

The UNICORE Grid System

Tutorial

Klaus-Dieter Oertel, Pallas GmbH
kdoertel@pallas.com

Mathilde Romberg, Forschungszentrum Jülich
m.romberg@fz-juelich.de

Euro-Par 2002, Paderborn



Outline

1. Introduction UNICORE
 - UNICORE Plus Project
 - Software Status
 - Architecture
 - [– Acronyms]
2. Client
 - “My first job”
 - Installation
 - Configuration
 - Job Preparation
 - Job Monitoring
 - Application Specific Interfaces (Plugins)



Outline (cont.)

3. Server
 - Overview (Packages, Components, Prerequisites)
 - Gateway (Installation, Configuration, Maintenance)
 - Network Job Supervisor
 - Installation, basic configuration
 - Incarnation Data Base
 - Maintenance
 - UNICORE User Data Base
 - Target System Interface (Installation, Configuration)
4. Discussion



The UNICORE Grid System

Tutorial

Part 1 Introduction



UNICORE Plus Project

UNiform Interface to COmputing REsources

- Follow-on of the UNICORE project
- Funded by German Ministry of Education and Research (bmb+f), Grant-id: 01 IR 001
- Grant period 1.1.2000 - 31.12.2002
- Development of a prototype for the seamless, secure, and intuitive access to distributed supercomputer resources
- <http://www.fz-juelich.de/unicoreplus>
- <http://www.unicore.de>



Partner

- Research Center Jülich (FZJ, Project Coordinator)
- German Weather Service (DWD)
- Computer Center University of Stuttgart (RUS)
- Pallas GmbH, Brühl
- Center for High Performance Computing at TU Dresden (ZHR)
- Computer Center University of Karlsruhe (RUKA)
- Konrad Zuse Center, Berlin (ZIB)
- Leibniz Computer Center, Munich (LRZ)
- Paderborn Center for Parallel Computing (PC²)
- Fujitsu Laboratory of Europe (former fecit)



Goals

- Closer cooperation of HPC Centers
- Support for computational science
- Overcome the seams created by
 - Incompatible system software
 - Site policies and practices
- Exploit existing and emerging technologies
 - Java, Web-techniques
 - X.509 certificates
- Minimal interference with local site policies and administration



Project Plan

- Based on results of UNICORE project
- Resource Modelling (static)
- Application Specific Interfaces (CPMD, NASTRAN, ..., generic / services)
- Extended Work Flow (repeat, if-then-else, ...)
- Data Management Enhancements (high-speed, high-throughput, file transfer independent of job)
- Metacomputing (Co-scheduling, MPI and PACX integration, Vampir extensions)
- Advanced administration



Software Status

- Current version 3.6
- User Client available for Windows and Unix (Linux,...) , runs also on Macintosh
- Servers to be run on Unix (Linux, ...)
- Target systems Unix (Linux, ...) only
- T3E, SP3, VPP, hpcLine, SR 8000, SX-5, PC-Clusters, ... as targets
- Several NQS dialects, LL, LSF, PBS, CCS as target batch systems



Software Status (cont.)

- UNICORE is deployed at the UNICORE Plus and the EUROGRID project partner centers
- UNICORE Forum e.V. (www.unicore.org) distributes UNICORE under community source license
- Public test system (www.fz-juelich.de/unicore-test) for testing client functions



Deployment

- European Projects use UNICORE
 - EUROGRID (www.eurogrid.org)
Application Testbed for European GRID computing partly funded under EC grant IST-1999-20247
 - GRIP (www.grid-interoperability.org)
Grid Interoperability Project partly funded under EC grant IST-2001-32257
 - OpenMolGRID
Open Computing Grid for Molecular Science and Engineering partly funded by EC grant IST-2001-37238



Download Page

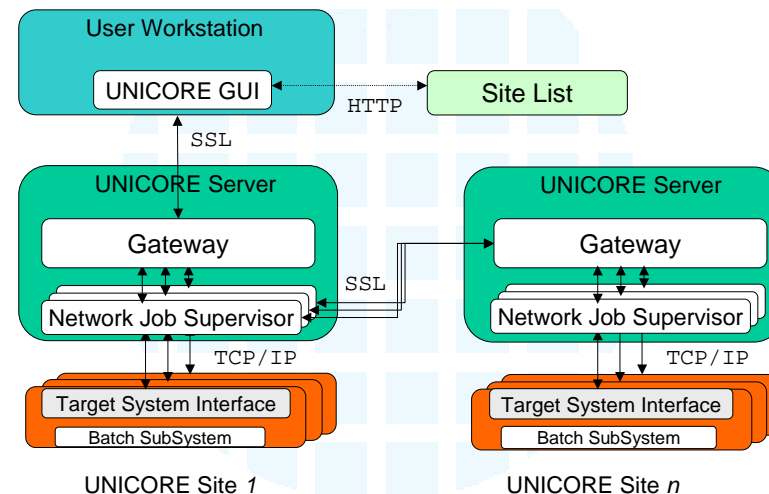
- UNICORE Software and Sources available at <http://www.unicore.org/downloads.htm>:
 - Introduction
 - Package bundles (Client, Server, full)
 - Abstract Job Object
 - Client
 - Plugins
 - Gateway
 - Server
 - Contributions
 - License
 - Certificate Authority



Highlights

- Support for Batch-Applications
- Heterogeneous metacomputing
- Transparent data staging / transfer
- Uniform user authentication and security mechanisms
- Uniform GUI for job creation and monitoring
- Easy integration of new applications through plugins
- Jobs in XML formats
- Command line interface

Architecture



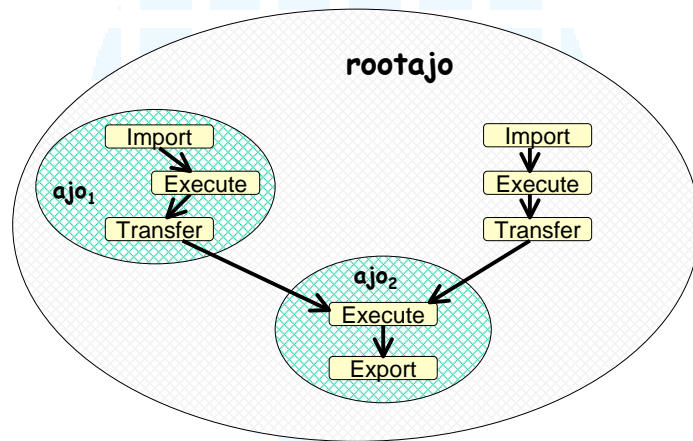
UNICORE Job

- Job contains
 - Sub-jobs and tasks
 - Dependency information
 - Target system
- Tasks are translated into batch jobs for the destination system

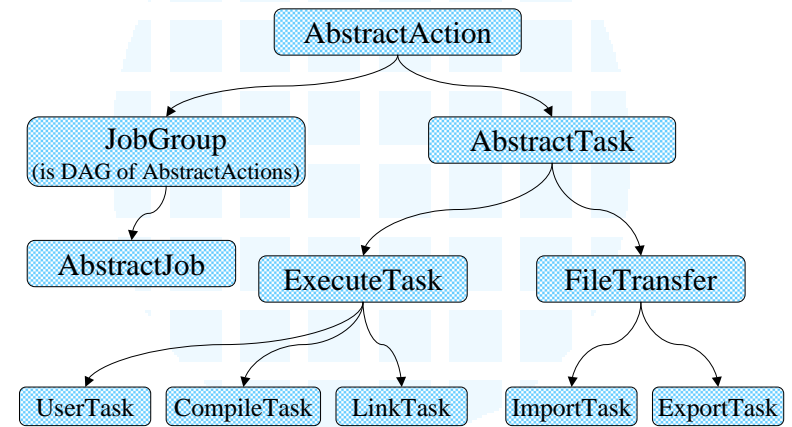
Abstract Job Object (AJO)

- Abstract representation of UNICORE job
- Recursive Java object
- Specifies all actions to be performed by UNICORE
 - Execute task
 - File transfer task
 - Control task
- Contains dependency graph
- Contains resource specification
- Contains data set descriptions for data to be streamed

AJO - Example



AJO – Class Hierarchy



Data Model

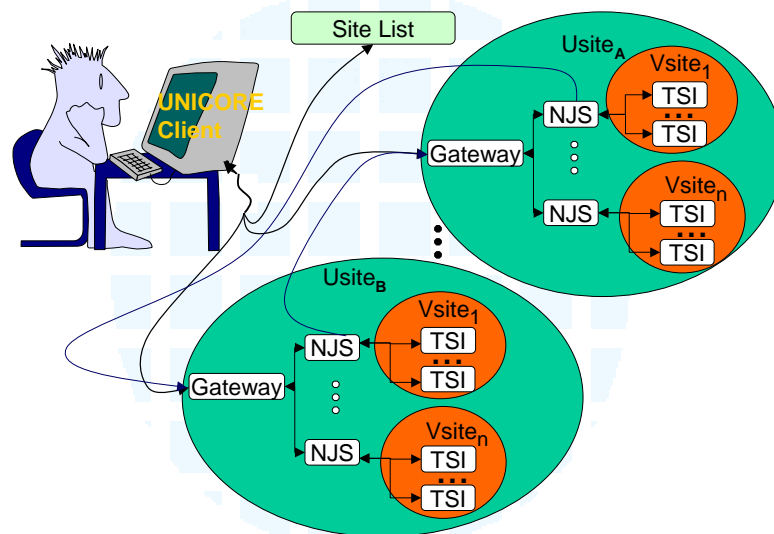
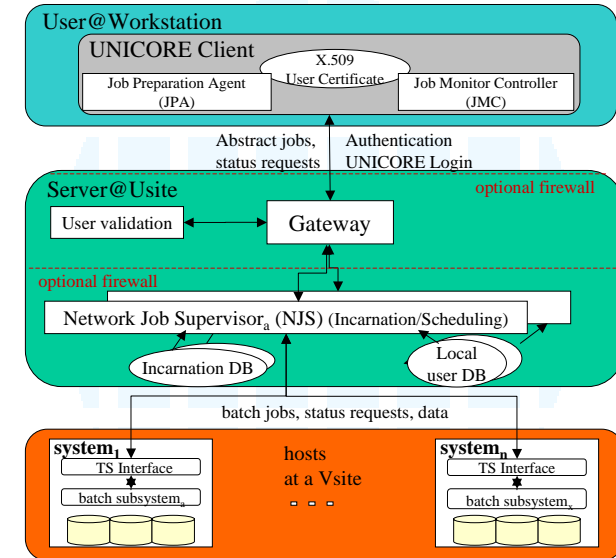
- UNICORE file space per UNICORE job
- Non-permanent
- User has to specify remote data location explicitly
- Data import / export / transfer

Security

- Secure Socket Layer (SSL)
- X.509v3 user and server certificates
- User's secret key stored in pw protected keystore at client
- UNICORE Login \equiv User certificate
- Authorization by mapping of certificate to local userid
- AJOs signed with user certificate
- Users may have multiple certs from different signers
- Server components accept certs from multiple signers
- Site Specific Security Objects (SSO)

UNICORE Protocol Layer

- Defines the structure of data sent between Client and Servers
- Layered on top of existing protocols
- Refers to Client – Gateway connection
- Coupled request / reply pairs
 - Contain serialized Java object and an optional stream of bytes



Acronyms

Usite	Site providing UNICORE services
Vsite	Computing resource, target system
Ujob	UNICORE Job
Ulogin	UNICORE Login, X.509 certificate
Xlogin	Unix Login at Vsite
Uspace	Temporary file space for Ujob at Vsite
Xspace	File space at the Vsite
Nspace	File space on the computer where Client runs



AJO	Abstract Job Object
UPL	UNICORE Protocol Layer (Client – Gateway)
NJS	Network Job Supervisor
Incarnation	Transl. of AJO into batch job using IDB
IDB	Incarnation Data Base
UUDB	UNICORE User Data Base
TSI	Target System Interface
JPA	Job Preparation Agent, part of Client
JMC	Job Monitor Controller, part of Client

The UNICORE Grid System

Tutorial

Part 2 Client

Outline

1. Introduction UNICORE
 - UNICORE Plus Project
 - Software Status
 - Architecture
 - [- Acronyms]
2. Client
 - “My first job”
 - Installation
 - Configuration
 - Job Preparation
 - Job Monitoring
 - Application Specific Interfaces (Plugins)

Client Versions

- Version 3.6
 - Available at www.unicore.org/downloads.htm
 - User guide included
- Tutorial: already version 4.0
 - Available late autumn 2002
 - Extended functionality (e.g. work flow constructs)
 - Improved GUI
 - GUI of versions differ but use Tutorial + User Guide for version 3.6

“My First Job”

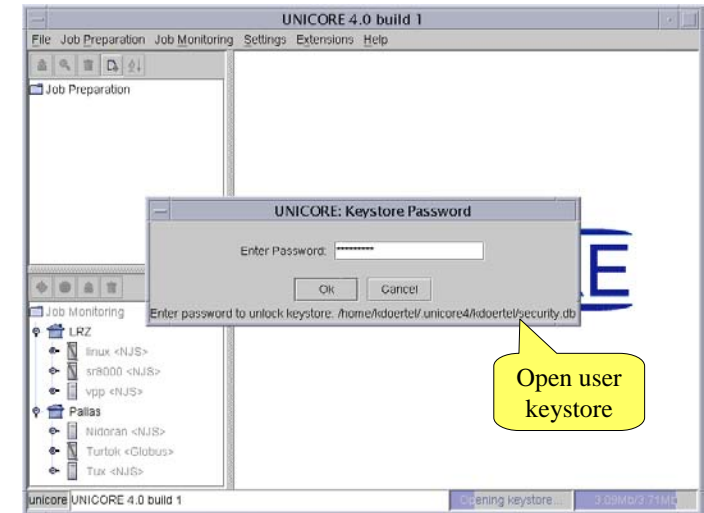
Assumption: Client installed & configured

