



CONTRACT NUMBER 031513

eDEISA
**EXTENDED DISTRIBUTED EUROPEAN
INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

First Outreach Activity Report

Deliverable ID: eDEISA-D-eN2-1

Due date: May, 31st, 2007

Actual delivery date: May, 25th, 2007

Lead contractor for this deliverable: IDRIS-CNRS, France

Project start date: June 1st, 2006

Duration: 2 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Content

Table of Content.....	1
1. Executive Summary.....	2
2. Objectives and organization	3
3. First three DEISA training sessions	5
3.1 The first DEISA Training session.....	5
3.2 The second DEISA training session	6
3.3 The third DEISA training session.....	8
4. Results of the feedback questionnaires from the three training session events organized during the 1 st project year	9
5. Realisation and dissemination of the training sessions	11
6. Annex 1: Lists of attendees	13
7. Annex 2: Description of the UNICORE exercises done during the Hands-on sessions	17
7.1 Prerequisites.....	17
7.2 Prepare and configure the Client.....	17
7.2.1 Install the Client.....	17
7.2.2 Import private certificate keystore (.pkcs12).....	18
7.2.3 Make DEISA Gateways known to the client	19
7.2.4 Configure your Home site as UNICORE Site	20
7.2.5 Refresh the actual settings	20
7.3 Use the Client	20
7.3.1 Run a simple script.....	20
7.3.2 Hold, resume, abort a job	22
7.3.3 File import.....	22
7.3.4 If-then-else job.....	23
7.3.5 File export.....	24
7.3.6 DEISA GPFS and UNICORE	24
7.4 Access the DCPE with UNICORE	26
7.5 Defining resources with UNICORE.....	27
7.6 Hints for further usage.....	27
7.6.1 What site resources are available?	27
8. Annex 3: Description of the DESHL exercises done during the Hands-on sessions	28
8.1 Prerequisites.....	28
8.2 Installation	28
8.2.1 Download the installer	28
8.2.2 Install and configure the DESHL via the installer GUI	29
8.2.3 Verify the settings	33
8.2.4 Help System	33
8.3 Using the DESHL client.....	34
8.3.1 Upload, submit and manage a simple job	34
8.3.2 Staging in data for a job, staging out results	37
9. References and Applicable Documents.....	40
10. List of Acronyms and Abbreviations	41

1. Executive Summary

This document describes the objectives of the DEISA Training Sessions and the organisation of the three first sessions, with a description of the content of each of them. A detailed analysis of the feedback questionnaires filled by the participants of each session is also included, as well as a description of the dissemination activities around the training sessions.

In annexes, the whole lists of the participants to each of the first three sessions are given. And the description of the exercises proposed to the attendees is included, both for the UNICORE and DESHL sections.

This document is publicly available.

2. Objectives and organization

The main objective of this service activity, as explained in the eDEISA *Description of work* Technical Annex, is *to enable fast development of user skills and know-how needed for the efficient utilisation of the DEISA infrastructure*. For this purpose, the DEISA contract planned the organization of six training sessions in various European countries between June 2006 and April 2008.

After initial discussions between the six partners involved in this activity (BSC, CINECA, CSC, FZJ, HLRS and IDRIS), the following schedule was defined at the beginning of 2006:

- Paris, July 2006
- Jülich, October 2006
- Barcelona, March 2007
- Helsinki, May 2007
- Bologna, October 2007
- Stuttgart, February 2008

These sessions are open to scientists from all European countries, and to participants from industrial organizations. The attendance for each session is limited to 30 participants.

This activity is driven by the DEISA SA4 “User Support” service activity (see the deliverables D-SA4-4 [2], D-SA4-4.2 [3] and D-SA4-5 [4]), with the support of the dissemination service activity. The User Support team is in charge of preparing the content of the training sessions and providing the speakers. The dissemination team is in charge of the promotion of the events and the preparation of the support material.

For each session, DEISA funded the travel and living expenses of up to 15 participants coming from another country than the one in which the event is organized. As a result of this approach the sessions were genuinely European events, with attendees from many countries.

According to both the feedback received after the first session as well as our wish to extend the opening of the infrastructure and allow more scientists to experiment with it, we decided, starting with the second training event, to add a special hands-on session. In this hands-on session the participants could experiment by themselves for a few hours with some of the major features of the hardware and software infrastructure, including especially the usage of the Global File Systems, UNICORE, DESHL and the DCPE. According to the feedback received from the participants to the first hands-on session, at the second session in Jülich, we decided to further reinforce this part during the next training sessions. The participants emphasized the importance and full interest of these hours of real practice. To give a detailed idea of the content of the hands-on sessions, we include the whole description of the exercises proposed to the attendees, both for the UNICORE [7] and DESHL [6] sections, as used during the third session in Barcelona in March 2007 in appendices.

Moreover, each participant to the training sessions becomes a *DEISA Test User* (DTU) and has a DTU account opened to be able to continue to experiment with the infrastructure for a period of three months after the session that he attended. These DTU accounts have only a small amount of resources usable. As decided by the *DEISA Executive Committee* in the summer of 2006, this new kind of access to the infrastructure has been opened to allow any European scientist to test specific

services like the global file systems and UNICORE, and to enable evaluation of the new opportunities that the DEISA infrastructure can offer to them, encouraging them to answer to the next DECI calls of proposals. In particular all the participants to the training sessions have a DTU account opened since the second session.

Each participant to a session receives a whole set of documents upon local registration on the first day of the session. These include the copy of all the presentations which will be given during the session, one exemplar of the current version of the DEISA Primer [5] and one DEISA brochure folder [1].

Each training session, with the exception of the first, which had no hands-on session, is divided in three parts:

- The first part gives a global description and introduction to the DEISA infrastructure and describes the general middleware services, the usage of the *DEISA Common Production Environment* (DCPE) and the detailed utilisation of UNICORE and DESHL.
- The second part consists of a *hands-on* session, which allows each user to experiment by himself the main services of the infrastructure (Global File Systems, UNICORE, DESHL, DCPE).
- The third part is dedicated to a different HPC topic each time. It is usually divided in two further parts, one technical and one based on *use cases*, to give pedagogical summaries of experiments in different scientific fields, mostly coming from achieved DECI projects.

When a session occurred in the supercomputing building of one partner, an optional visit of the computing room is also proposed to the participants during the session, to give the attendees more insight to the specific supercomputer, which is part of the DEISA infrastructure.

The specialized topics were:

- *Highly scalable parallel applications* for the first session in Paris,
- *Performance and portability* for the second session in Jülich,
- *Performance measurements and codes optimization* for the third one in Barcelona,
- *Heterogeneous environments and optimization* for the next planned one in Espoo.

During the reporting period, the fourth session was prepared, which will be organized by CSC and will occur in Espoo from May 30th to June 1st. The two remaining sessions are scheduled in October 2007 in Bologna (organized by CINECA) and in February 2008 in Stuttgart (organized by HLRS).

3. First three DEISA training sessions

Three DEISA training sessions have been organized from the eDEISA project budget during the 3rd project year: July 3-5, 2006 in Paris, France, October 23-25, 2006 in Jülich, Germany and March 7-9, 2007 in Barcelona, Spain. The preparations for the 4th DEISA training session to be held in Espoo, Finland in the end of May are proceeding currently.

In total, 68 scientists from 13 European countries (see figure 1 for details) have learned to exploit the European supercomputing environment at the three DEISA training sessions organized during the 1st eDEISA project year. In addition, 8 scientific speakers and 13 staff members have benefited from the event.

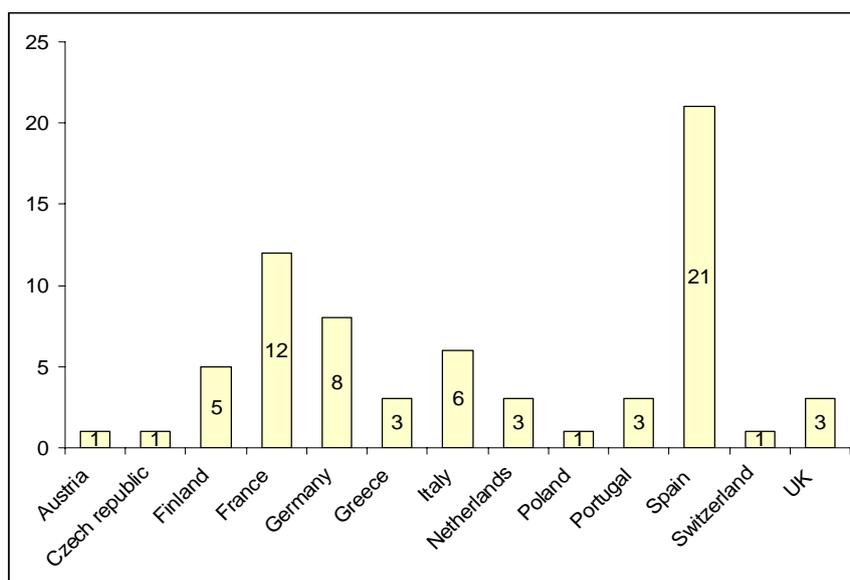


Figure 1: Distribution of the participants by country of the institutions to which the participants belong

3.1 The first DEISA Training session

The first DEISA training session was held on July 3-5, 2006 at the Concorde Saint-Lazare Hotel, Paris, France. In total, there were 28 participants, comprising 13 scientific users, 10 speakers, and 6 DEISA staff.



Figure 2: 1st DEISA training session

There were 10 lectures in this training session. These presentations can be found at http://www.deisa.org/news_events/deisa_events/paris_training.php :

- DEISA general presentations
 - [DEISA project and organization](#)¹, Victor Alessandrini, IDRIS, DEISA Project Director
 - [DEISA infrastructure and middleware](#), Denis Girou, IDRIS
 - [DEISA Common Production Environment](#), Denis Girou, IDRIS
 - [UNICORE \(UNiform Interface to COmputing RESources\)](#), Michael Rambadt, FZJ
 - [DESHL \(DEISA Services for the Heterogeneous management Layer\)](#), Malcolm Illingworth, EPCC
- Highly scalable parallel applications – Tools and techniques, Mark Bull and Gavin Pringle, EPCC
 - [introduction](#)
 - [performance analysis](#)
 - [communication optimisation](#)
 - [decomposition strategies and I/O optimisation](#)
 - [mixed programming models](#)
- Highly scalable parallel applications – Use cases
 - [The HORIZON Project: High Performance Computing and cosmology](#), Édouard Audit, Astrophysics Department, CEA/DAPNIA/SAp, Saclay, France
 - [Scaling ECMWF's IFS Forecast Model to large numbers of processors](#), George Mozdzynski, ECMWF, Reading, The Great Britain
 - [Channel 2000: A highly parallel algorithm for computing wall turbulence using 2K processors](#), Sergio Hoyas, School of Aeronautics, Madrid, Spain
 - [Domain Cloning – A new approach to parallelization of particle-in-cell \(PIC\) codes](#), Roman Hatzky, RZG, IPP & MPG, Garching, Germany

3.2 The second DEISA training session

The second DEISA training session was held at Forschungszentrum Jülich GmbH, in Jülich, Germany on October 23-25, 2006. In total, there were 36 participants, comprising 23 scientific users, 8 speakers, and 5 DEISA staff.

¹ Each presentation is available independently by clicking on each hyperlink.



