

SVR ENGINEERING COLLEGE

AYYALURU METTA(V), NANDYAL, KURNOOL DT.

ANDHRA PRADESH – 518502



2018–2019

LABORATORY MANUAL

OF

GRID & CLOUD COMPUTING LABORATORY
(15A05710)

(R-15 REGULATION)

Prepared by

Mr. M. SUBBA REDDY

Asst. Professor For

B. Tech IV YEAR–ISEM.(CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SVR ENGINEERING COLLEGE

(AFFILIATED TO JNTUA ANANTHAPURAM – AICITE - INDIA)

AYYALURUMETTA(V),NANDYAL,KURNOOL DT.ANDRAPRADESH–518502

LABMANUALCONTENT
GRID & CLOUD COMPUTING LABORATORY
(15A05710)

1. Institute Vision & Mission, Department Vision & Mission
2. PO, PEO & PSO Statements.
3. List of Experiments
4. CO-PO Attainment
5. Experiment Code and Outputs

1. Institute Vision & Mission, Department Vision & Mission Institute

Vision:

To produce Competent Engineering Graduates & Managers with a strong base of Technical & Managerial Knowledge and the Complementary Skills needed to be Successful Professional Engineers & Managers.

Institute Mission:

To fulfill the vision by imparting Quality Technical & Management Education to the Aspiring Students, by creating Effective Teaching/Learning Environment and providing State – of the – Art Infrastructure and Resources.

Department Vision:

To produce Industry ready Software Engineers to meet the challenges of 21st Century.

Department Mission:

1. Impart core knowledge and necessary skills in Computer Science and Engineering through innovative teaching and learning methodology.
2. Inculcate critical thinking, ethics, lifelong learning and creativity needed for industry and society.
3. Cultivate the students with all-round competencies, for career, higher education and self-employability.

2. PO, PEO& PSO Statements

PROGRAMME OUTCOMES (POs)

PO-1: Engineering knowledge - Apply the knowledge of mathematics, science, engineering fundamentals of Computer Science& Engineering to solve complex real-life engineering problems related to CSE.

PO-2: Problem analysis - Identify, formulate, review research literature, and analyze complex engineering problems related to CSE and reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3: Design/development of solutions - Design solutions for complex engineering problems related to CSE and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, cultural, societal and environmental considerations.

PO-4: Conduct investigations of complex problems - Use research-based knowledge and research methods, including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

PO-5: Modern tool usage - Select/Create and apply appropriate techniques, resources and modern engineering and IT tools and technologies for rapidly changing computing needs, including prediction and modeling to complex engineering activities, with an understanding of the limitations.

PO-6: The engineer and society - Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the CSE professional engineering practice.

PO-7: Environment and Sustainability - Understand the impact of the CSE professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics - Apply ethical principles and commit to professional ethics and responsibilities and norms of the relevant engineering practices.

PO-9: Individual and team work - Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication - Communicate effectively on complex engineering activities with the engineering community and with the society-at-large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

PO-11: Project management and finance - Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12: Life-long learning - Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological changes.

SVR ENGINEERING COLLEGE

Department:

COMPUTER SCIENCE & ENGINEERING

Course Outcome Attainment - Internal Assessments

Name of the faculty :

M. SUBBA REDDY, V SAMBA SIVA RAO

Academic Year:

2018-19

Branch & Section:

COMPUTER SCIENCE & ENGINEERING

Exam:

15A05710

Course:

Grid & Cloud Computing Laboratory

Semester:

IV-I

Course Outcomes	Internal Lab		Internal Lab	University Exam
15A05710.1	3		3	3
15A05710.2	3		3	3
15A05710.3	3		3	3
15A05710.4	3		3	3
15A05710.5	3		3	3

Course Outcomes		Attainment Level
15A05710.1	Design and Implement applications on the Cloud.	3
15A05710.2	Design and implement applications on the Grid.	3
15A05710.3	Use the grid and cloud tool kits.	3
15A05710.4	Be familiar with developing web services/Applications in grid framework	3
15A05710.5	use tool kits for grid and cloud environment	3
Average Attainment		3

Overall Course Attainment

3

SVR ENGINEERING COLLEGE

DEPARTMENT		COMPUTER SCIENCE & ENGINEERING				
PROGRAM OUTCOME ATTAINMENT						
Name of Faculty:		M. SUBBA REDDY, V SAMBA SIVA RAO			Academic Year	2018-19
Branch & Section:		COMPUTER SCIENCE & ENGINEERING			SUB CODE:	15A05710
Course:		Grid & Cloud Computing Laboratory			Semester:	IV-I

COURSE OUTCOME ATTAINMENT

Course outcome attainment	Internal lab		Internal lab	External lab
15A05710.1	3		3	3
15A05710.2	3		3	3
15A05710.3	3		3	3
15A05710.4	3		3	3
15A05710.5	3		3	3

COURSE OUTCOMES AND PROGRAM OUTCOMES MAPPING

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
15A05710.1	3	3	2	2	1	2			1		1	2	3	1
15A05710.2	3	2	1	1		1		1				1	2	2
15A05710.3	3	3	2	2	2	1	1			1			2	1
15A05710.4	3	3	1	2		2			2		2	2	2	2
15A05710.5	3	2	2	2		2		2		1			3	1
AVERAGE	3.0	2.6	1.6	1.8	1.5	1.6	1.0	1.5	1.5	1.0	1.5	1.7	2.4	1.4

PO- ATTAINMENT

		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
INTERNAL	15A05710.1	9	9	6	6	3	6			3		3	6	9	3
	15A05710.2	9	6	3	3		3		3				3	6	6
	15A05710.3	9	9	6	6	6	3	3			3			6	3
	15A05710.4	9	9	3	6		6			6			6	6	6

INTE RNA L	15A0571 0.5	9	6	6	6		6		6		3			9	3	
	UNIVERSITY	15A0571 0.1	9	9	6	6	3	6			3		3	6	9	3
		15A0571 0.2	9	6	3	3		3		3				3	6	6
		15A0571 0.3	9	9	6	6	6	3	3			3			6	3
		15A0571 0.4	9	9	3	6		6			6		6	6	6	6
		15A0571 0.5	9	6	6	6		6		6		3			9	3
		OVERALL	15A0571 0.1	3	3	3	3	3	3			3		3	3	3
15A0571 0.2	3		3	3	3		3		3				3	3	3	
15A0571 0.3	3		3	3	3	3	3	3			3			3	3	
15A0571 0.4	3		3	3	3		3			3		3	3	3	3	
15A0571 0.5	3		3	3	3		3		3		3			3	3	
Attainment	3		3	3	3	3	3	3	3	3	3	3	3	3	3	3

Faculty: M. SUBBA REDDY

Head of the Department

LIST OF EXPERIMENTS (SYLLABUS)

(15A05701) GRID AND CLOUD COMPUTING LABORATORY (Common to CSE&IT)

(15A05710) GRID AND CLOUD COMPUTING LABORATORY

Course Objectives:

- The student should be made to:
- Be familiar with developing web services /Applications in grid framework.
- Be exposed to tool kits for grid and cloud environment.
- Learn to use Hadoop
- Learn to run virtual machines of different configuration.

Course Outcomes:

- The student should be able to
- Design and Implement applications on the Cloud.
- Design and implement applications on the Grid.
- Use the grid and cloud tool kits.

GRID COMPUTING PROGRAMS USING GRID SIM

Installations Steps :-

Requirements:

1. JDK
2. Grid Sim Toolkit

Steps:

4. Install JDK toolkit
5. Set path for JDK toolkit
Path=C:/jdk1.8/binClasspath=C:/jdk1.8/jre/lib/rt.jar;
6. Download Grid Sim 5.2
7. Extract Grid Sim in to one folder.
8. Set path=C:/gridsim/bin;
9. Set Class path=C:/gridsim/jar/*;
10. Set Class path=C:/gridsim/examples;
11. Set variable GridSim=C:/gridsim

GRID COMPUTING PROGRAMS USING GRIDSIM

- 1 Program to creates one Grid resource with three machines
- 2 Program to to create one or more Grid users. A Grid user contains one or more Gridlets
- 3 Program to shows how two GridSim entities interact with each other ; main(ie example3) class creates Gridlets and sends them to the other GridSim entities, i.e. Test class
- 4 Program shows how a grid user submits its Gridlets or tasks to one grid resource entity
- 5 Program to show how a grid user submits its Gridlets or task to many grid resource entities
- 6 Program to show how to create one or more grid users and submits its Gridlets or task to many grid resource entities
- 7 Program to creates one Grid resource with three machines

Grid computing programs using Use Globus Toolkit or equivalent:

- 1 Develop a new Web Service for Calculator.
- 2 Develop new OGSA-compliant Web Service.
- 3 Using Apache Axis develop a Grid Service.
- 4 Develop applications using Java or C/C++ Grid APIs
- 5 Develop secured applications using basic security mechanisms available in Globus Toolkit.

Develop a Grid portal, where user can submit a job and get the result. Implement it with and without GRAM concept.

CLOUD COMPUTING

Programs on SaaS

- 1 Create an word document of your class time table and store locally and on the cloud with doc,and pdf format . (use www.zoho.com and docs.google.com)
- 2 Create a spread sheet which contains employee salary information and calculate gross and total sal using the formula
DA=10% OF BASIC HRA=30% OF BASIC
PF=10% OF BASIC IF BASIC<=3000 12% OF BASIC IF BASIC>3000 TAX=10% OF BASIC IF BASIC<=1500
=11% OF BASIC IF BASIC>1500 AND BASIC<=2500
=12% OF BASIC IF BASIC>2500
(use www.zoho.com and docs.google.com)
NET_SALARY=BASIC_SALARY+DA+HRA-PF-TAX
- 3 Prepare a ppt on cloud computing -introduction , models, services ,andarchitecture
Ppt should contain explanations, images and at least 20 pages
(use www.zoho.com and docs.google.com)
- 4 Create your resume in a neat format using google and zoho cloud

Programs on PaaS

- 1 Write a Google app engine program to generate n even numbers and deploy it to google cloud
- 2 Google app engine program multiply two matrices
- 3 Google app engine program to validate user ; create a database login(username, password) in mysql and deploy to cloud
- 4 Write a Google app engine program to display nth largest no from the given list of numbers and deploy it into google cloud
- 5 Google app engine program to validate the user
Use mysql to store user info and deploy on to the cloud

Implement Prog 1-5 using Microsoft Azure

CASE STUDY- cloud computing

Sr. No.	Title of Experiment	Aim of the Experiment	Demonstration Equipments/ Components to be required	Type of Experiment/ Demonstration (Lab/Classroom)
1	Case Study of Amazon	To understand the services of Amazon elastic cloud.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
2	Case Study of Azure	To understand the services of Microsoft azure.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
3	Case Study of Hadoop	To understand the services of hadoop.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
4	Case Study of Aneka	To understand the services of aneka elastic cloud.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
5	Case Study of Google Apps	To understand the services of google apps engine.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
6	Google apps business solution for data access and data upload	To understand the business solution application of Google apps.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.
7	Control panel software manager Application of hypervisors	To understand the application of hypervisors.	Computers with Internet Connection	Experiment: Student perform practical under supervision of faculty and Lab technician.

GRID COMPUTING PROGRAMS USING GRIDSIM

1. Program to create one Grid resource with three machines

```
import
java.util.Calendar; im
port
java.util.LinkedList;

import gridsim.*;

class Example1
{
    public static void main (String [] args)
    {
        System.out.println ("Starting example of how to create one Grid"+"resource");

        try
        {
            int num_user=0;
            Calendar calendar =
            Calendar.getInstance (); boolean tr
            ace_flag=true;

            String [] exclude_from_file = {
            ""}; String [] exclude_from_proce
            ssing={" "};

            String report_name=null;
```

```

        System.out.println("Initializing GridSim
package");GridSim.init(num_user,calendar,trace_fla
g,exclude_from_file,
        exclude_from_processing,report_name);

GridSim.init(num_user,calendar,trace_flag);

GridResourcegridResource=createGridRes
ource();System.out.println("Finishthel
stexample");

    }
catch (Exceptione)
{
    e.printStackTrace();System.out.print
ln("Unwanted errorhappens");
}
}

privatestaticGridResourcecreateGridResource()
{
    System.out.println("Starting tocreateoneGridresourcewith"+"3 Machines...");

    MachineList mList = new
    MachineList();System.out.println("
CreatesaMachinelist");

    intmipsRating=377;

```

```

mList.add( new Machine(0, 4, mipsRating)); // First
MachineSystem.out.println("Create the 1st Machine that has 4
PEs and " + stores it into the
        Machine list");
mList.add( new Machine(1, 4, mipsRating)); // Second
MachineSystem.out.println("Create the 2nd Machine that has
4 PEs and " +
        "stores it into the Machine list");
mList.add( new Machine(2, 2, mipsRating)); // Third
MachineSystem.out.println("Create the 3rd Machine that has
2 PEs and " +
        "stores it into the Machine list");

String arch="SunUltra";
        //system architectureString os="Solaris";
        //operating system
double time_zone=9.0;
        //time zone of this resource location
double cost=3.0;
        //the cost of using this resource

ResourceCharacteristics resConfig=new ResourceCharacteristics(arch, os, mList,
        ResourceCharacteristics.TIME_SHARED, time_zone, cost);

System.out.println();

```

```
System.out.println("Create the properties of a Grid resource and store the machine list");
```

```
String name = "Resource_0"; // resource
```

```
namedouble baud_rate = 100.0;
```

```
        // communication speed
```

```
ed
```

```

longseed=11L*13*17*19*23+1;

doublepeakLoad=0.0; // the resource load during
peak hourdoubleoffPeakLoad=0.0;

        //theresourceloadduringoff-
peakhrdoubleholidayLoad=0.0;

        //theresourceloadduring
holiday

LinkedList<Integer>Weekends=newLinkedList<Integer>()
;Weekends.add(new
Integer(Calendar.SATURDAY));Weekends.add(newInteger(
Calendar.SUNDAY));

// incorporates holidays. However, no holidays are
set in this
exampleLinkedList<Integer>Holidays=newLinkedList<Int
eger>();

GridResourcegridRes=
null;try
{
    gridRes = new GridResource(name, baud_rate,
        seed,resConfig,peakLoad,offPeakLoad,holidayL
        oad,Weekends,Holidays);
}
catch (Exception
    e)
    {e.printStackTrace()
    };

```

```
}
```

```
System.out.println("Finally, createsoneGridresourcean  
dstores"+"thepropertiesofaGridresource");
```

```
        returngridRes;  
    }  
}
```

Output:

```
E:\GridSim\examples\gridsim\example01>java Example1  
Starting example of how to create one Grid resource  
Initializing GridSim package  
Initialising...  
Starting to create one Grid resource with 3 Machines ...  
Creates a Machine list  
Creates the 1st Machine that has 4 PEs and stores it into the Machine list  
Creates the 2nd Machine that has 4 PEs and stores it into the Machine list  
Creates the 3rd Machine that has 2 PEs and stores it into the Machine list  
  
Creates the properties of a Grid resource and stores the Machine list  
Finally, creates one Grid resource and stores the properties of a Grid resource
```

2. Program to create one or more Grid users. A Grid user contains one or more Gridlets.

```
import
```

```
java.util.*;i
```

```
import gridsim.
```

```
*/;
```

```
class Example2
```

```
{
```

```
    public static void main (String [] args)
```

```
    {
```

```
        System.out.println ("Starting example of how to create Grid users");  
        System.out.println ();
```

```
    }  
}
```

```

{
    GridletList list =
        createGridlet();System.out.println("Creat
        ing"+list.size()+"Gridlets");ResourceUser
        ListuserList=createGridUser(list);
        System.out.println("Creating"+userList.size()+"Gridus
        ers");printGridletList(list);

        System.out.println("Finishtheexample");
    }
    catch (Exception e)
    {
        e.printStackTrace();System.out.prin
        tln("Unwanted errorhappens");
    }
}
privatestaticGridletListcreateGridlet()
{
    GridletList list = new
    GridletList();intid=0;
    doublelength=3500
    .0;long file_size
    =
    300;longoutput_si
    ze=300;
    Gridletgridlet1=newGridlet(id,length,file_size,out
    put_size);id++;
    Gridletgridlet2=newGridlet(id,5000,50
    0,500);id++;

```

```

Gridletgridlet3=newGridlet(id,9000,90
0,900);list.add(gridlet1);
list.add(gri
dlet2);list.
add(gridlet3
);
Random random = new
Random();GridSimStandardPE
.setRating(100);intcount=5
;
double min_range =
0.10;double max_range
= 0.50;for(inti=1;
i<count+1;i++)
{
length=GridSimStandardPE.toMIs(random.nextDouble()*ou
tput_size);file_size=(long)GridSimRandom.real(100,min
_range,max_range,
random.nextDouble());
output_size=(long)GridSimRandom.real(250,min_range,ma
x_range,random.nextDouble());
Gridletgridlet=newGridlet(id+i,length,fil
e_size,output_size);
list.add(gridlet);
}

returnlist;
}
privatestaticResourceUserListcreateGridUser(GridletListlist)

```

{

```

ResourceUserList userList = new ResourceUserList();

userList.add(0); //userID starts from 0
userList.add(1);
userList.add(2);

int userSize = userList
.size(); int
gridletSize =
list.size(); int id = 0;
for (int i = 0; i < gridletSize; i++)
{
    if (i != 0 && i
        % userSize == 0) id++;

    ((Gridlet) list.get(i)).setUserID(id);
}

return userList;
}

private static void printGridletList (GridletList list)
{
    int size = list.
size(); Gridle
t gridlet;

```

```

String indent =
"";System.out.p
rintln();
System.out.println("GridletID"+indent+"UserID"+ind
ent+"length"+indent+" filesize"+indent+
"outputsize");

for(inti =0;i <size;i++)
{
gridlet=(Gridlet)list.get(i);
System.out.println(indent
+
gridlet.getGridletID() + indent +indent +
indent + gridlet.getUserID() + indent +
indent +(int)gridlet.getGridletLength()
+indent+indent+
(int) gridlet.getGridletFileSize() +
indent + indent
+(int)gridlet.getGridletOutputSize())
;
}
}
} // endclass

```

3. Program to show how two GridSim entities interact with each other; main (ie example3) class creates Gridlets and sends them to the other GridSim entities, i.e. Testclass

```

import
java.util.*;i
mport gridsim.

```

```
*;
```

```
classExample3extendsGridSim
```

```
{
```

```

privateStringentityName;
privateGridletListlist_;
privateGridletListreceiveList_;
Example3(Stringname,doublebaud_rate,GridletListlist)throwsException
{
    super(name
    );this.list
    _=list;
    receiveList_ = new
    GridletList();entityName
    e_="Test";
    newTest(entityName_,baud_rate);
}
publicvoidbody()
{
    intsize=list_.
    size();Gridlet
    obj,gridlet;
    for(inti =0;i <size;i++)
    {
        obj=(Gridlet)list_.get(i);
        System.out.println("InsideExample3.body()=>SendingGridlet"+obj.getGridletID());
        super.send(entityName_,
        GridSimTags.SCHEDULE_NOW,GridSimTags.GRIDLET_SUBMIT,obj);
        gridlet=super.gridletReceive();
        System.out.println("InsideExample3.body()=>ReceivingGridlet");
    }
}

```

```
ridlet"+gridlet.getGridletID());
```

```

        receiveList_.add(gridlet);
    }

    super.send(entityName_,
        GridSimTags.SCHEDULE_NOW,GridSimTags.END_OF_SIMULA
        TION);
}

publicGridletListgetGridl
    etList(){returnreceiveL
        ist_;
    }

publicstaticvoidmain(String[]args)
{
    System.out.println("Starting
    Example3");System.out.println(
    );

    try
    {
        intnum_user=0;//numberofusersneedtobecreatedCale
        ndarcalendar=Calendar.getInstance();
        booleantrace_flag=true;//meantraceGridSim
        eventsString[]exclude_from_file={" "};
        String[]exclude_from_processing={
        " "};Stringreport_name=null;
        System.out.println("Initializing GridSim
        package");GridSim.init(num_user,calendar,trace_flag,e
        xclude_from_file,
            exclude_from_processing,
            report_name);GridletListlist=creat

```

```
eGridlet();
```

```

        System.out.println("Creating " + list.size() + "
        Gridlets");Example3obj=newExample3("Example3",56
        0.00,list);GridSim.startGridSimulation();
        GridletListnewList=obj.getGridletLi
        st();printGridletList(newList);Syst
        em.out.println("FinishExample3");
    }
    catch (Exceptione)
    {
        e.printStackTrace();System.out.prin
        tln("Unwantederrorshappen");
    }
}
privatestaticGridletListcreateGridlet()
{
    GridletList list = new
    GridletList();intid=0;
    doublelength=3500
    .0;long file_size
    =
    300;longoutput_si
    ze=300;
    Gridletgridlet1=newGridlet(id,length,file_size,out
    put_size);id++;
    Gridletgridlet2=newGridlet(id,5000,50
    0,500);id++;
    Gridletgridlet3=newGridlet(id,9000,900,900);

```

```

list.add(gridlet1);list.add(gridlet2);list.add(gridlet3);
long seed = 11L*13*17*19*23+1;Randomrandom=newRandom(seed);GridSimStandardPE.setRating(100);
intcount=5;
for(inti=1; i<count+1;i++)
{
length = GridSimStandardPE.toMIs(random.nextDouble()*50);file_size=(long)GridSimRandom.real(100,0.10,0.40,random.nextDouble());
output_size=(long)GridSimRandom.real(250,0.10,0.50,random.nextDouble());
Gridletgridlet=newGridlet(id+i,length,file_size,output_size);
list.add(gridlet);
}

returnlist;
}
privatestaticvoidprintGridletList(GridletListlist)
{
int size =

```

```
list.size();Gr  
idletgridlet;
```

```

String indent =
"";System.out.p
rintln();
System.out.println("===== OUTPUT
=====");System.out.println("GridletID
"+indent+"STATUS");

for(inti =0;i <size;i++)
{
gridlet=(Gridlet)list.get(i);
System.out.print(indent+gridlet.getGridletID()+indent
+indent);

if (gridlet.getGridletStatus() ==
Gridlet.SUCCESS)System.out.println
("SUCCESS");
}
}
}

```

4. Programshowshowagriduser submitsitsGridletsortaskstoonegridresourceentity

```

import
java.util.*;i
mportgridsim.
*;
classExample6extendsGridSim
{
private Integer
ID_;private
String
name_;privateGrid
letListlist_;
privateGridletListrecei
veList_;privateinttotal
Resource_;

Example6(Stringname,doublebaud_rate,inttotal_re
source)throwsException
{
super(name,
baud_rate);this.na
me_=name;
this.totalResource_=
total_resource;this.receiveList_
= new
GridletList();this.ID_=newIntege
r(getEntityId(name));
System.out.println("Creatingagriduserentitywithnam
e="+name+",andid="+this.ID_);
this.list_= createGridlet(

```

```
this.ID_.intValue()  
);System.out.println(name+":Creating"+this.  
list_.size()+  
    "Gridlets");  
}  
publicvoidbody()
```

```

{
    int resourceID[] = new int[this.totalResource_];
    double resourceCost[] = new
    double[this.totalResource_]; String resourceN
    ame[] = new String[this.totalResource_];

    LinkedList
    resList; ResourceCharacte
    ristics
    resChar; while (true)
    {
        super.gridSimHold(1.0); //hold by 1sec
        ondresList = super.getGridResourceLis
        t();
        if (resList.size() ==
            this.totalResource_) break;
        else
        {
            System.out.println(this.name_
                + ":Waiting
                to get list of resources...");
        }
    }
    inti = 0;
    for (i = 0; i < this.totalResource_; i++)
    {
        resourceID[i] = ( (Integer) resList.get(i)
        ).intValue(); super.send(resourceID[i],
        GridSimTags.SCHEDULE_NOW,

```

```
GridSimTags.RESOURCE_CHARACTERISTICS,this.ID_);
```

```

resChar=(ResourceCharacteristics) super.receiveEventOb
ject();resourceName[i]=resChar.getResourceName();
resourceCost[i]=resChar.getCostPerSec();
System.out.println(this.name_+":ReceivedResourceCharacteristicsf
rom"+resourceName[i]+",withid="+resourceID[i]);
super.recordStatistics("\Received
ResourceCharacteristics "+"from"
+resourceName[i)+"\","");
}
Gridlet
gridlet;St
ringinfo;
intid=0;
for(i=0;i <this.list_.size();i++)
{
gridlet=(Gridlet)this.list_.get(i);
info="Gridlet_"+gridlet.getGridletID();
id =
GridSimRandom.intSample(this.totalResource_);System.o
ut.println(this.name_+":Sending"+info+"to"+
resourceName[id]+"withid="+resourceID
[id]);super.gridletSubmit(gridlet,
resourceID[id]);super.recordStatistics("\
Submit"+info+"to"+
resourceName[id)+"\","
");gridlet=super.gridl
etReceive();
System.out.println(this.name_+":ReceivingGridlet"+gri
dlet.getGridletID());

```

```
super.recordStatistics("\Received"+info+"from"+
```

```

        resourceName[id] + "\",
        gridlet.getProcessingCost());this.receiveLi
        st_.add(gridlet);
    }
    super.shutdownGridStatisticsEntity();s
    uper.shutdownUserEntity();super.termi
    nateIOEntities();

    System.out.println(this.name_+":%%%Exitingbody()");
}

publicGridletListgetGridl
    etList(){returnthis.rec
    eiveList_;
}

privateGridletListcreateGridlet(intuserID)
{
    GridletList list = new
    GridletList();intid=0;
    doublelength=3500
    .0;long file_size
    =
    300;longoutput_si
    ze=300;
    Gridletgridlet1=newGridlet(id,length,file_size,out
    put_size);id++;
    Gridletgridlet2=newGridlet(id,5000,50
    0,500);id++;
    Gridletgridlet3=newGridlet(id,9000,900,900);

    //settingtheowneroftheseGridlets