- Login to Client
- Create a new job
- Look for Resource Information
- Specify script task to be executed
- Submit the job
- Query the job status
- Retrieve job output
- Delete job from Vsite

Client “My First Job”



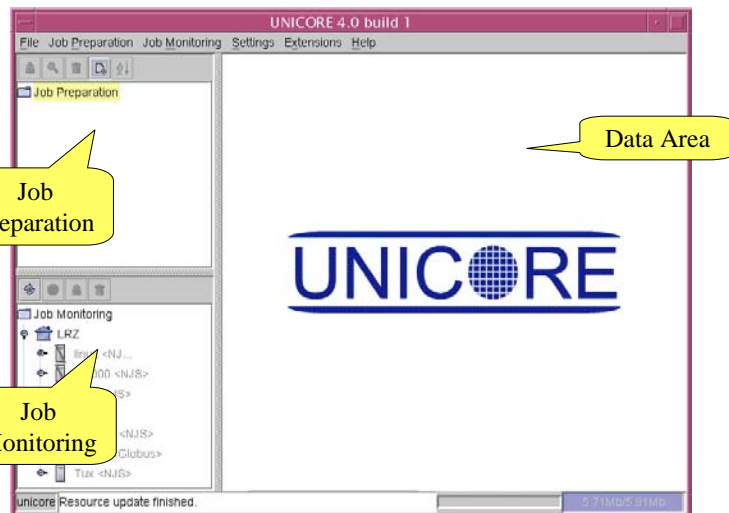
Login



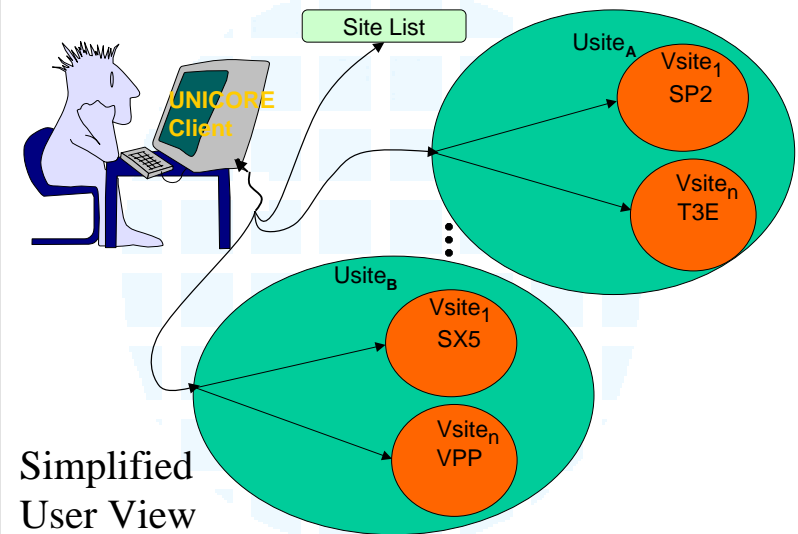
Client “My First Job”



Client GUI



Client “My First Job”

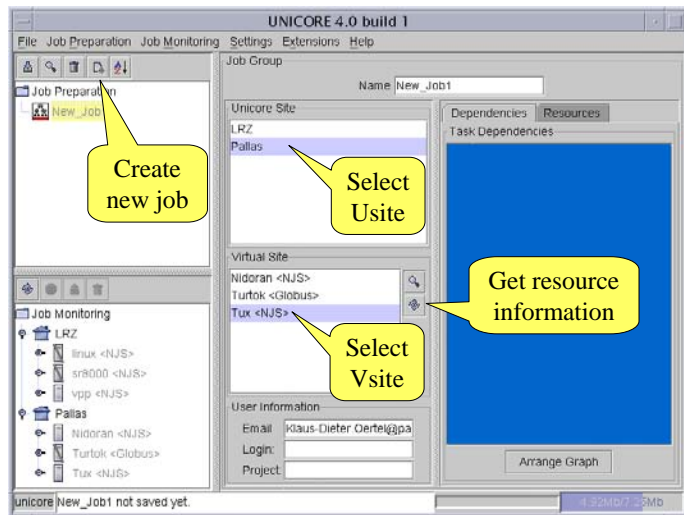


Client “My First Job”



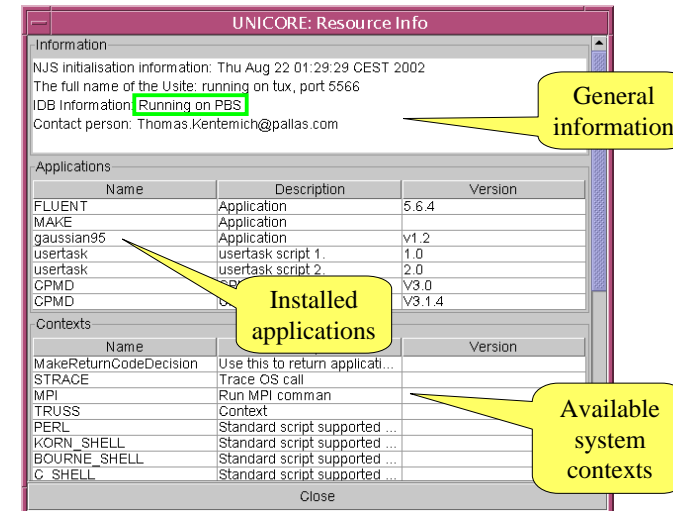
New Job

Client
• "My First Job"



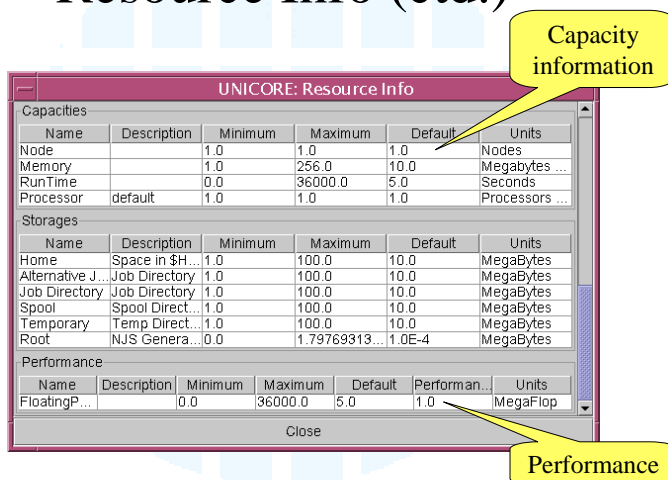
Resource Info

Client
• job preparation



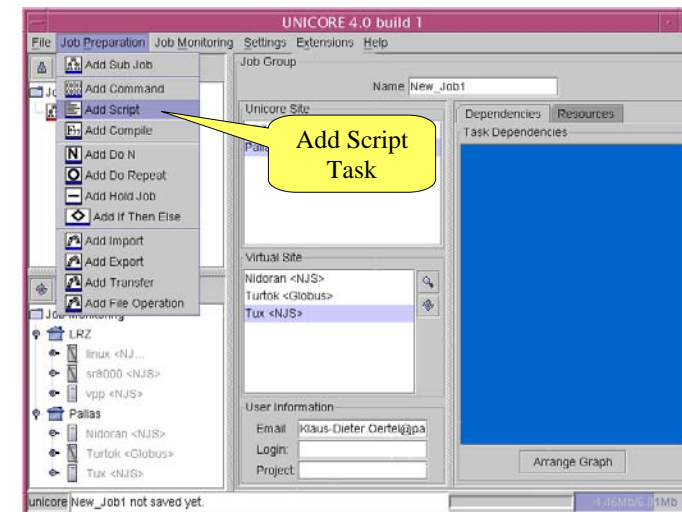
Resource Info (ctd.)

Client
• job preparation



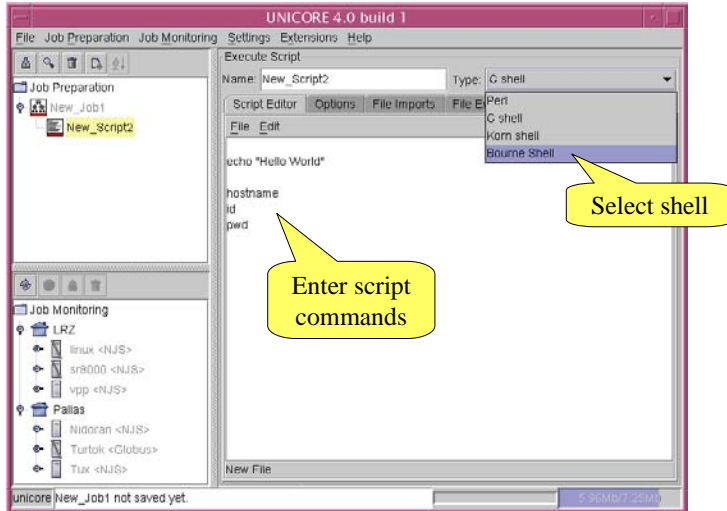
Add Script Task

Client
• "My First Job"



Edit Script

Client
• "My First Job"

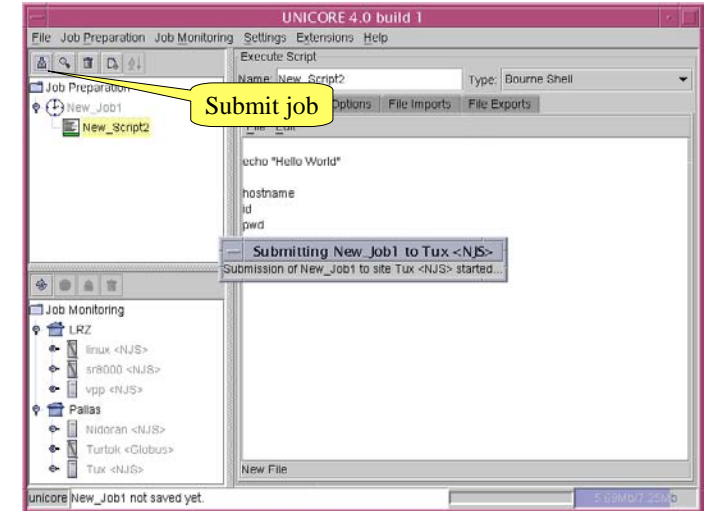


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

37

Submit Job

Client
• "My First Job"

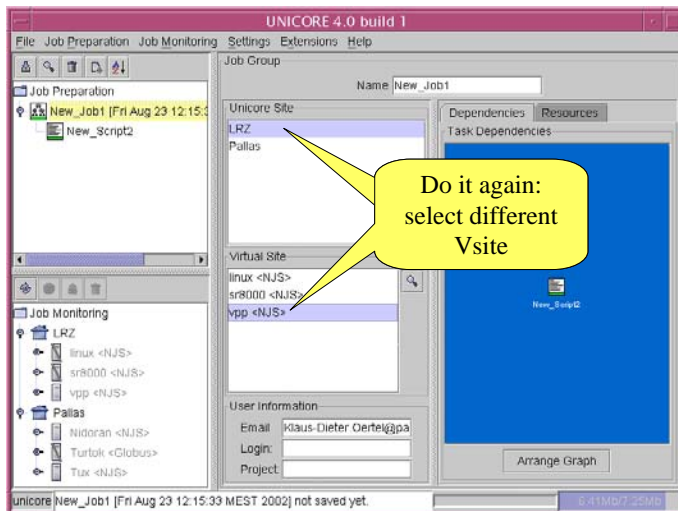


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

38

Select VPP from LRZ

Client
• "My First Job"

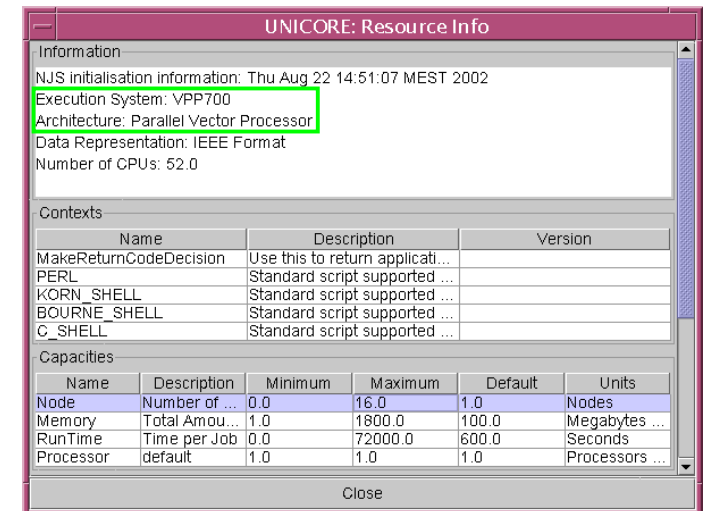


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

39

Resource Info of VPP

Client
• "My First Job"

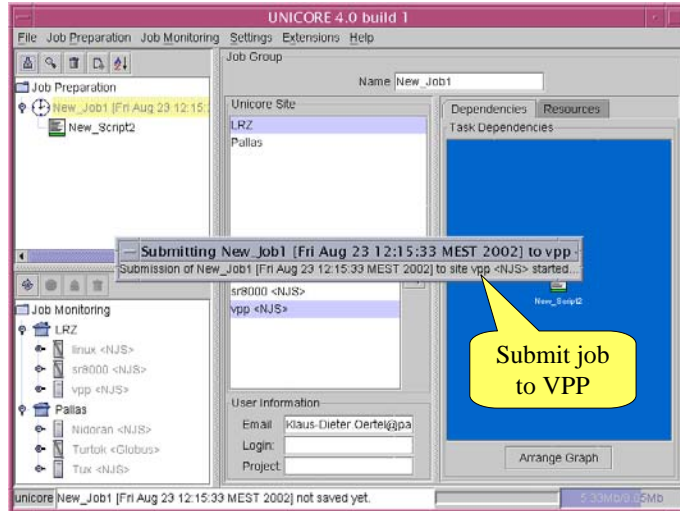


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

40

Submit to VPP

Client
• "My First Job"