Figure 3: 2nd DEISA training session

There were 9 lectures in this training session. These presentations can be found at http://www.deisa.org/news_events/deisa_events/juelich_training.php :

- DEISA general presentations
 - [DEISA Overview](#), Victor Alessandrini, IDRIS, and Achim Streit, FZJ
 - [DEISA Access, filesystems and job submission](#), Denis Girou, IDRIS
 - [DEISA Common Production Environment](#), Denis Girou, IDRIS
 - [UNICORE – Introduction](#), Michael Rambadt, FZJ
 - [Using UNICORE](#), Michael Rambadt, FZJ
 - [The DESHL Client](#), Malcolm Illingworth, EPCC
- Performance and portability: Tools and techniques for achieving optimal scaling on a cluster of share memory nodes, Gavin Pringle, EPCC
 - [Introduction](#)
 - [Communication](#)
 - [Decomposition](#)
 - [Mixed mode](#)
 - [Performance analysis](#)
- Performance and portability, use cases
 - [Scaling ECMWF's IFS Forecast Model to large numbers of processors](#), George Mozdzynski, ECMWF
 - [Coupling Applications and Visualization](#), Wolfgang Frings and Morris Riedel, FZJ

As a result of the experience and feedback from the 1st DEISA training, the *Access to DEISA* hands-on session was introduced with this training session. The participants had the opportunity to experience the DEISA infrastructure first-hand and they were granted a three-months-long test user account for DEISA (DTU account) in order to be able to continue experimenting with the DEISA infrastructure, once returned to their institutes.

3.3 The third DEISA training session

The third DEISA training session was held at Barcelona Supercomputing Center, in Barcelona, Spain on March 7-9, 2007. In total, there were 48 participants, comprising 33 scientific users, 9 speakers and 6 members of the DEISA staff.



Figure 4: 3rd DEISA training session

There were 9 lectures in this training session. The *Access to DEISA* hands-on-session took place the second day of the training. The presentations can be found at http://www.deisa.org/news_events/deisa_events/barcelona_training.php :

- DEISA General Presentations
 - [DEISA Overview](#), Victor Alessandrini, IDRIS
 - [DEISA Access, filesystems and job submission](#), Denis Girou, IDRIS
 - [DEISA Common Production Environment](#), Denis Girou, IDRIS
 - [UNICORE](#), Michael Rambadt, FZJ
 - [The DESHL client](#), Malcolm Illingworth, EPCC
- [Special Topic – Performance measurements and codes optimization](#), Jesús Labarta, BSC
- Use Cases
 - [FEARLESS – A new modelling approach for turbulent astrophysical flows](#), Wolfram Schmidt, University of Würzburg
 - [Improving Quantum Computer Simulations: DEISA project results](#), Marcus Richter and Guido Arnold, FZJ
 - [Monte Carlo Simulation of a Twisted Nematic Display](#), Matteo Ricci, Dipartimento di Chimica Fisica e Inorganica – Università di Bologna

As it was the case for the Jülich session and will be the case for all the next sessions, all scientific participants were granted with a three-months-long test user account for DEISA (DTU account).

4. Results of the feedback questionnaires from the three training session events organized during the 1st project year

In total, 68 scientists from 13 European countries attended the three DEISA training sessions organized during the 1st eDEISA project year, with in addition 9 scientific speakers and 13 DEISA staff members. Among the 68 scientists, 55 participants have answered to the feedback questionnaire.

Of the respondents 62 per cent were from higher/further education, 18 per cent from government laboratories, 16 per cent from supercomputing centers, 2 per cent from the industry and 2 per cent from other sectors, mostly research institutes.

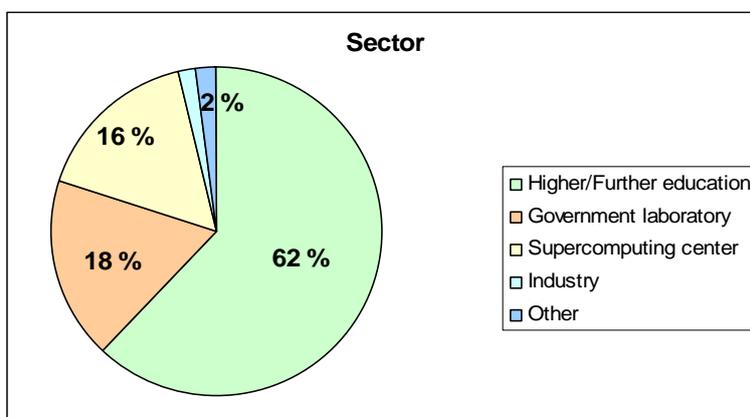


Figure 5: Sectors, in which the participants work

39 per cent of the respondents were research fellows, 21 per cent belonged to the academic staff, 21 per cent were post-graduate students, and 19 per cent belonged to the group "other", comprising research engineers, research collaborators, and Postdocs.

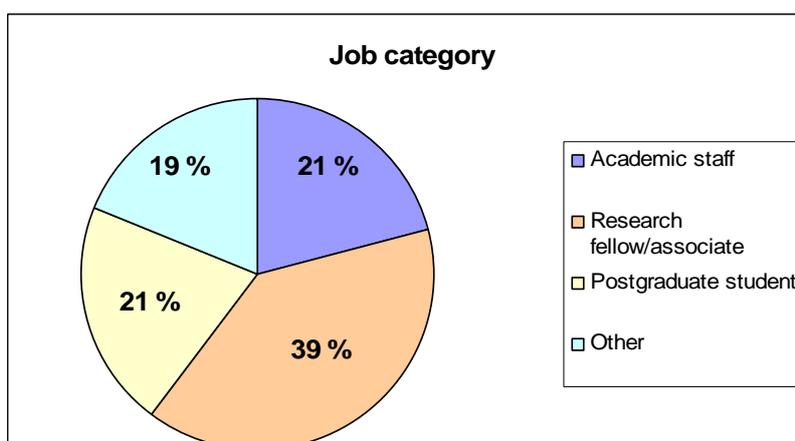


Figure 6: Participants' job categories

The respondents areas of research were Bioinformatics and biomedicine (9), Computational chemistry and quantum chemistry (7), HPC (7), Grid computing (5), Fluid, Stellar and Molecular Dynamics (4), Fluid mechanics and quantum mechanics (4), Climate modeling and weather forecasting (3), Turbulence (2), Condensed Matter Physics (2), Astrophysics (2), Remote sensing (2), Computational linguistics,

Geosciences, Plasma simulation, Simulation of QC, Multiscale modelling, Amorphous materials, Cold Atoms, BEC, Material Science, Material Physics, Solid state Physics.

38 per cent of the respondents were experienced users of e-Infrastructures, 15 per cent were new users, and 39 per cent wanted to understand how or whether its techniques are relevant to respondents' research and work.

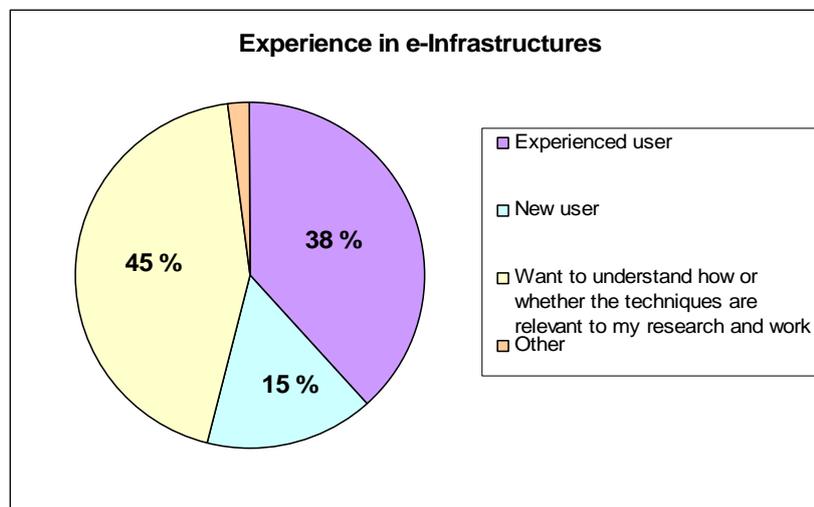


Figure 7: Participants' experience un e-Infrastructures

The objectives that respondents named for participation were: Acquiring detailed information on DEISA and possibilities to apply a DECI project (17), learning to work with DEISA (12), learning to use UNICORE (5), code optimization (2), improve code performance in DEISA (1), learning about possibilities for industry (1), and learning about new supercomputing technologies (6), learning about European HPC and grid infrastructures (3).

The overall assessment of the training sessions was 'very good', scoring an average 4 on an evaluation criteria scale from 1-5). 98 per cent of the respondents thought the training to be helpful in achieving their objectives. Concerning the contribution in support to participants' research activities, half of the respondents said that the training had either exceeded or achieved expectations. The rest stated that the training made a positive contribution to their research activity.

As unexpected benefits in attending the training, participants mentioned learning about submission policies at supercomputing centers, becoming aware of code analysis techniques with immediate application for the codes used, new scientific contacts and meeting people from many other research fields.

The great majority of the participants learned about the DEISA training either by an email call (19) or through colleagues (26). Seven respondents found out about the training by reading the DEISA eNewsletter and by visiting the DEISA website.

5. Realisation and dissemination of the training sessions

DEISA project's N2 dissemination activity has assisted in the realization and dissemination of the three DEISA Training sessions that took place on July 3-5, 2006 in Paris, France; on October 23-25, 2006 in Jülich, Germany and on March 7-9, 2007 in Barcelona, Spain. The organization of the 4th DEISA training session to be held on May 30 – June 1, 2007 is in process.

The main channel of dissemination has been the DEISA website. Webpages have been realized for the upcoming training at <http://www.deisa.org/training>. After the training, all presentations and pictures of the training have been made available at the event specific webpages. In addition training sessions have been highlighted in the DEISA website's front page. In total there have been 2239 different visitors to the training related webpages, in the period of June 2006 (establishment of the 1st training pages) to April 2007. In figures 8, 9 and 10 web traffic in the training related webpages are presented.

Webpages	Number of different visitors
Paris training	476
Jülich training	503
Barcelona training	439
Upcoming Espoo training	702

Figure 8: Number of different visitors to the training related webpages, period June 2006 – April 2007

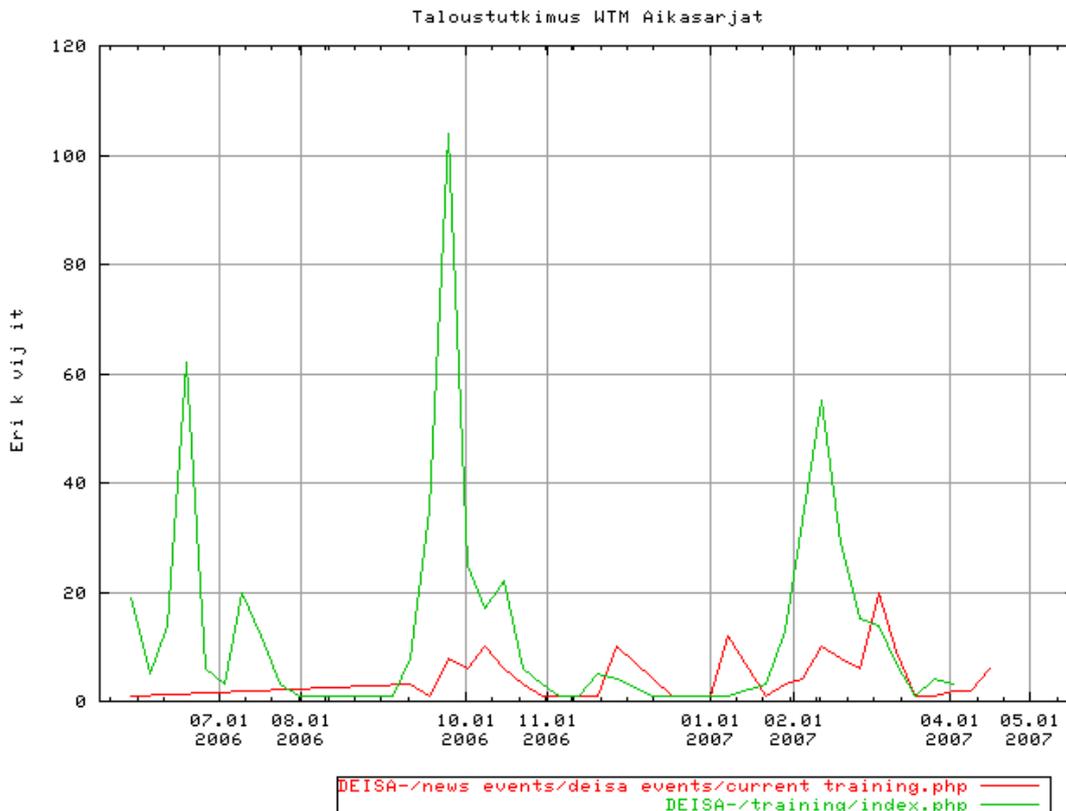


Figure 9: Different visitors to the webpage advertising the upcoming training session in a week, period June 2006 – April 2007