```

```

gridlet1.setUserID(u
serID);gridlet2.setU
serID(userID);gridle
t3.setUserID(userID)
;

//StoretheGridletsinto
alistlist.add(gridlet1
);list.add(gridlet2);l
ist.add(gridlet3);
GridSimStandardPE.setRating(100);

// creates 5
Gridletsintmax=
5;
intcount=GridSimRandom.intSample(
max);for(inti =1;i <count+1;i++)
{
length =
GridSimStandardPE.toMIs(GridSimRandom.doubleSample()*50)
;file_size=(long)GridSimRandom.real(100,0.10,0.40,
GridSimRandom.doubleSample());outp
ut_size=(long)GridSimRandom.real(250,0.10,0.50,
GridSimRandom.doubleSample());
Gridletgridlet=newGridlet(id+i,length
,file_size,
output_si
ze);gridlet.setUser
ID(userID);list.add

```

```
(gridlet);  
}
```

```

returnlist;
}
publicstaticvoidmain(String[]args)
{
System.out.println("StartingExample6");

try
{
int num_user = 3; // number of
grid
usersCalendarcalendar=Calendar.g
etInstance();
booleantrace_flag=false;//meandon'ttraceGridSime
ventsString[]exclude_from_file={" "};
String[]exclude_from_processing={
" "};Stringreport_name=null;
GridSim.init(num_user, calendar, trace_flag,
exclude_from_file,exclude_from_processing,re
port_name);
GridResourceresource0=createGridResource("Reso
urce_0");GridResourceresource1=createGridResou
rce("Resource_1");GridResourceresource2=create
GridResource("Resource_2");inttotal_resource=3
;
Example6 user0 = new Example6("User_0", 560.00,
total_resource);Example6 user1 = new
Example6("User_1", 250.00,
total_resource);Example6 user2 = new

```

```
Example6("User_2", 150.00,  
total_resource);GridSim.startGridSimulation();
```

```

GridletList newList =
null;newList=user0.getGr
idletList();
printGridletList(newList,"User_0");

newList =
user1.getGridletList();printGridlet
List(newList,"User_1");

newList =
user2.getGridletList();printGridlet
List(newList,"User_2");

System.out.println("FinishExample6");
}
catch (Exception e)
{
e.printStackTrace();System.out.prin
tln("Unwantederrorshappen");
}
}
private static GridResource createGridResource (String name)
{
MachineList mList=new MachineList
t();int mipsRating =377;
mList.add( new Machine(0, 4, mipsRating)); //
First Machine mList.add( new Machine(1, 4,
mipsRating)); // Second
Machine mList.add(new Machine(2,2,mipsRating)); //

```

ThirdMachine

```

Stringarch="SunUltra";          //
system
architectureStringos="Solaris";
                                //operating system
doubletime_zone=9.0;
                                //timezonethisresourcelocat
doublecost=3.0; //thecostofusing
thisresource

ResourceCharacteristicsresConfig=newResourceCharac
    teristics(arch, os, mList,
    ResourceCharacteristics.TIME_SHARED,time_zone,
    cost);

// 5. Finally, we need to create a
GridResource object.doublebaud_rate=100.0;
                                // communication
speedlongseed=11L*13*17*19*23+1;
doublepeakLoad=0.0; // the resource load during
peak hourdouble offPeakLoad = 0.0; // the
resource load during off-peak hrdouble
holidayLoad = 0.0; // the resource load during
holidayLinkedListWeekends=newLinkedList();
Weekends.add(newInteger(Calendar.SATUR
DAY));Weekends.add(new
Integer(Calendar.SUNDAY));LinkedList
Holidays = new
LinkedList();GridResourcegridRes=null;
try{

```

```
gridRes = new GridResource(name, baud_rate,  
    seed,resConfig,peakLoad,offPeakLoad,holidayL  
    oad,Weekends,Holidays);  
}
```

```

catch (Exception
    e)
    {e.printStackTrace()
    }

System.out.println("CreatesoneGridresourcewithname="+name);returngridRes;
}
privatestaticvoidprintGridletList (GridletListlist,Stringname)
{
    int size =
    list.size();Gridletgridlet;

    String indent =
    "";System.out.println();
    System.out.println("=====OUTPUTfor"+name+"=====
    ==");System.out.println("GridletID"+indent+"STATUS"+indent+
    "ResourceID"+indent+"Cost");

    for(inti =0;i <size;i++)
    {
        gridlet=(Gridlet)list.get(i);
        System.out.print(indent+gridlet.getGridletID()+indent
        +indent);
    }
}

```

```
if (gridlet.getGridletStatus () == Gridlet.SUCCESS)
```

```

        System.out.print("SUCCESS");

        System.out.println(indent+indent+gridlet.getResourceID()+indent+indent+gridlet.getProcessingCost());
    }
}

} //endclass

```

5. Program to show how a grid user submits its Gridlets or task to many grid resource entities

```

import
java.util.*;
import gridsim.*;
public class
Test
{
    private static final int MIN = 1; // min number
    of test
    cases private static final int MAX = 8; // max number of test
    cases

    /**
     * Usage in Unix/ Linux:
     *   javac -classpath $GRIDSIM/gridsim.jar:. Test.java
     *   java Test [policy:space|time] [testcase number:1-8]
     *
     * For example: java Test space 7 --> runningSpace-Shared for test case #7
     *               java Test time 3 --> runningTime-Shared for test case #3
     *
     * The operation of these Test Cases offer are:
     * Test Case 1: Submit Gridlets - then wait until all finish to collect
     * Test Case 2: Submit Gridlets - Cancel some of them - Finish
     * Test Case 3: Submit Gridlets - Pause some of them - Cancel - Finish
     * Test Case 4: Submit Gridlets - Pause - Resume - Cancel - Finish
     * Test Case 5: Submit Gridlets - Move some of them - Finish
     * Test Case 6: Submit Gridlets - Pause - Move - Finish
     * Test Case 7: Submit Gridlets - Pause - Resume - Move - Finish

```

* TestCase8:SubmitGridlets-Pause-Resume -Move-Cancel-Finish

```

*
* NOTE:
* -TestCase1isthesimplestandTestCase8isthemostcomplicated.
*
* -TheseTestCasesarequiteflexible,meaning,youcanadjusthowbig
* theseexperimentsarebyincreasing/decreasingtotalUser,totalPE,etc
* frommain()only.Youdon'tneedtomodify
* anyoftheTestCasesclasses.
*
* -Becarefulwhensettingthenumberstoohigh(above200)
* sinceyoumightgetJava"OutOfMemory"exception.
*
* -ForeffectiveexperimentforGridletorJobmigration,youneedto
* havealargenumberofGridResourceentities,saymorethan6.
*/
publicstaticvoidmain(String[]args)
{
    System.out.println("Starting
    Test Cases");try
    {
        // Parse the command
        line argsintpolicy=0;
        if ( args[0].equals("t") ||
            args[0].equals("time") )
            {policy=ResourceCharacteristics.TIME
            _SHARED;
            }
        else if ( args[0].equals("s") ||
            args[0].equals("space") )
            {policy=ResourceCharacteristics.SPACE_S
            HARED;
            }
        else {
            System.out.println("Error -- Invalid
            allocation policy... ");return;
        }

        //determinewhichtestcasenumbertocho
        oseinttestNum=Integer.parseInt(args
        [1]);
        if(testNum <MIN||testNum
            >MAX){testNum=MIN;
        }

        //////////////////////////////////////
        //Firststep:InitializetheGridSimppackage.Itshouldbecalled

```

```

//beforecreating anyentities. Wecan'trunthisexamplewithout
//initializingGridSimfirst. Wewillgetrun-timeexception
//error.
Calendarcalendar=Calendar.getInstance();
booleantrace_flag=false;//truemeanstracingGridSimevents

//listoffilesorprocessingnamestobeexcludedfromany
//statisticalmeasures
String[]exclude_from_file ={
""};String[]exclude_from_proce
ssing={" "};

//thenameofareportfiletobe written. Wedon'twanttowrite
//anything here.
Stringreport_name=null;

//initializeallrevelantvariables
doublebaudRate[]={1000,5000};//bandwidthforeven
,oddintpeRating[]={10,50};// PERating
foreven,odd
double price[] = {3.0, 5.0}; //
resource for even,
oddintgridletLength[]={1000,2000,3000,
4000,5000};

//InitializetheGridSimpackage
int totalUser = 2; // total Users for this
experimentGridSim.init(totalUser,calendar,trace_fl
ag,exclude_from_file,
    exclude_from_processing,report_name);

////////////////////////////////////
//Secondstep:CreatesoneormoreGridResourceobjects
int totalResource = 3; // total GridResources
for this experimentint totalMachine = 1;// total
Machines for each GridResourceinttotalPE=3; //
total PEs for each
MachinecreateResource(totalResource,totalMachine
,totalPE,baudRate,
    peRating,price,policy);

////////////////////////////////////
//Thirdstep:Createsgridusers
inttotalGridlet=4; // total Gridlets for
each
UsercreateUser(totalUser,totalGridlet,gridle
tLength,baudRate,
    testNum);

```

////////////////////////////////////

```

        //Fourthstep:Startsthesimu
        lationGridSim.startGridSim
        ulation();
    }
    catch (Exception e)
    {
        System.out.println("Unwanted errors
        happen");System.out.println( e.getMessage()
        );System.out.println("Usage:javaTest[time|spa
        ce][1-8]");
    }
    System.out.println("=====ENDOFTEST=====");
}

/**
 * Createsmany GridResources
 */
publicstaticvoidcreateResource(inttotalRes,intto
    talMachine,int totalPE, double[]
    baudRate, int[]
    peRating,double[]price,intpolicy)
{
    doublebandwidth=0;
    doublecost=0.0;

    //aloopthatcreatesoneormoreGridResou
    rcesfor(inti =0;i <totalRes;i++)
    {
        Stringname="GridResource_
        "+i;if(i %2==0)
        {
            bandwidth =
            baudRate[0];cost=pr
            ice[0];
        }
        else
        {
            bandwidth =
            baudRate[1];cost=pr
            ice[1];
        }

        //createsaGridResource
        createGridResource(name,totalMachine,totalPE,
            bandwidth,peRating,policy,cost);
    }
}

```

```

}

/**
 * CreatesmanyGridUsers
 */
public static void createUser(int totalUser,
                             int
                             totalGridlet,int []glLength,double [
                             ]baudRate,inttestNum)
{
    try
    {
        doublebandwidth=
        0;doubledelay=0.
        0;

        for(inti=0;i<totalUser;i++)
        {
            String name =
            "User_" + i;if(i
            %2==0){
                bandwidth=baudRate[
                0];delay=5.0;
            }
            else {
                bandwidth=baudRate[1];
            }

            //createsaGriduser
            createTestCase(name, bandwidth, delay,
                           totalGridlet, glLength,testNum);
        }
    }
    catch (Exceptione) {
        //...ignore
    }
}

/**
 * Aselectionofdifferenttestcases
 */
privatestaticvoidcreateTestCase (Stringname,doubleband
                                width,doubledelay,inttotalGridlet,int []glLe
                                ngth,
                                inttestNum)throws Exception
{

```

```
switch (testNum)
{
    case1:
        newTestCase1 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    case2:
        new TestCase2 (name, bandwidth, delay,
        totalGridlet, glLength);break;

    case3:
        newTestCase3 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    case4:
        newTestCase4 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    case5:
        newTestCase5 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    case6:
        newTestCase6 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    case7:
        newTestCase7 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    case8:
        newTestCase8 (name,bandwidth,delay,totalGridlet,g
        lLength);break;

    default:
        System.out.println("Notarecognizedtestc
        ase.");break;
}
}

/**
```

```

* Creates one Grid resource. A Grid resource contains one or more
* Machines. Similarly, a Machine contains one or more PEs (Processing
* Elements or CPUs).
*/
private static void createGridResource (String name, int total
    alMachine, int totalPE, double
    bandwidth, int [] peRating,
    int policy, double cost)
{
    // Here are the steps needed to create a Grid resource:
    // 1. We need to create an object of MachineList to store one or more
    // Machines
    MachineList mList = new MachineList ();

    int rating = 0;
    for (int i = 0; i < totalMachine; i++)
    {
        // Even machines have different PE ratings compared to
        // odd ones if (i % 2 == 0) {
            rating = peRating [0];
        }
        else {
            rating = peRating [1];
        }

        // 2. Create one machine with its id, number of PEs and rating
        mList.add (new Machine (i, totalPE, rating));
    }

    // 3. Create a Resource Characteristics object that stores the
    // properties of a Grid resource: architecture, OS, list of
    // Machines, allocation policy: time- or space-shared, time zone
    // and its price (G$/PE time unit).
    String arch = "SunUltra";
        // system architecture
    String os = "Solaris";
        // operating system
    double time_zone = 0.0; // time zone this resource is located

    ResourceCharacteristics resConfig = new ResourceCharacteristics (arch, os, mList, policy, time_zone, cost);

    // 4. Finally, we need to create a Grid Resource object.
    long seed = 11L * 13 * 17 * 19 * 23 + 1;
}

```

```

double peakLoad = 0.0; // the resource load
during peak hour double offPeakLoad = 0.0; // the
resource load during off-peak
hour double holidayLoad=0.0; // the resource load during h
oliday

// incorporates weekends so the grid resource is
on 7 days a
weekLinkedListWeekends=newLinkedList();
Weekends.add(new Integer(Calendar.SATURDAY));Weekends
.add(new Integer(Calendar.SUNDAY));

// incorporates holidays. However, no holidays are
set in this exampleLinkedListHolidays
=newLinkedList();
try
{
    GridResourcegridRes=newGridResource(name,bandwidth,se
ed,resConfig, peakLoad, offPeakLoad, holidayLoad,
Weekends,Holidays);
}
catch(Exceptione)
{
    System.out.println("ErrorincreatingGridResource.")
;System.out.println(e.getMessage());
}

System.out.println("CreatesoneGridresourcewithname="+n
ame);return;
}

} //endclass

```

6. Program to show how to create one or more grid users and submit its Gridlets to many grid resource entities

```

import
java.util.*;i
mport gridsim.
*;

/**
* This is the example main program that demonstrates how to

```

* submit/cancel /resume/pause/moveGridletstodifferentGridResources.

* Youcanplayaroundwiththisclassbyadjustingfewparametersinmain()

* suchastotalUser,totalGridlet,etc.

*/

```
publicclassTest
```

```
{
```

```
private static final int MIN = 1; // min
```

```
number of test casesprivatestaticfinal
```

```
intMAX=8;//maxnumberoftestcases
```

```
/**
```

```
* UsageinUnix/ Linux:
```

```
* javac-classpath$GRIDSIM/gridsim.jar:.Test.java
```

```
* javaTest[policy:space|time][testcasenumber:1-8]
```

```
*
```

```
* Forexample:javaTestspace7-->runningSpace-Sharedfortestcase#7
```

```
* javaTesttime3-->running Time-Sharedfortestcase#3
```

```
*
```

```
* TheoperationoftheseTestCasesofferare:
```

```
* TestCase1:SubmitGridlets-thenwaituntil allFinishtocollect
```

```
* TestCase2:SubmitGridlets-Cancelsofthem-Finish
```

```
* TestCase3:SubmitGridlets-Pausesomeofthem -Cancel-Finish
```

```
* TestCase4:SubmitGridlets-Pause-Resume-Cancel-Finish
```

```
* TestCase5:SubmitGridlets-Movesomeofthem-Finish
```

```
* TestCase6:SubmitGridlets-Pause-Move-Finish
```

```
* TestCase7:SubmitGridlets-Pause-Resume-Move-Finish
```

* TestCase8:SubmitGridlets-Pause-Resume -Move-Cancel-Finish

*

* NOTE:

* -TestCase1isthesimplestandTestCase8isthemostcomplicated.

*

* -TheseTestCasesarequiteflexible,meaning,youcanadjusthowbig

* theseexperimentsarebyincreasing/decreasingtotalUser,totalPE,etc

* frommain()only.Youdon'tneedtomodify

* anyoftheTestCaseclasses.

*

* -Becarefulwhensettingthenumberstoohigh(above200)

* sinceyoumightgetJava"OutOfMemory"exception.

*

* -ForeaneffectiveexperimentforGridletorJobmigration,youneedto

* havealargenumberofGridResourceentities,saymorethan6.

*/

```
publicstaticvoidmain(String[]args)
```

```
{
```

```
    System.out.println("Starting
```

```
    Test Cases");try
```

```
    {
```

```
        // Parse the command
```

```
        line argsintpolicy=0;
```

```
        if ( args[0].equals("t") ||
```

```
            args[0].equals("time") )
```

```
            {policy=ResourceCharacteristics.TIME_
```

```
            SHARED;
```

```

}

else if ( args[0].equals("s") ||
        args[0].equals("space") )
    {policy=ResourceCharacteristics.SPACE_S
    HARED;
}

else {
    System.out.println("Error--
    Invalidallocationpolicy...");return;
}

//determinewhichtestcasenumbertochosen
inttestNum=Integer.parseInt(args
[1]);
if(testNum<MIN ||testNum
    >MAX){testNum=MIN;
}

////////////////////////////////////
//Firststep:InitializetheGridSimpackage.Itshouldbecalled
//beforecreating anyentities.Wecan'trunthisexamplewithout
//initializingGridSimfirst.Wewillgetrun-timeexception
//error.
Calendarcalendar=Calendar.getInstance();
booleantrace_flag=false;//truemeanstracing GridSimevents

//listoffilesorprocessingnamestobeexcludedfromany
//statisticalmeasures

```

```

String[]exclude_from_file ={
""};String[]exclude_from_proce
ssing={" "};

//thenameofareportfiletobewritten. Wedon'twanttowrite
//anything here.
Stringreport_name=null;

//initializeallrevelantvariables
doublebaudRate[]={1000,5000};//bandwidthforeve
n,oddintpeRating[]={10,50};// PERating
foreven,odd
doubleprice[]={3.0,5.0};//resourcefore
ven,oddintgridletLength[]={1000,2000,3
000,4000,5000};

//InitializetheGridSimpackage
int totalUser = 2; // total Users for this
experimentGridSim.init(totalUser,calendar,trace_flag,
exclude_from_file,
    exclude_from_processing,report_name);

////////////////////////////////////
//Secondstep:CreatesoneormoreGridResourceobjects
int totalResource = 3; // total GridResources
for this experimentint totalMachine = 1;// total
Machines for each GridResourceinttotalPE=3;
    // total PEs for each
MachinecreateResource(totalResource,totalMachine

```

```
,totalPE,baudRate,  
    peRating,price,policy);
```

```

////////////////////////////////////
//Thirdstep:Createsgridusers
inttotalGridlet=4;    // total Gridlets for
each
UsercreateUser(totalUser,totalGridlet,gridle
tLength,baudRate,
    testNum);

////////////////////////////////////
//Fourthstep:Startsthesimu
lationGridSim.startGridSim
ulation();
}
catch (Exceptione)
{
    System.out.println("Unwanted errors
happen");System.out.println( e.getMessage()
);System.out.println("Usage:javaTest[time|s
pace][1-8]");
}
System.out.println("=====ENDOFTEST=====");
}

/**
 * Createsmany GridResources
 */
publicstaticvoidcreateResource(inttotalRes,intto
    talMachine,inttotalPE,double[]baudRate

```

```
,int[]peRating,
```

```

        double[]price,intpolicy)
{
    doublebandwidth=0;dou
blecost=0.0;

    //aloopthatcreatesoneormoreGridResou
rcesfor(inti =0;i <totalRes;i++)
{
    Stringname="GridResource_"
"+i;if(i %2==0)
    {
        bandwidth =
        baudRate[0];cost=pr
        ice[0];
    }
    else
    {
        bandwidth =
        baudRate[1];cost=pr
        ice[1];
    }

    //createsaGridResource
    createGridResource(name, totalMachine,
        totalPE,
        bandwidth,peRating,policy,cost);
}
}

```

```

/**
 * CreatesmanyGridUsers
 */
public static void createUser(int totalUser,
                               int
                               totalGridlet,int []glLength,double[]
                               ]baudRate,inttestNum)
{
    try
    {
        doublebandwidth=
        0;doubledelay=0.
        0;

        for(inti=0;i<totalUser;i++)
        {
            String name =
            "User_" + i;if(i
            %2==0) {
                bandwidth=baudRate[
                0];delay=5.0;
            }
            else {
                bandwidth=baudRate[1];
            }

            //createsaGriduser
            createTestCase(name,bandwidth,delay,totalGridlet,glLength,

```

```

        testNum);
    }
}
catch (Exception e) {
    //...ignore
}
}

/**
 * A selection of different test cases
 */
private static void createTestCase (String name, double ba
        ndwidth, double delay, int totalGridlet, int []
        glLength,
        int testNum) throws Exception
{
    switch (testNum)
    {
        case 1:
            new TestCase1 (name, bandwidth, delay, totalGridlet, gl
                Length); break;

        case 2:
            new TestCase2 (name, bandwidth, delay, totalGridlet, gl
                Length); break;
    }
}

```

```
case3:
    newTestCase3(name,bandwidth,delay,totalGridlet,gl
    Length);break;

case4:
    newTestCase4(name,bandwidth,delay,totalGridlet,gl
    Length);break;

case5:
    newTestCase5(name,bandwidth,delay,totalGridlet,gl
    Length);break;

case6:
    newTestCase6(name,bandwidth,delay,totalGridlet,gl
    Length);break;

case7:
    newTestCase7(name,bandwidth,delay,totalGridlet,gl
    Length);break;

case8:
    newTestCase8(name,bandwidth,delay,totalGridlet,gl
    Length);break;

default:
```

```

        System.out.println("Notarecognizedtestc
        ase.");break;
    }
}

/**
 * CreatesoneGridresource.AGridresourcecontainsoneormore
 * Machines.Similarly,aMachinecontainsoneormorePEs(Processing
 * ElementsorCPUs).
 */
privatestaticvoidcreateGridResource(Stringname,intto
    talMachine,inttotalPE,double
    bandwidth,int[]peRating,
    intpolicy,doublecost)
{
    //HerearethestepsneededtocreataGridresource:
    //1. WeneedtocreateanobjectofMachineListtostoreoneormore
    //Machines
    MachineListmList=newMachineList();

    intrating=0;
    for(inti =0; i <totalMachine;i++)
    {
        //evenMachineshavedifferentPERatingcompareto
        oddonesif(i %2==0){
            rating=peRating[0];

```

```

    }

    else {
        rating=peRating[1];
    }

    //2.CreateoneMachinewithitsid,numberofPEsandratingmLi
    st.add(newMachine(i,totalPE,rating));
}

//3.CreateaResourceCharacteristicsobjectthatstoresthe
//propertiesofaGridresource:architecture,OS,listof
//Machines,allocationpolicy:time-orspace-shared,timezone
//anditsprice (G$/PEtimeunit).
Stringarch="SunUltra";          //
system
architectureStringos="Solaris";
                //operating system
doubletime_zone=0.0; //timezonethisresourcelocated

ResourceCharacteristicsresConfig=newResourceCharac
    teristics(arch,os,mList,policy,time_zone,cost)
    ;

//4.Finally,weneedtocreateaGridResourceo
bject.longseed=11L*13*17*19*23+1;
double peakLoad = 0.0;// the resource load
during peak hourdouble offPeakLoad = 0.0; // the
resource load during off-peak
hrdoubleholidayLoad=0.0;//the

```

resource load during holiday

```

// incorporates weekends so the grid resource is
on 7 days a
weekLinkedListWeekends=newLinkedList();
Weekends.add(newInteger(Calendar.SATURDAY))
;Weekends.add(newInteger(Calendar.SUNDAY));

// incorporates holidays. However, no holidays are
set in this exampleLinkedListHolidays
=newLinkedList();
try
{
    GridResourcegridRes=newGridResource(name,bandwidth,
        seed,resConfig, peakLoad, offPeakLoad,
        holidayLoad, Weekends,Holidays);
}
catch(Exceptione)
{
    System.out.println("ErrorincreatingGrid
Resource.");System.out.println(e.getMes
sage());
}

System.out.println("CreatesoneGridresourcewithname="+na
me);return;
}

} //endclass

```

7. Program to create one Grid resource with three machines

Grid computing programs using Use Globus Toolkit
it is equivalent:

GRID COMPUTING PROGRAMS USING USE GLOBUS TOOLKIT IS EQUIVALENT

1. Develop a new Web Service for Calculator.

OBJECTIVE:

To develop a new Web service for Calculator applications.

PROCEDURE:

When you start Globus toolkit container, there will be number of services starts up. The service for this task will be a simple Math service that can perform basic arithmetic for a client.

The Math service will access a resource with two properties:

1. An interval value that can be operated upon by the service
2. A string value that holds string describing the last operation

This service itself will have three remotely accessible operations that operate upon *value*:

- (a) *add*, that adds *a* to the resource property *value*.
- (b) *subtract* that subtracts *a* from the resource property *value*.
- (c) *getValueRP* that returns the current value of *value*.

Usually, the best way for any programming task is to begin with an overall description of what you want the code to do, which in this case is the service interface. The service interface describes how what the service provides in terms of names of operations, their arguments and return values. A Java interface for our service is:

```

public interface Math
{
public void add(int
a);
public void
subtract(int
a);
public int getValueR
P();
}

```

It is possible to start with this interface and create the necessary WSDL file using the standard Webservice tool called Java2WSDL. However, the WSDL file for GT 4 has to include details of resource properties that are not given explicitly in the interface above. Hence, we will provide the WSDL file.