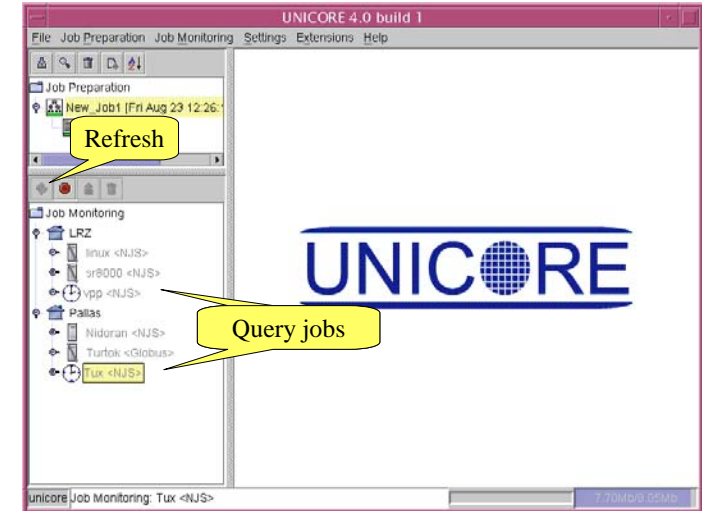


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

41

Query Jobs at Vsite

Client
• "My First Job"

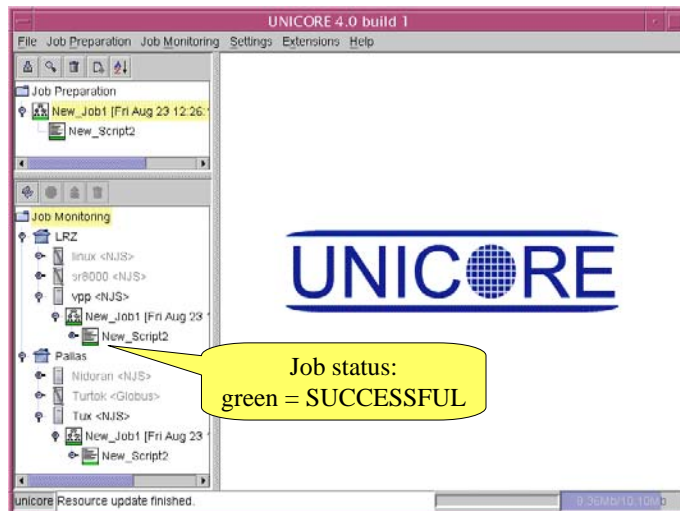


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

42

Job Status

Client
• "My First Job"

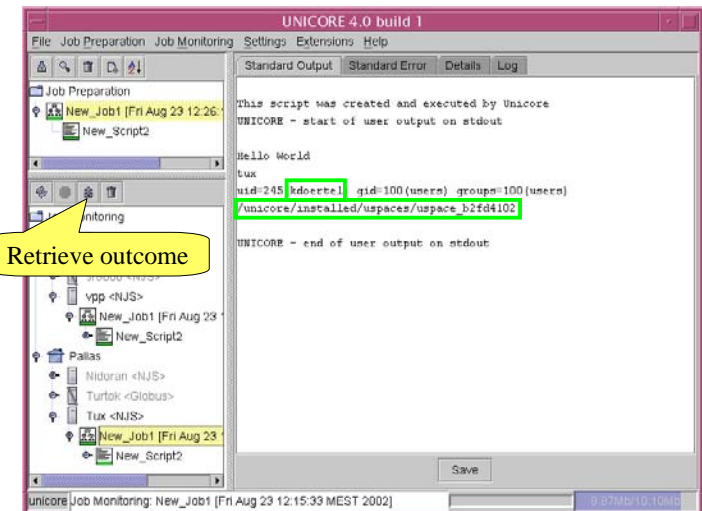


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

43

Retrieve Outcome Tux

Client
• "My First Job"

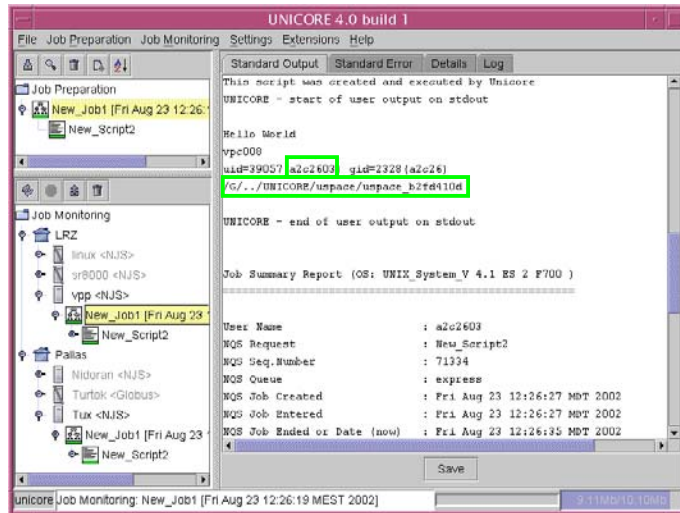


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

44

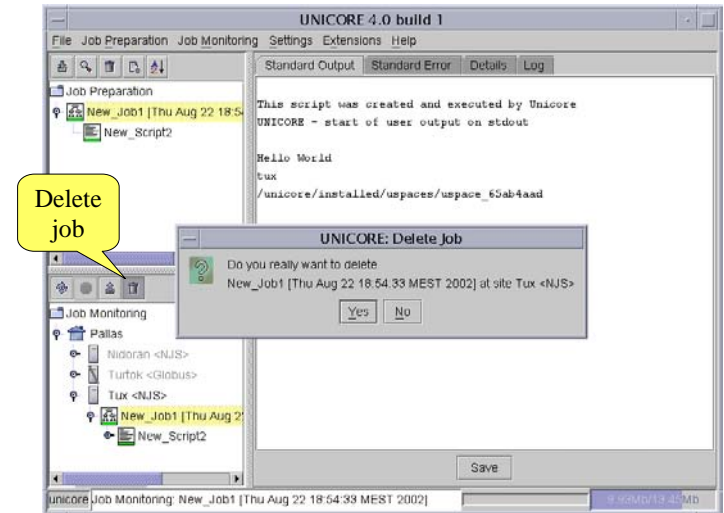
Retrieve Outcome VPP

Client
• "My First Job"



Delete Job from Vsite

Client
• "My First Job"



Lessons Learned

Client
• "My First Job"

Use of UNICORE is

- secure
 - Password protected access to keystore
- intuitive
 - Jobs generated from components in a GUI
- seamless
 - Abstract jobs are portable
 - One password to access systems at different sites
 - Transparent use of batch systems (here: PBS, NQS)



Installation Prerequisites

Client
• Installation

- Java 1.4.x for Client 4.0
- X509v3 certificates
 - Public CA signer certificate of Usite's (Gateway's) certificate
 - Available from Usite
 - User keystore containing public user certificate (+ private user key)
- At the sites:
 - Mapping of the User certificate to a UNIX login



Getting User Certificates?

- Test certificates available from test system www.fz-juelich.de/unicore-test
- Ask UNICORE site manager for user certificates accepted at that site
- Install OpenSSL to setup in-house UNICORE system with own



Installation

- Unix: `tar -xvf UNICORE_Client.tar`
- Windows: self-extracting UNICORE_Client.exe

```

UNICORE_Client/.java.policy          UNICORE_Client/lib/commandPlugin.jar
UNICORE_Client/CHANGES              UNICORE_Client/lib/compilePlugin.jar
UNICORE_Client/LICENSE.jakarta-oro  UNICORE_Client/lib/iaik_javac_crypto.jar
UNICORE_Client/LICENSE.xerces       UNICORE_Client/lib/iaik_jce_full.jar
UNICORE_Client/README               UNICORE_Client/lib/iaik_ssl.jar
UNICORE_Client/bin/unicore          UNICORE_Client/lib/jakarta-oro.jar
UNICORE_Client/doc/unicore-guide.pdf UNICORE_Client/lib/jh.jar
UNICORE_Client/gateways.xml         UNICORE_Client/lib/scriptPlugin.jar
UNICORE_Client/lib/SystemDefaults.txt UNICORE_Client/lib/smallServicePlugin.jar
UNICORE_Client/lib/ajo.jar          UNICORE_Client/lib/whitespace.xml
UNICORE_Client/lib/autoupdatePlugin.jar UNICORE_Client/lib/xalan.jar
UNICORE_Client/lib/bc-jce.jar       UNICORE_Client/lib/xerces.jar
UNICORE_Client/lib/client.jar

```



Java Policy File

Client 4.0 uses Java security manager:

- Access to keystore, file system etc. controlled by entries in file `$HOME/.java.policy`:

```

grant codeBase "${unicore.system.plugin.dir}" {
    permission java.io.FilePermission "<<ALL FILES>>", "read, write";
    permission java.net.SocketPermission "*", "connect";
    permission java.net.SocketPermission "*", "resolve";
    permission ...
}

```

- Copy UNICORE_Client/.java.policy into your local home directory
- Or: add the contents to an existing `$HOME/.java.policy`



Client Start

- Unix
 - execute UNICORE_Client/bin/unicore
- Windows
 - double-click on UNICORE_Client/lib/client.jar
- Command line execution
 - Only for prepared jobs:

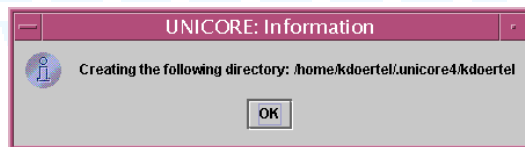

```
java -jar client.jar -nogui -password <password>
-jobs <job1#job2 #job3...>
```



Config Directory

Starting from scratch:

- Creation of directory \$HOME/.unicore4/\$USER



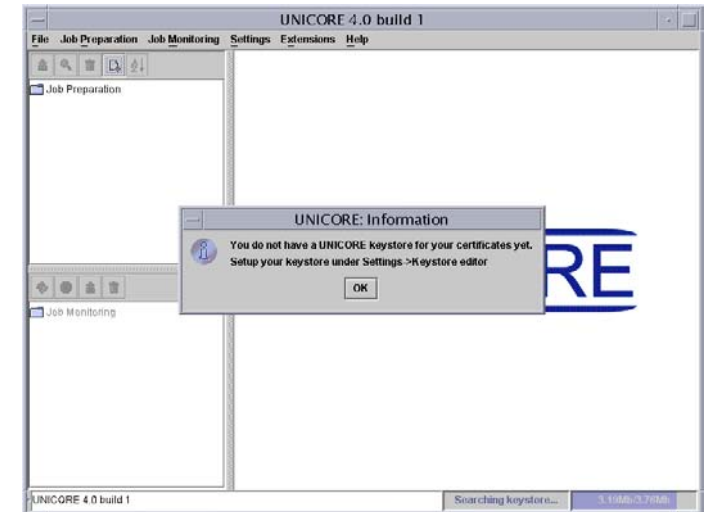
Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

53

Keystore warning



Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

54

Keystore Editor

Used to create a new UNICORE keystore and to:

- Import Usite's CA certificate
- Import the user keystore

Also used to:

- Query certificate details
- Export the user certificate
- Generate private + public key + corresponding certification request

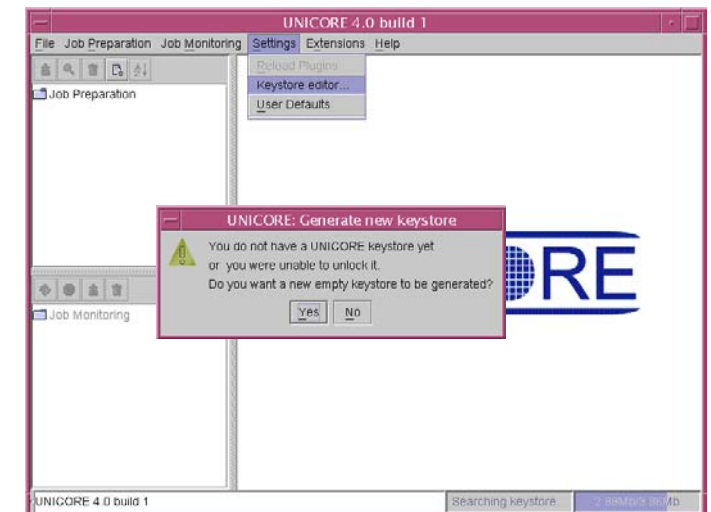
Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

55

Open Keystore Editor



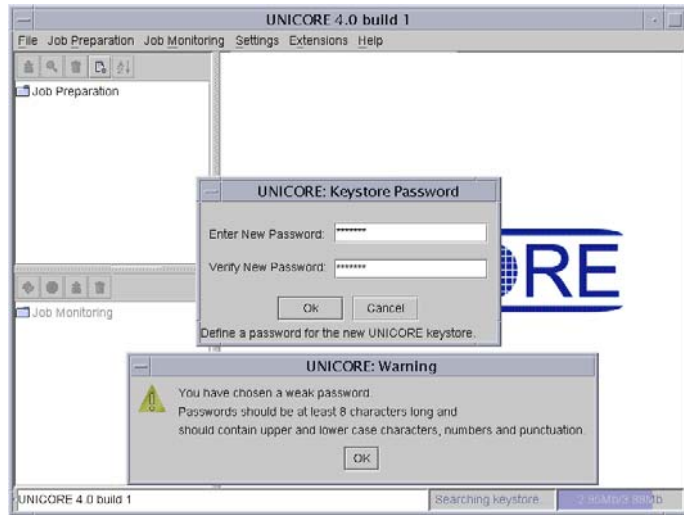
Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