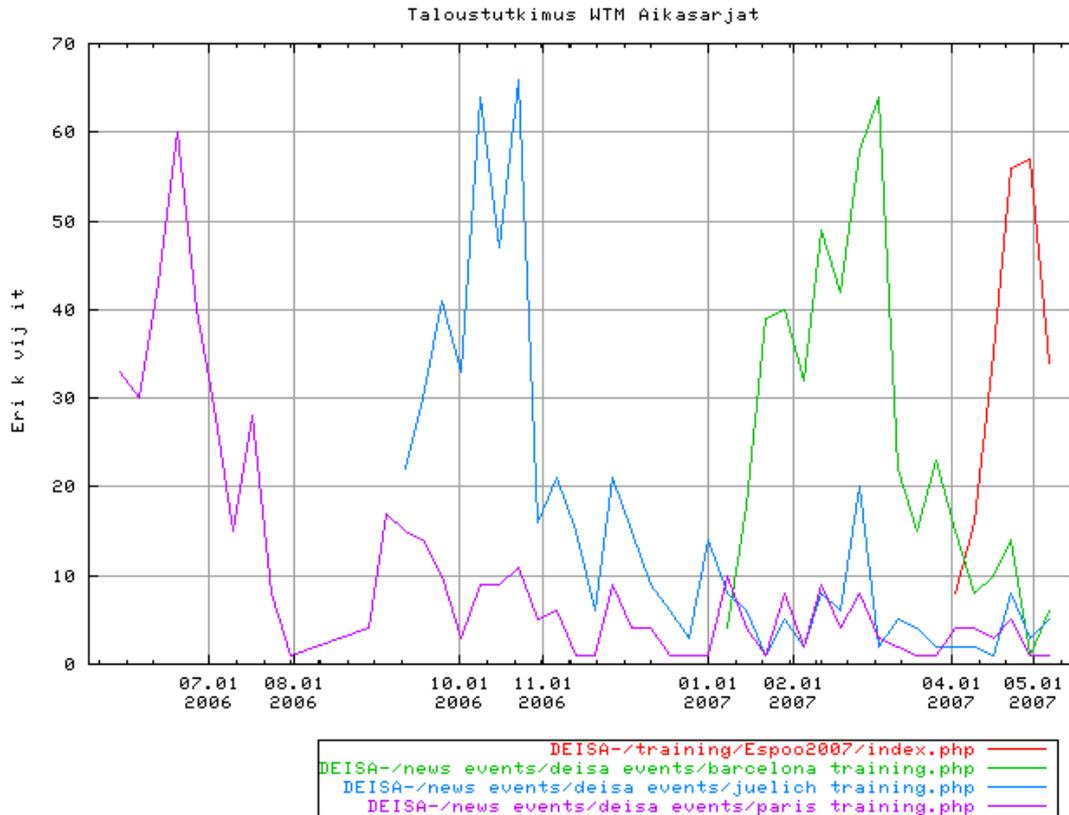


Figure 10: Different visitors to the training specific webpages in a week, period June 2006 – April 2007

Other means of promotion have been:

- production of the electronic invitation card and its distribution by all the partner sites to their users and usual, appropriate contacts
- direct contacts to current DECI in collaboration with the members of the Application Task-Force (ATaskF)
- articles in the DEISA eNewsletter, which has over 600 subscribers and which is distributed locally by all partners to about 6200 email contacts:
 - 1st DEISA training: highly scalable parallel applications, Vol. 2, 2006
 - New tools from DEISA training, Vol. 3, 2006
 - DEISA Training session in Jülich, Germany, Vol. 4, 2006
 - DEISA Training: Positive Contribution to Research, Vol. 1, 2007
 - DEISA Training announce, Vol. 2, 2007
 - DEISA Training: First hand experience of the infrastructure, Vol. 3, 2007

DEISA project's N2 dissemination activity has also produced the supporting material for the training sessions: badges, handout covers and feedback questionnaires.

6. Annex 1: Lists of attendees

Table 1: Participants of the 1st DEISA training

Name	Company	Country
Nicole Audiffren	CINES	France
Marc Baaden	CNRS UPR9080/IBPC	France
J. Behrend	WSI	Germany
Jaroslav Bijak	Grenoble Institute of Technology	France
Juan Escobar	CNRS/LA	France
Ari Hietanen	University of Helsinki (Div. of Theor. Phys.)	Finland
Drissa Houatra	France Telecom – R&D Division	France
Sergio Hoyas	ETSI Aeroneicos – UPM	Spain
Evangelos Kotsalis	CSE Lab ETH Zurich	Switzerland
Zouhair Lachkar	CEA/LSCE	France
Matthieu Marquillie	Laboratoire de Mécanique de Lille (UMR 8107)	France
Jose Carlos Mourin-Allego	CESGA	Spain
Kevin Roy	University of Manchester	UK
Oleg Zabaydullin	LIXAM	France
DEISA STAFF		
Thierry Goldmann	IDRIS-CNRS	France
Marc-André Hermanns	Forschungszentrum Jülich	Germany
Petri Nikunen	CSC	Finland
Dirk Petry	LRZ	Germany
Saara Värttö	CSC – Scientific Computing Ltd	Finland
David Vicente	BSC-CNS	Spain
SPEAKERS		
Victor Alessandrini	IDRIS-CNRS	France
Édouard Audit	CEA/DAPNIA/Sap	France
Mark Bull	EPCC	UK
Denis Girou	IDRIS-CNRS	France
Roman Hatzky	RZG, IPP & MPG	Germany
Sergio Hoyas	Madrid School of Aeronautics	Spain
Malcom Ilingworth	EPCC	UK
George Mozdzynski	ECMWF	UK
Gavin Pringle	EPCC	UK
Michaël Rambadt	FZJ	Germany

Table 2: Participants, of the 2nd DEISA training

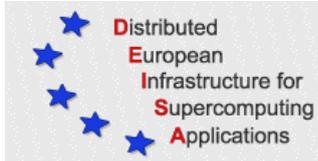
Name	Company	Country
Paulo Abreu	Departamento de Quoca	Portugal
Francisco Calvo	University of Manchester	UK
Enrico Calzavarini	University of Twente, The Netherlands	The Netherlands
Giancarlo Cicero	Politecnico of Torino	Italy
Dimitris Dellis	University of Athens	Greece
Alessandro G. Di Nuovo	University of Catania	Italy
Jonathan Downing	Astronomisches Rechen-Institut	Germany
Michael Eckart	Universität Ulm	Germany
Andreas Ernst	Astronomisches Rechen-Institut	Germany
George Lithoxoos	University of Athens	Greece
Matthieu Hautreux	CEA	France
Waclaw Kus	Silesian University of Technology DSMCM	Poland
Joop van Lenthe	Theoretical Chemistry Utrecht	The Netherlands
Claudio Maruccio	University of Minho	Portugal
Andre Severo Pereira Gomes	Laboratoire de Chimie Quantique	France
Jannis Samios	University of Athens	Greece
Jörg Schumacher	Technische Universität Ilmenau	Germany
Ulrich Sperhake	TPI Universität Jena	Germany
Toma Susi	Helsinki University of Technology	Finland
Timo Vehviläinen	Laboratory of Physics (HUT)	Finland
Willy Weisz	University of Vienna, VCPC	Austria
DEISA STAFF		
Ivan Giroto	CINECA	Italy
Marc-André Hermanns	Forschungszentrum Jülich	Germany
Andrea Tarsi	CINECA	Italy
Pascal Voury	IDRIS-CNRS	France
Michele Weiland	EPCC, University of Edinburgh	UK
SPEAKERS		
Wolfgang Frings	Forschungszentrum Jülich	Germany
Denis Girou	IDRIS-CNRS	France
Malcolm Illingworth	EPCC, University of Edinburgh	UK
George Mozdzyński	ECMWF	UK
Gavin Pringle	EPCC, University of Edinburgh	UK
Michael Rambadt	Forschungszentrum Jülich	Germany
Morris Riedel	Forschungszentrum Jülich	Germany
Achim Streit	Forschungszentrum Jülich	Germany

Table 3: Participants of the 3rd DEISA training session

Name	Company	Country / Nationality
Roman Affentranger	Institute of Biotechnology and Biomedicine	Spain / Swiss
Alexander Alijah	University of Coimbra	Portugal / German
Jose Luis Alvarez-Perez	Universitat Politècnica de Catalunya	Spain / Spanish
Ignasi Belda	Institut de Recerca Biomedica	Spain / Spanish
Arrigo Calzolari	INFM-CNR S3, National Center	Italy / Italian
Javier Castaneda	Universitat de Barcelona	Spain / Spain
Albert Compte	IDIBAPS	Spain / Spanish
Stefano Corni	INFM-CNR National Research Center S3	Italy / Italy
Xavier Daura	Universitat Autònoma de Barcelona	Spain / Spanish
Sergio Espeja	IULA Universitat Pompeu Fabra	Spain / Spanish
Alejandra Flores	UPC	Barcelona / Mexican
Alfredo Guardo	Universitat Politècnica de Catalunya	Spain / Colombia
Mark Halling-Brown	Birkbeck College	UK / British
Ondrej Jakl	Czech Academy of Sci., Institute of Geonics	Czech Republic / Czech
Damien Lacroix	Universitat Politècnica de Catalunya	Spain / French
Xavier Luri	Univ. de Barcelona	Spain / Spanish
Sergio Madurga	PCB-UB	Spain / Spanish
Emilio Mastroiani	Universita di Catania	Italy / Italian
Stéphane Montesino	LEGI	France / French
Francisco Ogando	Helsinki University of Technology	Finland / Spanish
Ignacio Pagonabarraga	Departament de Física Fonamental (UB)	Spain / Spanish
Marzio Pennisi	University of Catania	Italy / Italian
Horacio Emilio Perez Sanchez	ForschungsZentrum Karlsruhe	Germany / Spanish
Jordi Portell	Universitat de Barcelona	Spain / Spanish
J. Daniel Prades	Universitat de Barcelona	Spain / Spain
Adlane Sayede	LPCIA – Université d'Artois	France / Algerian
Alain Schenkel	Institute of Biotechnology	Finland / Swiss
Francisco Martinez Serrano	Universidad Miguel Hernandez	Spain / Spanish
Mina Talati	Université Claude Bernard Lyon 1 and CNRS	France / Indian
David Valencia	Universidad de Extremadura	Spain / Spanish
Angel de Vicente	Instituto de Astrofísica de Canarias	Spain / Spanish
Tsjerk Wassenaar	University of Utrecht	The Netherlands / Dutch
Lei Zhang	IRB Barcelona – Institut de Recerca Biomèdica	Spain / China

DEISA STAFF		
Stefan Berner	LRZ Munich	Germany / German
Carlo Cavazzoni	CINECA	Italy / Italian
Andre Giesler	Forschungszentrum Jülich GmbH (FZJ)	Germany / German
Anh-Khai Nguyen	IDRIS-CNRS	France / French
Saara Värttö	CSC – Finnish IT center for science	Finland / Finnish
David Vicente	BSC-CNS	Spain / Spanish
SPEAKERS		
Victor Alessandrini	IDRIS-CNRS	France / French
Guido Arnold	Forschungszentrum Jülich GmbH (FZJ)	Germany
Denis Girou	IDRIS-CNRS	France / French
Malcolm Illingworth	EPCC	UK / British
Jesús Labarta	Barcelona Supercomputing Center	Spain / Spanish
Michael Rambadt	Forschungszentrum Jülich GmbH (FZJ)	Germany / German
Matteo Ricci	Dipartimento di Chimica Fisica e Inorganica	Italy / Italian
Marcus Richter	Forschungszentrum Jülich GmbH (FZJ)	Germany / German
Wolfram Schmidt	University of Würzburg	Germany / Austrian

7. Annex 2: Description of the UNICORE exercises done during the Hands-on sessions



Hands-out DEISA UNICORE Training Session BSC Barcelona, 2007-03-08

7.1 Prerequisites

The UNICORE Client is a Java Application that requires at least **Java** Run Time Environment version **1.5** to be installed.