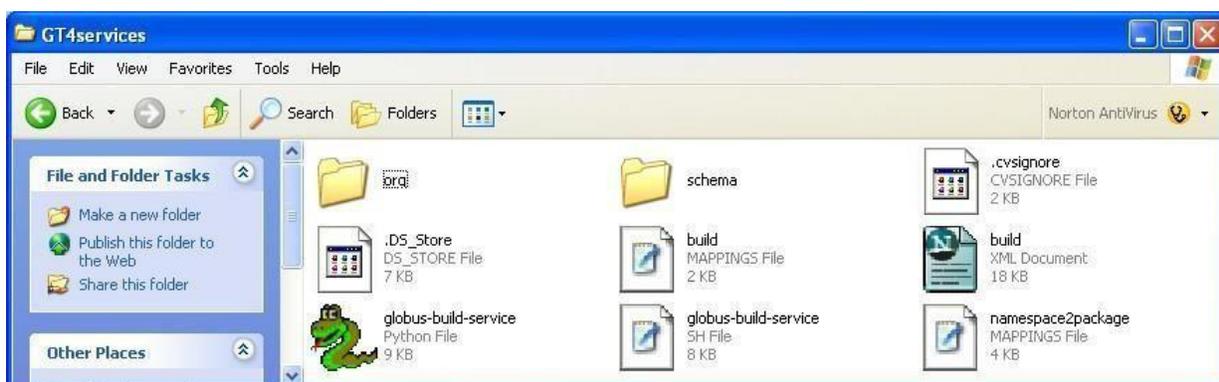
Step 1 Getting the Files

All the required files are provided and come directly from [1]. The MathService source code files can be found from <http://www.gt4book.com>

(<http://www.gt4book.com/downloads/gt4book-examples.tar.gz>)

A Windows zip compressed version can be found at

<http://www.cs.uncc.edu/~abw/ITCS4146S07/gt4book-examples.zip>. Download and uncompress the file into a directory called **GT4services**. Everything is included (the java source WSDL and deployment files, etc.):



WSDLserviceinterfacedescriptionfile--

TheWSDLserviceinterfacedescriptionfileisprovidedwithintheGT4servicesfolderat:

**GT4Services\schema\examples\MathService_
instance\Math.wsdl**

This file, and discussion of its contents, can be found in Appendix A. Later on we will need to modify this file, but first we will use the existing contents that describe the Math service above.

Service code in Java -- For this assignment, both the code for service operations and for the resource properties are put in the same class for convenience. More complex services and resources would be defined in separate classes. The Java code for the service and its resource properties is located within the GT4services folder at:

GT4services\org\globus\examples\services\core\first\impl\MathService.java.

DeploymentDescriptor--

The deployment descriptor gives several different important sets of information about the service once it is deployed. It is located within the **GT4services** folder at:

GT4services\org\globus\examples\services\core\first\deploy-server.wsdd.

Step 2--Building the Math Service

It is now necessary to package all the required files into a GAR (Grid Archive) file. The build tool ant from the Apache Software Foundation is used to achieve this as shown overleaf:

Generating a GAR file with Ant (from

<http://gdp.globus.org/gt4-tutorial/multiplehtml/ch03s04.html>)

Ant is similar in concept to the Unix make tool but a Java tool and XML based.

Build scripts are provided by Globus 4 to use the ant build file. The windows version of the build script for Math Service is the Python file called **globus-build-service.py**, which is held in the **GT4services** directory. The build script takes one argument, the name of your service that you want to deploy. To keep with the naming convention in [1], this service will be called **first**.

In the *Client Window*, run the build script from the **GT4services** directory with:

globus-build-service.py first

The output should look similar to the following:

Buildfile: build.xml

.
. .
. .

BUILDSUCCESSFUL

Totaltime:8seconds

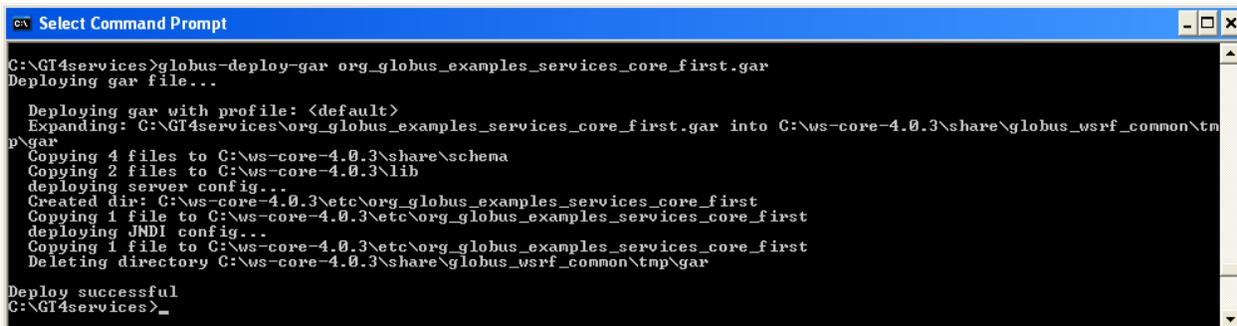
Duringthebuildprocess,anewdirectoryiscreatedinyour**GT4Services**directorythatisnamed**build**.All of your stubs and class files that were generated will be in that directory and its subdirectories. Moreimportantly, there is a GAR (Grid Archive) file called **org_globus_examples_services_core_first.gar**. TheGAR file is the package that contains every file that is needed to successfully deploy your Math Serviceinto the Globus container. The files contained in the GAR file are the Java class files, WSDL, compiledstubs,andthedeploymentdescriptor.

Step3–Deploying theMath Service

If the container is still running in the Container Window, then stop it using Control-C. To deploytheMathService,youwilluseatoolprovidedbytheGlobusToolkitcalled**globus-deploy-gar**. Inthe*ContainerWindow*, issuethecommand:

globus-deploy-garorg_globus_examples_services_core_first.gar

Successfuloutputof thecommandis:



```
 Select Command Prompt
C:\GT4services>globus-deploy-gar org_globus_examples_services_core_first.gar
Deploying gar file...
  Deploying gar with profile: <default>
  Expanding: C:\GT4services\org_globus_examples_services_core_first.gar into C:\ws-core-4.0.3\share\globus_usrf_common\tmp\gar
  Copying 4 files to C:\ws-core-4.0.3\share\schemata
  Copying 2 files to C:\ws-core-4.0.3\lib
  deploying server config...
  Created dir: C:\ws-core-4.0.3\etc\org_globus_examples_services_core_first
  Copying 1 file to C:\ws-core-4.0.3\etc\org_globus_examples_services_core_first
  deploying JNDI config...
  Copying 1 file to C:\ws-core-4.0.3\etc\org_globus_examples_services_core_first
  Deleting directory C:\ws-core-4.0.3\share\globus_usrf_common\tmp\gar
Deploy successful
C:\GT4services>
```

Theservicehasnowbeendeployed.

Check service is deployed by starting container from the *Container*

Window:Youshouldseetheservicecalled**MathService**.

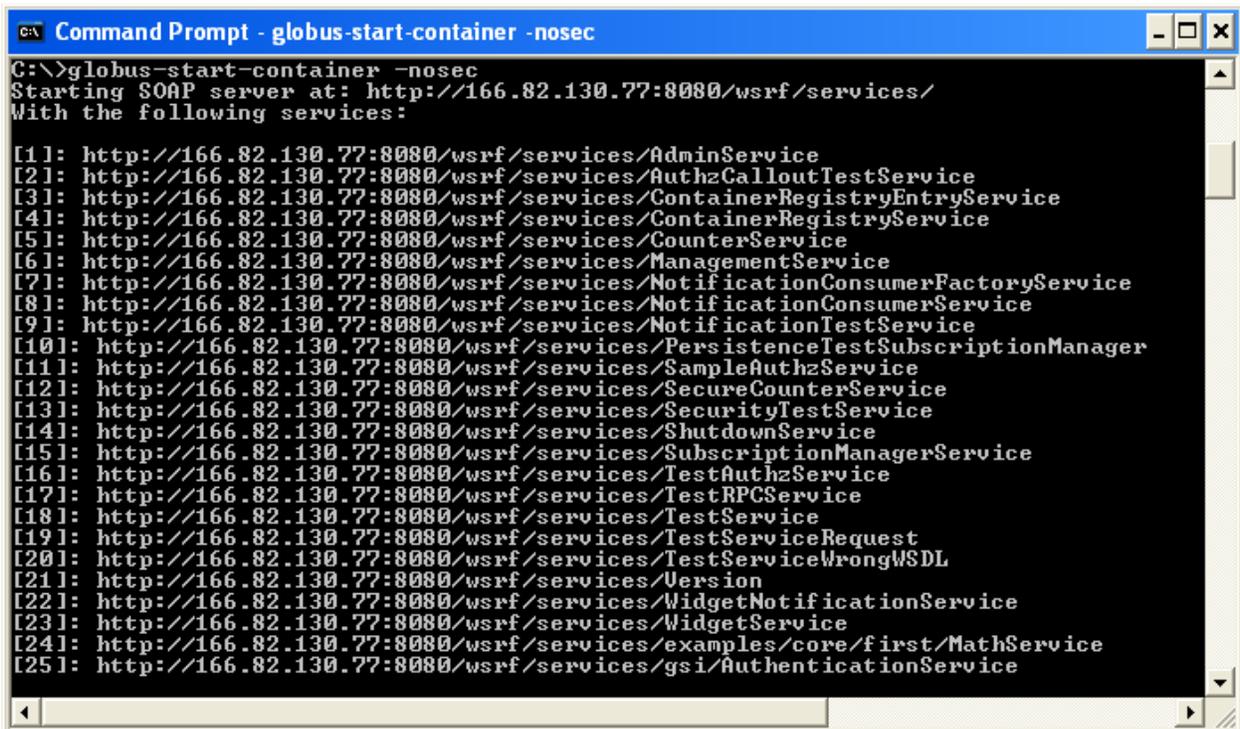
Step4–CompilingtheClient

AclienthasalreadybeenprovidedtotesttheMath

ServiceandislocatedintheGT4Services directory

at:GT4Services\org\globus\examples\clients\MathService_instance\Client.java

andcontains



```
C:\>globus-start-container -nosec
Starting SOAP server at: http://166.82.130.77:8080/wsrf/services/
With the following services:

[1]: http://166.82.130.77:8080/wsrf/services/AdminService
[2]: http://166.82.130.77:8080/wsrf/services/AuthzCalloutTestService
[3]: http://166.82.130.77:8080/wsrf/services/ContainerRegistryEntryService
[4]: http://166.82.130.77:8080/wsrf/services/ContainerRegistryService
[5]: http://166.82.130.77:8080/wsrf/services/CounterService
[6]: http://166.82.130.77:8080/wsrf/services/ManagementService
[7]: http://166.82.130.77:8080/wsrf/services/NotificationConsumerFactoryService
[8]: http://166.82.130.77:8080/wsrf/services/NotificationConsumerService
[9]: http://166.82.130.77:8080/wsrf/services/NotificationTestService
[10]: http://166.82.130.77:8080/wsrf/services/PersistenceTestSubscriptionManager
[11]: http://166.82.130.77:8080/wsrf/services/SampleAuthzService
[12]: http://166.82.130.77:8080/wsrf/services/SecureCounterService
[13]: http://166.82.130.77:8080/wsrf/services/SecurityTestService
[14]: http://166.82.130.77:8080/wsrf/services/ShutdownService
[15]: http://166.82.130.77:8080/wsrf/services/SubscriptionManagerService
[16]: http://166.82.130.77:8080/wsrf/services/TestAuthzService
[17]: http://166.82.130.77:8080/wsrf/services/TestRPCService
[18]: http://166.82.130.77:8080/wsrf/services/TestService
[19]: http://166.82.130.77:8080/wsrf/services/TestServiceRequest
[20]: http://166.82.130.77:8080/wsrf/services/TestServiceWrongWSDL
[21]: http://166.82.130.77:8080/wsrf/services/Version
[22]: http://166.82.130.77:8080/wsrf/services/WidgetNotificationService
[23]: http://166.82.130.77:8080/wsrf/services/WidgetService
[24]: http://166.82.130.77:8080/wsrf/services/examples/core/first/MathService
[25]: http://166.82.130.77:8080/wsrf/services/gsi/AuthenticationService
```

You should see the service called **MathService**.

Step4–CompilingtheClient

AclienthasalreadybeenprovidedtotesttheMath

ServiceandislocatedintheGT4Services directory

at:GT4Services\org\globus\examples\clients\MathService_instance\Client.java

andcontains the following code:

```

packageorg.globus.examples.clients.MathService_instan
ce;importorg.apache.axis.message.addressing.Address;
import
org.apache.axis.message.addressing.EndpointReferenceType;importorg.gl
obus.examples.stubs.MathService_instance.MathPortType;import
org.globus.examples.stubs.MathService_instance.GetValueRP;import
org.globus.examples.stubs.MathService_instance.service.MathServiceAddressi
ngLocator;
publicclassClient{
publicstaticvoidmain(String[]args)
{MathServiceAddressingLocatorlocator
=newMathServiceAddressingLocator()
try{
StringserviceURI=args[0];
// Create endpoint reference to
serviceEndpointReferenceType endpoint =
newEndpointReferenceType();endpoint.setAdd
ress(newAddress(serviceURI));MathPortTypep
ath;
//GetPortType
math =locator.getMathPortTypePort(endpoint);
// Perform an
additionmath.add(
10);
// Perform another

```

`additionmath.add(5);`

```

// Access
valueSystem.out.println("Current value:");
+math.getValueRP(new GetValueRP());
//Performasubtractio
nmath.subtract(5);
// Access
valueSystem.out.println("Current value:");
+math.getValueRP(new GetValueRP());
}catch(Exceptione){e.
printStackTrace();
}
}
}

```

When the client is run from the command line, you pass it one argument. The argument is the URL that specifies where the service resides. The client will create the end point reference and incorporate this URL as the address. The end point reference is then used with the `getMathPortTypePort` method of a `MathServiceAddressingLocator` object to obtain a reference to the `Math` interface (`portType`). Then, we can apply the methods available in the service as though they were local methods. Notice that the call to `toService(addAndSubtractMethodCalls)` must be in a `try{}catch(){}` block because a `RemoteException` may be thrown. The code for the `MathServiceAddressingLocator` is created during the build process. (Thus you don't have to write it!)

(a) Setting the Classpath

To compile the new client, you will need the JAR files from the Globus toolkit in your CLASSPATH. Do this by executing the following command in the Client Window:

```
%GLOBUS_LOCATION%\etc\globus-devel-env.bat
```

You can verify that this sets your CLASSPATH, by executing the command:

echo%CLASSPATH%

You should see a long list of JAR files.

Running **\gt4\etc\globus-devel-**

env.bat only need to be done *once* for each *Client Window* that you open. It does *not* need to be done each time you compile.

(b) Compiling Client

Once your CLASSPATH has been set, then you can compile the Client code by typing in the following command:

javac -

classpathbuild\classes\org\globus\examples\services\core\first\impl:%

CLASSPATH%org\globus\examples\clients\MathService_instance\Cl

ient.java

Step 5—Start the Container for your Service

Restart the Globus container from the *Container Window* with:

globus-start-container -nosec

if the container is not running.

Step 6—Run the Client

To start the client from your **GT4Services** directory, do the following in the *Client Window*, which passes the GSH of the service as an argument:

java -

classpathbuild\classes\org\globus\examples\services\core\first\impl:%

CLASSPATH%org.globus.examples.clients.MathService_instance.Cl

ienthttp://localhost:8080/wsrf/services/examples/core/first/MathServi

ce which should give the output:

Current value: 15

Current value: 10

Step7–UndeploytheMathServiceandKillaContainer

Before we can add functionality to the Math Service (Section 5), we must undeploy the service. In the *ContainerWindow*, kill the container with a `Control-C`. Then to undeploy the service, type in the following command:

```
globus-undeploy-garorg_globus_examples_services_core_first
```

which should result with the following output:

```
Undeploying
```

```
gar...Deleting/.
```

```
.
```

```
.
```

```
Undeploysuccessful
```

6 Adding Functionality to the Math Service

In this final task, you are asked to modify the Math service and associated files so the service supports the multiplication operation. To do this task, you will need to modify:

Service code (**MathService.java**)

WSDL file (**Math.wsdl**)

The exact changes that are necessary are not given. You are to work them out yourself. You will need to fully understand the contents of service code and WSDL files and then modify them accordingly. Appendix A gives an explanation of the important parts of these files. Keep all file names the same and simply redeploy the service afterwards. You will also need to add a code to the client code (**Client.java**) to test the modified service to include multiplication.

Result:

2. Develop new OGSA-compliant Web Service

OBJECTIVE:

To develop a new OGSA-compliant web service.

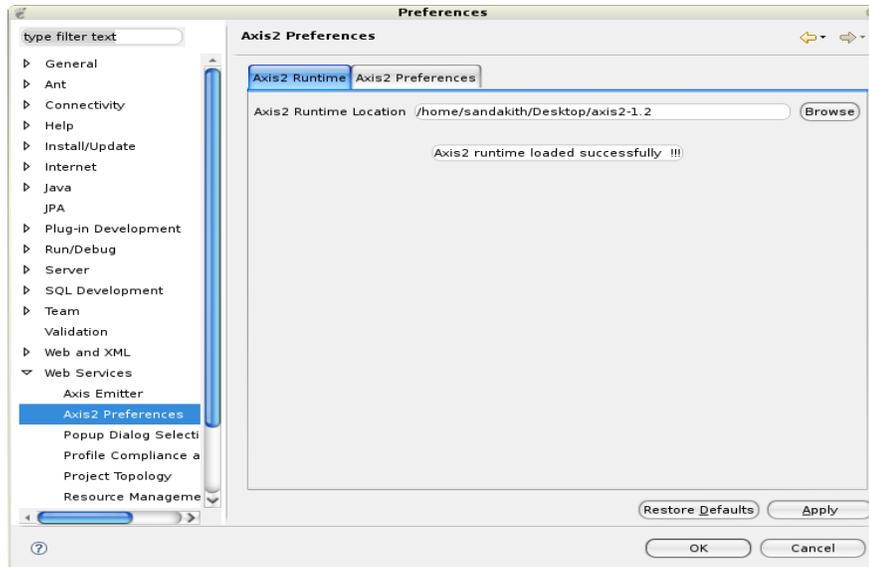
PROCEDURE:

Writing and deploying a WSRF Web Service is easier than you might think. You just have to follow five simple steps

1. Define the service's interface. This is done with *WSDL*
2. Implement the service. This is done with *Java*.
3. Define the deployment parameters. This is done with *WSDD* and *JNDI*
4. Compile everything and generate a *GAR* file. This is done with *Ant*
5. Deploy service. This is also done with a *GT4tool*

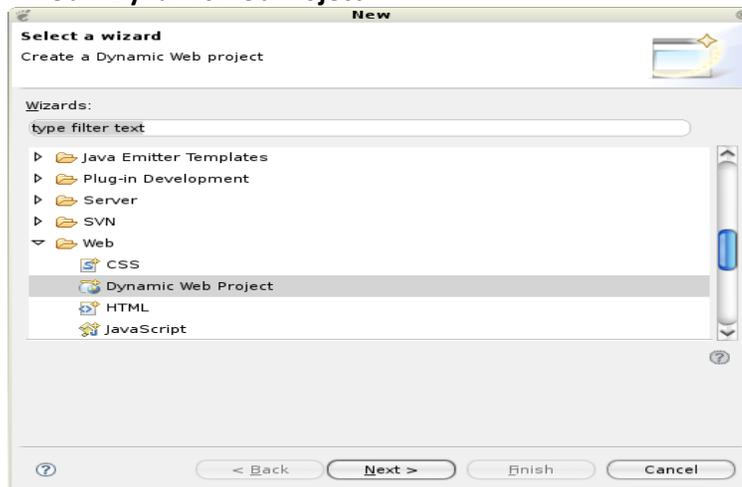
To run this program, as a minimum you will be required to have installed the following prerequisite software

- a. Download the latest Axis2 runtime from the above link and extract it. Now we point Eclipse WTP to downloaded Axis2 Runtime. Open **Window -> Preferences -> Web Services -> Axis2 Emitter**



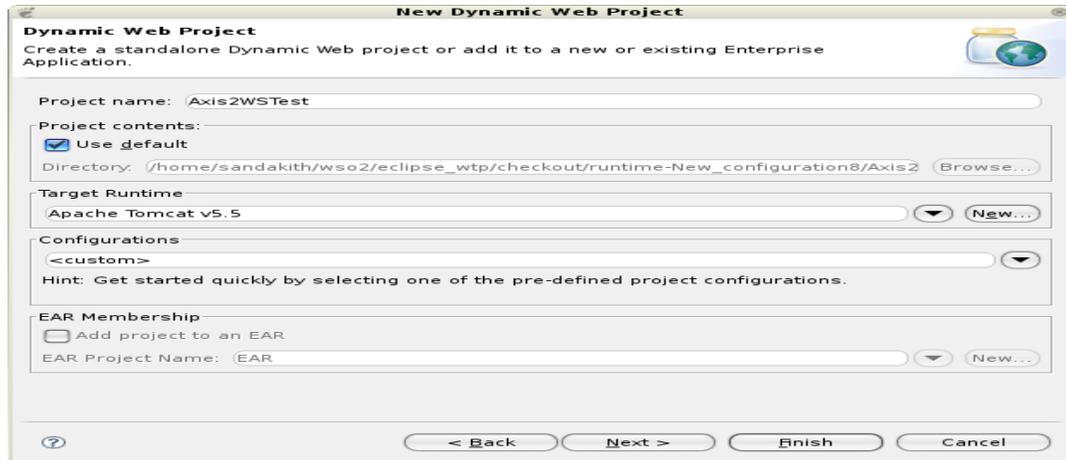
Select the Axis2 Runtime tab and point to the correct Axis2 runtime location. Alternatively at the Axis2 Preference tab, you can set the default setting that will come up on the WebServicesCreation wizards. For the moment we will accept the default settings.

- b. Click OK.
- c. Next we need to create a project with the support of Axis2 features. Open **File->New->Other...->Web ->Dynamic Web Project**



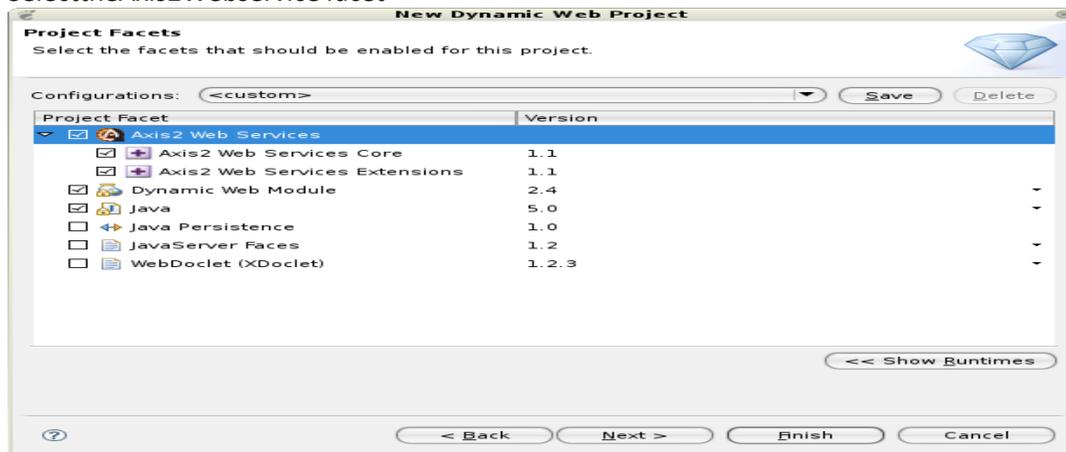
Click next

- d. Select the name **Axis2WSTest** as the Dynamic Web project name (you can specify any name you prefer), and select the configured Tomcat runtime as the target runtime.



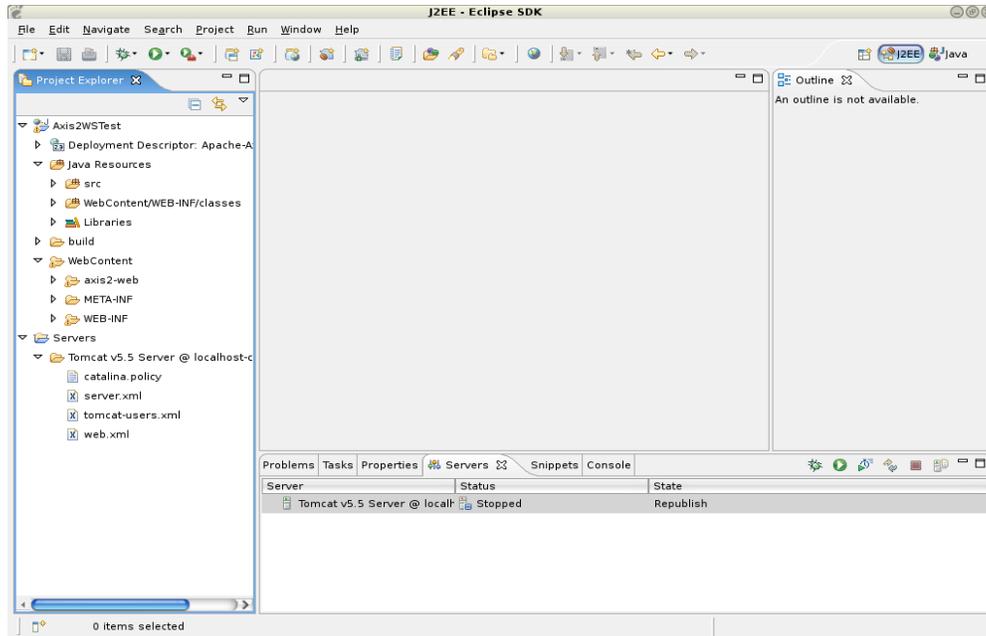
Clicknext.