56

Password Protected Keystore



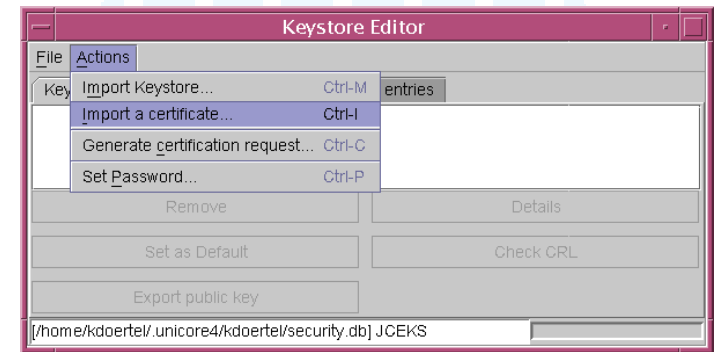
Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

57

Certificate Import



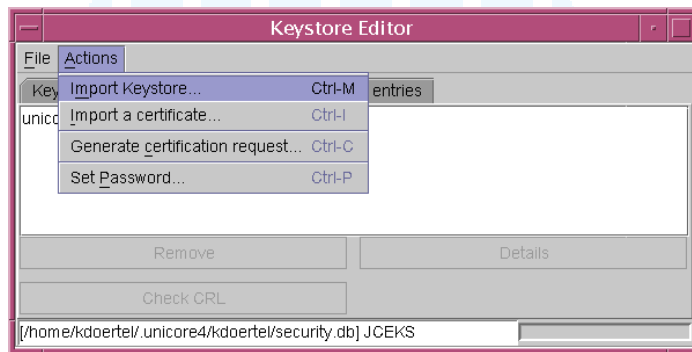
Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

58

Keystore Import



Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

59

Understanding Certificates

- Basis: private-public key encryption
- Why to trust the public key if we do not know the issuer?
- Create a certificate: the public key is signed by a certificate authority (CA)
- Trust the certificate because we trust the included CA certificate
- Certificate chain: we trust the CA certificate because we trust its signing CA
- Analogon: we trust a passport because we trust the issuing local government and the chain of government levels

Client
• Installation



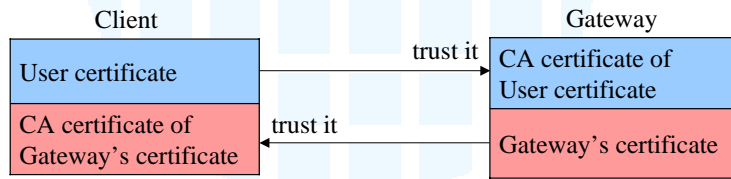
Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

60

UNICORE and Certificates

- Why are two certificates (User, Gateway's CA) needed?
- On connection, Client and Gateway exchange certificates for authentication
- To trust a certificate the CA certificate is required for comparison

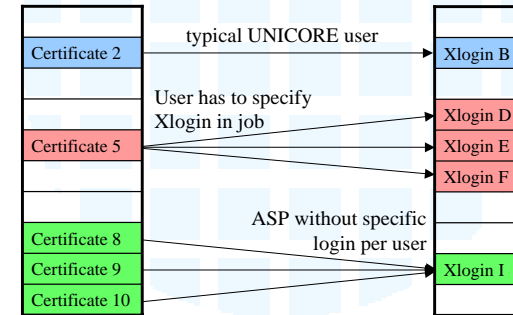
Client
• Installation



Certificate Mapping

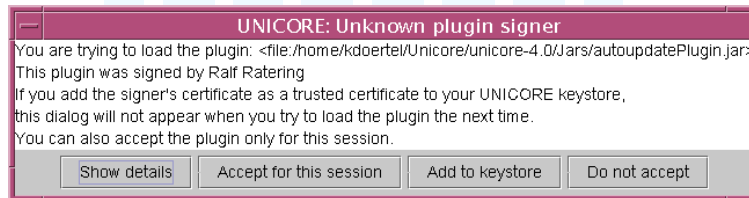
- At Vsites: Certificates are mapped in the UUDB (UNICORE User Database) to UNIX logins

Client
• Installation



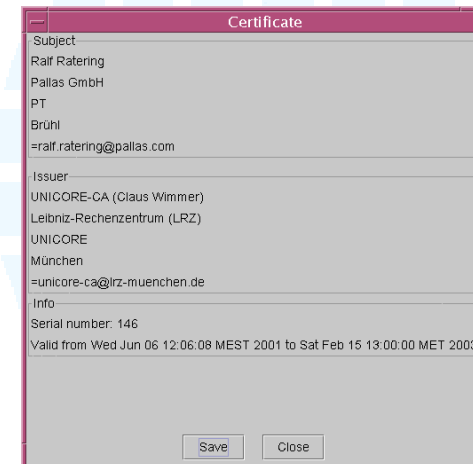
Trusted Plugin Signer

Client
• Installation

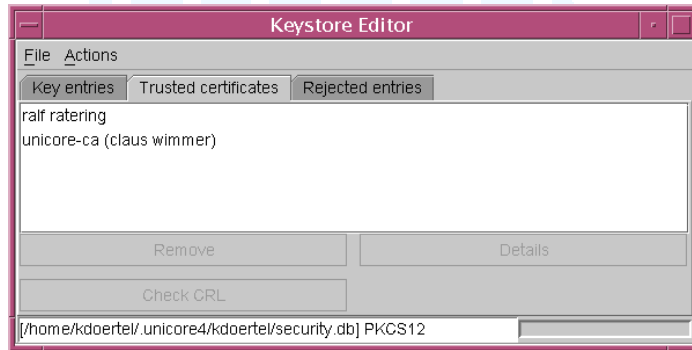


Plugin Signer Details

Client
• Installation



Trusted Certificates



Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

65

Config Directory

Files in \$HOME/.unicore4/\$USER after keystore setup and loaded plugins:

```
CommandDefaults.txt  
CompileDefaults.txt  
ScriptDefaults.txt  
UserDefaults.txt  
clientlog.txt  
clientlog.txt.lck  
clientlog.xml  
clientlog.xml.lck  
resourceCache.bin  
security.db  
security.db.ssl
```

Client
• Installation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

66

Configuration

- Essential configuration in User Defaults:
 - URL to XML file specifying Usites' addresses
 - Default from UNICORE Plus project:
www.unicore.de/unicoreSites.xml
 - Overwrite by local file
- Other configurations for convenience
 - Default paths for job storing/loading, etc.
 - Look and feel
 - Client logging level and format
- Plugin specific default settings

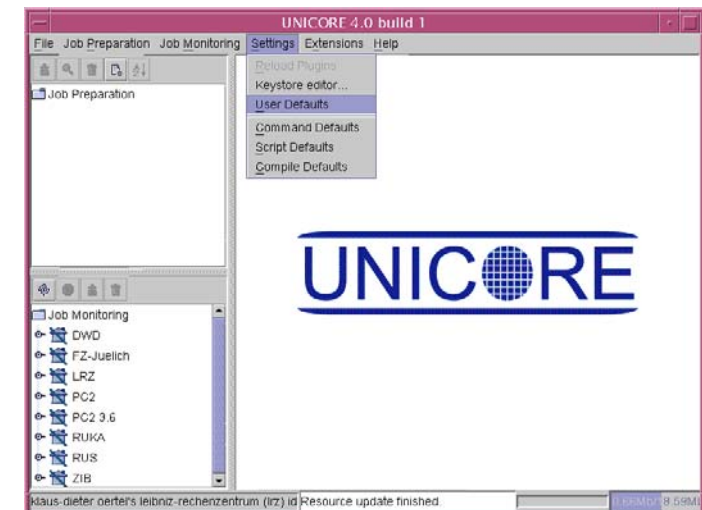
Client
• Configuration



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

67

User Defaults



Client
• Configuration



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

68

User Defaults: New Site List

Client
• Configuration

UNICORE: User Defaults

General Paths System Settings User Interface Logging Settings

User Email: klaus-Diet

Login:

Project:

URL for UNICORE Site Server: home/kdoertel/unicore4/kdoertel/unicorePallas40.xml

Certificate Revocation List: https://unicore-ca.lrz-muenchen.de/unicore-ca.crl

Proxy Server Address: webcheck.pallas.com Port: 8080 On/Off:

Ok Cancel

XML file specifying Usites' addresses



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

69

User Defaults: sites XML file

Client
• Configuration

```
<?xml version="1.0" ?>
<!DOCTYPE UsiteList [
  <!ELEMENT UsiteList (Usite+)>
  <!ELEMENT Usite EMPTY>
  <!ATTLIST Usite
    name CDATA #REQUIRED
    description CDATA #IMPLIED
    address CDATA #REQUIRED
    port CDATA #REQUIRED
  >
]>

<!-- List of UNICORE sites -->

<UsiteList>
  <Usite name = "Pallas" description = "Pallas"
    address = "tux" port = "4000">
  </Usite>
  <Usite name = "LRZ" description = "Leibniz Rechenzentrum Muenchen"
    address = "unicore.lrz-muenchen.de" port = "4433">
  </Usite>
</UsiteList>
```



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

70

User Defaults: Paths

Client
• Configuration

UNICORE: User Defaults

General Paths System Settings User Interface Logging Settings

Plugin directory: /home/kdoertel/Unicore/unicore-4.0/plugin

Job directory: /home/kdoertel/Unicore/jobs

Resource directory: /home/kdoertel/Unicore/resources

Import directory: /home/kdoertel/Unicore/imports

Export directory: /home/kdoertel/Unicore/exports

Stdout/Stderr and Log directory: /home/kdoertel/Unicore/exports

Scratch directory: /home/kdoertel/tmp

Ok Cancel



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

71

Script Plugin Defaults

Client
• Configuration

UNICORE: Script Plugin Defaults

Script Directory: /home/kdoertel/Unicore/scripts

Script Type: C shell Line wrap

Editor Font: dialog Size: 12

Ok Cancel

Enter default values for UNICORE script plugin



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

72

Job Preparation

Client • job preparation

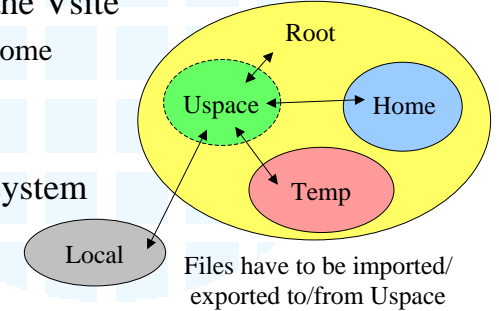
- UNICORE jobs are prepared in abstract seamless form
- Job contains
 - Sub-jobs and tasks
 - Resource requests
 - Dependency information
 - Without dependencies all tasks of a job may be executed in “parallel”



Unicore File Spaces

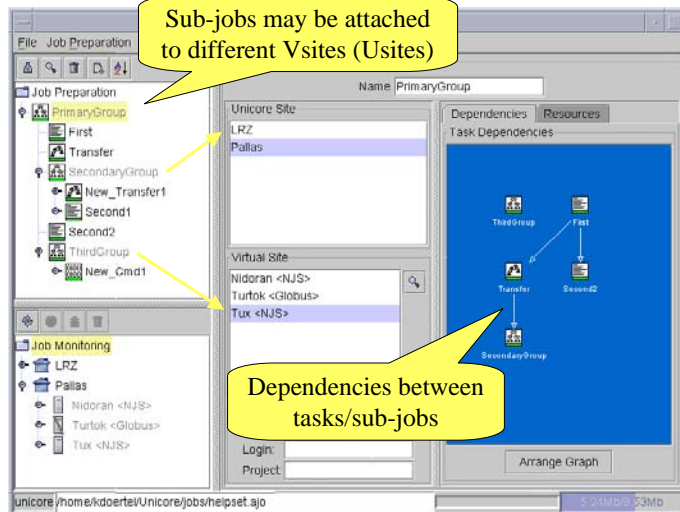
Client • job preparation

- Temporary at the Vsite
 - Job Directory (Uspace) for execution of a (sub-)job
- Permanent at the Vsite
 - Home: user home
 - Root
 - Temp
- At the Client system
 - Local