- Check if the **SUN-Java Runtime Environment** version higher than 1.5 is installed on your computer
 - **On MS Windows** type **java -version** in a DOS box
 - **On UNIX/Linux** type **java -version** in a console
- If the execution of the command fails or if you have a java version older than 1.5 go to the **SUN web page** to download and install the latest JRE 1.5.x on your local computer, following the instructions given there.

7.2 Prepare and configure the Client

7.2.1 Install the Client

- Download **client_5.6.5_installer.jar** from <http://www.unicore.eu> ->Download ->Core ->Client ->Download
- Execute the installer and start the client

Linux:

- Open a console and type:
java -jar client_5.6.5_installer.jar
- Start client via:
cd ~/UNICORE/bin
./unicore &

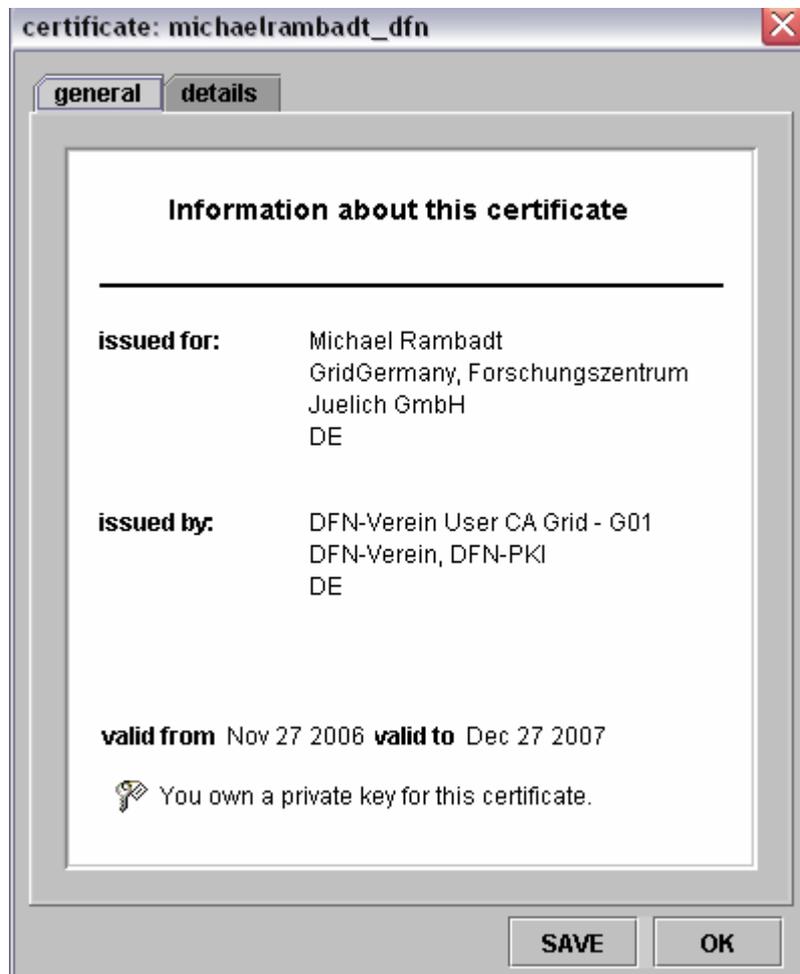
Windows:

- Double-click on **client_5.6.5_installer.jar**
- By double clicking on the UNICORE desktop symbol or execute the file **UNICORE\bin\unicore.bat**

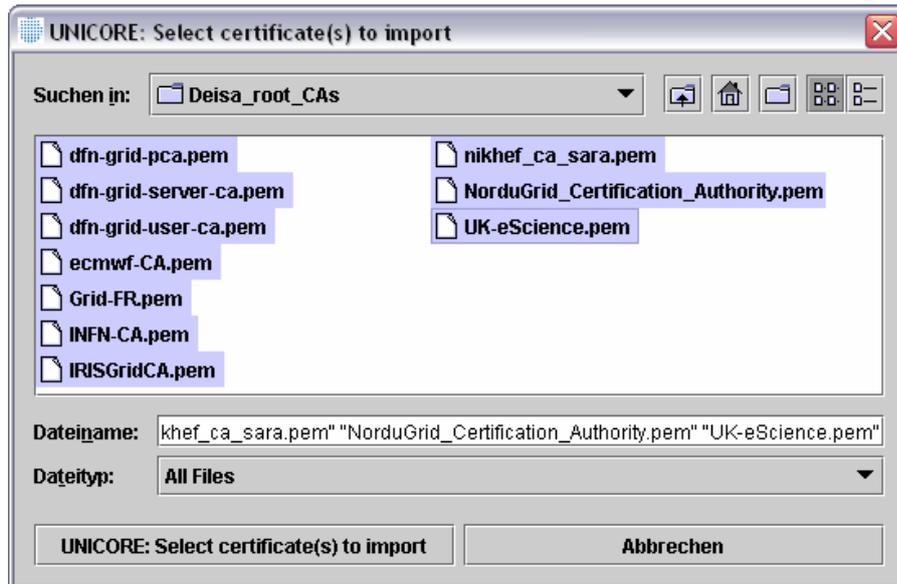
- Confirm creation of a new UNICORE keystore and choose a password.
- The UNICORE password checking is quite strict so a warning might appear. Click on **-> OK**

7.2.2 Import private certificate keystore (.pkcs12)

- **Prerequisites:** you have got already a signed certificate from your national CA and the corresponding .pkcs12 file on your local computer. Assuming this is called **user1.p12**
- Select **Settings -> Keystore Editor, Action -> Import Keystore** in the UNICORE client and select **user1.p12** within the upcoming filebrowser
- Provide your password the **user1.p12** is encrypted with
- Click **OK**. Your user certificate will be shown in the list of **Key entries**. If there is a problem you will be informed, otherwise you won't get any message.
- Mark your certificate and click on **Details**. They should look similar to the example.



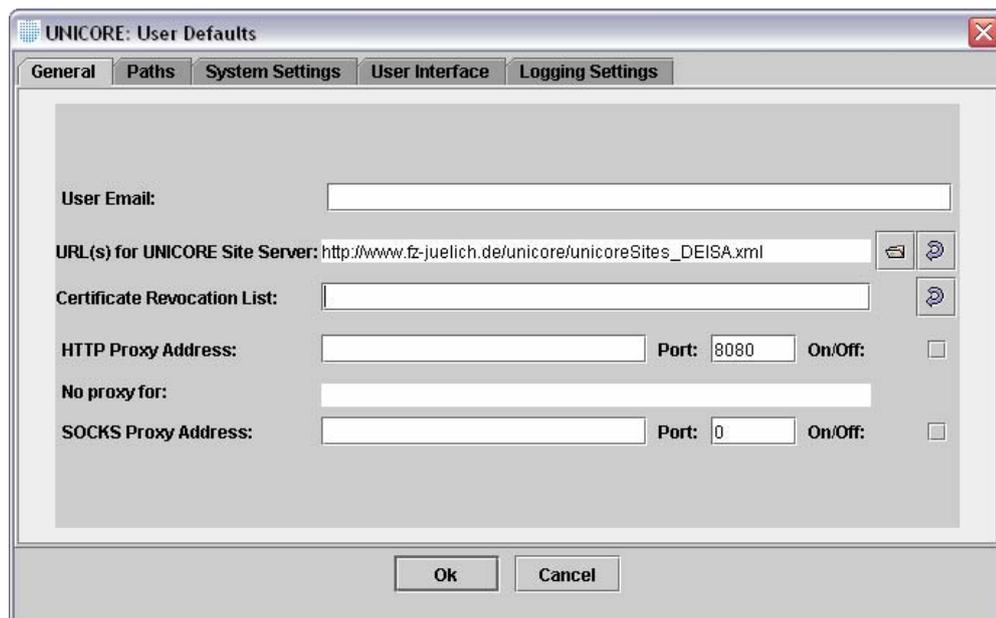
- Download the EU-GridPMA certificates from <http://winnetou.matrix.sara.nl/deisa/certs>
->ZIP package with certs from CAcerts file->unicerts.zip
- Extract the certificates to a local folder
- Import the EU-GridPMA certificates with the **UNICORE Keystore Editor**
- Choose from the menu **Action -> Import Certificate** and select all stored certificates



- Confirm with **OK**
- Save your keystore: **File -> Save**
- Exit the keystore: **File->Exit**

7.2.3 Make DEISA Gateways known to the client

- Select **Settings -> User Defaults General->URL(s) for UNICORE Site Server**
- Remove the existing URL (if any)
- Type http://www.fz-juelich.de/unicore/unicoreSites_DEISA.xml in this field



- Confirm with **OK**

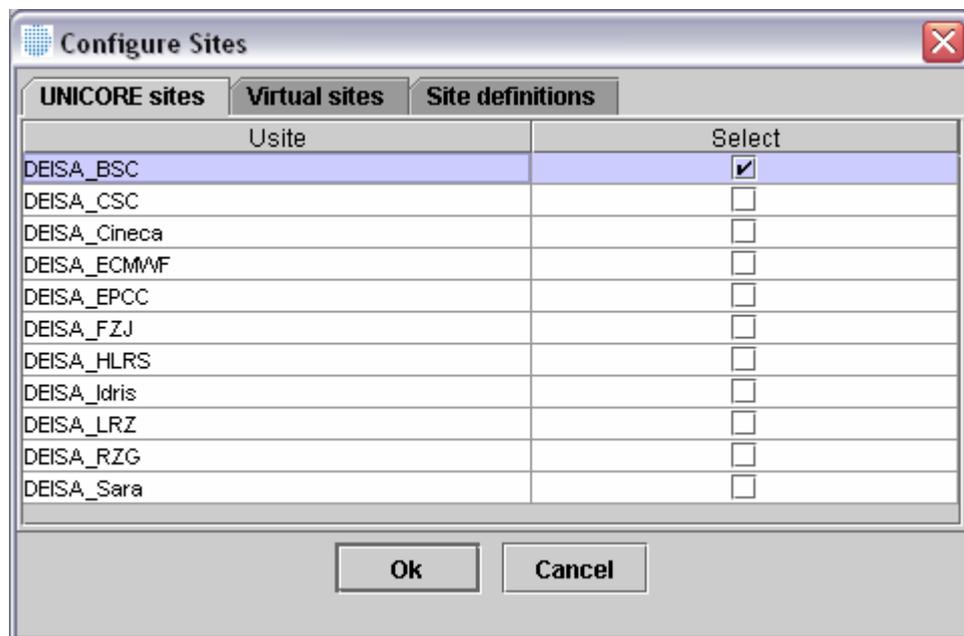
7.2.4 Configure your Home site as UNICORE Site

All DEISA users are expected to use their Home site as UNICORE Site (e.g. DEISA_BSC) for job submission. Only if the Home Site is not available other UNICORE Sites might be used instead. Each UNICORE Site contains the same Virtual Sites or target supercomputer resources.