- e. SelecttheAxis2Webservice facet

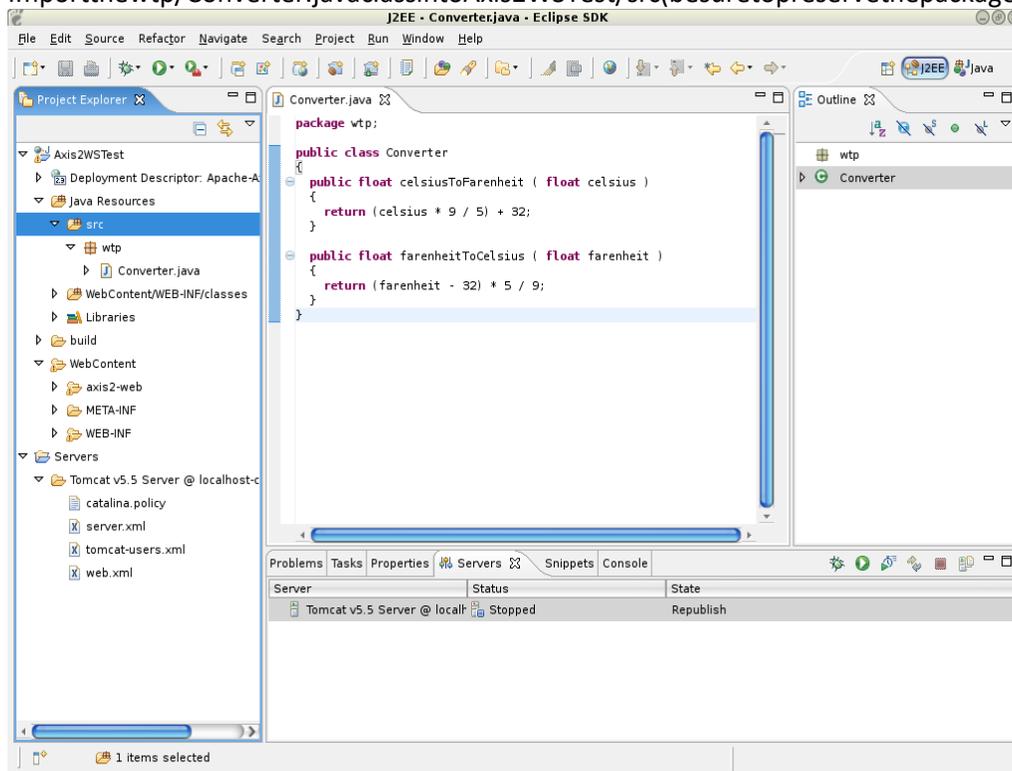


ClickFinish.

- f. Thiswill createdynamicWebprojectintheworbench

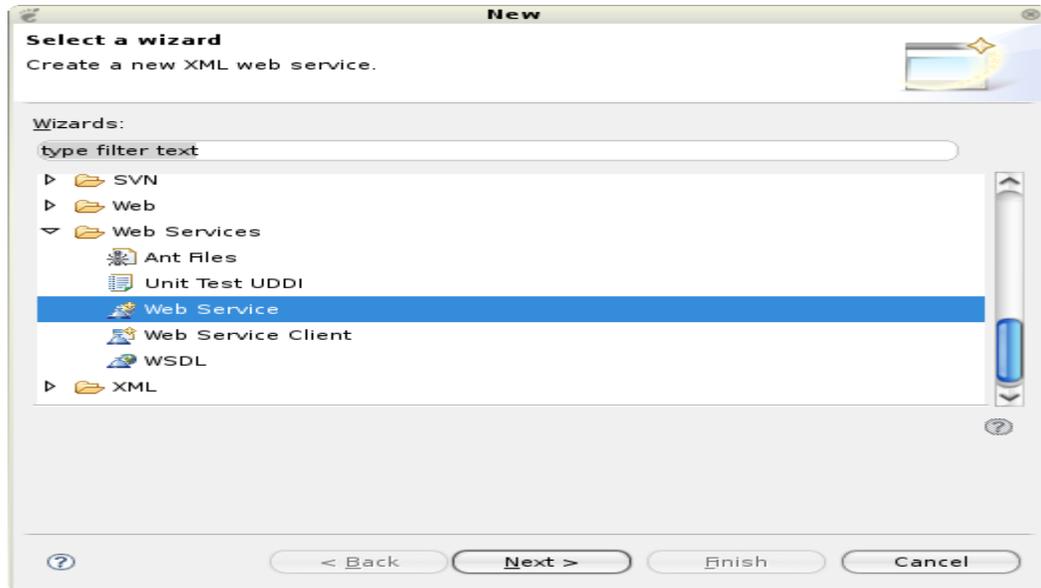


g. Import the wtp/Converter.java class into Axis2WSTest/src (be sure to preserve the package).



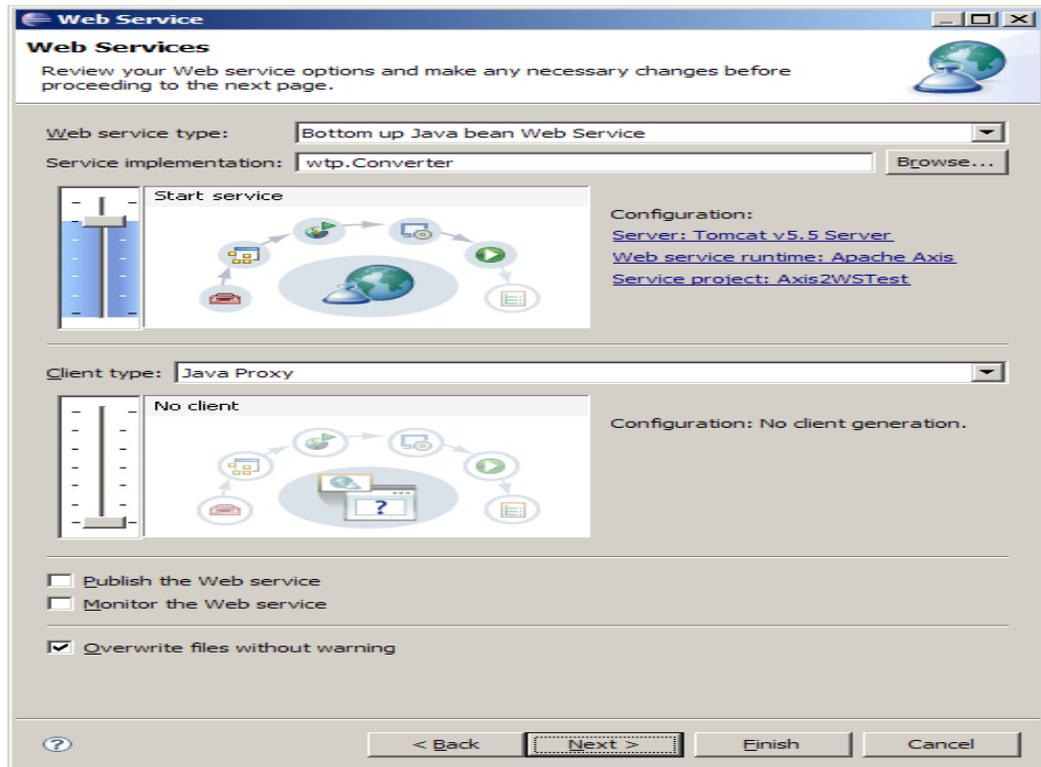
Build the Project, if it's not auto build.

h. Select Converter.java, open File->New->Other... ->WebServices->WebService

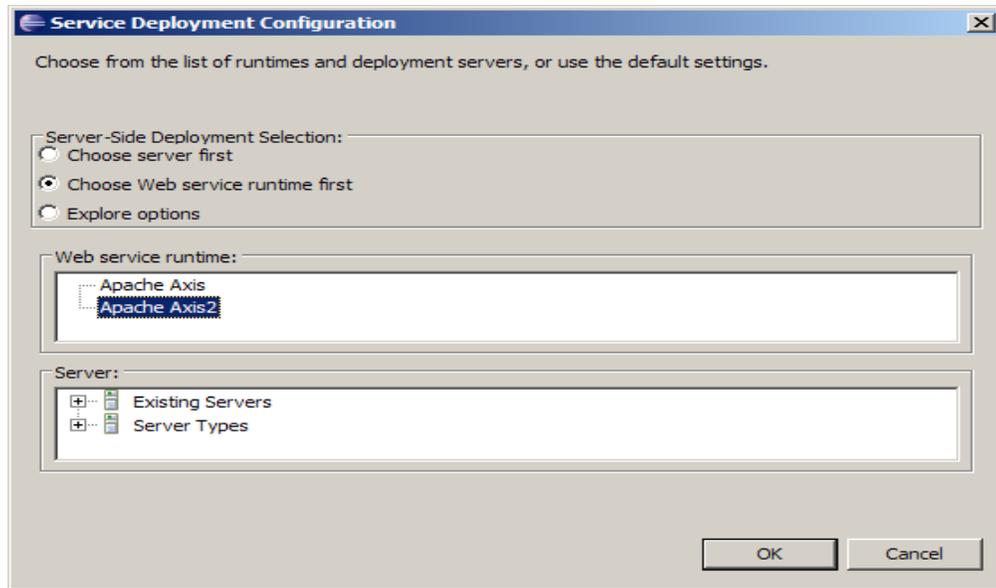


Click next.

- i. The Web service wizard would be brought up with Web service type set to **Bottom up JavabeanWebService** with the service implementation automatically filled in. Move the servicescale to **Start service**.

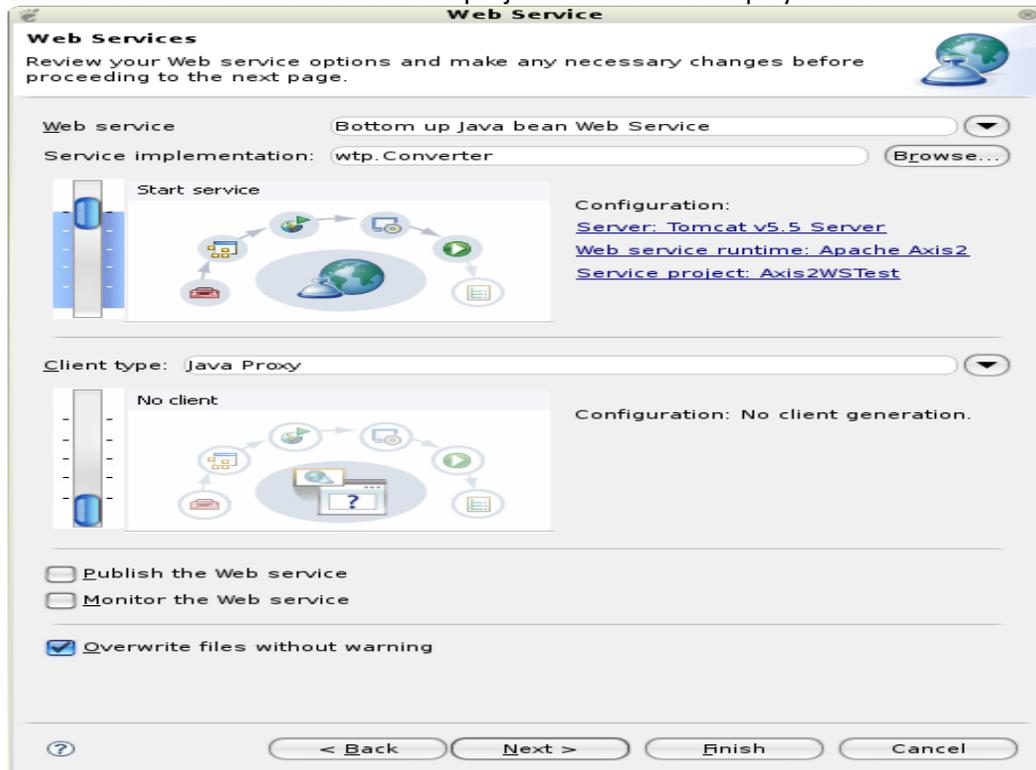


- j. Click on the **Web Service runtime** link to select the Axis2 runtime.



Click OK.

- k. Ensure that the correct server and service project are selected as displayed below.



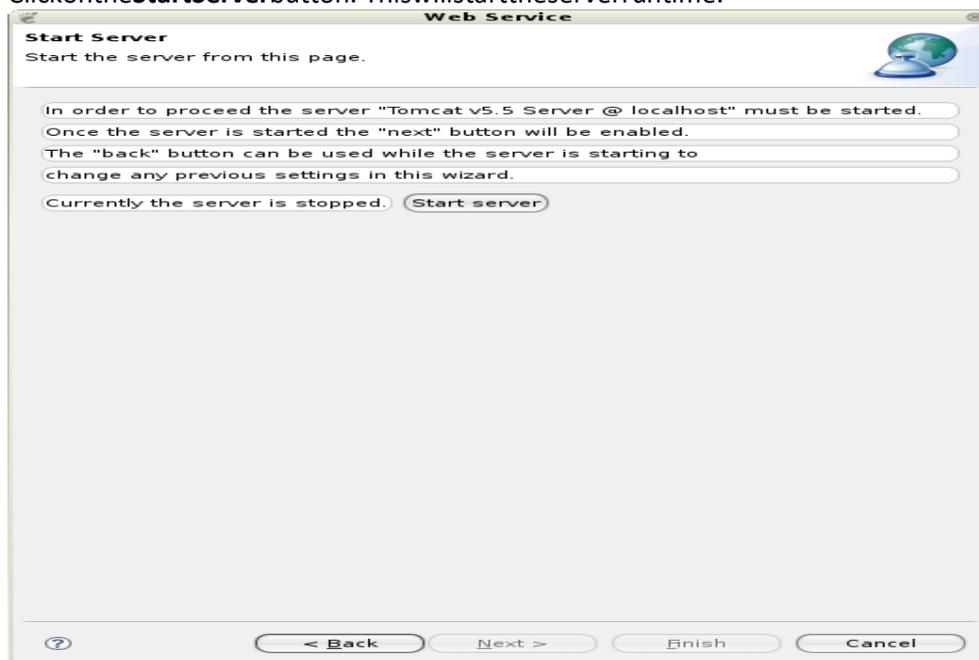
Click next.

- l. This page is the service.xml selection page. If you have a custom services.xml, you can include that by clicking the **Browse** button. For the moment, just leave it at the default.



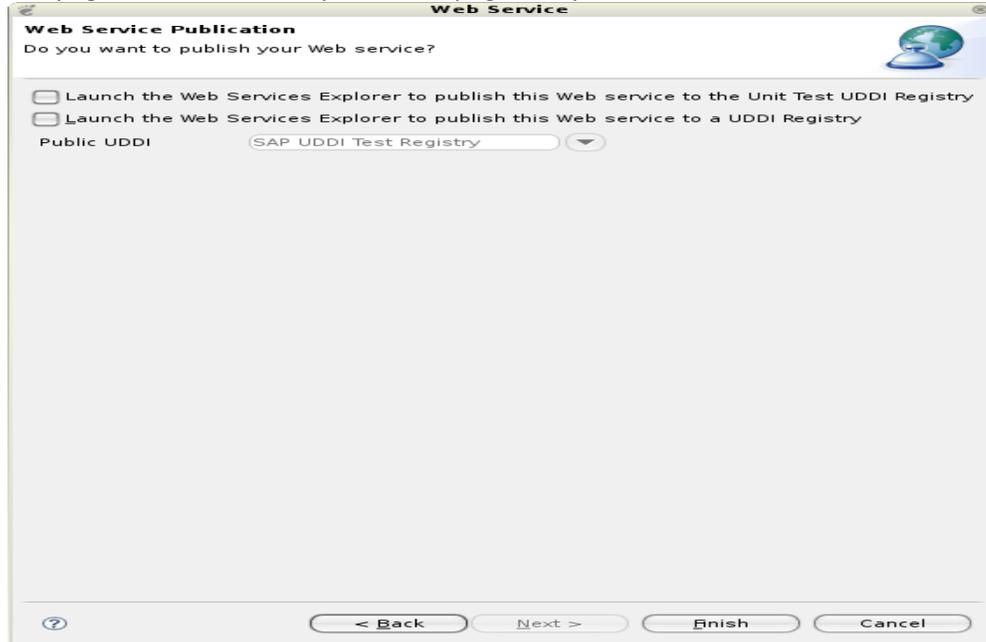
Click next.

- m. This page is the Start Server page. It will be displayed if the server has not been started. Click on the **Start Server** button. This will start the server runtime.



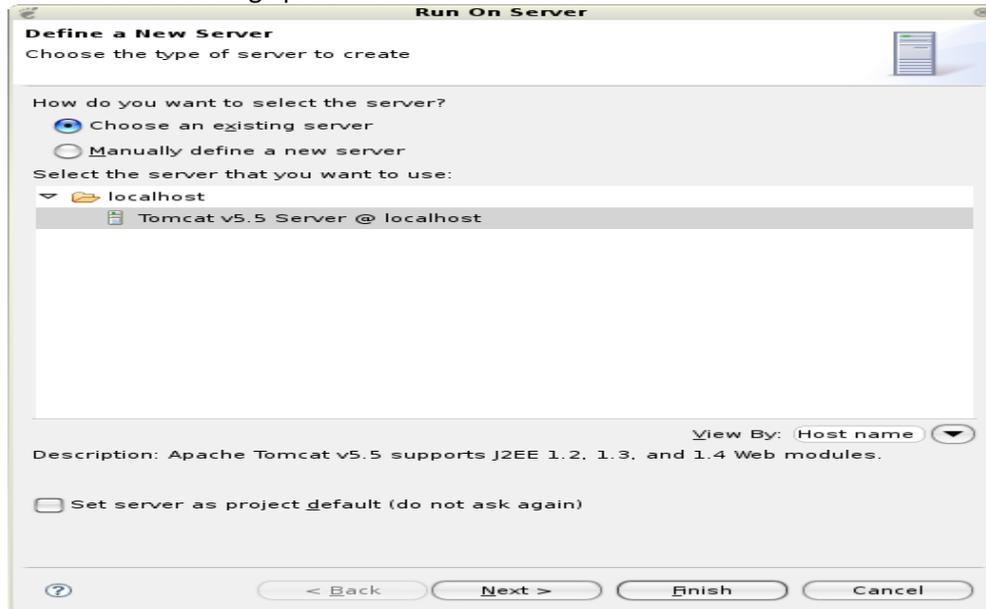
Clicknext.

- n. ThispageistheWebservicespublicationpage,acceptthedefaults.



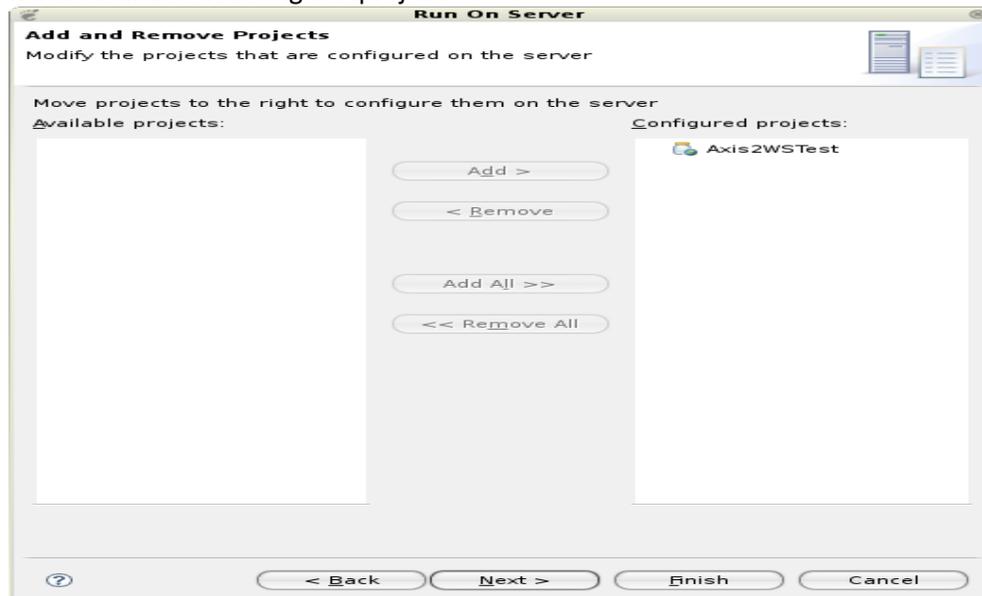
ClickFinish.

- o. Now,selectthe **Axis2WSTest** dynamicWebproject,right-clickandselectRun->RunAs->RunOnServertobring uptheAxis2servlet.



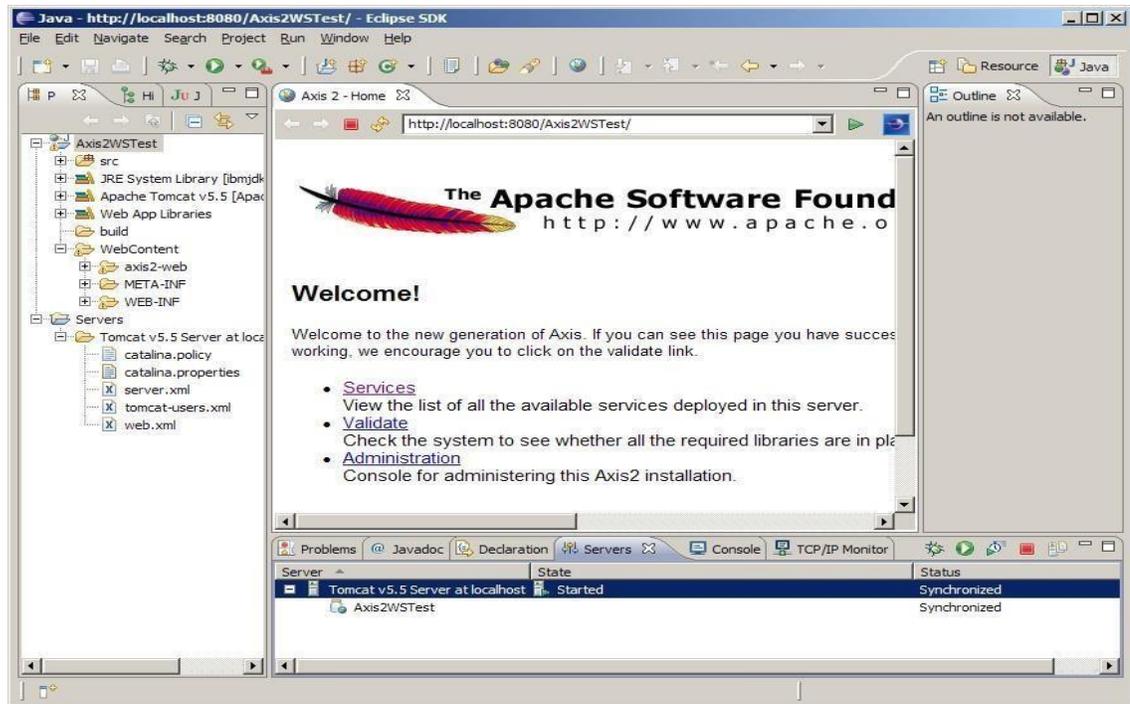
ClickNext.

- p. Make sure you have the **Axis2WSTest** dynamic Web project on the right-hand side under the Configured project.

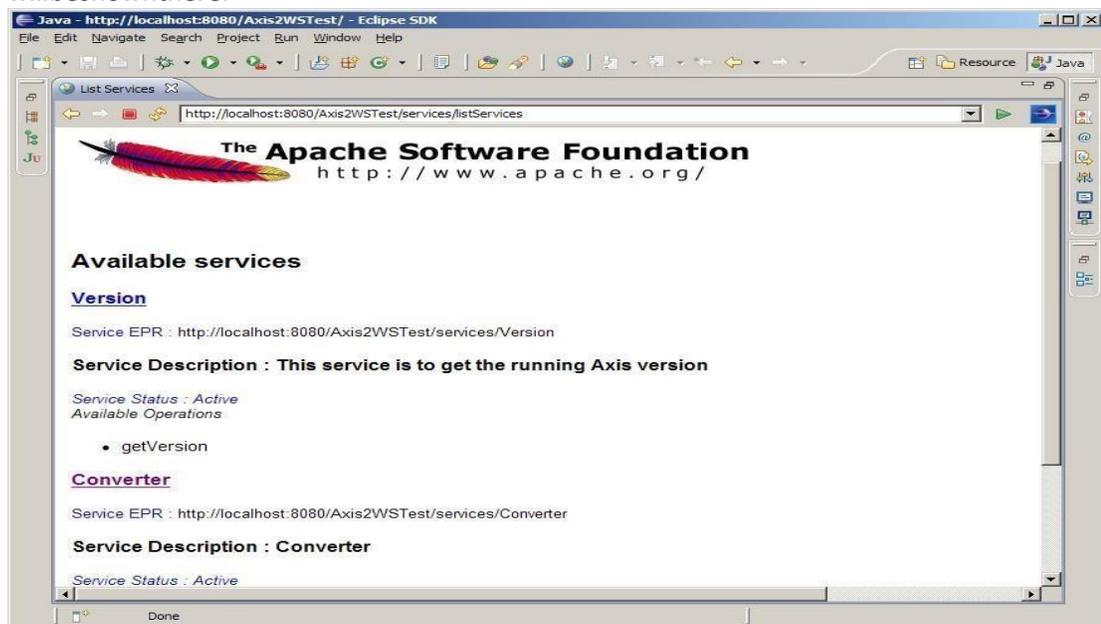


ClickFinish.