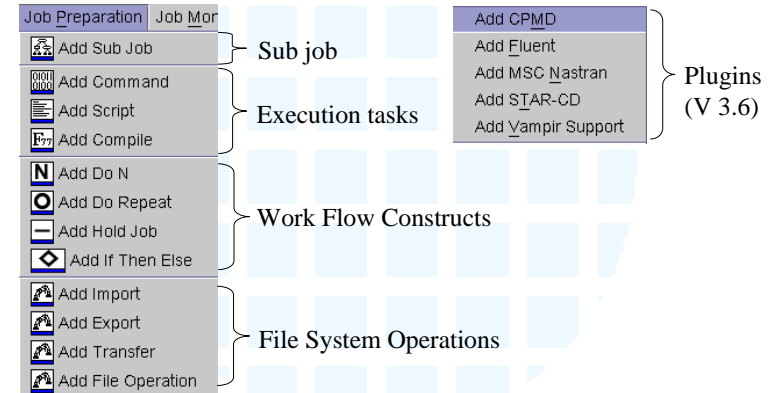
Job Structure

Client • job preparation



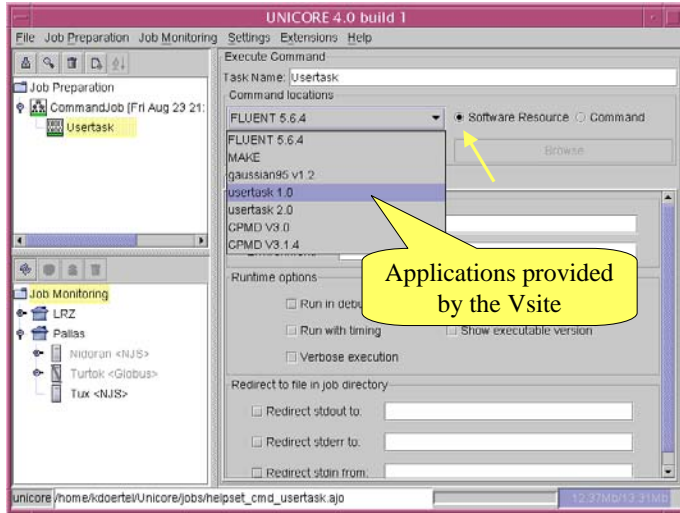
Job Preparation Menu

Client • job preparation



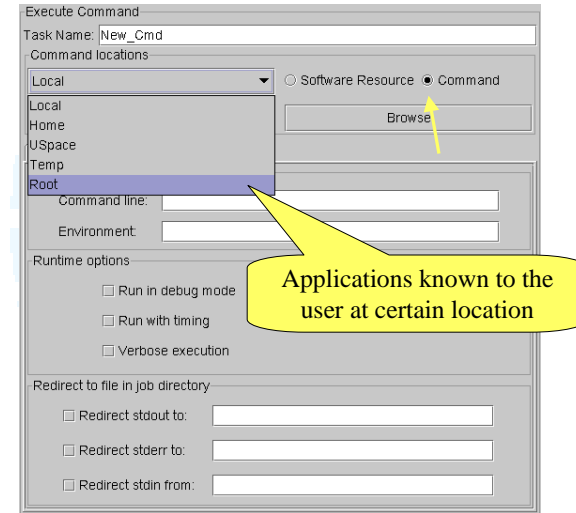
Command: Software Resource

Client
• job preparation



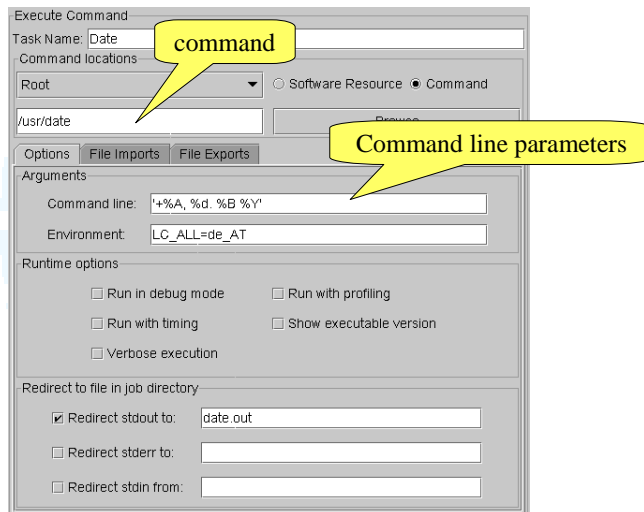
Command Location

Client
• job preparation



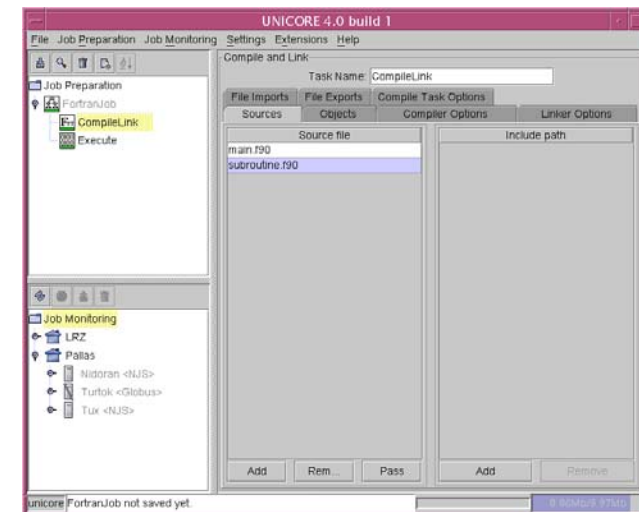
Command Options

Client
• job preparation



Compile & Link Task

Client
• job preparation



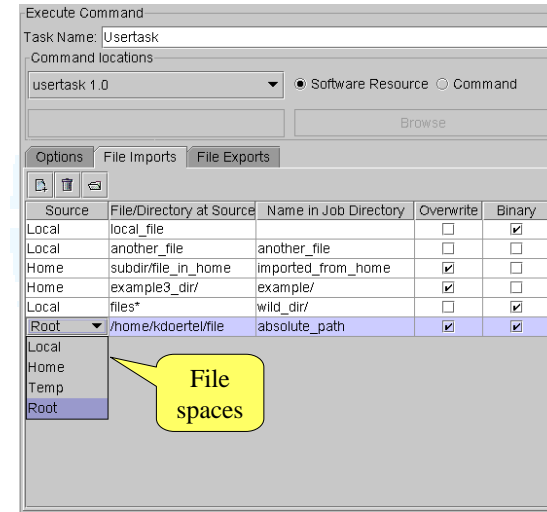
File System Operations

- Import/exports to/from job directory
 - Attached to execution tasks: dependencies are set automatically
 - Independent: specify dependencies
- Transfer of files between sub-jobs
 - Jobs are executed in distinct job directories
- Other operations:
 - copy, rename, mkdir, delete, chmod

Client
• job preparation



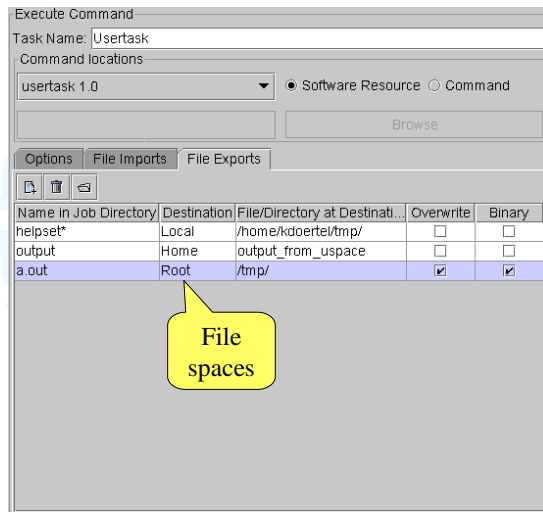
Command Import



Client
• job preparation



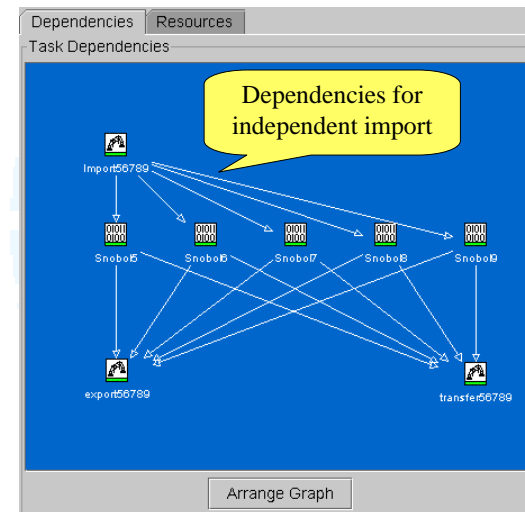
Command Export



Client
• job preparation



Explicit Import & Export Task

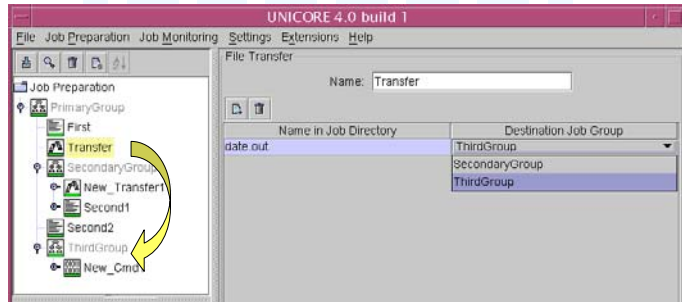


Client
• job preparation



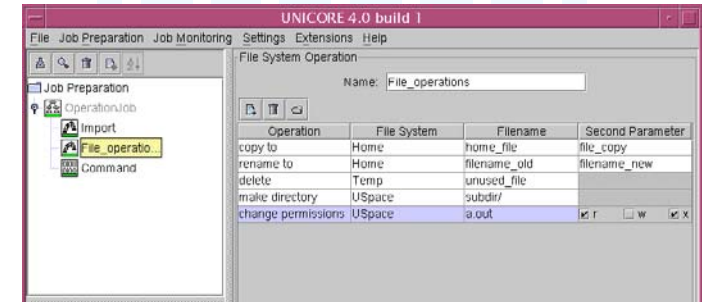
Transfer Task

Client
• job preparation



File Operations

Client
• job preparation



Resource Editor

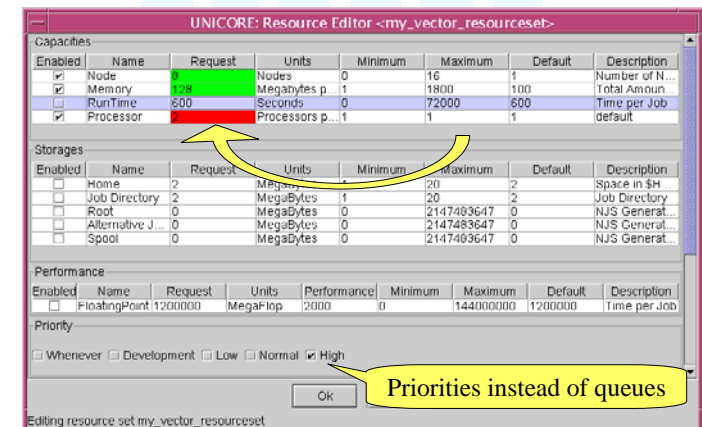
Client
• job preparation

- Used to specify resource requests
- Resource Editor combines resource information from Vsite with user requests
 - Check for correctness: minimum and maximum values not exceeded?
- Client in offline modus: Vsites' resource information taken from resource cache file.
- Resource requests are attached to tasks
- Store/load requests (as templates) to disk
 - e.g. typical parallel requests (e.g. context MPI)
 - e.g. typical vector processor requests



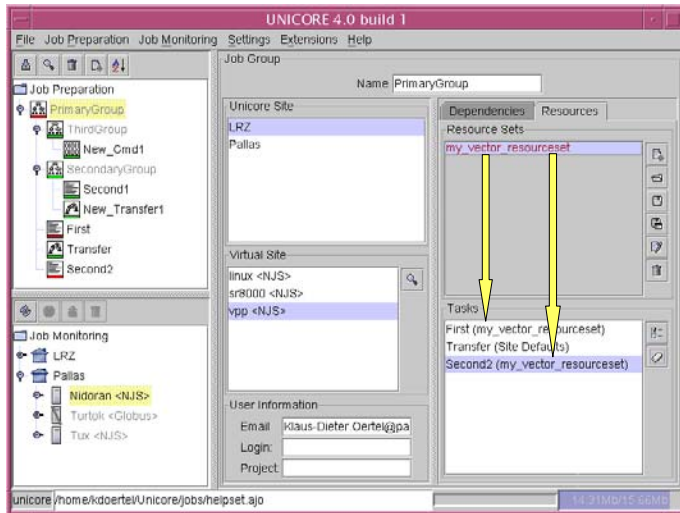
Resource Requests

Client
• job preparation



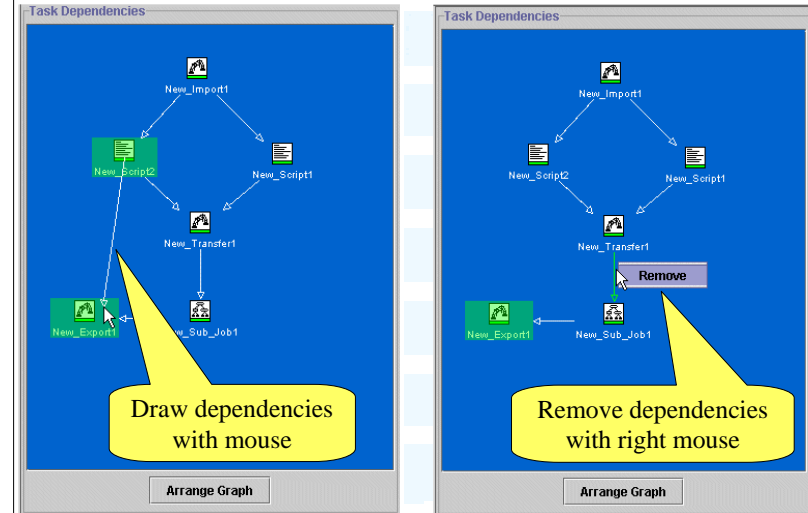
Resources Attached to Tasks

Client
• job preparation



Dependency Editor

Client
• job preparation



Remote File Access

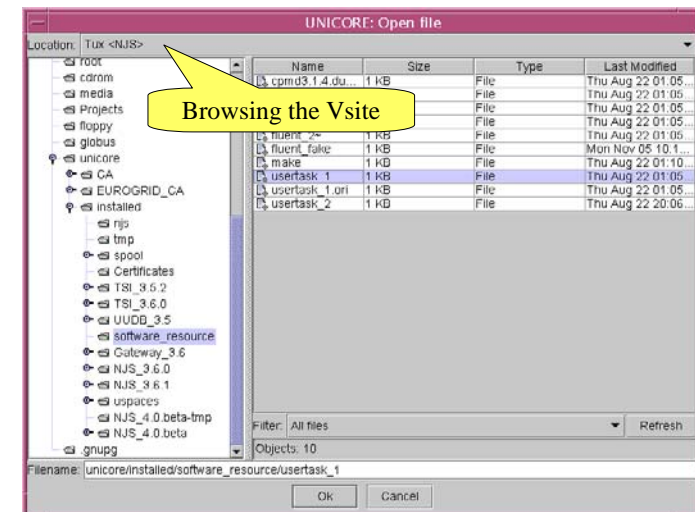
Client
• job preparation

- Use Remote File Chooser to select files/directories for import/export etc.
- Use Remote Text Editor to open remote script in editor, and to save modified file back