Therefore configure the UNICORE client just to show your Home site as UNICORE Site as follows:

Select from the menu of the UNICORE client **Settings->Configure sites**

- In the “Configure Sites editor” deselect all UNICORE Sites **except from your Home site**
- For a user with Home site BSC the UNICORE sites configuration should look as follows:



- Confirm with **OK**

7.2.5 Refresh the actual settings

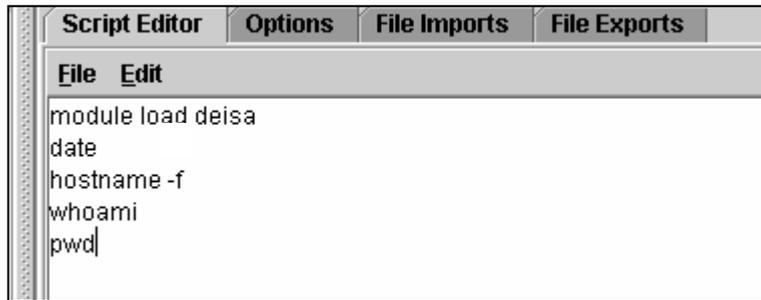
To refresh the actual settings of the UNICORE client with the actual configuration, select **Settings -> Clear Cache**

7.3 Use the Client

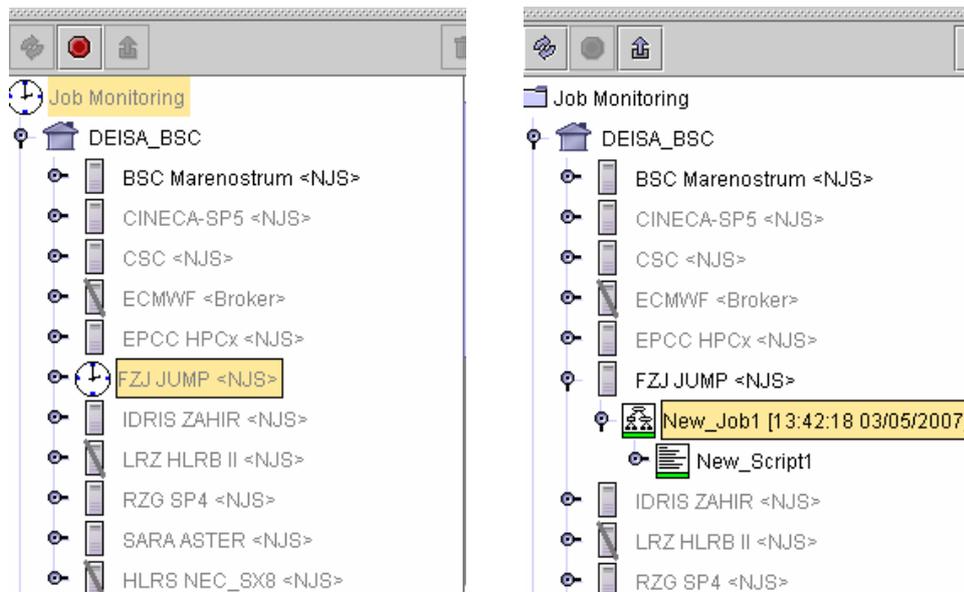
7.3.1 Run a simple script

- Select from the menu of the UNICORE client **File -> New Job**
- Select your DEISA Home Site (**UNICORE Site e.g. DEISA_BSC**) and a **Virtual Site (e.g. FZJ JUMP)**
 - **Important note:** BSC Marenostrum, ECMWF and HLRS_NEC_SX8 are **not** available for the training session use!
- Select **Job Preparation -> Add Script**

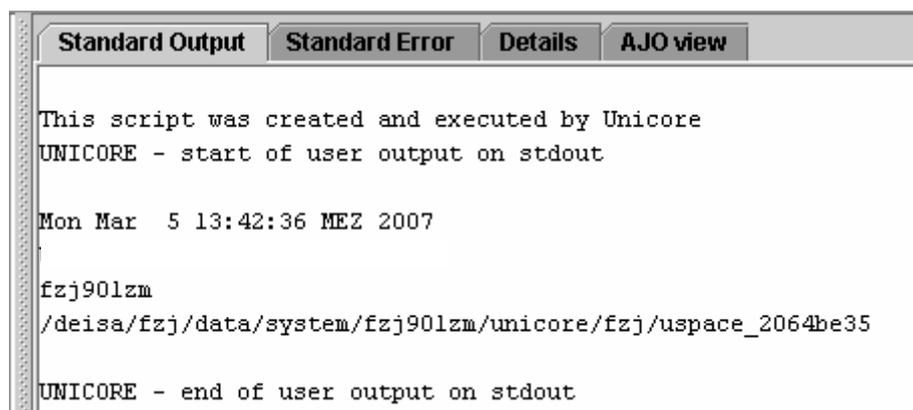
- Type a simple Bash script in the UNICORE script editor



- Right-click on the job and choose **Submit** or click on . The job will be submitted to the supercomputer of your Home site.
- Right-click on the target system the job has been submitted to in the lower left **Job Monitoring** window and choose **Refresh** or click on . You should see your job now coloured in yellow (indicating that the job is queued), red (indicating that the job has been failed) or green (indicating that the job has run successfully).



- Right-click on the job in the lower left window and choose **Fetch output** or click on . In the right part of the screen you should now see the output of your script.



7.3.2 Hold, resume, abort a job

- To create a job that runs a little longer, add to the script `sleep 20` and submit the job as before. Right-click on the target system in the **Job Monitoring** Window and choose **Refresh**. You should see your second job coloured in yellow and orange, indicating that the job is being executed/run. Right-click on the job and choose **Hold Job**. Then right-click on the job and choose **Resume Job**. You can try **Abort Job** (after job is hold) as well.

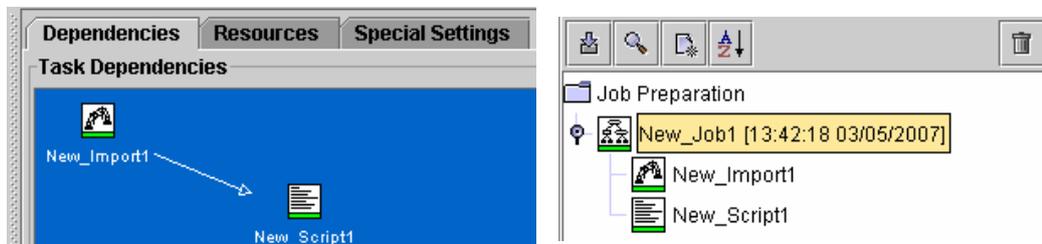
7.3.3 File import

- Create an arbitrary text file **foo.txt**. Create a new job (**File -> New Job**) and select a site where it should run.
- Select **Job Preparation-> Add Import**. In the right part of the screen, click on the leftmost icon to create a new file entry 
 - In the first column, choose e.g. **Local**
 - Choose the file to import from the UNICORE filebrowser  or enter the complete path to your text file (e.g. `/home/deisa111/foo.txt`) into the second column.
 - In the third column enter the new name of the text file, use the same name **foo.txt**.

Source	File at Source	File in Job Directory	Overwrite File(s)	Binary
Local	/home/deisa111/foo.txt	foo.txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>

- To display the content of the imported file add a UNICORE script (**Job Preparation-> Add Script**) to the job with the following content:


```
cat foo.txt
```
- As it is, the **script task** and the **import task** would run independent of each other in arbitrary order. But the script should be executed after the import has finished, so we have to create a **workflow**.
- Click on the job and look at the blue window on the right hand side.
- Click on the file import with the right mouse button (don't release the button yet) and drag an arrow to the script.



- Now submit the job and fetch the output as above.

7.3.4 If-then-else job

- Create a copy of foo.txt (foo2.txt) and change the copy a little.
- Create a new job and select the Site where it should run. Add an **import** task as before (**Job Preparation-> Add File Import**), but there create a task for both foo.txt and foo2.txt.

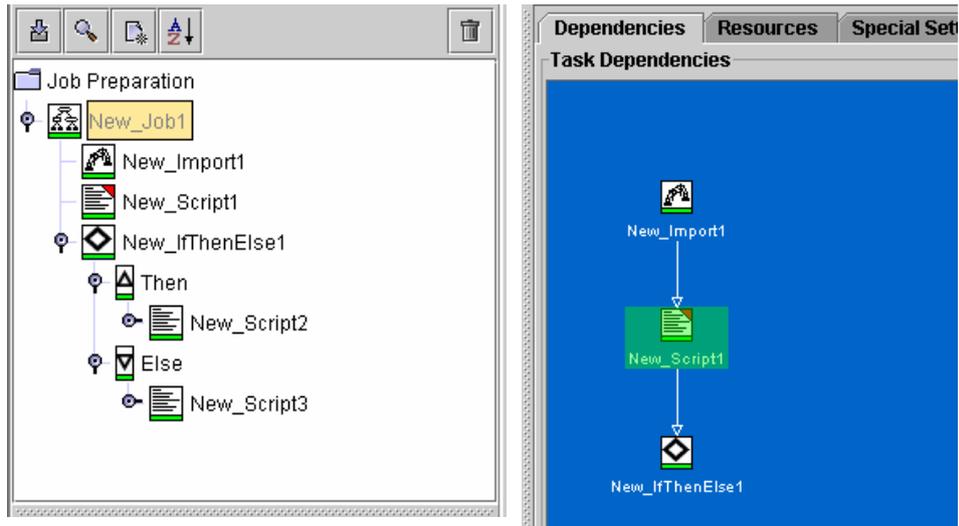
Source	File at Source	File in Job Directory	Overwrite File(s)	Binary
Local	/home/deisa111/foo.txt	foo.txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Local	/home/deisa111/foo2.txt	foo2.txt	<input checked="" type="checkbox"/>	<input type="checkbox"/>

- Add a UNICORE script (**Job Preparation-> Add Script**) which contains the line


```
diff foo.txt foo2.txt > diff_output
```
- To add an if-then-else task, select **Job Preparation-> Add If Then Else**.
- In the right part of the screen you can now configure the **if-then-else task**.
- In the uppermost part labelled "Return Code Test" choose from the first drop down menu **return code equals** and leave the according value at zero.
- From the second drop down menu choose your diff-script.

- The if-then-else task has two subjobs: Then and Else.
 - Select the Then-part and add a script **Job Preparation-> Add Script**.
 - Enter: `echo "Files are equal!"` in the UNICORE script editor.
 - Add a script to the Else-part as well with the following content:


```
echo "Files differ!"
```
 - Now go to the workflow edit window and drag an arrow from the Import Task to the diff-script and from the diff script to the If-Then-Else-Task. Submit your job.



7.3.5 File export

- Add an export task to the above job (**Job Preparation-> Add Export**)
- Create a new entry and enter *diff_output* in the first column. Choose **Local** from the drop down menu and enter something like *diff_output* in the third column. If you want to run the job multiple times, select “**Overwrite File(s)**” as well.
- In the workflow editor, drag an arrow from the **if-then-else** task to the **file export** task.
- Submit your job. You should now have a copy of *diff_output* with the output of your diff-script on your local workstation.