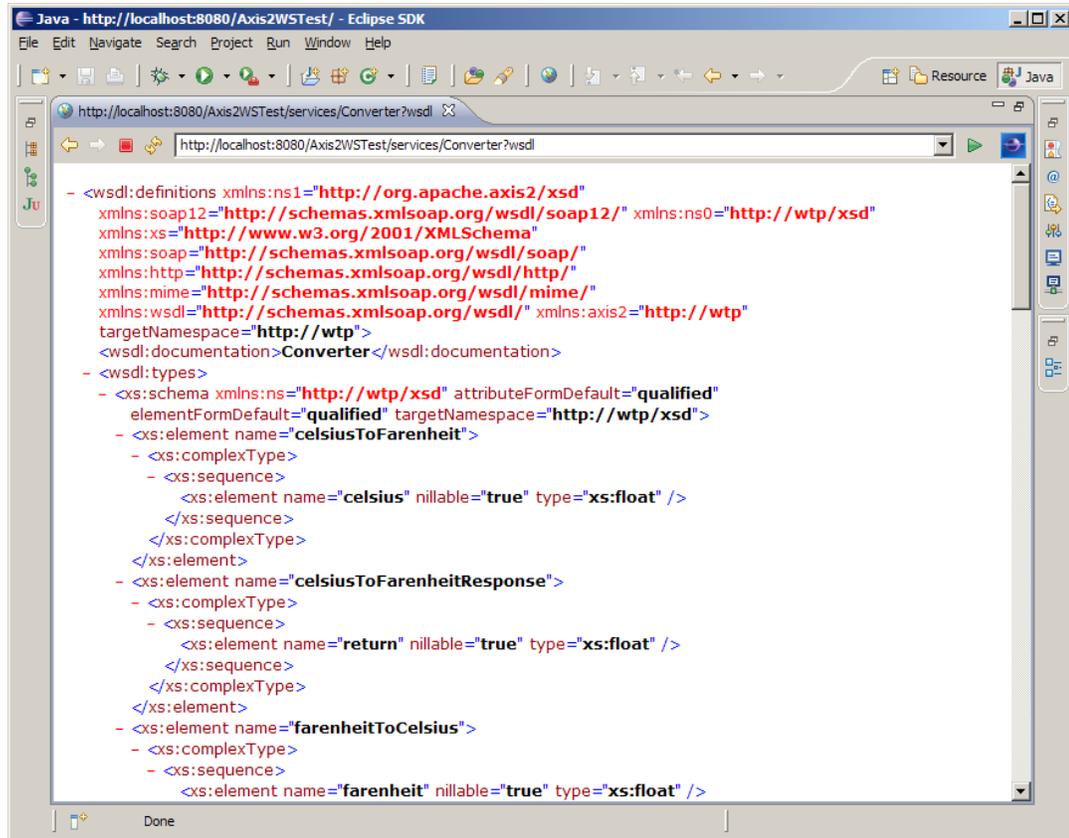
- q. This will deploy the Axis2 server webapp on the configured servlet container and will display the Axis2 home page. Note that the servlet container will start up according to the Server configuration files on your workspace.



- r. Click on the **Services** link to view the available services. The newly created converter Webservice will be shown there.



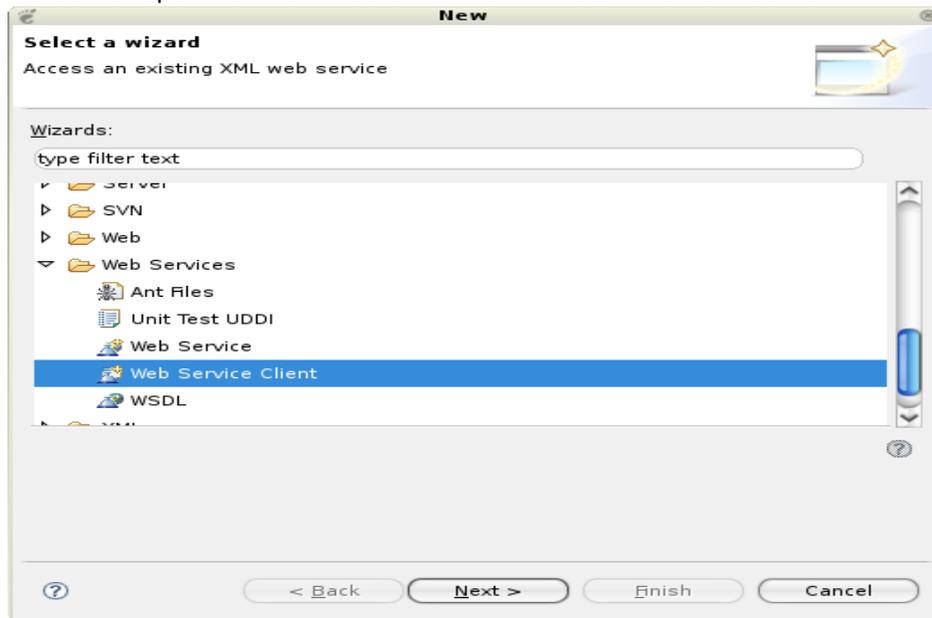
- s. Click on the **Converter Service** link to display the wsdl URL of the newly created Webservice. Copy the URL.



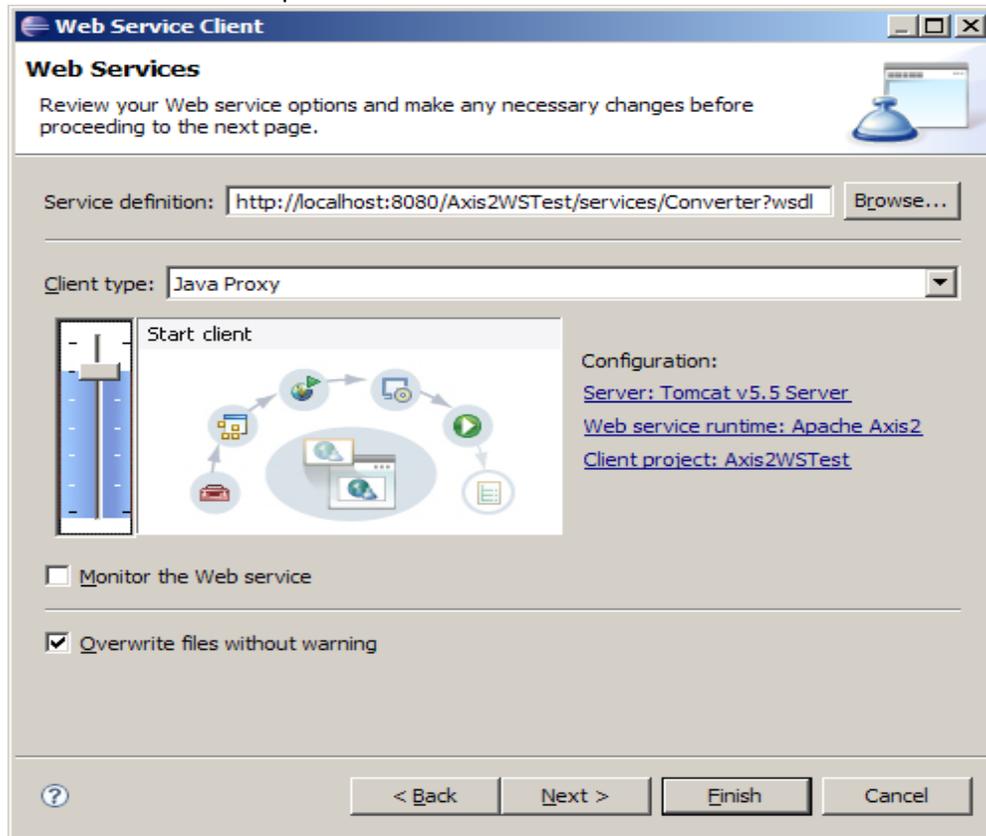
The screenshot shows the Eclipse IDE with a browser window displaying a WSDL file. The WSDL content is as follows:

```
<?xml version='1.0' encoding='UTF-8' ?>
<wsdl:definitions xmlns:ns1="http://org.apache.axis2/xsd"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:ns0="http://wtp/xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:axis2="http://wtp"
  targetNamespace="http://wtp">
  <wsdl:documentation>Converter</wsdl:documentation>
  <wsdl:types>
  <xs:schema xmlns:ns="http://wtp/xsd" attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://wtp/xsd">
    <xs:element name="celsiusToFahrenheit">
    <xs:complexType>
    <xs:sequence>
    <xs:element name="celsius" nillable="true" type="xs:float" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    <xs:element name="celsiusToFahrenheitResponse">
    <xs:complexType>
    <xs:sequence>
    <xs:element name="return" nillable="true" type="xs:float" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
    <xs:element name="fahrenheitToCelsius">
    <xs:complexType>
    <xs:sequence>
    <xs:element name="fahrenheit" nillable="true" type="xs:float" />
    </xs:sequence>
    </xs:complexType>
    </xs:element>
  </xs:schema>
  </wsdl:types>
</wsdl:definitions>
```

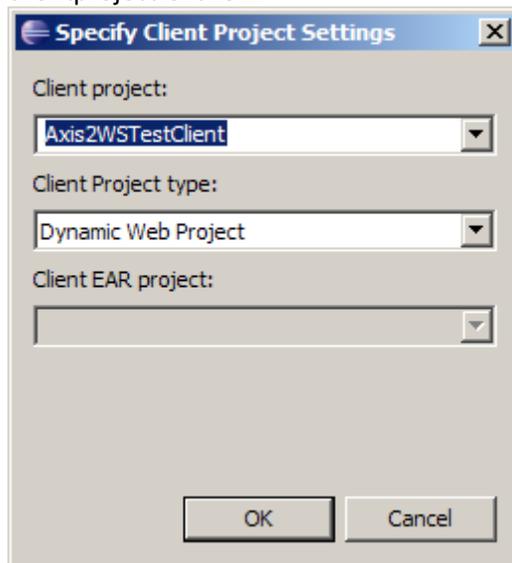
- t. Now we'll generate the client for the newly created service by referring the ?wsdl generated by the axis2Server. Open File->New->Other...->Web Services->Web Service Client



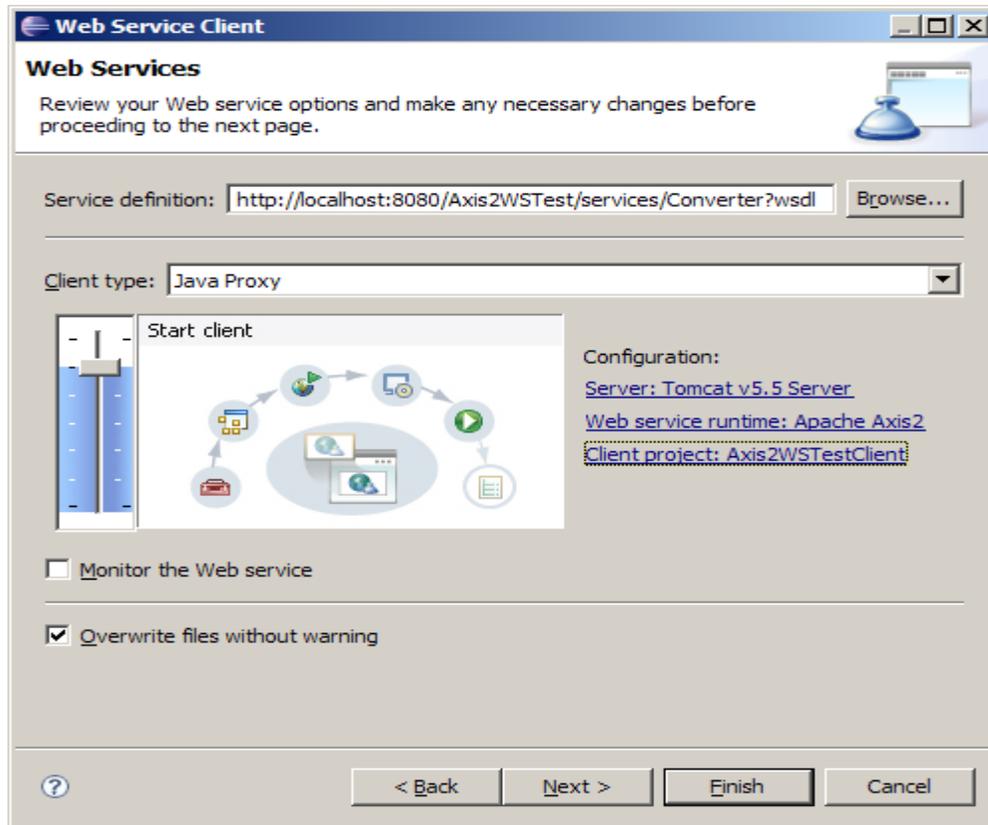
- u. Paste the URL that was copied earlier into the service definition field.



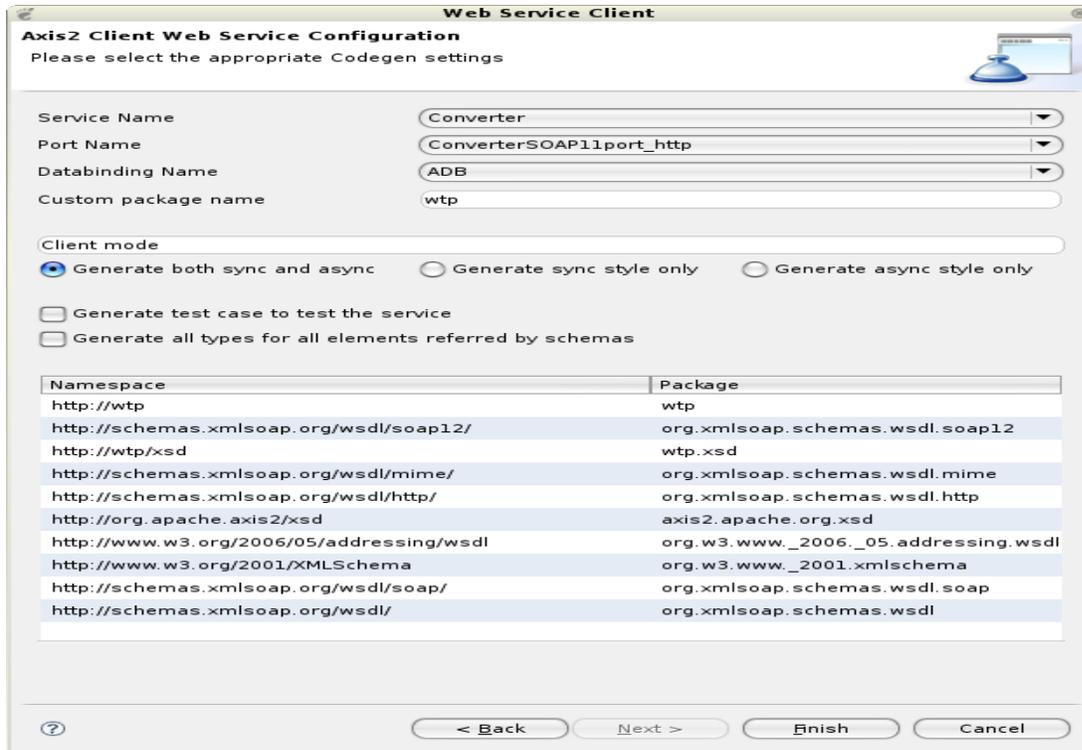
- v. Click on the **Client project** hyperlink and enter **Axis2WSTestClient** as the name of the client project. Click OK.



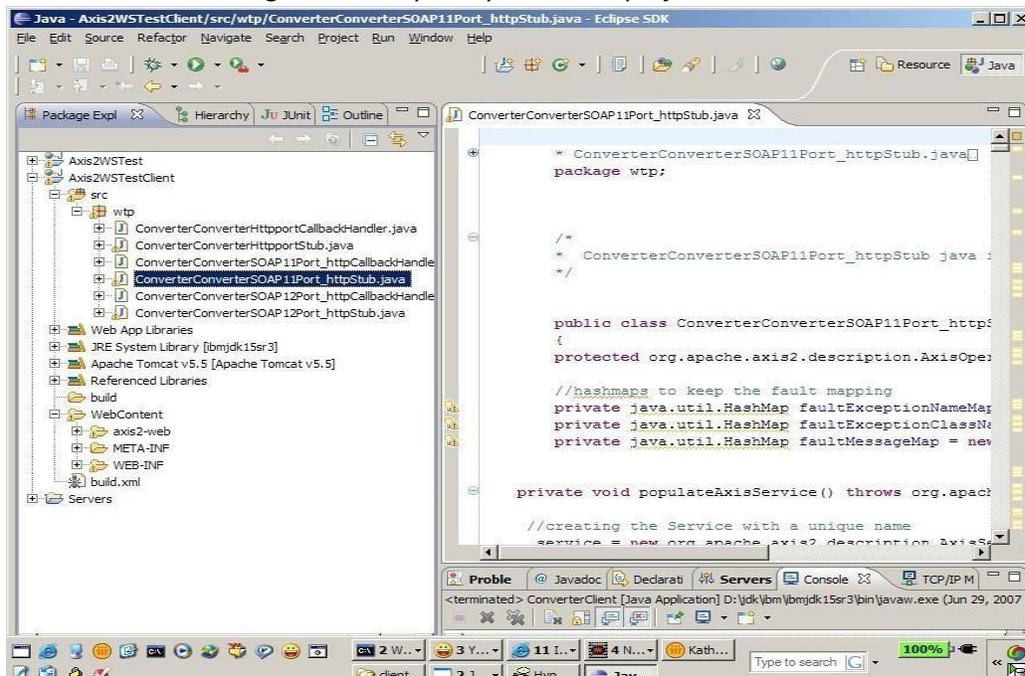
Back on the Web Services Client wizard, make sure the Web service runtime is set to Axis2 and the server is set correctly. Click Next.



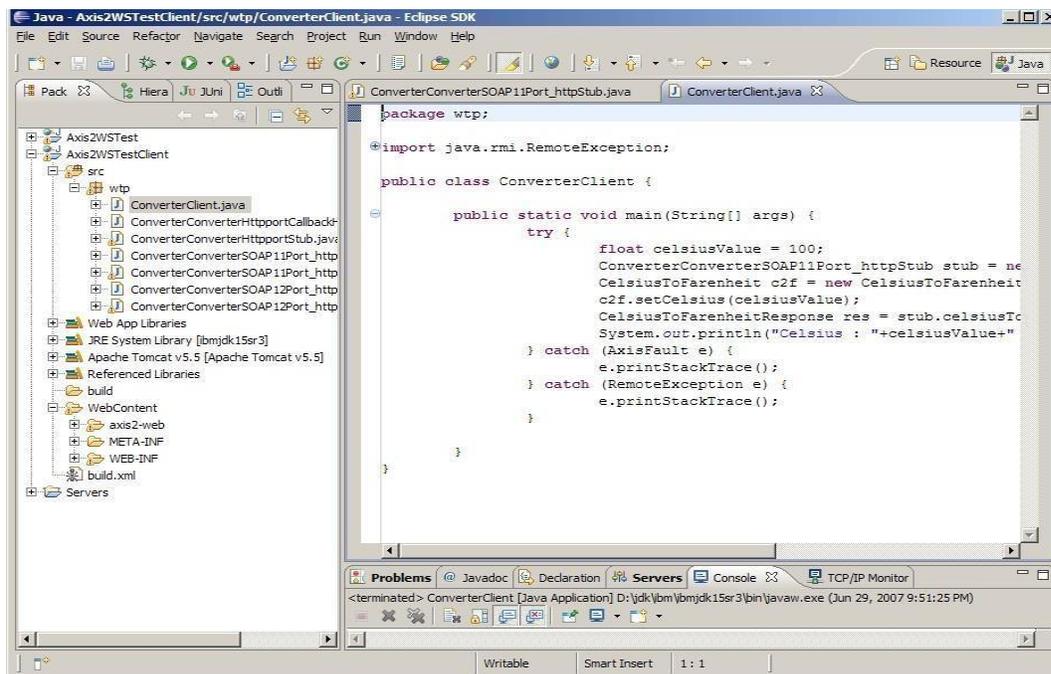
NextpageistheClientConfigurationPage.AcceptthedefaultsandclickFinish.



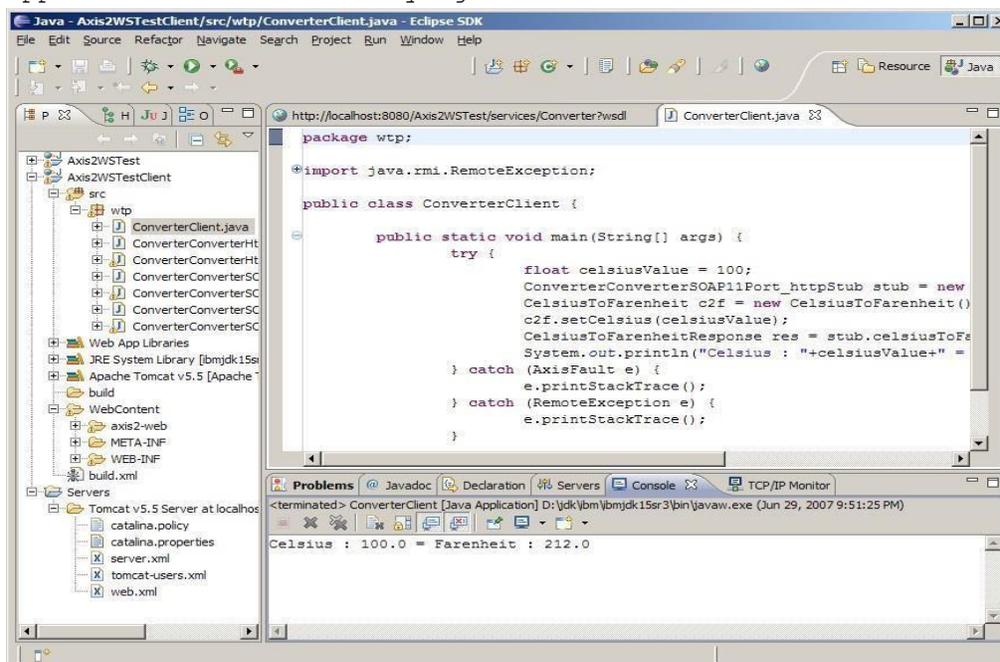
The Clients stubs will be generated to your Dynamic Web project Axis2WSTestClient.



Now we are going to write Java main program to invoke the client stub. Import the `ConverterClient.java` file to the workspace into the `wtp` package in the `src` folder of `Axis2WSTestClient`.



Then select the `ConverterClient` file, right-click and select `Run As -> Java Application`. Here's what you get on the server console:



Another way to test and invoke the service is to select **Generate test case to test the service** checkbox on the Axis2 Client Web Service Configuration Page when going through the Web Service Client wizard.

Axis2 Client Web Service Configuration
Please select the appropriate Codegen settings

Service Name: Converter
 Port Name: ConverterHttpport
 Databinding Name: ADB
 Custom package name: wtp

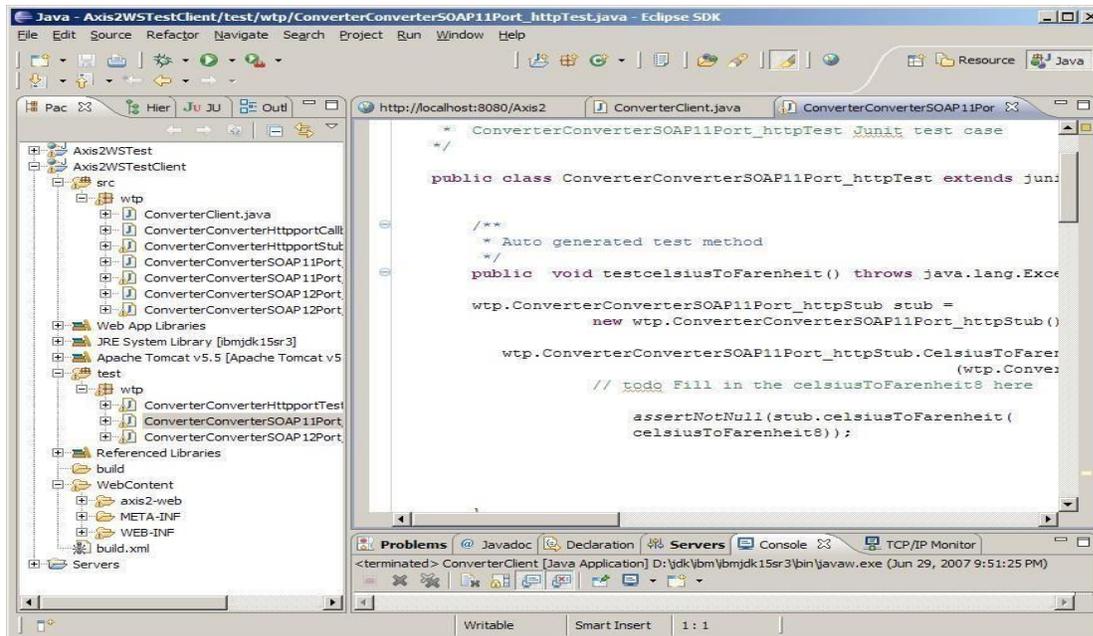
Client mode:
 Generate both sync and async
 Generate sync style only
 Generate async style only

Generate test case to test the service
 Generate all types for all elements referred by schemas

Namespace	Package
http://wtp	wtp
http://schemas.xmlsoap.org/wsdl/soap12/	org.xmlsoap.schemas.wsdl.soap12
http://wtp/xsd	wtp.xsd
http://schemas.xmlsoap.org/wsdl/mime/	org.xmlsoap.schemas.wsdl.mime
http://schemas.xmlsoap.org/wsdl/http/	org.xmlsoap.schemas.wsdl.http
http://org.apache.axis2/xsd	axis2.apache.org.xsd
http://www.w3.org/2006/05/addressing/wsdl	org.w3.www._2006_05.addressing.wsdl
http://www.w3.org/2001/XMLSchema	org.w3.www._2001.xmlschema
http://schemas.xmlsoap.org/wsdl/soap/	org.xmlsoap.schemas.wsdl.soap
http://schemas.xmlsoap.org/wsdl/	org.xmlsoap.schemas.wsdl

< Back Next > Finish Cancel

If that option is selected, the Axis2 emitter will generate JUnit testcases matching the WSDL we provide to the client. These JUnit testcases will be generated to a newly added source directory to the **Axis2WSTestClient** project called **test**.