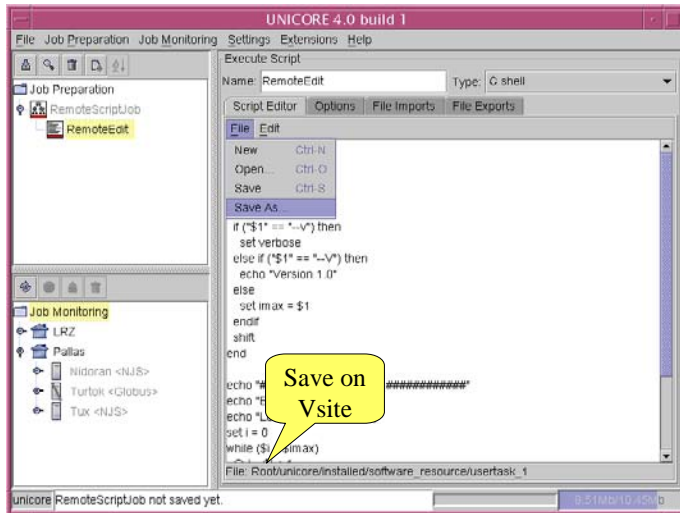
Remote File Chooser

Client
• job preparation



Remote Script Edit

Client
• job preparation



Work Flow Constructs

Client
• job preparation

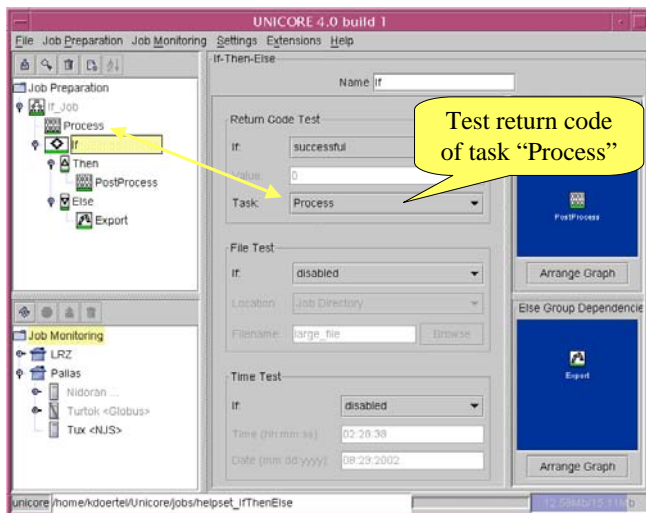
New in Client 4.0:

- If-Then-Else task checks for:
 - Status or return code of predecessor
 - File existence or file permissions
 - Execution time stamp
- Do-N (do loop) task
- Do Repeat Until: analog to if-then-else checks
- Hold task: check for expiration of time



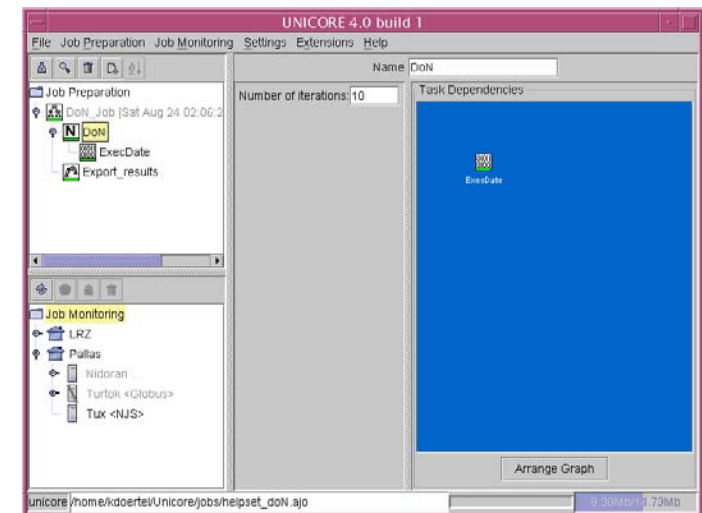
If-Then-Else Task

Client
• job preparation



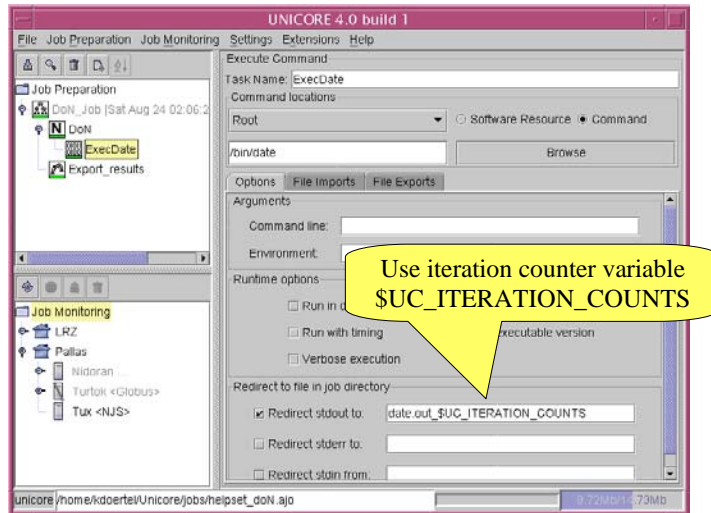
Do-N

Client
• job preparation



Do-N Body

Client
• job preparation

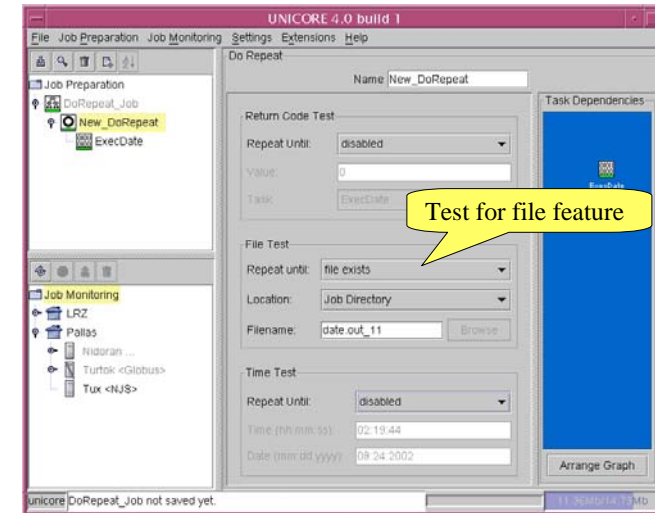


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

97

Do Repeat

Client
• job preparation

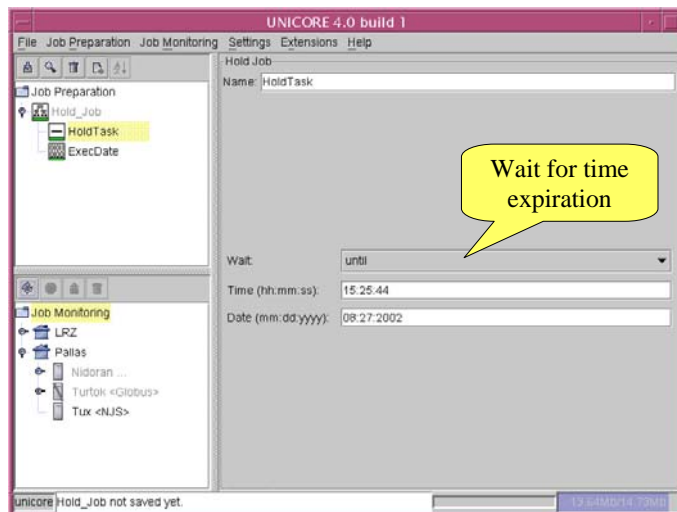


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

98

Hold Task

Client
• job preparation



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

99

Job Monitoring

Client
• job monitoring

- Display job status
- Retrieve job output
 - Standard output/error
 - Files exported to Local client system
- Trouble shooting: provide job details and Vsite log
- Abort and remove jobs
- Hold and resume jobs if supported by batch sub-system

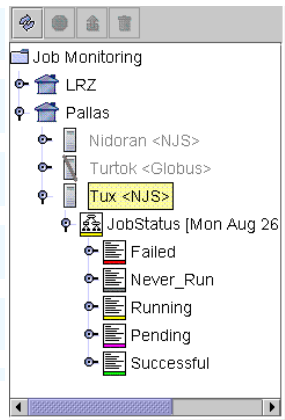


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

100

Colour codes

Client • job monitoring

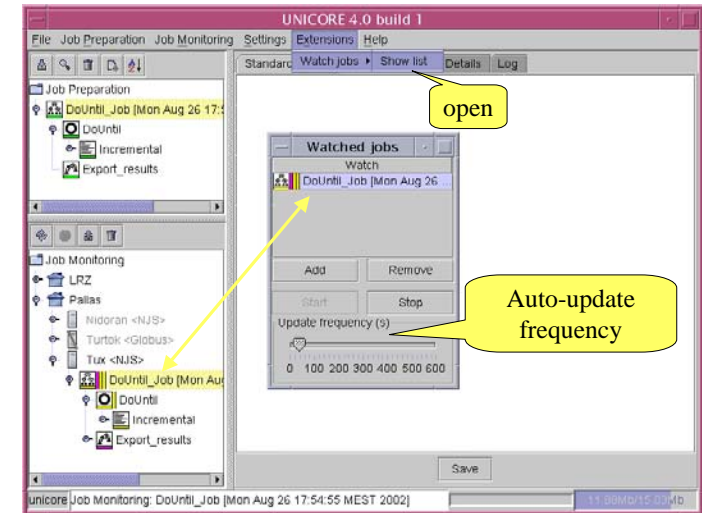


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

101

Watching Job Progress

Client • job monitoring

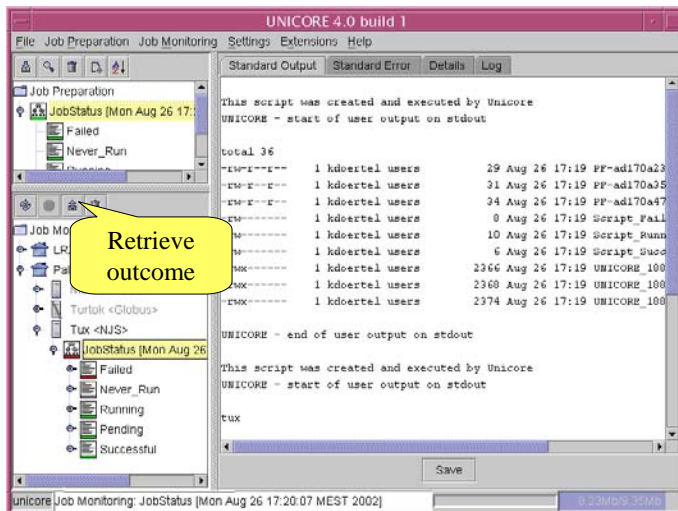


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

102

Standard Output/Error

Client • job monitoring

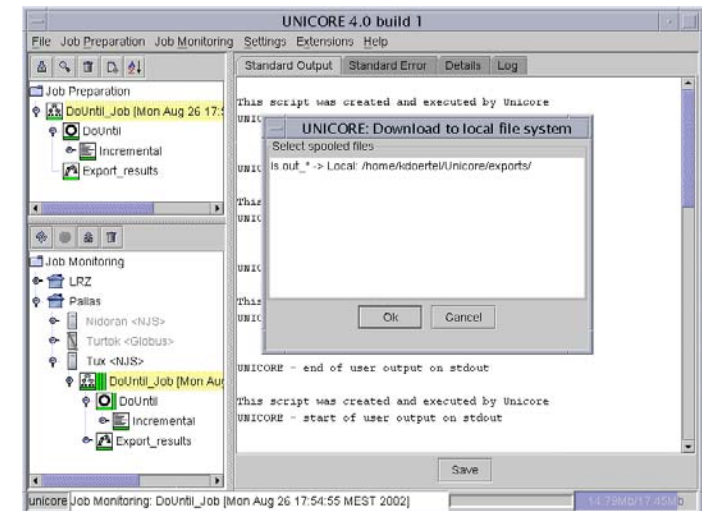


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

103

Export to Local

Client • job monitoring



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

104

Job Details

```

Standard Output Standard Error Details Log
<Unkown Time>: Job JobStatus [Mon Aug 26 17:20:07 MEST 2002] execut
ing at Tux <NJS>/Pallas

##
<Unkown Time>: Task Failed not_successful at Tux <NJS>/Pallas
Execution: not_successful
BSS ID: 1886900

##
<Unkown Time>: Task Never_Run never_run at Tux <NJS>/Pallas
Execution: never_run

##
<Unkown Time>: Task Running executing at Tux <NJS>/Pallas
Execution: successful

##
<Unkown Time>: Task Pending pending at Tux <NJS>/Pallas
Execution: executing
BSS ID: 1888000

##
<Unkown Time>: Task Successful successful at Tux <NJS>/Pallas
Execution: successful
    
```