7.3.6 DEISA GPFS and UNICORE

This example is to show the functionality and strength of DEISA GPFS within UNICORE.

Step 1:

- Create a new UNICORE workflow with **File->New Job**
- Rename the job to “**DEISA_job**”
- Select your Home site as **UNICORE Site** (e.g. **DEISA_BSC** for the BSC users) and the **Virtual Site FZJ JUMP**
- Click on “**DEISA_job**”
- Add a subjob to the job (**Job Preparation->Add Sub Job**)
- Rename the first subjob from “**New_Sub_Job1**” to “**IDRIS**”
- Click on “**IDRIS**” and select the **UNICORE Site DEISA_BSC** and the **Virtual Site IDRIS ZAHIR**. This means that this subjob will be executed on the IDRIS supercomputer
- Click on “**IDRIS**” and select **Job Preparation->Add Script**
- Rename the script from “**New_Script1**” to “**start_at_IDRIS**”

- Enter the following script:

```

Name: New_Script3
Script Editor Options File Imports File Exports
File Edit
module load deisa
cd $DEISA_HOME
date
hostname
pwd
ls -al
echo "*****" >> working
echo `hostname` @ "`date`>> working
  
```

Step 2:

- Click on “DEISA_job”
- Add a 2nd subjob to “DEISA_job” (**Job Preparation->Add Sub Job**)
- Rename the 2nd subjob from “New_Sub_Job2” to “FZJ”
- Click on “FZJ” and select your Home site as **UNICORE Site** and the **Virtual Site FZJ JUMP**. This means that this 2nd subjob will be executed on the FZJ supercomputer
- Click on “FZJ” and select **Job Preparation->Add Script**
- Rename the script to “continue_at_FZJ”
- Enter the above script

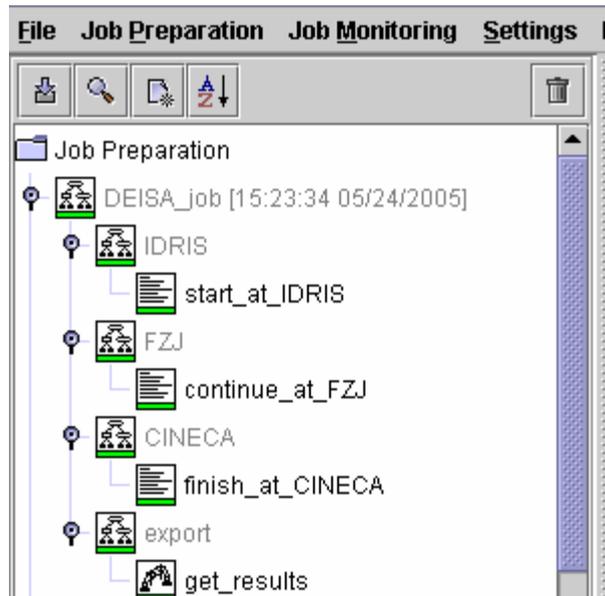
Step 3:

- Click on “DEISA_job”
- Add a third subjob to “DEISA_job” (**Job Preparation->Add Sub Job**)
- Rename the third subjob from “New_Sub_Job3” to “CINECA”
- Click on “CINECA” and select your Home site as **UNICORE Site** and the **Virtual Site CINECA-SP5**. This means that this third subjob will be executed on the CINECA supercomputer
- Click on “CINECA” and select Job **Preparation->Add Script**
- Rename the script to “finish_at_CINECA”
- Enter the above script

Step 4:

- Click on “DEISA_job”
- Add a fourth subjob to “DEISA_job” (**Job Preparation->Add Sub Job**)
- Rename the fourth subjob from “New_Sub_Job4” to “export”
- Click on “export” and select your Home site as **UNICORE Site** and the **Virtual Site RZG SP4**. This means that this fourth subjob will be executed on the RZG supercomputer
- Click on “export” and select **Job Preparation->Add Export**
- Rename the new export task to “get_results”
 - Create a new export task by clicking on 
 - Type in File in Job Directory: \$DEISA_HOME/working
 - Choose as Destination: Local
 - Choose as File at Destination your local desktop directory

- Click on “DEISA_job” It should look as follows:



- Now submit the job and fetch the output as above.
- Have a look in the local copy of “working”. The data from all sites the job has been submitted to should be available.

7.4 Access the DCPE with UNICORE

- Create a new job with following script

```
module load deisa
cd $DEISA_HOME
pwd
ls
module list
module avail
```

- Create a new job with the following script:

```
module load cpmd
module list
echo "CPMD-PATH:" $PATH
module unload cpmd
module list
echo "CPMD-PATH:" $PATH
```

```
module load nag
module list
```

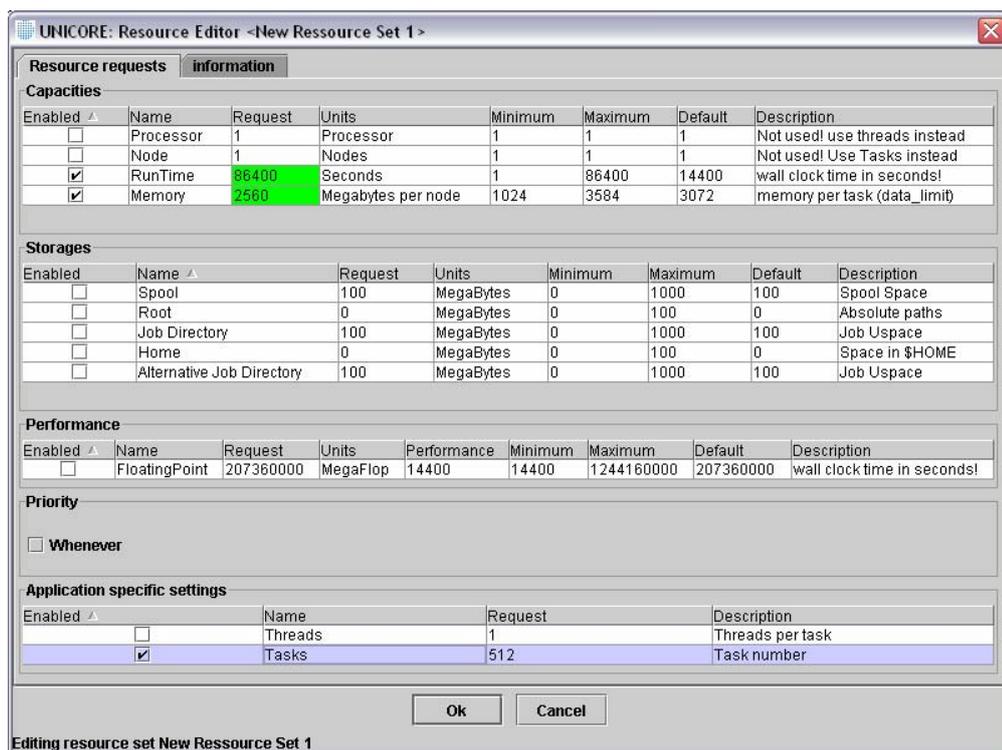
```
module load fftw
module list
echo $DEISA_FFLAGS
echo $DEISA_LDFLAGS
module switch fftw fftw/3
module list
```

```
echo $DEISA_FFLAGS
echo $DEISA_LDFLAGS
```

```
module whatis fortran
module help fortran
module display fortran
```

7.5 Defining resources with UNICORE

- Mark the job and click on **Resources** in the upper right window
- Add a new resource set by clicking on 
- Edit the resource set by clicking on 
 - the **Resource Editor** appears – enable the values you want to change by clicking in the corresponding squares of the first row
 - you can change the values by double clicking the **Request** row
 - your values have to be between the given **Minimum** and **Maximum**
 - when you have finished editing the resource set click **OK**
 - for each task in your job you can create an own resource set
 - to merge the resource set with the corresponding task select the lower right **Tasks** window and click on the task



UNICORE: Resource Editor -New Ressource Set 1

Resource requests information

Capacities

Enabled	Name	Request	Units	Minimum	Maximum	Default	Description
<input type="checkbox"/>	Processor	1	Processor	1	1	1	Not used! use threads instead
<input type="checkbox"/>	Node	1	Nodes	1	1	1	Not used! Use Tasks instead
<input checked="" type="checkbox"/>	RunTime	86400	Seconds	1	86400	14400	wall clock time in seconds!
<input checked="" type="checkbox"/>	Memory	2560	Megabytes per node	1024	3584	3072	memory per task (data_limit)

Storages

Enabled	Name	Request	Units	Minimum	Maximum	Default	Description
<input type="checkbox"/>	Spool	100	MegaBytes	0	1000	100	Spool Space
<input type="checkbox"/>	Root	0	MegaBytes	0	100	0	Absolute paths
<input type="checkbox"/>	Job Directory	100	MegaBytes	0	1000	100	Job Uspace
<input type="checkbox"/>	Home	0	MegaBytes	0	100	0	Space in \$HOME
<input type="checkbox"/>	Alternative Job Directory	100	MegaBytes	0	1000	100	Job Uspace

Performance

Enabled	Name	Request	Units	Performance	Minimum	Maximum	Default	Description
<input type="checkbox"/>	FloatingPoint	207360000	MegaFlop	14400	14400	1244160000	207360000	wall clock time in seconds!

Priority

Whenever

Application specific settings

Enabled	Name	Request	Description
<input type="checkbox"/>	Threads	1	Threads per task
<input checked="" type="checkbox"/>	Tasks	512	Task number

Ok Cancel

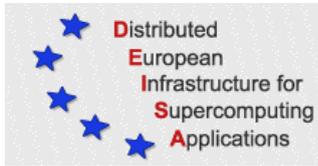
Editing resource set New Ressource Set 1

7.6 Hints for further usage

7.6.1 What site resources are available?

You can have a look at the resources of a site by clicking on the magnifying glass next to the Vsites List. Next to the tab of the workflow editor you can find a tab called "Resources" where you can specify the requirements of your job.

8. Annex 3: Description of the DESHL exercises done during the Hands-on sessions



DEISA Training Session – DESHL Exercises

BSC Barcelona, 2007-03-08

8.1 Prerequisites

- The DESHL Client is a Java Application that requires Java Run Time Environment version 1.5 to be installed. The DESHL client has been tested with the SUN Java Runtime Environment.

Check which version of Java is installed on your computer:

- On MS Windows type `java -version` in a command window
- On UNIX/Linux type `java -version` in a console window

If the execution of the command fails or if you have a Java version older than 1.5, then go to the SUN web page to download and install the latest release on your local workstation.

- To use the DESHL client, you will need a valid certificate in `.pem` format, and the root certification authority certificate, also in `.pem` format. For more information on certificates please see Appendix I, at the end of this document.
- You will also need to download the CA certificate for each of the DEISA gateways. These are available at <http://winnetou.matrix.sara.nl/deisa/certs/unicerts.tar.gz>, this must be unpacked into the directory containing your user certificate.

8.2 Installation

The following steps are required to install and configure the DESHL v4.0 client:

1. Download the installer
2. Install and configure the DESHL
3. Verify the settings

8.2.1 Download the installer

Download the latest DESHL installer from the JRA7 project site:

<http://forge.nesc.ac.uk/download.php/152/DESHL-4.0-RC3-installer.jar>

Either save it or execute it directly if your browser supports execution of jar files.