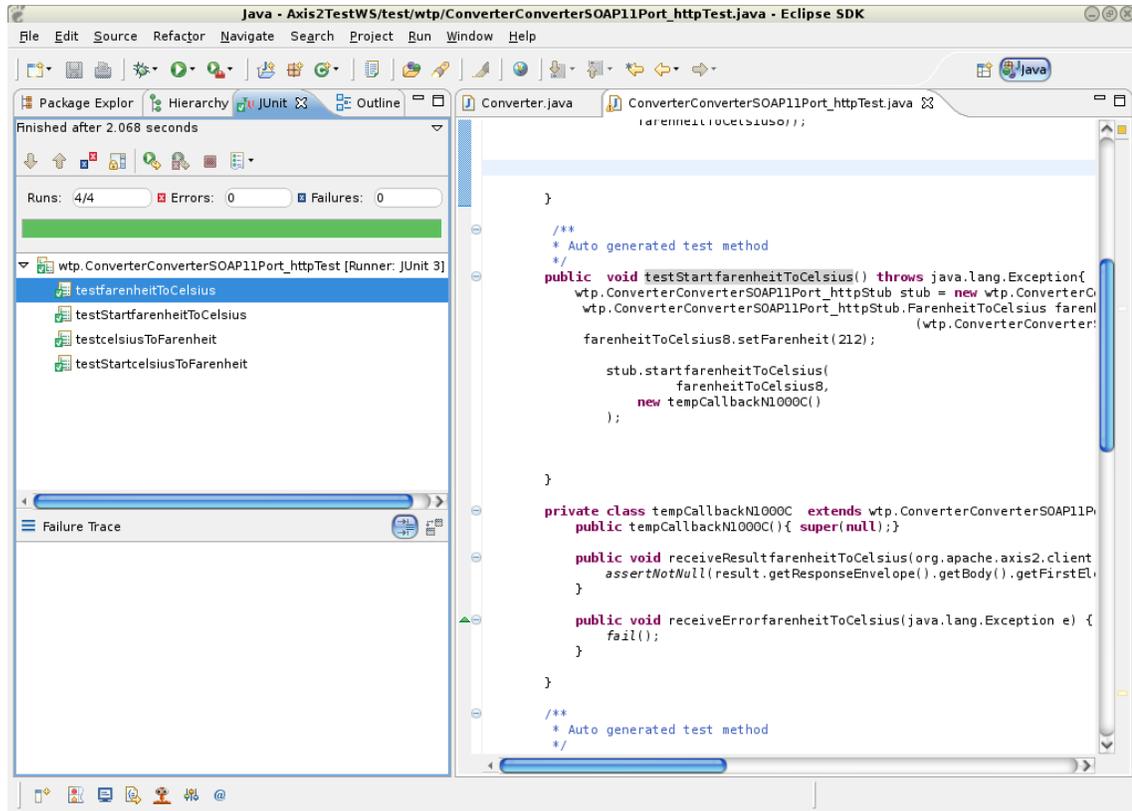


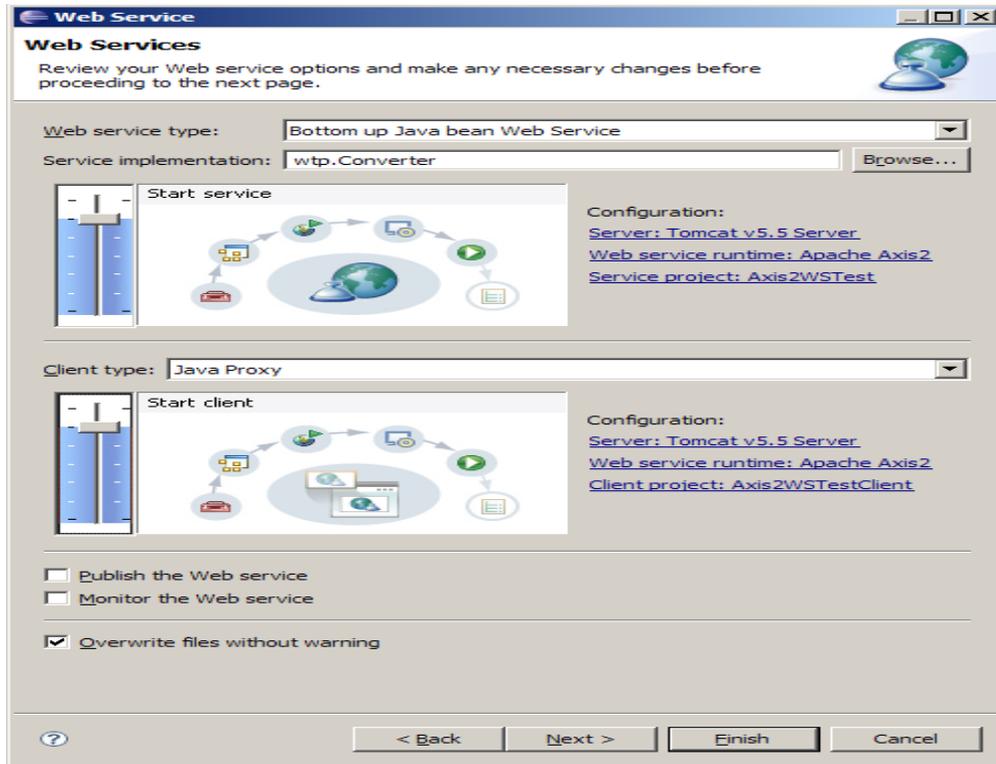
Next thing we need to do is to insert the test case with the valid inputs as the Web servicemethod arguments. In this case, let's test the ConverterConverterSOAP11Port_httpTest.java by provide values for Celsius and Farenheit for the temperature conversion. As an example, replacethegeneratedTODO statementineachtestmethodtofill inthedatawithvaluesas:

```
testfahrenheitToCelsius () ->
fahrenheitToCelsius8.setFahrenheit (212);testStartfahrenheitToCelsius () -
>fahrenheitToCelsius8.setFahrenheit (212);testcelsiusToFahrenheit () ->
celsiusToFahrenheit10.setCelsius (100);testStartcelsiusToFahrenheit ()-
>celsiusToFahrenheit10.setCelsius (100);
```

Herethetestcasesweregeneratedtotestboththesynchronousandasynchronousclients.

- w. Afterthat,selectthetestcase,right-click,selectRunAs->JUnitTest.YouwillbeabletoruntheunittestsuccessfullyinvokingtheWebservice.





The Web Service wizard orchestrates the end-to-end generation, assembly, deployment, installation and execution of the Web service and Web service client. Now that your Web service is running, there are a few interesting things you can do with this WSDL file. Examples:

- You can choose Web Services -> Test with Web Services Explorer to test the service.
- You can choose Web Services -> Publish WSDL file to publish the service to a public UDDI registry.

RESULT:

Thus the development of a new OGSA-compliant web service was executed successfully.

3.UsingApacheAxisdevelopaGridService

OBJECTIVE:

TodevelopaGridServiceusingApacheAxis.

PROCEDURE:

Youwillneedtodownloadandinstallthefollowing software:

1. Java2SDKv1.4.1,<http://java.sun.com/j2se/1.4.1/download.html>

2. ApacheTomcatv4.124

<http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.1.24/bin/jakartatomcat4.1.24.exe>.

3. XMLSecurityv1.0.4,

<http://www.apache.org/dist/xml/security/java-library/xmlsecuritybin1.0.4.zip>

4. Axisv1.1,http://ws.apache.org/axis/dist/1_1/axis-1_1.zip

1. Java2SDK

- Runthedownloaddexecutable(j2sdk-1_4_1-windows-i586.exe)whichwillinstallthe
- SDKinC:\j2sdk1.4.1.SettheJAVA_HOMEenvironmentvariabletopointtothisdirectoryasfollows:
- ClickonSTART->CONTROL PANEL->SYSTEM
- ClickontheAdvancedtab
- ClickontheEnvironmentVariablesbutton
- ClickontheNew...buttonintheuservariablesectionandenterthedetails
- AddtheJavabinariestoyourPATHvariableinthesamewaybysettingauservariablecalledPATHwiththevalue"%PATH%;C:\j2sdk1.4.1\bin"

2. Apache Tomcat

- Run the downloaded executable (jakarta-tomcat-4.1.24.exe), and assume the installation directory is C:\jakarta-tomcat-4.1.24.
- Edit C:\jakarta-tomcat-4.1.24\conf\tomcat-users.xml and create an "admin" and "manager" role as well as a user with both roles. The contents of the file should be similar to:

```
<?xml version='1.0' encoding='utf8'?>
<tomcat-users>
<role rolename="manager"/>
<role rolename="admin"/>
<user username="myuser" password="mypass" roles="admin,manager"/>
</tomcat-users>
```

- Start Tomcat by running C:\jakarta-tomcat-4.1.24\bin\startup.bat and test it by browsing <http://localhost:8080/>
- Stop Tomcat by running C:\jakarta-tomcat-4.1.24\bin\shutdown.bat.

3. XML Security

- Download and unzip http://www.apache.org/dist/xml/security/java/library/xmlsecurity-bin1_0_4.zip
- Copy xml-sec.jar to C:\axis-1_1\lib\
- Set your CLASSPATH environment variable to including the following: C:\axis1_1\lib\xml-sec.jar;

4. Apache Axis

- Unzip the downloaded Axis archive to C: (this will create a directory C:\axis-1_1).

- Extract the file `xmlsec.jar` from the downloaded security archive to

C:\axis1_1\webapps\axis\WEB-INF\lib.

- Set-

your CLASSPATH environment variable to include the following:

• Set your current working directory

- o All the AXIS jar files as found in

C:\axis-1_1\libC:\jakarta-tomcat-4.1.24\common\lib\servlet.jar

- Your CLASSPATH should therefore look something

like:C:\axis-1_1\lib\axis.jar;

C:\axis1_1\lib\axis-ant.jar;

C:\axis-1_1\lib\commons-

discovery.jar;C:\axis-

1_1\lib\commons-

logging.jar;C:\axis-

1_1\lib\jaxrpc.jar;

C:\axis-1_1\lib\log4j-

1.2.8.jar;C:\axis-

1_1\lib\saa.jar;C:\ax

is-

1_1\lib\wsdl4j.jar;C:\

axis-

1_1\lib\xercesImpl.jar

C:\axis-1_1\lib\xmlParserAPIs.jar;

C:\jakarta-tomcat-

4.1.24\common\lib\servlet.jarC:\axis

-1_1\lib\xml-sec.jar;

- Now tell Tomcat about your Axis web application by creating the

fileC:\jakarta-tomcat-

4.1.24\webapps\axis.xmlwiththefollowingcontent:

```
<Context path="/axis" docBase="C:\axis-  
1_1\webapps\axis" debug="0"privileged="true">  
<LoggerclassName="org.apache.catalina.logger.FileLogger"prefix="  
axis_log."suffix=".txt"timestamp="false"/>
```

5. Deploy a Sample Web Service Packaged within Axis Installations

Deploy one of the sample Web Services to test the system and to create the C:\axis-1_1\webapps\axis\WEB-INF\server-config.wsdd file. From C:\axis-1_1 issue the command (on one line):

```
java org.apache.axis.client.AdminClient
http://localhost:8080/axis/services/AdminService/samples/stock/deploy.wsdd
```

This should return the following:

```
.-Processing files samples/stock/deploy.wsdd
.-<Admin>Done processing</Admin>
```

RESULT:

Thus the development of a Grid Service using Apache Axis is executed successfully.

4. Develop applications using Java or C/C++ Grid APIs

OBJECTIVE:

To develop an application using Java or C/C++ Grid APIs.

SAMPLE CODE:

```
import
AgentTeamwork.Ateam.*;imp
ortMPJ.*;
public class UserProgAteam extends AteamProg{
private int phase;
public UserProgAteam(Ateam a)
{}
public UserProgAteam()
```

```

{}
//realconst
publicUserProgAteam(String[]
args) {phase=0;
}
//phaserecovery
private void
userRecovery( )
{phase=ateam.getSnapshot
Id();
}
privatevoidcompute() {
for (phase=0;phase<10;phase++
){try{
Thread.currentThread().sleep(1000);
}
catch(InterruptedException){
}
ateam.takeSnapshot(phase);
System.out.println( "UserProgAteam at rank " + MPJ.COMM_WORLD.Rank( ) + " :
took a snapshot " +phase);
}
}
public static void main( String[] args )
{System.out.println("UserProgAteam:gotsta
rted");MPJ.Init(args,ateam);
UserProgAteamprogram=null;
//Timertimer=newTimer(
);if(ateam.isResumed()
){

```

```

program = ( UserProgAteam
)ateam.retrieveLocalVar("pr
ogram");program.userRecover
y();
}
else
{
program = new UserProgAteam( args
);ateam.registerLocalVar("program",progr
am);
}
program.compute
(
);MPJ.Finalize(
);
}
publicclassUserProgAteamextendsAteamProg{
//applicationbody
privatevoidcompute() {for (phase=0;
phase<10;phase++) {
try{
Thread.currentThread().sleep(1000);
}
catch(InterruptedExeptione) {
}
ateam.takeSnapshot (phase);
System.out.println ("UserProgAteamatrank"+MPJ.COMM_WORLD.Rank()+
:tookasnapshot"+phase);
}}
}

```

Socketsamplecode-withinsomefunctionbody

```
import
```

```
AgentTeamwork.Ateam.GridTcp.*;p  
rivatefinalintport=2000;
```

```

private GridSocket socket; private
GridServerSocketserver;privateInputStreamin
put;privateOutputStreamoutput;

for(inti=start;i
<start+trans;i++){try{
output.write(i%128);
}catch( IOException) {
}

System.out.println ("Socketswith"+myRank+": "+" output["+i +"]="+i % 128);
}

for(inti=start;i<start+trans;
i++){try{
System.out.println("Socketswith"+myRank+": "+"input["+i +"]="+input.read());}
catch( IOException ) {
}}

```

MPIsamplecode

```

import
AgentTeamwork.Ateam.*;imp
ortMPI.*;

publicclassUserProgAteamextendsAteamProg{
//applicationbody privatevoidcompute() {
}

publicstaticvoidmain(String[
]args){MPJ.Init( args, ateam
);program.compute();MPJ.Fina
lize();
}
}

```

C/C++compile.sh–HelloWorld.cpp

```
#!/bin/s
hrm-
f*.class
javac-classpathMPJ.jar:Ateam.jar:*.java
#jarcvfGridJNI.jar*.class jar-
cvfGridJNI.jar*.classjavah-jniJavaToCpp
g++-rdynamicJavaToCpp.cpp-o_libJavaToCpp.so_-shared-ldlg++-shared-
o_libHelloWorld.so_GridJNI_library.cpp
HelloWorld.cpp
```

C/C++MPIsamplecode–HelloWorld.cpp

```
#include
<iostream.h>usin
gnamespacestd;
typedefintMPI_Request,MPI_Status,MPI_Comm;
externvoidtakeSnapshot(intargc);
externintMPI_Init(int*argc,char*
**argv);externvoidMPI_Finalize()
;
externintMPI_Comm_rank(MPI_Commcomm,int*rank
);extern int MPI_Comm_size(MPI_Comm comm,
int *size);intmain(intargc,char**argv){
cerr<<"main"<<endl;
cerr<<"argc="<<argc<<endl;
cerr<<"argv[0]="<<argv[0]<<endl;cerr<<"argv[1]="<<arg
v[1]<<endl;MPI_Init(&argc,&argv);
cout<<"MPIInitSuccessful!"<<endl;
cout<<"[HelloWorld.cpp]CallingRank()andSize()"<<endl;
```

```
inrank, size;

MPI_Comm_rank(0, &rank);
MPI_Comm_size(0, &size);

cout<<"[HelloWorld.cpp]Rank= "<<rank<<endl;

cout<<"[HelloWorld.cpp]Size= "<<
size<<endl; cerr<<"CallingMPI_Finalize()"<<endl; MPI_Finalize();

cerr<<"finished"<<endl;
}
```

RESULT:

The development of applications using Java or C/C++ Grid APIs is executed successfully.

5. Develop secured applications using basic security mechanisms

available in Globus Toolkit

OBJECTIVE:

To develop secured applications using basic security mechanisms available in Globus.

PROCEDURE:

The Globus Toolkit's Authentication and Authorization components provide the default standard for the "core" security software in Grid systems and applications. These software development kits (SDKs) provide programming libraries, Java classes, and essential tools for a PKI, certificate-based authentication system with single sign-on and delegation features, in either Web Services or non-Web Services frameworks. ("Delegation" means that once someone accesses a remote system, he can give the remote system permission to use his credentials to access other systems on his behalf.)

WEBSERVICES AUTHENTICATION AND AUTHORIZATION-

A Web services implementation of the Grid Security Infrastructure (GSI), containing the core libraries and tools needed to secure applications using GSI mechanisms. The Grid is a term commonly used to describe a distributed computing infrastructure which will allow "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations". The protocols and middleware to enable this Grid infrastructure have been developed by a number of initiatives, most notably the Globus Project.

Web Services are simply applications that interact with each other using Web standards, such as the HTTP transport protocol and the XML family of standards. In particular, Web Services use the SOAP messaging standard for communication between service and requestor. They should be self-describing, self-contained and modular; present a platform and implementation neutral connection layer; and be based on open standards for description, discovery and invocation.

The Grid Security Infrastructure (GSI) is based on the Generic Security Services API (GSS-API) and uses an extension to X509 certificates to provide a mechanism to authenticate subjects and authorize resources. It allows users to benefit from the ease of use of a single sign-on mechanism by using delegated credentials, and time-limited proxy certificates. GSI is used as the security infrastructure for the Globus Toolkit.

Recently, a new proposal for an Open Grid Services Architecture (OGSA) was announced which marries the Grid and Web Services to create a new Grid Services model. One problem, which has not yet been explicitly addressed, is that of security. A possible solution is to use as suitably secure retransport binding, e.g. TLS, and extend it to incorporate appropriate support for proxy credentials. It would be useful to test out some of the principles of Grid Services using the currently available frameworks and tools for developing Web Services. Unfortunately, no standards currently exist for implemented proxy credentials support to provide authenticated communication between web services. A number of XML/Web Services security standards are currently in development, e.g. XML Digital Signatures, SAML, XKMS, XACML, but the remainder of this document describes an approach proposed by ANL to use GSI over an SSL link.

A generic Job Submission environment, GAP enables researchers and scientists to execute their applications on Grid from a conventional web browser. Both Sequential and Parallel jobs can be submitted to GARUDA Grid through Portal. It provides a web interface for viewing the resources, and for submitting and monitoring jobs.



GARUDA - India's National Grid Computing Initiative will unleash a comprehensive computational capability to enable the user community to participate in increasingly interdisciplinary scientific experiments and also conduct simulations on an unprecedented scale.

User's Login Form

User Id :

Password :

I accept Garuda User Policy

[Sign Up ?](#)

[Forgot User Id or Password?](#)

[Help](#)



© 2006 - C-DAC, All rights reserved.

Pre-requisites for using GAP

```
Portal users need to set the following in their ~/.bash
rcfile. export GLOBUS_LOCATION=/opt/asvija/GLOBU
```

S-4.0.7/

```
source/opt/asvija/GLOBUS-4.0.7/etc/globus-user-env.sh

export
PATH=/usr/local/jdk1.6.0_10/bin:GW_LOCATION/bin:/opt/garud
aresv/bin:/opt/voms_client/bin:$PATH

exportLD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/voms_client/lib:
```

AccessingGAP

Type <http://192.168.60.40/GridPortal1.3/> (to access the Portal through GARUDA Network) or <http://203.200.36.236/GridPortal1.3> (to access the Portal through Internet) in the address bar of the web browser to invoke the Portal. It is preferable to access the Portal through GARUDA Network, since it is much faster than the Internet.

In order to access the facilities of Grid Portals such as Job Submission, Job Status tracking, Storing (Uploading) of Executables and View Output/Error data, the user has to login into the

Portal using the User's Login Form in the Homepage of the Portal.

a) New users are required to click Sign up in the User Login Form, which leads them to home page of Indian Grid Certification Authority (IGCA) (<http://ca.garudaindia.in/>). Click on Request Certificate and acquire the required user/host certificate(s), details are provided in IGCA section.

b) Registered users are required to provide User Id and Password for logging into the Portal and access various facilities.

Job Management

User can submit their job, monitor the status and view output files using the Job Management interfaces. Types of job submission (Basic and Advanced) and Job information are covered under this section.

Basic Job Submission

This interface can be used to submit sequential as well as parallel jobs. The user should provide the following information:

1. OptionalJobName-Usercanprovideasuitable(alias)namefortheirjob.
2. TypeofJobuserwanttoexecute,
3. OperatingSystem-RequiredfortheirJob,
4. 'Have you reserved the Resources' - An optional parameter contains the Reservation Id's thatcanbeusedforjobsubmissioninsteadofchoosingtheOperatingSystem/Processorparameter.
5. No.ofprocessesrequiredforthejob - Thisparameterisonlyforthe parallel applicationsthatrequiremorethanoneCPU.
6. CorrespondingExecutables-uploadedfromeitherlocalorremotemachine,
7. Input file, if required - The executable and the input file can either be uploaded from the localmachineorcanbeselectedfromtheRemoteFileList,ifitisavailableintheSubmitNode
8. STDIN-Required whentheuser wantstoprovide anyinputstotheapplicationduringtheruntime.
9. OptionalExecutionTime-Here,theExecutionTimeistheanticipatedjobcompletiontime.
10. AnyCommandLineargumentsorEnvironmentVariables,ifrequired.
11. UserSpecificOutput/Errorfiles-Iftheapplicationgeneratesoutput/errorfilesratherthanstandard output/error files and its entries should be separated by comma's or single empty space incaseofmultiplefiles.

All those fields marked with * are mandatory fields and should be filled before submitting a job. By clicking on submit button, the portal submits the job to GridWay Meta Scheduler, which then schedules the job for execution and returns the Job Id. The Job Id has to be noted for future referencetothisjob. In the event of unsuccessful submission, the corresponding error message is displayed.

All those fields marked with * are mandatory fields and should be filled before submitting a job. By clicking on submit button, the portal submits the job to GridWay Meta Scheduler, which then schedules the job for execution and returns the Job Id. The Job Id has to be noted for future referencetothisjob.

AdvancedJobSubmission

Job Management

Resources

File Browser

Accounting

Myproxy

Voms

Notices

Help/User Manual

FAQs

Partner site

Basic Submission
Advanced Submission
Job Info

Select OS/Processor*	----- Select -----	
Select Executable*	Remote	-- Select Remote File --
Command Line Arguments	<input type="text"/>	
Input File	Remote	-- Select Remote File -- <input type="button" value="Add to List"/>
STDIN	Remote	-- Select Remote File --
Environment	<input type="text"/>	
Application Specific Output File	<input type="text"/>	
Application Specific Error File	<input type="text"/>	
Memory	<input type="text"/>	
No. Of Nodes*	1	
No. Of Process*	1	

Note : All fields marked with * are mandatory

[Help](#)

This interface is provided for the user to submit their Sequential and Parallel Jobs. The difference from Basic job submission being: it is using GT4 Web Services components for submitting job to the Grid instead of Gridway scheduler.

The user is provided with two modes in this interface:

1. Default mode - Portal creates the XML file for the user.
2. Second mode, recommended for advanced users - The user can provide their own XML file as the executable, provided the required files are available in the submit node.

The user can view the status of the job submitted through Portal and the output file of the job by specifying the Job Id. The option for downloading the Output/ Error file is also provided, after the job execution. To cancel any of the queued jobs, the user has to select the job and click Cancel Job button, following which the acknowledgment for the job canceled is provided.

JobInfo

The user can view the status of the job submitted through Portal and the output file of the job by specifying the Job Id. The option for downloading the Output/ Error file is also provided, after the job execution. To cancel any of the queued jobs, the user has to select the job and click

Cancel Job button, following which the acknowledgment for the job canceled is provided.



Resources

The GridWay meta-scheduler provides the following information - NodeName, HeadNode, OS, ARCH, Load Average, Status, Configured Process and Available Process. This information aids user to select a suitable cluster and reserve them in advance for job submission.

The screenshot displays the Garuda Job Submission web interface. At the top, there is a header with the CDAC logo and the Garuda logo. Below the header, there is a navigation menu on the left with options like Job Management, Resources, File Browser, Accounting, Myproxy, voms, Notices, Help, FAQs, and Partner site. The main content area is titled "Resources available in the Grid" and shows a table of available nodes. Below the table is a "Reservation of Resources" form with fields for StartTime, EndTime, No of cpus, and OS NAME, along with "Show Availability" and "Reset" buttons. At the bottom, there is a copyright notice: "© 2008 - C-DAC, All rights reserved."

Resources available in the Grid
Total number of Nodes = 7
Total number of Available Processors = 75

Node Name	OS	ARCH	Memory	JobManager	Conf Procs	Available Procs
gg-blr.tfg	Linux2.6.18-53.	x86_6	8164	PBS	320	36
xn02.ctsf.cdac.org.in	Linux2.6.9-67.E	x86	2273	PBS	3	3
rrihpc1.rii.local.net	Linux2.4.21-20.	x86_6	11578	PBS	32	7
gg-hyd.cdac.org.in	Linux2.6.18-53.	x86_6	14154	PBS	8	0
gg-che.local	Linux2.6.18-8.e	x86_6	11130	PBS	312	7
tf73.ctsf.cdac.org.in	AIX5.1.0.67	Power	7660	PBS	80	9
master.iitg.ernet.in	Linux2.6.9-42.E	x86_6	7431	PBS	128	13

Reservation of Resources

StartTime:
 EndTime:
 No of cpus:
 OS NAME:

© 2008 - C-DAC, All rights reserved.