Client
• job monitoring



Expert Job Log

```

Standard Output Standard Error Details Log
>>> ScriptOptions (ExecuteScriptTask_Outcome@c6d5a690)
>>> 21:51:28 29/04 C Arrived at ELV1536two
>>> 21:51:28 29/04 T Known to NJS
>>> 21:51:30 29/04 S Status is now READY
>>> 21:51:30 29/04 T Starting execution
>>> 21:51:30 29/04 T Selecting a default incarnation <execution_ts>
>>> 21:51:30 29/04 T Syntax checks passed
>>> 21:51:30 29/04 T Syntax and external checks passed
>>> 21:51:30 29/04 C EXECUTED COMMAND:ARCH=SunOS; export ARCH
>>> 21:51:30 29/04 C LD_LIBRARY_PATH=/opt/SUNWspr/lib; export LD_LIBRARY_PATH
>>> 21:51:30 29/04 C # Incarnation of ExecuteTask, UserTask or ExecuteScriptTask
>>> 21:51:30 29/04 C SHELL=/usr/bin/sh; export SHELL; UC_VERBOSE=-v; $$SHELL $USP
>>> 21:51:30 29/04 C Incarnated Resources: nodes <1> processors <1> memory <10.0>
>>> 21:51:30 29/04 C Incarnated Resources: Time <20.0> large <1.0> fast <0.0>
>>> 21:51:30 29/04 C Incarnated Resources: home large <0.0> home fast <10.0>
>>> 21:51:30 29/04 C Incarnated Resources: priority <whenever>
>>> 21:51:30 29/04 C
>>> 21:51:30 29/04 C
>>> 21:51:30 29/04 C Selected queue: standard
>>> 21:51:31 29/04 S Incarnation started on BSS with identifier <2266401>
>>> 21:51:31 29/04 S Status is now EXECUTING
>>> 21:51:31 29/04 S Status is now RUNNING
>>> 21:51:39 29/04 S Status is now SUCCESSFUL. Message: Script reported no errors
    
```

Client
• job monitoring



Application Specific Interfaces (Plugins)

Applications under UNICORE

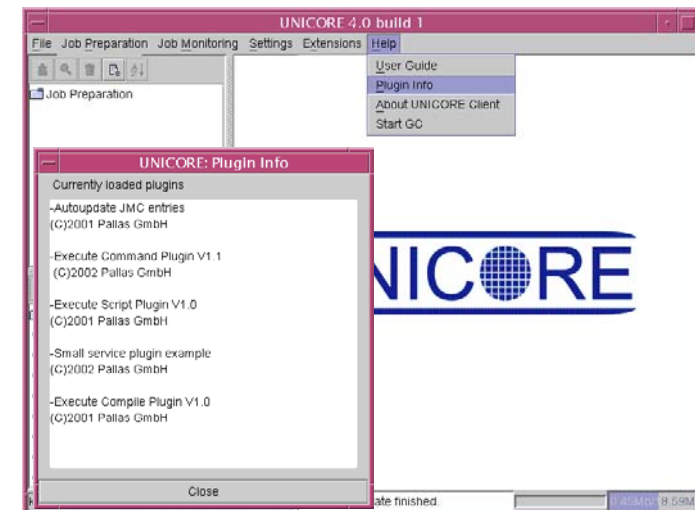
- Very seamful: execute from scripts
- Seamful: specify path to application in command task
- Seamless: select application as software resource in command task
- Optimum: provide application specific interface, a plugin



Client
• plugins



Plugin Info

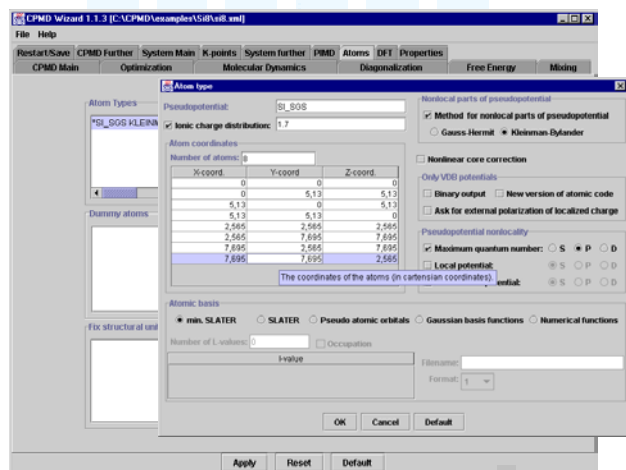


Client
• plugins



CPMD Plugin + Wizzard

Client
• plugins

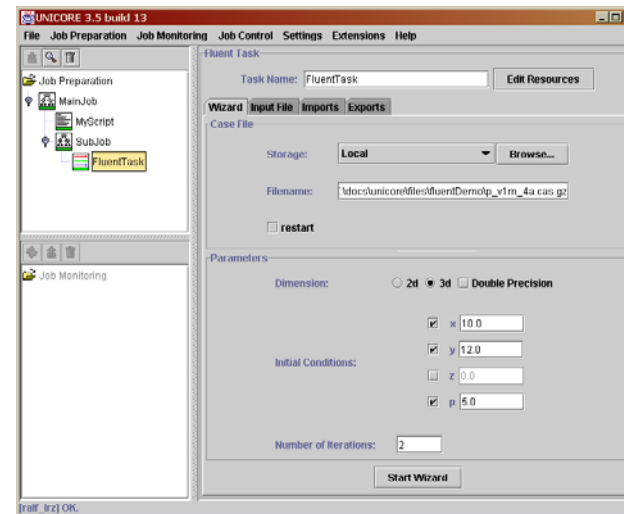


Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

109

Fluent Plugin (Client 3.6)

Client
• plugins



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

110

MSC.Nastran Plugin (V 3.6)

Client
• plugins



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

111

Gaussian 98 Plugin (V 3.6)

Client
• plugins



Klaus-Dieter Oertel, Pallas GmbH
Mathilde Romberg, Forschungszentrum Jülich

112

Plugin Development

Client
• plugins

- Implementation of at least three Java classes (extensions of abstract classes)
 - Start/Stop of plugin
 - Data container
 - User interface
- Plugins use the infrastructure of the core Client
 - Import/export factories
 - Resource management
- Because of the Java security manager (.java.policy file) external plugins have restricted access rights
- Example implementations provided by Command Task and other “system” plugins



The UNICORE Grid System

The UNICORE Grid System

Tutorial

Part 3
Server



Outline

server

3. Server
 - Overview (Packages, Components, Prerequisites)
 - Gateway (Installation, Configuration, Maintenance)
 - Network Job Supervisor
 - Installation, basic configuration
 - Incarnation Data Base
 - Maintenance
 - UNICORE User Data Base
 - Target System Interface (Installation, Configuration)
4. Discussion



Server packages

server
• Overview

- Version 3.6
- Server bundle contains
(from <http://www.unicore.org/downloads.htm> - Packages bundle)
 - gateway.tar (Gateway)
 - njs.tar (Network Job Supervisor)
 - UUDB.tar (UNICORE User Data Base)
 - tsi.tar (Target System Interface)
 - README_SERVERS

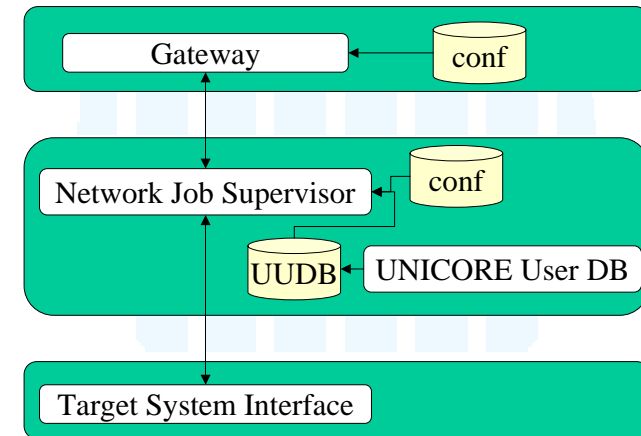


Server Documentation

contained in the packages

- Gateway
 - docs/using.pdf
 - README_GATEWAY
- NJS
 - docs/Using_3.6.2.pdf
 - docs/IDB_3.6.1.pdf
 - README_INSTALL
 - README *for UUDB*
- TSI
 - docs/TSI_api.pdf
 - README

Server Components



Server Prerequisites

- Gateway and NJS:
 - Java 1.3.x
 - Sun JSSE (Java Secure Socket Extension)
 - X.509 certificates for Gateway and NJS
 - Signer certificate(s)
- TSI:
 - Perl \geq 5.004

Gateway

- Entry point of a UNICORE Site
- Accepts SSL connections from Clients and NJSs
- Accepts valid certificates from all signers known to it (authentication)
- Talks UNICORE Protocol Layer (UPL) on connections to the outside world
- Sends/receives AJOs to/from the local NJSs

Installation

Server
• Gateway - Installation

- tar -xvf gateway.tar
 - README_GATEWAY
 - gateway/bin/change_log_level
 - gateway/bin/invalidate_crls
 - gateway/bin/list_log_files
 - gateway/bin/start_gateway
 - gateway/bin/start_gateway_p
 - gateway/bin/stop_gateway
 - gateway/bin/change_log_file
 - gateway/conf/logs/
 - gateway/conf/gateway.properties
 - gateway/conf/connections
 - gateway/docs/using.pdf
 - gateway/lib/gateway.jar symbolic link to Classes_gateway_3.6.1.jar
 - gateway/lib/Classes_ajo_3.6.1-build-2.jar
 - gateway/lib/ajo.jar symbolic link to Classes_ajo_3.6.1-build-2.jar
 - gateway/lib/Classes_gateway_3.6.1.jar



gateway.properties

Server
• Gateway - Configuration

Major configuration data set, contains

- gw.gateway_host_name= *fully qualified hostname*
- gw.port = *port to listen to for client connections*
- gw.identity= *path to gateway's private X.509 key*
- gw.password= *password for gw.identity file (optional)*
- gw.trusted_cas= *list of signer certificates separated by :*
- gw.connections = *connections data set with info on NJSs*
- gw.change_log_files=daily | hourly | *n*
- gw.conn_timeout = *timeout for idle connections (minutes)*



connections

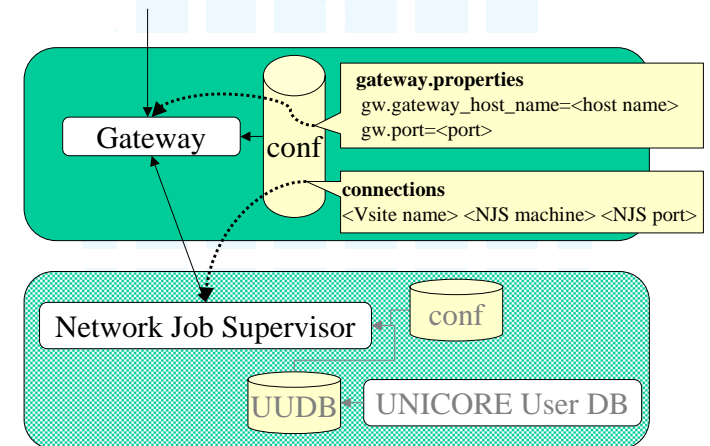
Server
• Gateway - Configuration

- Data set which contains information for the connected local NJS server(s)
- Is expected to be in the **conf** directory
- Format
 <Vsite name> <NJS machine> <NJS port>
- Example
 myVsite system.domain.tld 5555



Gateway connections

Server
• Gateway - Configuration



Preparations

- Check **java path** (1.3.x) in gateway/bin/start_gateway
- **lib** contains ajo.jar, gateway.jar, jsse.jar, jnet.jar, and jcert.jar
- Gateway port is opened for SSL connections to your gateway machine on your **firewall**
- Gateway machine should be used for running the Gateway only (no logins for general users etc.)

Maintenance

- **start_gateway** [<conf_dir>] *starts the Gateway*
- **start_gateway_p** [<conf_dir>] *starts the Gateway and prompts for the gateways password*
- **stop_gateway** [<conf_dir>] *checks for LAST_PID and stops that process*
- **list_log_files** type [<conf_dir>] *list log file names and pathes*
- **change_log_level** {S|W|I|C|T|D} [<conf_dir>]
- **change_log_file** [<conf_dir>] *starts new log file*

Network Job Supervisor

- UNICORE scheduler
- Receives/sends AJOs from/to local Gateway
- Translates AJO into batch job for target
- Maps the user's Ulogin to Xlogin
- Sends sub-AJOs to corresponding Gateway according to dependencies
- Polls for status and output of sub-AJOs
- Sends batch jobs and requests to TSI
- Polls TSI for job status and output

NJS Installation

- **tar -xvf njs.tar**
README_INSTALL
njs/bin/start_njs
njs/bin/list_log_files
njs/bin/njs_admin
njs/conf/example_idb
njs/conf/njs.properties
njs/docs/NJS_interface_doc_3.6.2build3.tar
njs/docs/IDB_3.6.1.pdf
njs/docs/Using_3.6.2.pdf
njs/docs/Changes_3.6.2.pdf
njs/lib/Classes_ajo_3.6.1-build-2.jar
njs/lib/ajo.jar symbolic link to Classes_ajo_3.6.1-build-2.jar
njs/lib/Classes_njs_3.6.2build3.jar
njs/lib/njs.jar symbolic link to Classes_njs_3.6.2build3.jar

njs.properties

Configuration data set, contains

- Specifications for connection to Gateway
 - **njs.vsite_name**= the name of the Vsite, same as in Gateway's connections file
 - **njs.gateway_port**= Port number the NJS listens on for Gateway connection (same as in GW's connections file)
 - **njs.gateway**= fully qualified host name of the gateway system
 - **njs.gateway_ssl**=true | false protocol for GW - NJS connection
 - **njs.use_ssl**=true | false protocol for connections to other Vsites

Server
• NJS - Configuration



njs.properties (cont.)