8.2.2 *Install and configure the DESHL via the installer GUI*

Run the GUI installer

If the installer did not immediately run in the previous step then run it now:

In a file explorer window double click on the installer jar file to start the installation. Alternatively, to install from the command line on UNIX:

```
$ java -jar DESHL-4.0-RC3-installer.jar
```

Alternatively, to install from the command line on Windows open a DOS command window and run the following:

```
C:\>java -jar DESHL-4.0-RC3-installer.jar
```

If problems are experienced when running the installer, extra debug information can be obtained by running with these additional arguments:

```
$ java -DTRACE=true -DSTACKTRACE=true -jar DESHL-4.0installer.jar
```

Configure the DESHL using the GUI installer

The graphical installer opens up with a welcome screen for DEISA JRA7 followed by a license agreement. The user is then requested to accept the license agreement before being allowed to install DESHL.

When prompted for an installation path, enter the following if installing on Windows:

```
C:\Program Files\DESHL-4.0
```

If installing on UNIX, specify a DESHL-4.0 directory in your local workstation home directory, for example:

```
/home/malcolm/DESHL-4.0
```

Next choose the packages to install as shown in Figure 1.

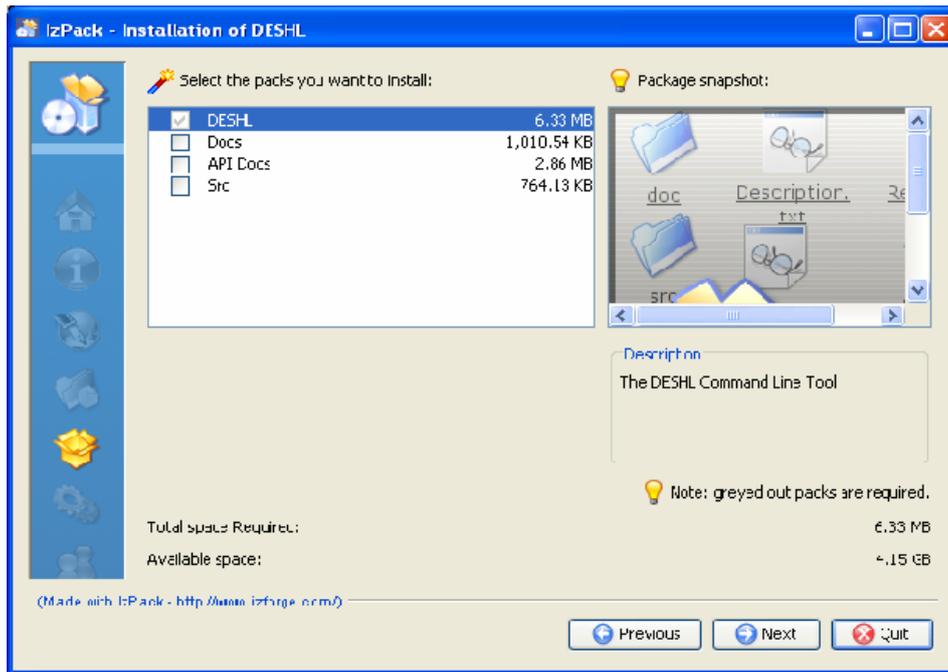


Figure 1: DESHL Package selection.

The base DESHL package is always required and cannot be deselected. Additional material can be selected or deselected for installation. The packages are:

- Docs – manuals for DESHL including the user manual and design document.
- API Docs – the Javadocs for the DESHL code.
- Src – the Java source for DESHL.

The next window, see Figure 2, shows how to configure one UNICORE NJS that can be used to interact with the DESHL infrastructure.

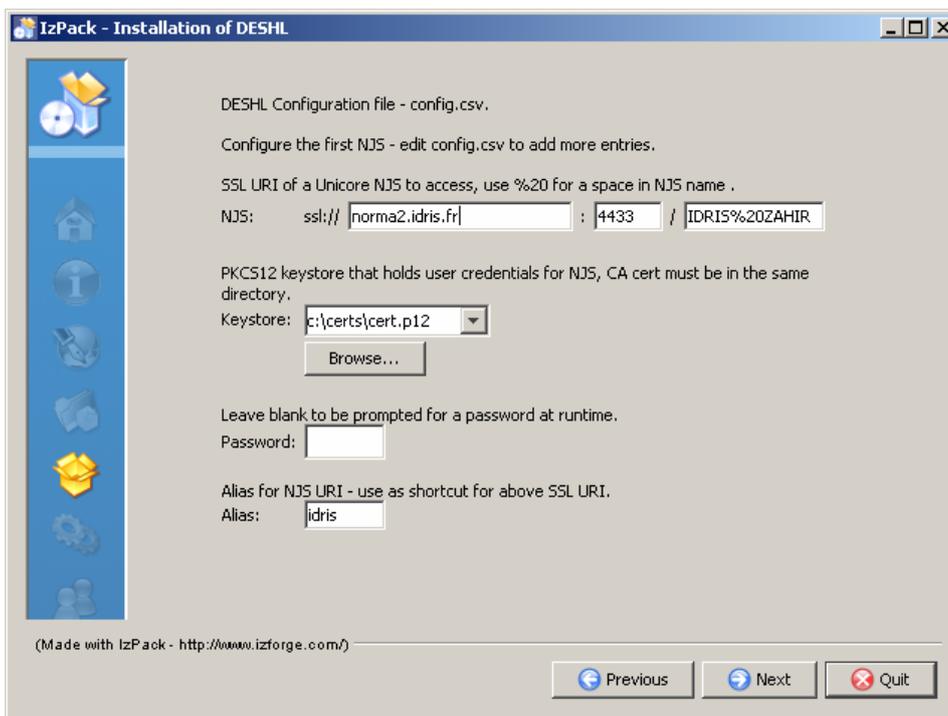


Figure 2: UNICORE NJS details.

The following information is expected (obtainable from the appropriate local site administration):

- The UNICORE NJS URI, this consists of:
 - The address and port of the UNICORE Gateway for that NJS
 - The name of the NJS (note that '%20' must be used instead of any spaces in the name)
- Use the 'Browse' button to navigate to the PKCS12 file that holds the user credentials. The directory that this file resides in must also contain the CA certificate for the issuer of the credential, in '.pem' format.
- If the password for the PKCS12 file is supplied here it is added to the configuration file in plaintext. This allows DESHL to access the PKCS12 keystore without prompting the user at every command. This file has user read only permissions on UNIX. Alternatively, this may be left blank and the DESHL client will ask for the password to be entered every time that it is required. For the purposes of the hands-on session, please leave this blank.
- An alias for the NJS URI. Define a shortcut for the NJS URI when using data staging and job management commands, this will make using the DESHL much easier than typing long path names. For example, "idris", "fzj", etc. The alias must not contain any spaces.

It should be possible to access any DEISA site from any DEISA gateway; however it is strongly recommended that you configure the DESHL client to always use your home site's gateway. A list of the current DEISA gateways is shown in the table below:

Sitename	Gateway Address	NJS name	Port
FZJ	zam177.zam.kfa-juelich.de	FZJ%20JUMP	4000
CINECA	fe04-ext.fec.cineca.it	CINECA-SP5	4433
RZG	unigate.rzg.mpg.de	RZG%20SP4	4433
SARA	uni-gw1.sara.nl	SARA%20ASTER	4004
IDRIS	norma2.idris.fr	IDRIS%20ZAHIR	4433
CSC	uni.csc.fi	CSC	4000
BSC	opsuni01.bsc.es	BSC%20Marenostrum	4433
LRZ	unicore.lrz.muenchen.de	LRZ%20HLRB%20II	4433
EPCC	admin.hpcx.ac.uk	EPCC%20HPCx	4433

(The BSC gateway will normally be accessible to those users who have BSC as their home site, but the supercomputer MareNostrum will not be available for the exercises nor during the DEISA Test User validity period. Please also note that IDRIS is unavailable for non-EU attendees.)

The installer attempts to validate and verify the supplied credentials in the following window.

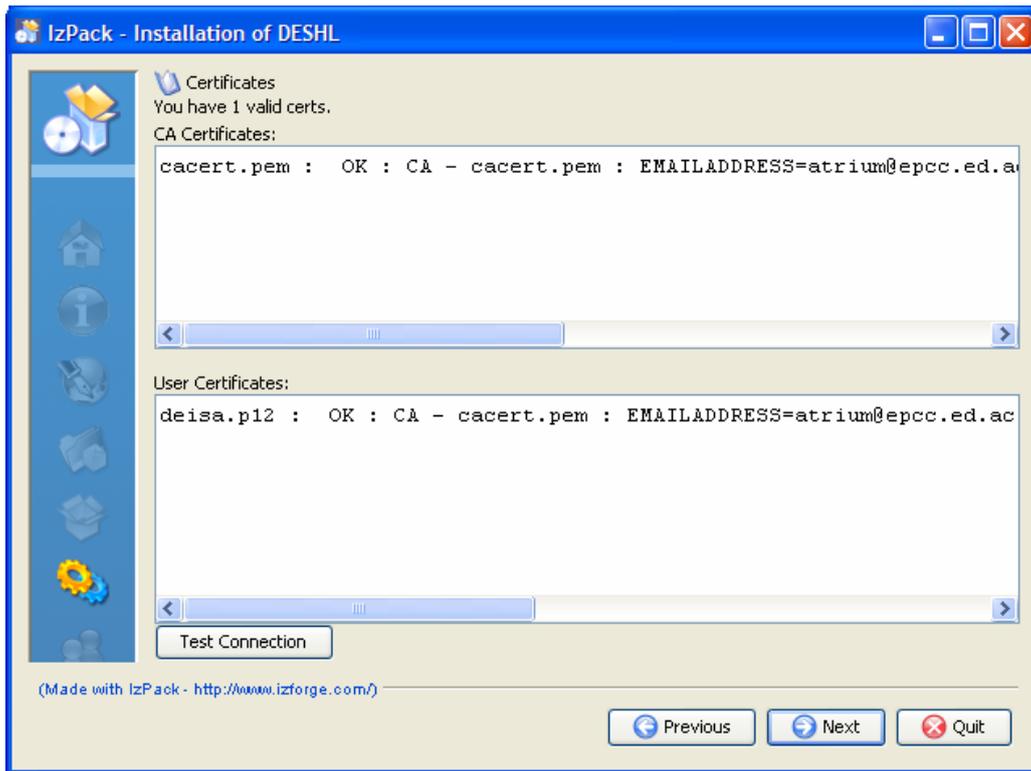


Figure 3: Installation on a Windows system

If password was left blank in the previous screen then the installer will request the password so that the credentials can be validated – this password is NOT saved in the configuration. (If a password is not entered here, the installer will report the user certificate as invalid as it will not be able to unlock it without the password.) (Do not use the “Test Connection” button; this will be removed in a future version.)

Using this information, the installer installs all selected materials into the chosen directory. It also sets up the DESHL configuration file ‘config.csv’ with the details given for the NJS URI. (This configuration file can be manually edited after installation to change the settings, but this should not be necessary for the exercises.)

Once complete all the necessary files are installed and configured to allow the DESHL to be used with the chosen NJS.

The installer does not add the DESHL executable to the path – add this manually as follows:

In UNIX, from the DESHL install directory:

```
$ export PATH=$PWD/bin:$PATH
```

In Windows, if DESHL install directory is *C:\deshl*:

```
C:\deshl>set PATH=C:\deshl\bin;%path%
```

8.2.3 Verify the settings

A simple test to verify that the DESHL client is correctly installed and configured is to try to list the contents of the home storage at the configured site. For example:

```
$ deshl list ssl://norma2.idris.fr:4433/IDRIS%20ZAHIR/home/
```

Or if using shortcuts:

```
$ deshl list idris/home
```

For the site, the DESHL client should display a list of the files and directories in the top level of the HOME storage. (The HOME storage is the site-specific home directory for the user defined in the UNICORE incarnation database.)

