik zadania uzyskany w wyniku polecenia `run -a` bądź też End Point Reference (EPR) otrzymany z listy zadań (`list-jobs`):

Przykład asynchronicznego uruchomienia zadania:

```
$ bin/ucc shell

ucc>run -a -b ./samples/date.u

/home/golbi/unicore/test/ucc-distribution-7.3.0/.4a88e886-8ff3-479c-a52e-a64277d87e19.job

/home/golbi/unicore/test/ucc-distribution-7.3.0/.4a88e886-8ff3-479c-a52e-a64277d87e19.properties

ucc>job-status 4a88e886-8ff3-479c-a52e-a64277d87e19.job
https://unicore.grid.task.gda.pl:8080/TASK-GALERAPLUS/services/JobManagement?res=4a88e886-8ff3-479c-a52e-a64277d87e19 SUCCESSFUL exit code: 0
ucc>get-output 4a88e886-8ff3-479c-a52e-a64277d87e19.job
SUCCESSFUL exit code: 0
stdout
stderr
```

Import/Export plików

Aby importować pliki z lokalnego komputera do katalogu zadania (lub eksportować z katalogu zadania do lokalnego komputera), użytkownik powinien wskazać w skrypcie pliki źródłowe i docelowe:

```
{
  Imports: [
    { From: "u6://StorageName/fileName", To: "remoteFileName" },
  ]
  Exports: [
    { From: "remoteFileName", To:"u6://StorageName/fileName" },
  ]
}
```

Plik może również zostać wysłany na Stoarge i pobrany z niego za pomocą poleceń `ucc put-file` oraz `ucc get-file`. Komenda `ls` pozwala na wypisanie wszystkich plików z katalogu:

```
$ ucc put-file -s fileName -t u6://StorageName/fileName
$ ucc get-file -s u6://StorageName/fileName -t newFileName
$ ucc ls u6://StorageName/fileName
```

Zasoby

W części `Resources` skryptu opisującego zadanie użytkownik może określić zasoby, z którymi zadanie zostanie uruchomione na zdalnym systemie. Sekcja ta może wyglądać następująco:

```
Resources: {
  Memory: 128000000,
  Nodes: 1,
  CPUs: 8 ,
}
```

Przykład

W poniższym przykładzie plik `localScript.sh` jest skryptem wypisującym do pliku o nazwie `newFile` słowo `Hello`:

```
echo "Hello" >> newFile
```

Użytkownik może utworzyć plik `localScript.sh` na lokalnym komputerze. Aby wysłać go na Storage powinien najpierw sprawdzić nazwę dostępnych Storage'y (tutaj: `SHARE-ACK`). Po przesłaniu pliku można wyświetlić znajdujące się w nim pliki:

```
$ ucc list-storages
$ ucc put-file -s localScript.sh -t u6://SHARE-ACK/storageScript.sh
$ ucc ls u6://SHARE-ACK/Home
```

Plik `bash.u` jest skryptem importującym `storageScript.sh` z Home Stoarge do katalogu zadania i zmieniającym jego nazwę na `remoteScript.sh`. Skrypt tworzy plik `newFile`, który następnie jest pobierany na komputer użytkownika jako `localNewFile`. Plik ten jest również eksportowany do Home Storage systemu UNICORE.

```
{
  ApplicationName: "Bash shell",
  Environment: [
    "SOURCE=remoteScript.sh",
  ],
  Imports: [
    { From: "u6://SHARE-ACK/storageScript.sh", To: "remoteScript.sh"}
  ],
  Exports: [
    { From: "newFile", To: "localNewFile"},
    { From: "newFile", To: "u6://SHARE-ACK/storageNewFile"}
  ],
  Resources: {
    CPUs: 1 ,
  }
}
```

Zadanie `bash.u` może zostać uruchomione poleceniem `ucc run`. Na komputerze użytkownika powinien zostać utworzony plik `localNewFile`.

```
$ ucc run bash.u -v
$ ucc ls u6://SHARE-ACK
```

Co dalej?