- Security parameters
 - **njs.njs_cert_loc**= path to NJS's private X.509 key(s)
 - **njs.ssl_password**= password for njs.njs_cert_loc file (optional)
 - **njs.unicore_ca_loc**= list of accepted signers of gateway certificates separated by :
- Specification for Incarnation
 - **njs.incarnationdb**= path to Incarnation Data Base file
- Specifications for UUDB
 - **uudb.directory**= where to find the User Data Base
 - **uudb.class_name**= uudb implementation java class

Server
• NJS - Configuration



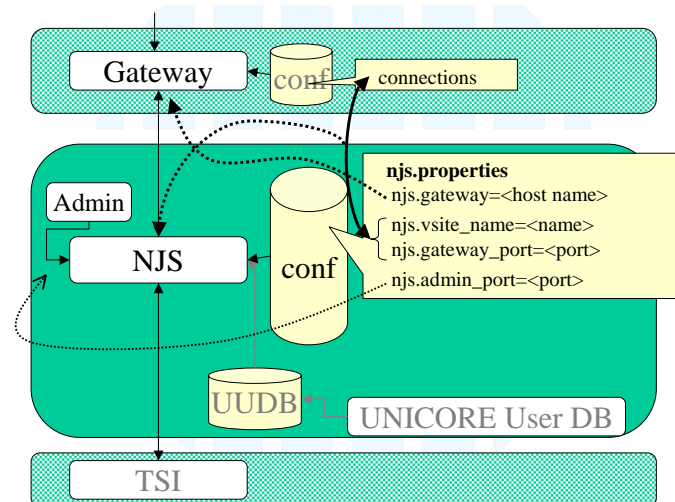
njs.properties (cont.)

- Logging options
 - **njs.logging_level**=S | W | I | C | T | D Severe - Warning - Information - Configuration - Talk - Debug
 - **njs.log_file_change_interval**=daily | hourly | n
- Administrator access
 - **njs.admin_port**= port number NJS listens to for administrator requests
- ...

Server
• NJS - Configuration



NJS connections



Server
• NJS - Configuration



Incarnation Data Base

Information service and translation table, contains definitions for

- GENERAL properties (file spaces, descriptions, ...)
- EXECUTION_TSI (host + ports, resources, batch queues, ...)
- STORAGE_TSI (for file transfers and management)
- RUN (translation rules for target)
- IMPORT, EXPORT, CLEANUP, LIST_DIRECTORY, RENAME, COPY_FILE, DELETE_FILE, CHANGE_PERMISSIONS
- FORTRAN, LINK



Examples

- GENERAL section
 USPACE_ROOT *path for job's temporary working dir*
 TextInfoResource *site defined information to be given to the user*
- CLEANUP section
 INVOCATION [cd ..; RM_CMD -rf <DIR>]
commands to be executed at the very end of a UNICORE job
- COPY_FILE section
 INVOCATION OVERWRITE [COPY_CMD -
 pf <SOURCE> <DESTINATION>] *commands to be executed for copying files with overwrite allowed*



EXECUTION_TSI Section

NAME <name> *identifier for this TSI*
 SOURCE <machine> <port1> <port2> *host address and port numbers for TSI*
 SOFTWARE_RESOURCE <application> <version>
 NODE [<comment>] [<default>] [<max>] [<min>]
 PROCESSOR ...
 CPUTIME ...
 MEMORY ...
 HOME [<comment>] [<default>] [<max>] [<min>]
 [<subspace>]
 ...



EXECUTION_TSI - Example

NAME my_tsi
 SOURCE target.domain.tld 6666 7777
 SOFTWARE_RESOURCE CPMD V3.0h
 NODE [Number of Nodes] DEFAULT [32] MAXIMUM [256] MINIMUM [1]
 PROCESSOR [Number of PEs per Node] DEFAULT ...
 CPUTIME [CPU Time per Job] DEFAULT [600]
 MAXIMUM [14400] MINIMUM [10] RATE [600]
 MEMORY [Amount of Memory per Node] DEFAULT ...
 HOME [\$HOME] DEFAULT [10] MAXIMUM [100]
 MINIMUM [1] SUBSPACE [HOME]



RUN Section

Translation of application into executable commands:

```
INVOCATION <application><version>[<run cmd's>]
INVOCATION PERL      [ SHELL=PERL_CMD;
  $SHELL <RUNCMD> <VERSION> <VERBOSE> ]
INVOCATION DEBUG [ touch IDB_DEBUG;
  <STANDARD> ]
...
#DEFINE TSI_LS /path_on_target/tsi_ls
```



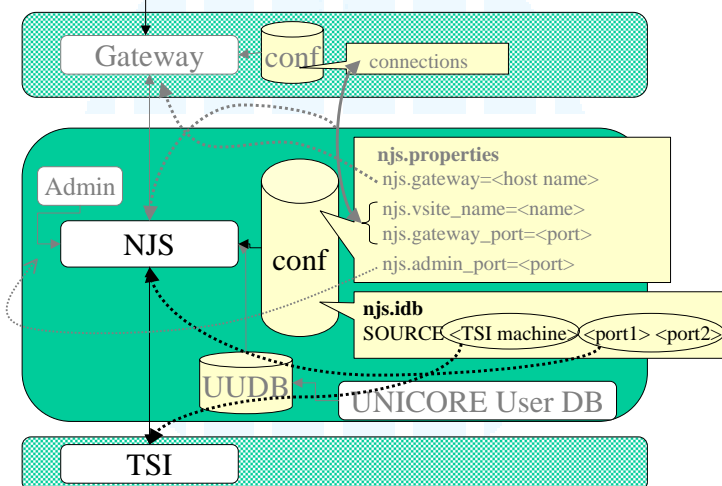
RUN - Example

Software resource:

```
INVOCATION CPMD-V3.0h [
  /dir/mpirun -np $UC_NODES /cpmd_dir/bin/cpmd30h.x
  $CPMD_FILE $PP_LIBRARY ]
INVOCATION VampirTrace-2.0 [
  export PAL_LICENSEFILE=/usr/local/vt/etc/license.dat;
  export PAL_ROOT=/usr/local/vt;
  /bin/mpprun -n $UC_NODES $EXEC_BIN
  $VT_CONFIG ]
```



NJS connections



Preparations

Check that

- njs/bin/start_njs has the correct **java path** (Java 1.3.x)
- **lib** contains ajo.jar, njs.jar, jsse.jar, jnet.jar, and jcert.jar
- NJS machine should be used for running the NJS servers only (no logins for general users etc.)



Startup / Shutdown

- `start_njs <conf_dir>`
starts NJS using the configuration given; creates LAST_PID in <conf_dir>; starts new log file in <conf_dir>/logs
- `njs_admin [-m <njs_machine>] [-p <port>] stop`
connects to NJS on given port; saves the current status to njs.save_dir; stops NJS process



njs_admin

- Perl script implementing administrator interface
- Connects to NJS on specified port
- Connection specification in script:

```
$njs_machine = "<host name>";
$njs_port = "<port>";
```

Can be overwritten by command line options -m, -p
- Requires admin userid to have write access to NJS's configuration directory
- Session and command mode



list

- `list [short|detailed|long] [<selection>]`
 – `<selection>`

```
ajos    all AJOs corresponding to batch jobs
all     all AJOs
ajo_id  specified AJO
```

`<expression>` *terms using & and | and () with type, status, user, ulogin, bssid, and rootajo selections*



list – example 1

- `njs_admin list detailed 885c2908`
- New_Job1 (885c2908 in 885c2908) AJO 10:23:42 25/06
SUCCESSFUL zdv038 () Ulogin: ...

 ACTION NAME: New_Job1
 OUTCOME DIR: /work/UNICORE/outcome_885c2908/
 LAST MESSAGE: Status is now SUCCESSFUL
 ENDORSER: EmailAddress=m.romberg@fz-juelich.de, CN=...
 CONSIGNOR: EmailAddress=m.romberg@fz-juelich.de, CN=...
 USPACE DIR: /work/UNICORE/uspace_885c2908/



list – example 2

```
njs_admin 'list detailed type USER & user zdv038'
```

```
New_Gaussi (885c290e in 885c2908) USER 10:23:41 25/06  
SUCCESSFUL zdv038 () Ulogin: EmailAddress=m.romberg@fz-  
juelich.de, CN= .... 66165
```

```
ACTION NAME: New_Gaussian1
```

```
ROOT AJO: New_Job1
```

```
OUTCOME DIR
```

```
/work/UNICORE/outcome_885c2908/AA885c2910/AA885c290f/AA885c290e/
```

```
LAST MESSAGE: Status is now SUCCESSFUL. Message: Script reported no  
errors
```

```
BSSID: 66165
```

```
EXECUTED COMMAND: ....
```



abort, cancel, hold, resume

- **abort <selection>**
aborts the execution of the selected actions
- **cancel <selection>**
aborts the selected actions and removes any output
- **hold <selection>**
holds the execution of the selected actions
- **resume <selection>**
resumes the execution of previously hold actions



ls, remove

```
ls [outcomes | uspaces] [<Xlogin>]
```

```
remove [outcome | uspace] <ajo_id><Xlogin>
```

```
njs_admin 'ls uspaces zdv038'
```

```
drwx----- 2 zdv038 root 4096 Aug 5 14:47
```

```
uspace_7d86d3a1 New_Job1 (7d86d3a1 in 7d86d3a1)
```

```
AJO 14:47:51 05/08 EXECUTING zdv038 ()
```

```
Ulogin: ...
```



logging

```
logging [new_file | level | interval | info]
```

- *Close current log file and open new one*
- *Set new logging level:*
logging level [S | W | I | C | T | D]
- *Set new interval for changing the log file*
logging interval [daily | hourly | *n*]
- *Gives current logging status (level, interval, log-file)*



tsi

Gives status of connected TSI

To be enhanced with UNICORE 4.0:

tsi [up | down | status | stop | refresh]

list_log_files

- list_log_files <type> [<conf_dir>]

<type>: a all files
l all since latest start
L all before latest start
n latest n files
-n all except latest n

UNICORE User Data Base

- Management of Ulogin – Xlogin mapping information
- NJS accesses this information
- Basic version allows to map one certificate to exactly one Xlogin
- NJS to UUDB interface defined to adapt to site specific user data bases (i.e. ldap)
- <http://www.unicore.org/downloads.htm> → contributions offers an alternative uudb with certificate-projectid pairs being mapped to Xlogins

UUDB Installation

1. tar -xvf UUDB.tar
db_release.tar
installer
README
2. installer <path to uudb dir>
[<path to lib dir with jsse.jar>]
unpacks db_release.tar into the specified UUDB directory
3. bin/install <path to uudb dir>
<path to lib dir with jsse.jar>
generates the UUDB commands from template files

Configuration (cont.)

Server
• TSI - configuration

```
$main::site_sez_no_holds = 1;
$main::hold_cmd = "$nqs_dir/qhold";
$main::resume_cmd = "$nqs_dir/qrls";

$main::qstat_cmd = "$nqs_dir/qstat -a";

$ENV{SHELL}="/bin/sh";
# $ENV{PATH}=":/bin:/usr/bin:/usr/ucb/"; # if this is set
# then only commands from these directories are found
# Add any other site required environment variables here

$main::default_job_name = "UnicoreJob";
# END OF CONFIGURATION <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```



Example: submit.pm

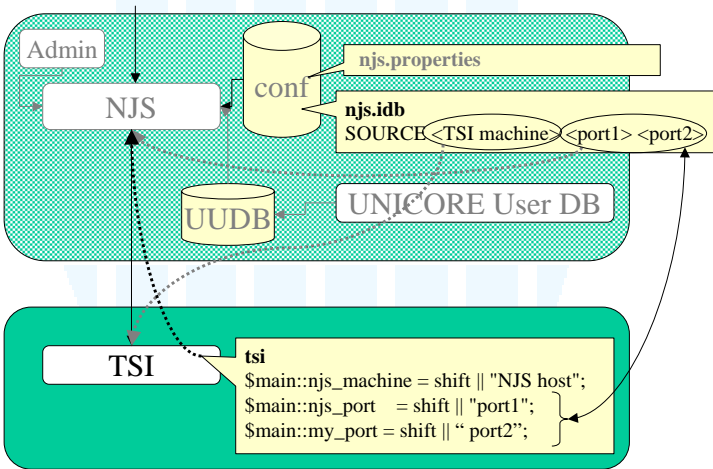
Server
• TSI - configuration

```
$jobname = $2 if $1 eq "JOBNAME";
$outcome_dir = $2 if $1 eq "OUTCOME_DIR";
$uspace_dir = $2 if $1 eq "USPACE_DIR";
$time = $2 if $1 eq "TIME";
$memory = $2 if $1 eq "MEMORY";
$nodes = $2 if $1 eq "NODES";
...
$memory = "-lM $memory"."Mb";
my $command = "$main::submit_cmd $queue $nodes
              $email $memory $time $jobname $stdout_loc
              $stderr_loc $$submit::tsi_unique_file_name";
```



TSI connections

Server
• TSI - configuration



Preparations

Server
• TSI - Maintenance

- Use Perl 5.004 or higher
- Make tsi_ls world readable
- Give Uspace root directory mode 1777
- For testing: TSI may be run unprivileged
 ⇒ commands will be executed as the user who started the TSI



Start / Stop

Server
• TSI - Maintenance

- **User: root**
- **tsi** [<njs_machine> <port1> <port2>]
*starts the TSI daemon which connects to port2 at NJS;
TSI worker processes use port1
(default: take NJS information from tsi configuration)*
- **kill** <pids>
*kill all tsi processes
(in version 4.0 this may be done through njs_admin)*

Server connections

Server
• communication