8.2.4 Help System

The DESHL client contains a help option, `-h` or `--help`. This can be used to display the set of commands which are currently supported, and the options for a specific command.

```
$ deshl -h

[-h] command [command-options]
Where:
  -h,--help  Print this message.

With commands:
  fetch
  copy
  isDir
  move
  status
  jobs
  remove
  makeDir
  list
  submit
  sites
  terminate
  isFile
  exists

Run deshl <command> -h for individual command options.
```

8.3 Using the DESHL client

The following exercises assume that a shortcut to the configured site was specified during installation. For purposes of illustration, this will be assumed to be "idris", however you should substitute your own shortcut in the examples.

8.3.1 Upload, submit and manage a simple job

Create a new directory on your local workstation and call it exercises

Go into this directory, and verify that the *deshl* executable has been correctly added to your path by running *deshl -h* from the command prompt.

In your exercises directory, create files called *hello-submit.sh* and *hello.sh*, and add the contents to the respective files as shown below:

```
hello-submit.sh

#!/bin/bash
# Test job script for DESHL using SAGA.
#
# SAGA JobDefinition based directives:
#$ SAGA_FileTransfer = file:///jobs/hello.sh#HOME > hello.sh
#$ SAGA_HostList = idris
#$ SAGA_JobCmd = hello.sh
```

```
hello.sh

#!/bin/bash

echo "Here I am on `hostname` in $PWD. Current date is `date`"
```

Please note that you must edit *hello-submit.sh* to specify the execution site:

```
#$ SAGA_HostList = idris
```

The *hello-submit.sh* file contains a set of SAGA directives for submitting and managing the execution of a job, the job in this case being the executable script, *hello.sh*.

The rest of this exercise will show you how to import the executable script to a DEISA site, and then use the SAGA script with the DESHL command line tool to submit the job and retrieve the output from the job.

The following steps will be performed:

1. Verify if a remote directory exists
2. Create a remote directory for the executable script
3. Upload the executable script to the
4. Edit the SAGA script
5. Submit the job
6. Get the job status
7. Cleanup the job and retrieve the job output

1. Verify if a remote directory exists

First we want to determine if the remote directory exists:

```
$ deshl exists idris/home/jobs
```

If the remote directory exists, the following message should be displayed:

```
exists: target idris/home/jobs exists
```

Otherwise, the following message is displayed:

```
exists: target idris/home/jobs does not exist
```

2. Create a remote directory for the script

If the remote directory does not exist, create it using the *deshl makeDir* command:

```
$ deshl makeDir idris/home/jobs
```

Verify that the remote directory now exists by listing the contents of its parent directory. This should show the jobs directory in the listing.

```
$ deshl list idris/home
...
drwx   32768   Feb 07 2007 01:22:01   jobs
...
```

3. Upload the executable script to the server

You now must upload the executable script, *hello.sh*, from your local machine to the DEISA site where the job will be run. To do this, use the *deshl copy* command:

```
$ deshl copy hello.sh idris/home/jobs/hello.sh
```

4. Make the script executable on the server.

Executable files uploaded using DESHL lose their execute permission; a further step is therefore required to make the file executable. (It is expected that most executables will already exist on the DEISA file system and so this step will not be encountered often.)

Create a file called *chmod.sh* in your *exercises* directory and add the contents shown below. (Again, you will need to change the value for *SAGA_HostList* to match the site to which you wish to submit jobs.)

```
chmod.sh

#!/bin/bash
#$ SAGA_FileTransfer = file:///bin/chmod#ROOT > chmod.sh
#$ SAGA_HostList = idris
#$ SAGA_JobCmd = chmod.sh
```

The absolute path of the copied file on the remote site must be known.

It is also possible to specify the path in terms of environment variables on the remote system. For example, submitting from a Windows workstation:

```
$ deshl submit chmod.sh u+x $HOME/jobs/hello.sh
```

Submitting from a UNIX workstation (please note that the `\` character is required for the path to be evaluated correctly):

```
$ deshl submit chmod.sh u+x \"$HOME/jobs/hello.sh
```

The job submission will return a job identifier similar to "idris%2F957383131".

The status of the job can be checked using *deshl status*:

```
deshl status idris%2F957383131
Job: idris%2F2127398328, has status: Running
```

Once the job has completed, the uploaded executable script should have execute permissions. This can be confirmed by listing the file properties:

```
$ deshl list idris/home/jobs/hello.sh
-rwx      88      Feb 07 2007 01:22:01      hello.sh
```

When *deshl* jobs have completed running, the resources associated with the job such as output to *stdout* and *stderr* remain on the remote server until explicitly fetched to your local workstation. When a job has been fetched, resources used by the job will be freed and the job identifier will become invalid. Subsequent attempts to get information about the job after it has been fetched will fail.

To free resources used by the `chmod` job, fetch the job using the *deshl fetch* command:

```
$ deshl fetch idris%2F957383131
```

5. Submit the job

Now that your script on the remote site has execute permissions, you can submit a job to run it.

```
$ deshl submit hello-submit.sh
```

This returns a job identifier, which we can use to monitor the job's status.

6. Get the job status

```
$ deshl status idris%2F957383131
Job: idris%2F957383131, has status: Running
```

You can request more details of the status using the `-f` flag:

```
$ deshl status -f idris%2F957383131

Job: idris%2F1875704394, has status: DoneOk
    ssl://norma2.idris.fr:4433/IDRIS%20ZAHIR
    Name: hello.sh
    DoneOk 06/21 06:00:52
    Running 06/21 05:58:38
```

Once the job has successfully completed, the status will be reported as *DoneOK*.

7. Cleanup the job and retrieve the job output

In your *exercises* directory, create a subdirectory called *output*.

Once the job has completed, the output from the job can now be retrieved and any resources consumed by the job released using the `fetch` command. The `fetch` command allows you to retrieve the job output into a specified directory; for this example you will retrieve into the local directory *output* that you have just created.

```
$ deshl fetch -d output idris%2F957383131
```

This will place two files in the specified directory; one of these is the job's output to standard output, and the other is the job's output to standard error. Look at the files in the *output* directory to verify this. Fetching the job also frees any resources used by the job.

8.3.2 Staging in data for a job, staging out results

In this part of the tutorial, you will stage in data required by a job, run the job and then retrieve data produced by the job. The example script in this case will perform a simple concatenate on two specified files and the file containing the joined files will be the final result.

In your *exercises* directory, create files called *catfiles_submit.sh* and *concat.sh*; add the contents as listed below.

```
catfiles-submit.sh

#!/bin/bash
# Test job script for DESHL using SAGA.
#
# SAGA JobDefinition based directives:
#$ SAGA FileTransfer = file:///jobs/concat.sh#HOME > concat.sh
#$ SAGA FileTransfer = file:///tutorial/filea#HOME > filea
#$ SAGA FileTransfer = file:///tutorial/fileb#HOME > fileb
#$ SAGA FileTransfer = file:///tutorial/filec#HOME < filec
#$ SAGA HostList = idris
#$ SAGA JobCmd = concat.sh
```

```
concat.sh

#!/bin/bash

cat filea fileb > filec
```

Again, please note that you will have to change the value of *SAGA_HostList* to specify the site that you are submitting the job to.

Use the *deshl copy* command to import the executable script file, *concat.sh*, to the jobs directory on the remote site and set the execute permissions using *chmod.sh*:

```
$ desh1 copy concat.sh idris/home/jobs/concat.sh
$ desh1 submit chmod.sh u+x $HOME/jobs/concat.sh
$ desh1 list idris/home/jobs/concat.sh
```

As before, check the status of the job and fetch the job once it has completed.

Use the *deshl makeDir* and *deshl copy* commands to create a *tutorial* directory on the remote host, then copy two files from your local machine into the remote *tutorial* directory as *filea* and *fileb*. Confirm that the files have been copied using the *deshl list* command.

For example:

```
$ desh1 makeDir idris/home/tutorial
$ desh1 copy localFileA.dat idris/home/tutorial/filea
$ desh1 copy localFileB.dat idris/home/tutorial/fileb
$ desh1 list idris/home/tutorial
```

You can now submit the job as before:

```
$ desh1 submit catfiles-submit.sh
```

You can monitor the job status using the *deshl status* command. When the job has completed, an output data file called *filec* will have been produced in the remote tutorial directory. Confirm that this file exists by listing its properties:

```
$ desh1 list idris/home/tutorial/filec
-rw-      63      Mar 01 2007 03:29:57      filec
```

Export this file to your local machine using the *deshl copy* command:

```
$ desh1 copy idris/home/tutorial/filec filec.dat
```

Inspect *filec.dat* to confirm it contains the concatenated contents of the two original files. Remember to cleanup the job using the *deshl fetch* command.

Appendix I – Certificate Management

A certificate compliant to the EU Grid PMA minimal requirements is required to use the DESHL. Details of how to obtain this certificate vary from country to country, therefore please contact the appropriate certificate issuing authority in your region for details on how to apply for a certificate. For each site that the user wishes to access, the DESHL client requires both the user's certificate and the issuing authority's root certificate to be made available in a directory which is accessible to the DESHL client at runtime (the directory location is set in the DESHL configuration). Both certificates must be X.509 (v3) compliant (DER encoded as base64 with additional header and footer lines). The user's certificate must also be packaged in a PKCS12 bundle with the same password as for the certificate (this is the same as for UNICORE).

Please also see the DEISA Primer:

https://www.deisa.org/userscorner/primer/access_to_grid.php#2.1.3

The public certificate of the CA (Certificate Authority) that signed a user certificate **MUST** be placed in the same directory as the PKCS12 file holding that user certificate and it **MUST** be a '.pem' file.

If you have an X.509 certificate, with the public key in *usercert.pem* and the private key in *userkey.pem*, you can convert it to the PKCS12 format using the *OpenSSL* (<http://www.openssl.org>) tool:

```
$ openssl pkcs12 -export -in usercert.pem -inkey userkey.pem  
-out user.p12 -certfile "CApublicKey-name "user1"
```

First, you will have to type the passphrase that you chose for your original PEM certificate when you requested it to the RA, then a new passphrase for the PKCS12 keystore. OpenSSL is available also for the Windows operating system, visit: <http://www.openssl.org/related/binaries.html>

You can also run *openssl* under *cygwin* on Windows, see <http://www.cygwin.com/>

It is also possible to export a user certificate and private key pair from a web browser in PKCS12 format.

For example in Internet Explorer from '*Tools/Internet Options...*' select the '*Content*' tab then '*Certificates*' button, choose the personal certificate to export and follow the export wizard selecting the PKCS12 format.

In Firefox choose '*Tools/Options...*' select '*Advanced*' then the '*Security*' tab, press the '*View Certificates*' button, and choose which of '*Your Certificates*' to backup then save as a PKCS12 file.

9. References and Applicable Documents

- [1] DEISA Brochure:
http://www.deisa.org/press/presentations_and_brochures.php
- [2] DEISA D-SA4-4 deliverable: *SA4 Second Annual Activity Report*
- [3] DEISA D-SA4-4.2 deliverable: *SA4 Mid-term Activity Report*
- [4] DEISA D-SA4-5 deliverable: *SA4 Third Annual Activity Report*
- [5] DEISA Primer Documentation: <http://www.deisa.org/userscorner/primer.php>
- [6] DEISA Services for the Heterogeneous Management Layer (DESHL):
<http://www.deisa.org/organisation/heterogeneous.php>
- [7] UNICORE (UNiform Interface to COmputing REsources):
<http://unicore.sourceforge.net/>

10. List of Acronyms and Abbreviations

DCPE	DEISA Common Production Environment
DECI	DEISA Extreme Computing Initiative
DESHL	DEISA Services for the Heterogeneous Management Layer
DTU	DEISA Test User
Execution site	The site where the jobs of a user are executed
Home site	The site where a user usually works, logs in and submits jobs