Steps for Reservation of Resources

1. Check the available free resources with valid parameters (StartTime and EndTime – duration for which the resource needs to be reserved). The input fields No. of CPUs and OS entries are optional.

Example: starttime=2009-04-02 17:06:53 endtime=2009-04-02 19:07:10 No. of CPUs=2 OSNAME=Linux

2. Choose the Available Process required for the job. Example: Available Procs=4

3. Select the required resource from the available list of resources.

4. Book the resources for reserving a resource for the requested period of time and process.

5. The reserved resources can be modified/canceled.

6. Once the reservation process is successfully completed, the Reservation ID is displayed and is made available in the Basic Job Submission page.

File browser

For the logged-in user, the File Browser lists files, such as the uploaded executables and Input/Output/Error files, along with their size and last modified information. It also allows deletion of files.

Job Management

Resources

File Browser

Accounting

Myproxy

Voms

Notices

Help

FAQs

Partner site

Remove Files

<input type="checkbox"/>	File Name	Size inBytes	Last Modified Date
<input type="checkbox"/>	BackUp2	12288	Jun 26, 2007 9:59:29 AM
<input type="checkbox"/>	BIO	4096	Mar 3, 2009 9:47:54 AM
<input type="checkbox"/>	Execs	4096	Feb 24, 2009 5:31:29 PM
<input type="checkbox"/>	logs	4096	Jun 26, 2007 11:28:50 AM
<input type="checkbox"/>	RSLFilesDir	4096	Feb 23, 2009 5:32:23 PM
<input type="checkbox"/>	TEST	4096	Feb 25, 2009 5:22:02 PM
<input type="checkbox"/>	XML	4096	Mar 11, 2009 5:34:32 PM
<input type="checkbox"/>	113048submit.xml	1530	Mar 12, 2009 2:42:47 PM
<input type="checkbox"/>	113049submit.xml	1487	Mar 12, 2009 2:43:04 PM
<input type="checkbox"/>	113050submit.xml	1549	Mar 12, 2009 2:45:54 PM

<< Prev

Delete

Reset

Next >>

© 2006 - CDAC, All rights reserved.

Accounting

This module provides Accounting information of the jobs that are submitted to GARUDA, such as no. of jobs submitted, and system parameters such as Memory usage, Virtual memory, Wall Time, and CPUtime. Lastonemonthdataisdisplayedbydefault.

MyProxy

MyProxy allows user to upload their Globus Certificates into Myproxy Server and the same can be used for initializing the Grid proxy on the Grid. If the certificate has been already generated for you, but you do not have access to the above-mentioned files, you can download it from GridFS machine (from \$HOME/.globus directory) using winscp/scp.

MyProxyInit

By default, the "Myproxy Init" option is enabled for the user. Upload proxy by entering valid inputs -User name, Grid-proxy Passphrase, User certificate file (usercert.pem), User key file (userkey.pem) andProxylifetime (168hoursisthedefaultvalue).

MyProxyGet

Grid proxy will be initialized on the Grid head node by providing the inputs - User name, MyproxyPassphraseandLifetimeofthecertificate.

VOMSProxy

The Virtual Organization Management System (VOMS) allows userstobelong to Virtual

Organizations (VOs), thereby allowing them to utilize resources earmarked for those VOs.

The user can also request for a new VO by using "Request for VO" link. VOMS proxy initialization with multiple roles is provided to the user, by selecting more than one entry on the Role combobox.

Steps to be followed to access GSRM from gridfs:

Login to
 gridfs (192.168.60.40) Upload
 your IGCA user certificate
 Initialize proxy with grid-
 proxy-init
 Set environmental variables, respectively for whichever client
 to be used. Run the SRM commands

GSRM Access points

pvfs2 (172.20.1.81) nodes should be used to just test all the available SRM
 client interfaces like StoRM, DPm, BestMan.

gridfs (192.168.60.40) nodes should, if the user wishes to use GSRM storage for job execution. Users can download/Upload input/output files into GSRM while submitting jobs from gridfs.

Following **Access mechanisms** are available at above mentioned nodes to access GSRM:

1. gridfs(192.168.60.40): gridfs is the Bangalore GARUDA head node. GSRM services can be accessed from the reusing StoRM command line interface.

If the user wants to use the clientSRM (StoRM Clients) from gridfs machine Create a valid user proxy using grid-proxy-init

Set the env variable for Globus location path

```
export GLOBUS_LOCATION=
```

```
GLOBUS_LOCATION:/usr/local/GARUDA/GLOBUS-
```

```
4.0.7/export PATH=$PATH:/opt/gsrM-client/srmv2storm/bin
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/gsrM-
```

```
client/cgsi_soap/libRun the clientSRM command
```

2. pvfs2 (172.20.1.81): pvfs2 is the GSRM testing node with the following client interfaces installed.

Bestman Java APIs

DPM APIs

3. GSRM WebClient is accessible from many of the user machines reachable to GSRM server (xn05.ctsf.cdac.org.in), using URL--<https://xn05.ctsf.cdac.org.in/>

GSRM Client

Interfaces StoRMComm

and Line Client

1. StoRM command line client

format: clientSRM <requestName> <requestO

ptions>

2. To get help for clientSRM commands: cli

entSRM-h

3. CommandtopingtoGSRMserver:cli

entSRMping-e<GSRMendpoint>

BestmanCommandLineClients

1. CommandtopingtoGSRMserver

srm-ping-serviceurlhttpg://xn05.ctsf.cdac.org.in:8446/dpm/ctsf.cdac.org.in/home/garud

2. UploadfiletoGSRMserver

srm-copy<srcurl><targeturl><serviceurl>

Pre-requisitesforusingSOAcompiler

1. JavaRunTimeEnvironment(JDK1.6+)
2. WebBrowserwithJavawebstartsupport

CompilerGUI

Compiler Entry Form

Project Path
(Project Dir should contains following dis: bin,lib,include,src)

Operating System

Compiler Name

Compiler Options

<ul style="list-style-type: none">-fcprop-registers-fcse-follow-jumps-fcse-skip-blocks-fcx-limited-range-fdata-sections	<input type="button" value="Add>>"/> <input type="button" value="Remove"/>	<ul style="list-style-type: none">-fmudflapth-fbranch-probabilities-fcx-limited-range
---	---	---

Special Libraries

<ul style="list-style-type: none">ThreeBook_2.5_IntelFourBook_2.5_IntelOneBook_2.5_IntelTwoBook_2.5_Intel	<input type="button" value="Add>>"/> <input type="button" value="Remove"/>	<ul style="list-style-type: none">FourBook_2.5_Intel
--	---	--

The values with yellow background are from different Resource Managers

The users are required to adhere to following directory structure.
Application ParentDir-src/,bin/,lib/,include/

1) Login

This method is for logging into the GARUDA.

Inputs

username	MyProxyUser Name
password	MyProxyPassword
lifetime	Indicateshowlongisthe proxy'slifetime

Output

Proxystring	ProxyissuedbytheMyproxyserver
Loginstatus	Indicates the status of the operationLastLoginTime
	Giveswhenthisuserwaslastlogg
edinCurrentLoginTime	
	Givesusersloggingintime

2) uploadProxy

This method uploads a proxy that is generated using other tools, to the MyProxyServer.

Inputs

username	MyProxyUser Name
password	MyProxyPassword
proxyBytes	Existing proxy file is given as bytearray

Output

uploadStatus	Indicates the status of the operation
--------------	---------------------------------------

3) storeCredential

This method is used for uploading the credential that is the PKCS12 certificate directly to the MyProxy Server. It will convert the PKCS12 to certificate and stores in server for users to download the proxy until it expires.

Inputs

username	MyProxyUser Name
password	MyProxyPassword
p12Bytes	PKCS12 file as bytearray

Output

storeStatus Indicatesthestatusoftheoperation

Result:

Thusthedevlopmentofsecuredapplicationsusingbasicsecuritymechanismsavailablein Globusisexecutedsuccessfully.

6.DevelopaGridportal,whereusercansubmitajobandgettheresult.

ImplementitwithandwithoutGRAMconcept

OBJECTIVE:

TodevelopaGridportal,whereusercansubmitajobandgettheresultandtoimplementitwith andwithoutGRAMconcept.

PROCEDURE:

1. Opening the workflow editor

The editor is a Java Web start application download and installation is only a click.

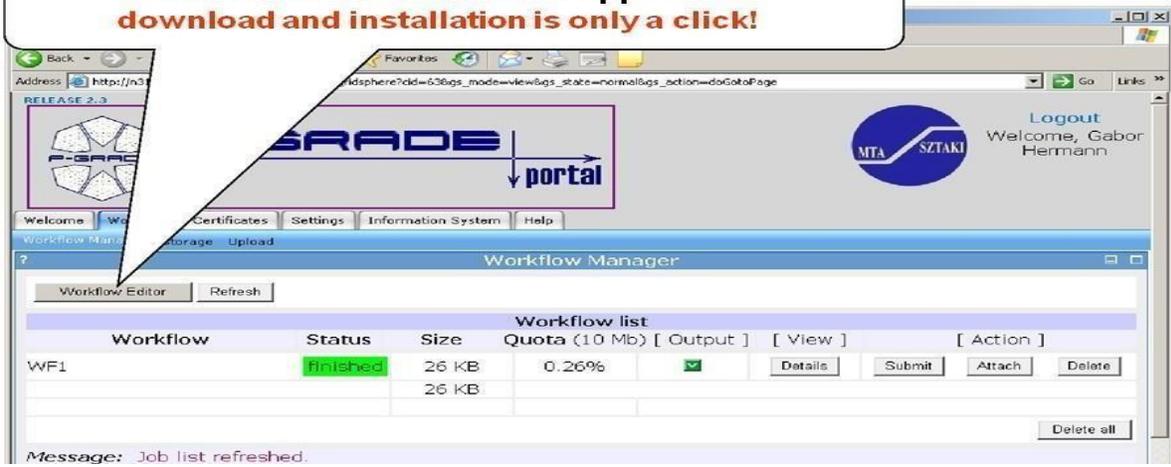


2. Java Webstart

applicationDownloadandins

tall

The editor is a Java Webstart application
download and installation is only a click!

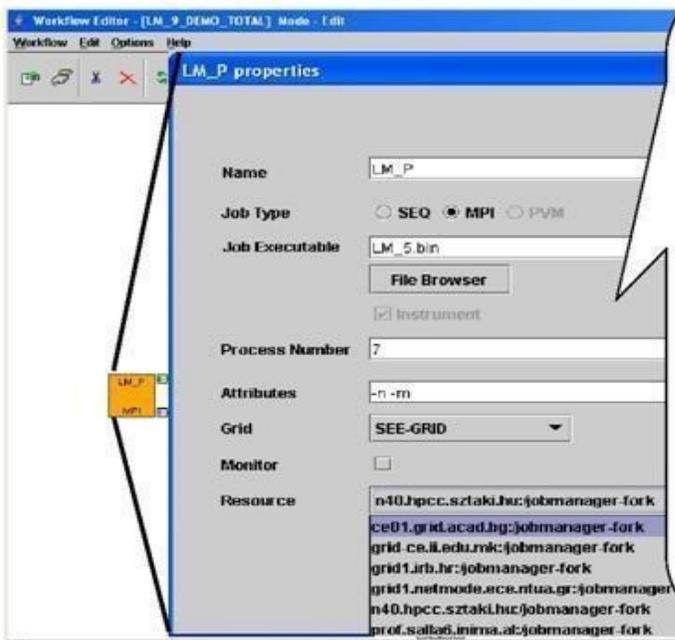


Workflow Manager

Workflow	Status	Size	Quota (10 Mb)	[Output]	[View]	[Action]
WF1	finished	26 KB	0.26%	<input checked="" type="checkbox"/>	Details	Submit Attach Delete
		26 KB				

Message: Job list refreshed.

3. Jobpropertywindow:



4. TheinformationsystemcanqueryEGEEandGlobusinformationsystems

The information system portlet can query EGEE and Globus information systems

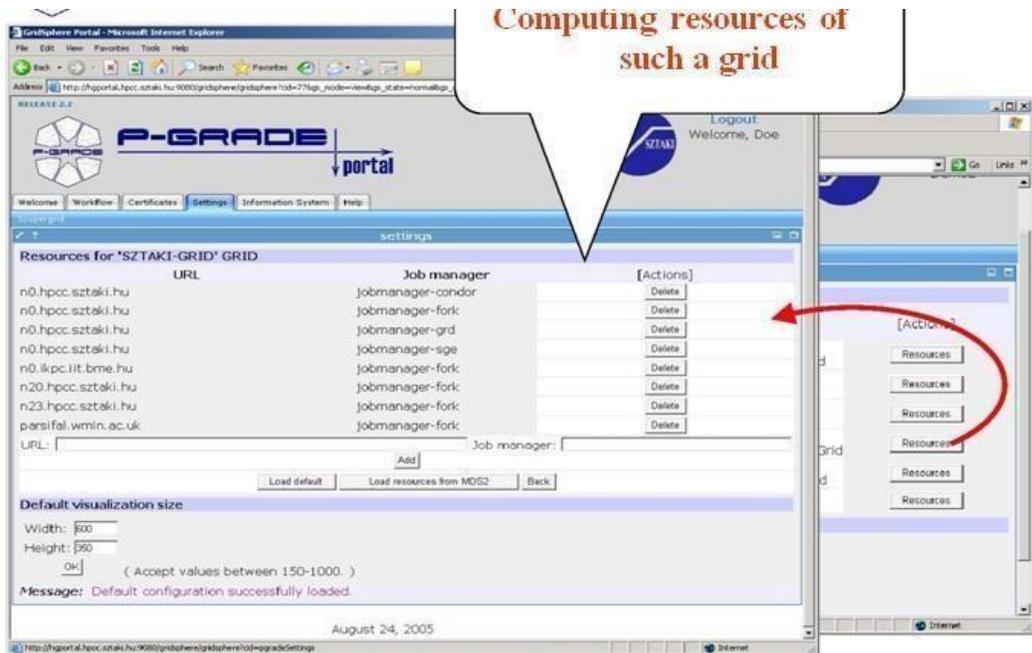
The screenshot shows the GridSphere Portal Information System Monitor. The interface includes a navigation menu with 'Welcome', 'Workflow', 'Certificates', 'Settings', 'Information System', and 'Help'. The main content area is titled 'Monitor' and displays a table of grid sites for the 'EGEE' grid and 'All' VO. The table is divided into 'Computing Element' and 'Storage Element' sections. A dropdown menu for 'Select Grid' is open, showing options like 'brwf', 'calice', 'cdf', 'cesga', 'cms', 'compass', 'compchem', 'CosmoGrid', 'id', 'scms', and 'tech'. The table lists various sites such as 'aegis01-phy', 'alberta-log2', 'beijing-cnic-log2-ig54', 'beijing-log2', 'belgrid-ud', 'bg-inme', 'bg01-ipp', 'bg02-im', 'bg04-acad', 'bham-log2', 'bifi', 'bitfabgs', 'bristol-pp-log', and 'budapest'.

Site Name	Computing Element						Storage Element		
	total	Free	Usage	Running	Waiting	Load	Total	Available	Usage
aegis01-phy	32	10	69%	12	0	0%	106,971 GB	79,263 GB	26%
alberta-log2	50	0	100%	0	0	0%	1,221 TB	308,592 GB	75%
beijing-cnic-log2-ig54	32	32	0%	0	0	0%	62,87 GB	56,992 GB	9%
beijing-log2	8	8	0%	0	0	0%	2 KB	1 KB	50%
belgrid-ud	12	12	0%	0	0	0%	N/A	N/A	-
bg-inme	20	20	0%	0	0	0%	37,355 GB	37,299 GB	0%
bg01-ipp	19	1	95%	13	5	28%	N/A	N/A	-
bg02-im	4	4	0%	0	0	0%	32,944 GB	20,169 GB	39%
bg04-acad	11	11	0%	0	0	0%	32,844 GB	27,149 GB	17%
bham-log2	132	107	19%	0	0	0%	1,639 TB	1,518 TB	7%
bifi	2	2	0%	0	0	0%	103,52 GB	98,274 GB	5%
bitfabgs	101	99	2%	0	4	100%	417,777 GB	407,123 GB	3%
bristol-pp-log	2	2	0%	0	0	0%	174,885 GB	164,261 GB	6%
budapest	95	24	75%	70	0	0%	1,36 TB	1,305 TB	4%

5. Listofavailablegrids



6. Computing resources of such a grid

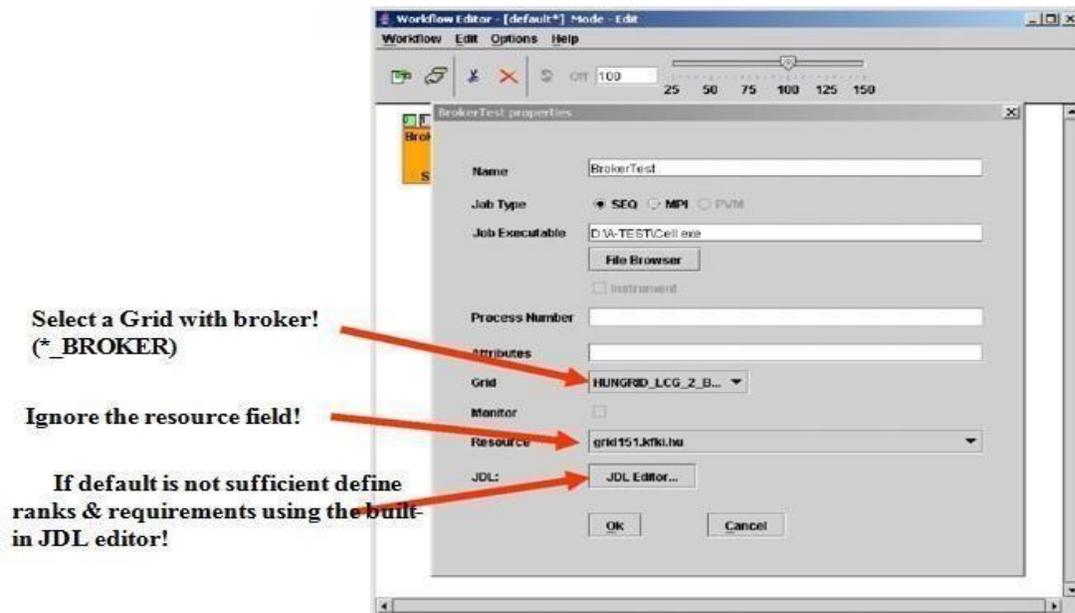


7. Broker resource selection

->Select a Broker Grid for the job

->Specify extra ranks and requirements for the job in Job description language.

->The broker will find the best resource for your job.



8. Defining input/output data for jobs

letype

Input: required by

the

jobOutput: produced by

hejobFilelocation:

local: mydesktop

remote: grid storage

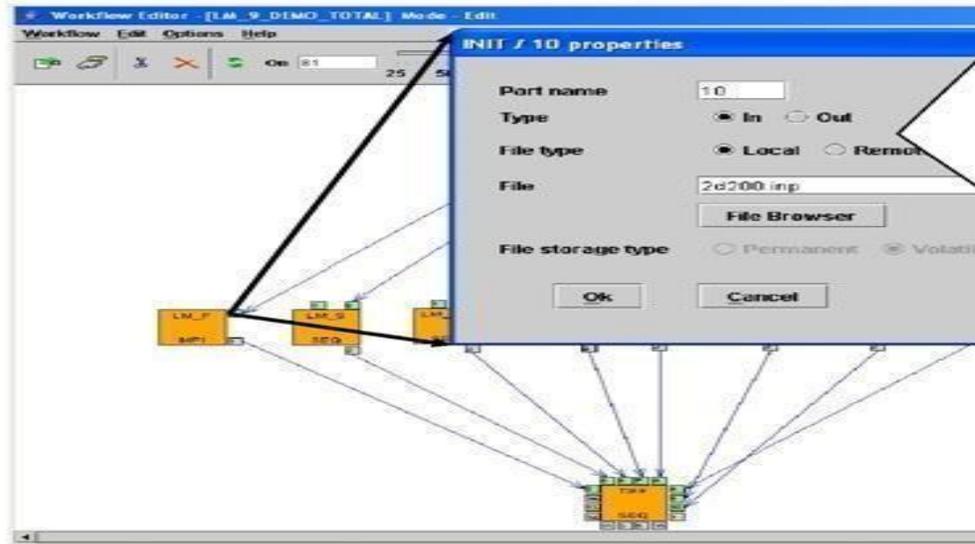
resourceFilename:

Uniquenameofthefile

Filestoragetype:

Permanent:final resultofWF

Volatile:onlyusedforinter-jobdatatransfer

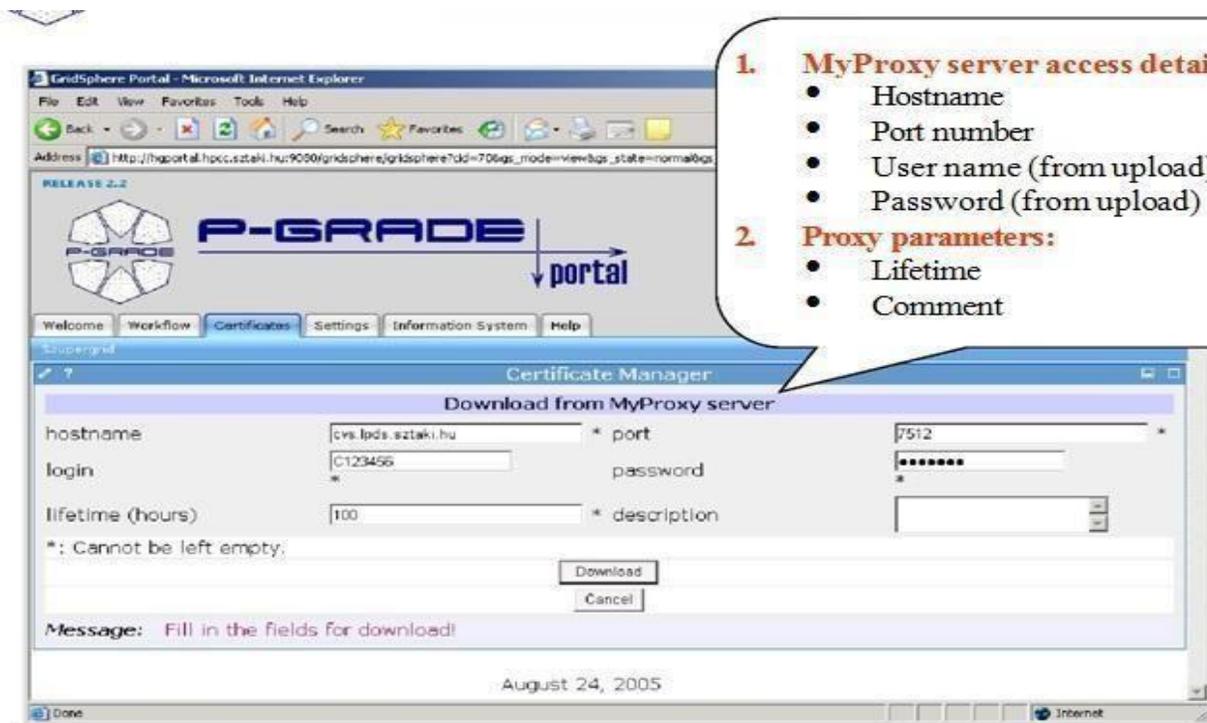


9. Executing workflows with the P-

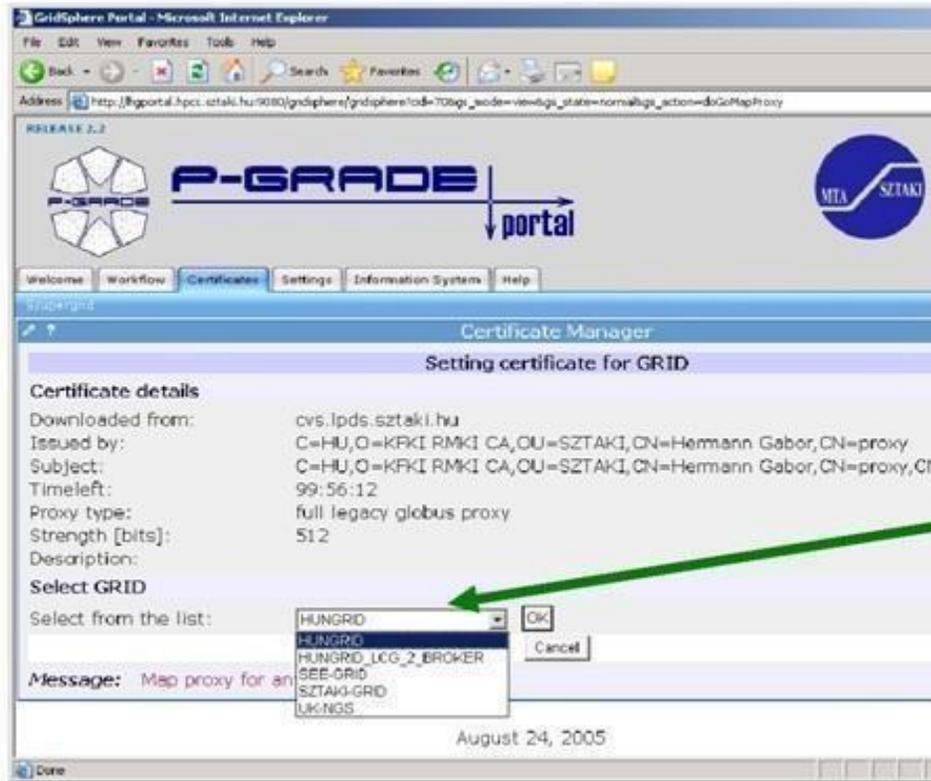
GradeportalDownloadproxies



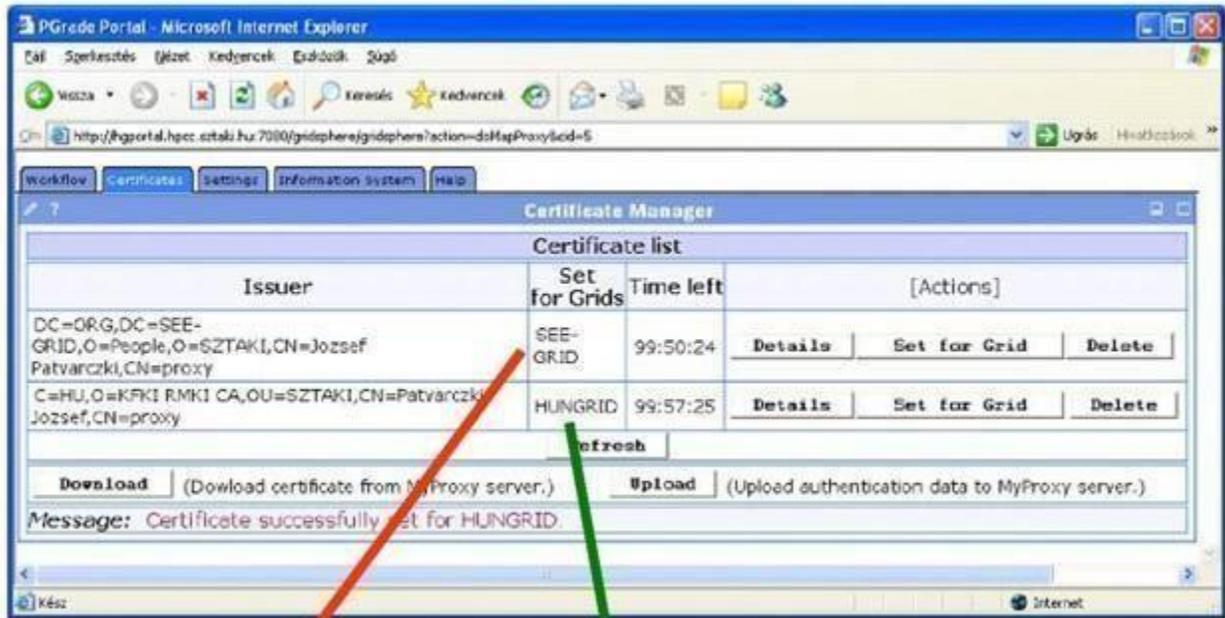
10. Downloading proxy



11. Associating the proxy with a grid



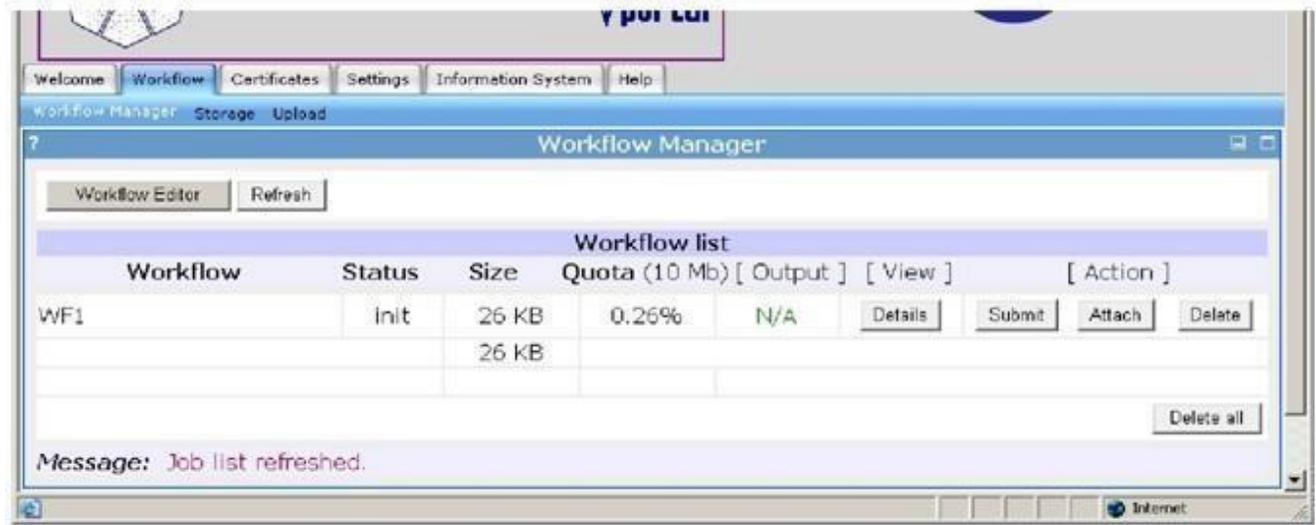
12. Browsing Proxies



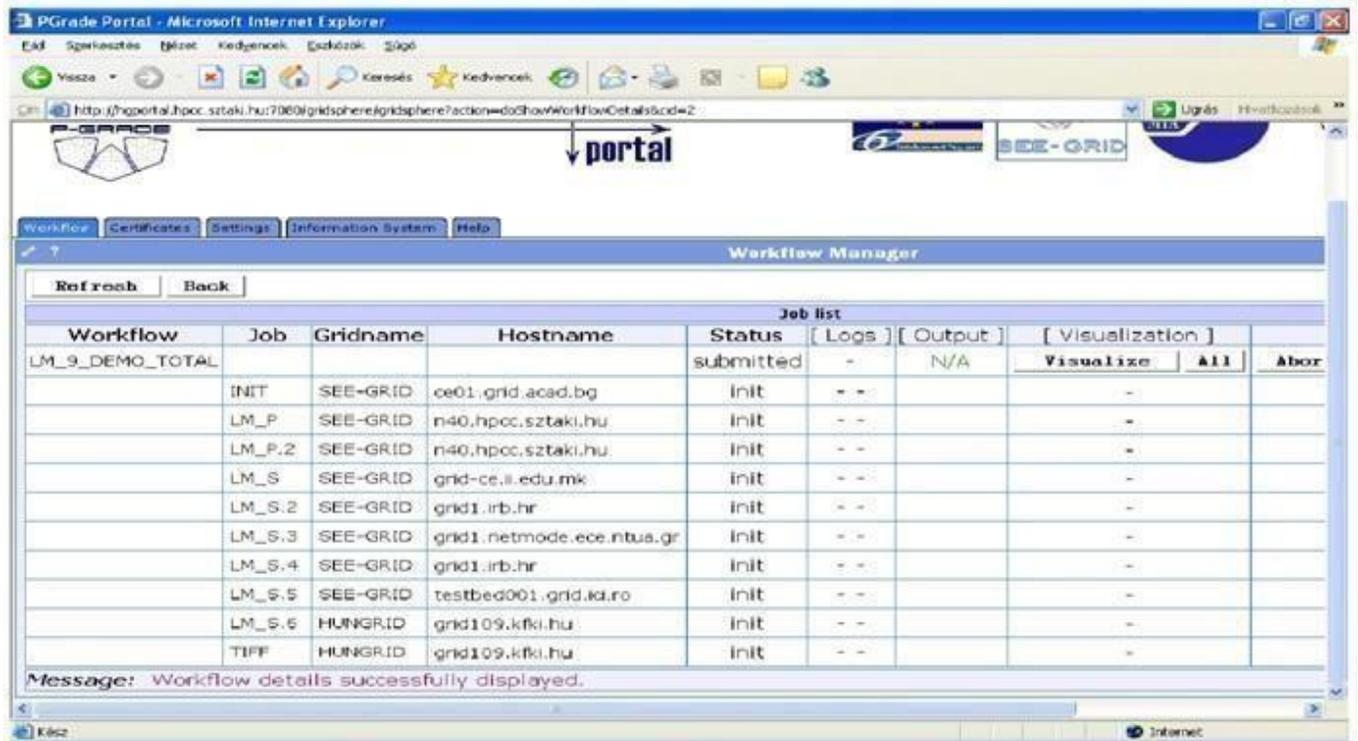
13. Workflow

executionWorkflowport

let



14. Observation by the workflow portlet



The screenshot displays the PGrade Portal interface in Microsoft Internet Explorer. The browser window title is "PGrade Portal - Microsoft Internet Explorer". The address bar shows the URL: <http://hgportal.hpcc.sztaki.hu:7060/gridisphere/gridisphere?action=doShowWorkflowDetails&cd=2>. The page features a navigation menu with "Workflow", "Certificates", "Settings", "Information System", and "Help". Below the menu is a "Workflow Manager" section with "Refresh" and "Back" buttons. The main content area displays a "Job list" table with columns for Workflow, Job, Gridname, Hostname, Status, Logs, Output, and Visualization. A message at the bottom states: "Message: Workflow details successfully displayed."

Workflow	Job	Gridname	Hostname	Status	Logs	Output	Visualization
LM_9_DEMO_TOTAL				submitted	-	N/A	Visualize All Abort
	INIT	SEE-GRID	ce01.grid.acad.bg	init	- -		-
	LM_P	SEE-GRID	n40.hpcc.sztaki.hu	init	- -		-
	LM_P.2	SEE-GRID	n40.hpcc.sztaki.hu	init	- -		-
	LM_S	SEE-GRID	grid-ce.ii.edu.mk	init	- -		-
	LM_S.2	SEE-GRID	grid1.irb.hr	init	- -		-
	LM_S.3	SEE-GRID	grid1.netmode.ece.ntus.gr	init	- -		-
	LM_S.4	SEE-GRID	grid1.irb.hr	init	- -		-
	LM_S.5	SEE-GRID	testbed001.grid.kd.ro	init	- -		-
	LM_S.6	HUNGRID	grid109.kfki.hu	init	- -		-
	TIFF	HUNGRID	grid109.kfki.hu	init	- -		-

PGrade Portal - Microsoft Internet Explorer

http://hgportal.hpc.sztaki.hu:7000/gridsphere/gridsphere?action=doGoToPage&id=2

Workflow Manager

Refresh Back

Workflow	Job	Gridname	Hostname	Status	Logs	Output	[Visualization]		
							Visualize	All	Abort
LM_9_DEMO_TOTAL	INIT	SEE-GRID	ce01.grid.acad.bg	running	-	N/A	-	-	-
	LM_P	SEE-GRID	n40.hpc.sztaki.hu	init	-	-	-	-	-
	LM_P.2	SEE-GRID	n40.hpc.sztaki.hu	init	-	-	-	-	-
	LM_S	SEE-GRID	grid-oe.ii.edu.mk	init	-	-	-	-	-
	LM_S.2	SEE-GRID	grid1.irb.hr	init	-	-	-	-	-
	LM_S.3	SEE-GRID	grid1.netmode.ece.ntua.gr	init	-	-	-	-	-
	LM_S.4	SEE-GRID	grid1.irb.hr	init	-	-	-	-	-
	LM_S.5	SEE-GRID	testbed001.grid.icl.ro	init	-	-	-	-	-
	LM_S.6	HUNGRID	grid109.kfki.hu	init	-	-	-	-	-
	TIFF	HUNGRID	grid109.kfki.hu	init	-	-	-	-	-

Message: Job list refreshed.

PGrade Portal - Microsoft Internet Explorer

http://hgportal.hpc.sztaki.hu:7000/gridsphere/gridsphere?action=doGoToPage&id=2

Workflow Manager

Refresh Back

Workflow	Job	Gridname	Hostname	Status	Logs	Output	[Visualization]		
							Visualize	All	Abort
LM_9_DEMO_TOTAL	INIT	SEE-GRID	ce01.grid.acad.bg	finished	-	N/A	-	-	-
	LM_P	SEE-GRID	n40.hpc.sztaki.hu	init	-	-	-	-	-
	LM_P.2	SEE-GRID	n40.hpc.sztaki.hu	init	-	-	-	-	-
	LM_S	SEE-GRID	grid-oe.ii.edu.mk	finished	-	-	-	-	-
	LM_S.2	SEE-GRID	grid1.irb.hr	finished	Out	-	-	-	-
	LM_S.3	SEE-GRID	grid1.netmode.ece.ntua.gr	finished	Out	-	-	-	-
	LM_S.4	SEE-GRID	grid1.irb.hr	finished	Out	-	-	-	-
	LM_S.5	SEE-GRID	testbed001.grid.icl.ro	finished	Out	-	-	-	-
	LM_S.6	HUNGRID	chemgrid3.chemres.hu	finished	Out	-	-	-	-
	TIFF	HUNGRID	grid109.kfki.hu	init	-	-	-	-	-

Message: Job list refreshed.

PGate Portal - Microsoft Internet Explorer

http://hportal.hpc.sztaki.hu:7000/gridschare/gridschare?action=doGotoPage&id=2

Workflow Certificates Settings Information System Help

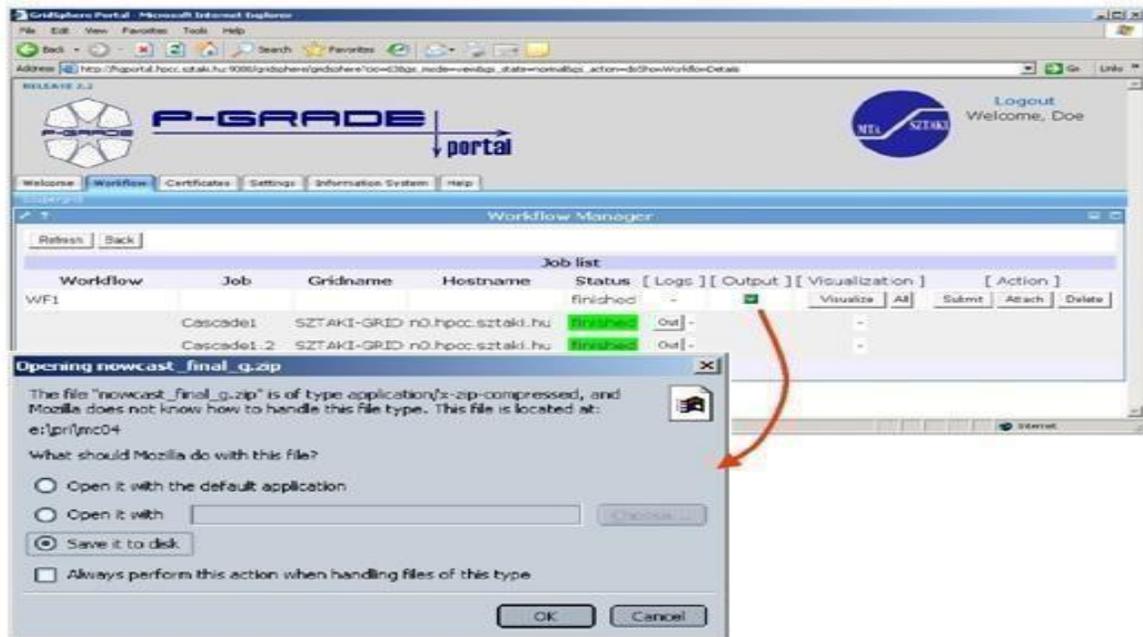
Workflow Manager

Refresh Back

Workflow	Job	Gridname	Hostname	Status	Job list		
					Logs	Output	Visualization
LM_9_DEMO_TOTAL	INIT	SEE-GRID	ce01.grid.acad.bg	finished	Err	Being zipped..	Visualize All
	LM_P	SEE-GRID	n40.hpc.sztaki.hu	Finished	Out	-	Visualize
	LM_P.2	SEE-GRID	n40.hpc.sztaki.hu	Finished	Out	-	Visualize
	LM_S	SEE-GRID	grid-ce.il.edu.mk	Finished	Out	-	-
	LM_S.2	SEE-GRID	grid1.irb.hr	Finished	Out	-	-
	LM_S.3	SEE-GRID	grid1.netmode.ece.ntua.gr	Finished	Out	-	-
	LM_S.4	SEE-GRID	grid1.irb.hr	Finished	Out	-	-
	LM_S.5	SEE-GRID	testbed001.grid.icl.ro	Finished	Out	-	-
	LM_S.6	HUNGRID	chemgrid3.chemres.hu	Finished	Out	-	-
	TIFF	HUNGRID	grid109.kfki.hu	Finished	Out	-	-

Message: Job list refreshed.

15. Downloading the results



RESULT

Thus the development of a Grid portal, where user can submit a job and get the result and to implement it with and without GRAM is executed successfully

CLOUD COMPUTING

Program on SaaS

1 Create a word document of your class timetable and store locally and on the cloud with doc and pdf format. (use www.zoho.com and docs.google.com)

Steps:

With Google Docs, you can create and edit text documents right in your web browser—no special software is required. Even better, multiple people can work at the same time, you can see people's changes as they make them, and every change is saved automatically.

To start, you need a document to work with.

In this section, you learn how to:

- Create a new document
- Import and convert old documents to Docs

Create a new document

You can

create a new document right in Docs or in Google Drive. In D

ocs, click **Create new document**.

In Drive, click **New > Google Docs > Blank document** or **From a template**.

Import and convert old documents to Docs

If you have existing text documents, such

as Microsoft® Word® or Adobe® PDF files, you can import and convert them to Docs.

- **Go to Drive.**
- **Click **New > File Upload** and choose a text document from your computer. Supported file formats include .doc, .docx, .dot, .html, plaintext (.txt), .odt, and .rtf.**
- **Right-click the file you want to convert and select **Open with > Google Docs**.**

Converting your document from another program creates a copy of your original file in Docs format. You can then edit it in your browser like any other document.

Create a class timetable

Shared documents

1. Open the file you want to share.
2. Click Share.
3. Enter the email addresses or Google Groups you want to share with.

Note: If you can't add people outside your company, see your GSuite administrator.

4. Choose what kind of access you want to grant people:
 - o **Can edit**—Collaborators can add and edit content as well as add comments.
 - o **Can comment**—Collaborators can add comments, but not edit content.
 - o **Can view**—People can view the file, but not edit or add comments.

Click Send.

Everyone you shared the document with receives an email with a link to the document.

2 Create a spreadsheet which contains employee salary information and calculate gross and total salary using the formula

DA=10% OF

BASIC HRA=30% OF BASIC

PF=10% OF BASIC IF

BASIC <= 3000 12% OF BASIC

IF

BASIC > 3000 TAX=10% OF BAS

IC IF BASIC <= 1500

=11% OF BASIC IF BASIC > 1500 AND BASIC <= 2500

=12% OF BASIC IF BASIC > 2500

(use www.zoho.com and

docs.google.com) **NET_SALARY=BASIC_SALAR**

Y+DA+HRA-PF-TAX

If you're accustomed to creating your spreadsheets using an office suite or software like Microsoft Excel, you won't have any issue in creating a Google Spreadsheet. Google Spreadsheet works

thesameasExcel, andyoucandomostoftheimportantspreadsheettaskswithit.YoucanuseGoogleSpreadsheetdirectlyfromyourwebbrowserorfromitsmobile app.

1. Sign into Google Sheets. Visit docs.google.com/spreadsheets and sign in with your Google or Gmail account. Your Gmail account gives you free access to Google Sheets.
2. View your existing sheets. Upon logging in, you will be brought to the main directory. If you already have existing spreadsheets, you can see and access them from here.
3. Create a new spreadsheet. Click the large red circle with a plus sign on the lower right corner. A new window or tab will be opened with the web-based spreadsheet.
4. Name the spreadsheet. "Untitled spreadsheet" appears on the top left corner. This is the current name of the spreadsheet. Click on it, and a small window will appear. Type in the name of the spreadsheet here, and click the "OK" button. You will see the name immediately change.
5. Work on the spreadsheet. You can work on Google Sheets much like how you would work on Microsoft Excel. There's a header menu and a toolbar with functions very similar to those of Microsoft Excel.
 1. calculate gross and totals using the formula
 2. $DA = 10\% \text{ OF BASIC}$
 3. $HRA = 30\% \text{ OF BASIC}$
 4. $PF = 10\% \text{ OF BASIC IF BASIC} \leq 3000$
 5. $12\% \text{ OF BASIC IF BASIC} > 3000$
 6. $TAX = 10\% \text{ OF BASIC IF BASIC} \leq 1500$
 7. $= 11\% \text{ OF BASIC IF BASIC} > 1500 \text{ AND BASIC} \leq 2500$
 8. $= 12\% \text{ OF BASIC IF BASIC} > 2500$
 9. (use www.zoho.com and docs.google.com)
 10. $NET_SALARY = BASIC_SALARY + DA + HRA - PF - TAX$
6. There's no need to save with Google Sheets as everything you do is automatically saved at regular intervals.
7. Exit the spreadsheet when you're finished. If you're done with your current document, you can just simply close the window or tab. Everything is saved automatically. You can access your document from Google Sheets or Google Drive.

3. Prepare a ppt on cloud computing –introduction , models, services ,and Architecture ppt should contain explanations, images and at least 20 pages (use www.zoho.com and docs.google.com)

Step 1: Login with your gmail id, at <http://docs.google.com/>

Step 2: Once you login, you will have a workspace area to work with your documents, spreadsheets and presentations. Just below the logo, you will find the "Create New" button, when you click on that, you will have all the available options. Select Presentation there.

Step 3: Now you have an empty presentation being created in your workspace. The look and feel doesn't look like a web application at all!

Step 4: You can go to the Format Menu item and change the presentation Theme or Background. There are a lot of templates and themes available!

Step 5: As you would do on any desktop office client, you can click on the new slide button, and select the Slide Design.

Step 6: If you need to insert a drawing, Shape or an Image, You can go to the Insert Menu item and select the same. Once you select image, You have options to select the image as a URL image or even a web upload!

Step 7: If you need to insert Tables to your slide deck, you can use the Table Menu item, and select the number of Rows and Columns.

Step 8: Following the above steps prepare presentation on cloud computing.

Step 9: You can now share the presentation through email attachment, give a link to the presentation.. so that one can watch it online in Google Docs and even Embed this presentation onto your website too..!

4. Create your resume in a neat format using Google and Zoho cloud Programs on PaaS

Steps:

With Google Docs, you can create and edit text documents right in your web browser—no special software is required. Even better, multiple people can work at the same time, you can see people's changes as they make them, and every change is saved automatically.

To start, you need a document to work with.

In this section, you learn how to:

- Create a new document
- Import and convert old documents to Docs

Create a new document

You can

create a new document right in Docs or in Google Drive. In D

ocs, click Create new document.

In Drive, click New > Google Docs > Blank document or From a template.

Import and convert old documents to Docs

If you have existing text documents, such

as Microsoft® Word® or Adobe® PDF files, you can import and convert them to Docs.

- Go to Drive.
- Click New > File Upload and choose a text document from your computer. Supported files include .doc, .docx, .dot, .html, plaintext (.txt), .odt, and .rtf.
- Right-click the file you want to convert and select Open with > Google Docs.

Converting your document from another program creates a copy of your original file in Docs format. You can then edit it in your browser like any other document.

Create/design a neat resume

Shared documents

1. Open the file you want to share.
2. Click Share.
3. Enter the email addresses or Google Groups you want to share with.

Note: If you can't add people outside your company, see your GSuite administrator.

4. Choose what kind of access you want to grant people:
 - o **Can edit**—Collaborators can add and edit content as well as add comments.
 - o **Can comment**—Collaborators can add comments, but not edit content.
 - o **Can view**—People can view the file, but not edit or add comments.

Click Send.

Everyone you shared the document with receives an email with a link to the document.

1 Write a Google app engine program to generate n even numbers and deploy it to google cloud

Steps:

1. Install Eclipse IDE for Java EE Developers, version 4.6 or later:
2. If you have the Google Plugin for Eclipse installed, complete the [migrating from GPE](#) procedures.
3. Installing Cloud Tools for Eclipse
4. To install the plugin:

5. Drag the install button into your running Eclipse workspace:



6. Or from inside Eclipse, select Help > Eclipse Marketplace... and search for Google Cloud.
7. Restart Eclipse when prompted.

App engine Program to generate even numbers using java servlet application

```
<!DOCTYPEhtml>
<htmlxmlns="http://www.w3.org/1999/xhtml"lang="en">
  <head>
    <metahttp-equiv="content-type"content="application/xhtml+xml;charset=UTF-8"/>
    <title>HelloAppEngine</title>
  </head>

  <body>
    <h1>HelloAppEngine!</h1>
    <formaction="/hello"method="get">
      Enter1stNumber<inputtype="text"name="n1"><br>
      <inputtype="Submit"value="Submit">
    </form>

  </body>
</html>

importjava.io.IOException;
```

```

import
javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import
javax.servlet.http.HttpServletRequest;
import
javax.servlet.http.HttpServletResponse
;

@WebServlet (
    name="HelloAppEngine",urlPatterns={"/h
ello"}
)
public class HelloAppEngine extends HttpServlet {

    @Override
    public void doGet (HttpServletRequest request,
        HttpServletResponse response) throws IOException {

        response.setContentType ("text/
plain"); response.setCharacterE
ncoding ("UTF-8");

        int a1 = Integer.parseInt (request.getParamet
er ("a1")); for (int i = 0; i < a1; ++i)
        {

```

```
if(i%2==0)
    response.getWriter().print("\t"+i);
```

```
}
```

```
}
```

```
}
```

2 Googleappengineprogrammultiplytwomatrices

[ReferProgram1Stepstoperformmatrixmultiplication]

3 Googleappengineprogramtovalidateuser;createdatabaselogin(username,password)inmysqlanddeploytocloud

[Followtheprogram1stepsto deployflexibleuservalidationsampleprogramfromgitrepository]

4 Write a Google App Engine program to display the n largest numbers from the given list of numbers and deploy it into Google Cloud

[Refer Program 1 Steps to find the n largest numbers from the given list of numbers]

EXPERIMENTNO:159

**5. GoogleappengineprogramtovalidatetheuserUsemysqltostoreuserinfoanddeploy
onthecloud**