



DELIVERABLE D4.1
APPENDIX D

MIDDLEWARE TEST PROCEDURE

WP4 International Testbed Organisation

Document Filename: **CG-4-D4.1-004-TEST.doc**
Work package: **WP4 International Testbed Organisation**
Partner(s): **CYFRONET, CSIC, DEMOKRITOS, LIP**
Lead Partner: **LIP**
Config ID: **CG-4-D4.1-004-TEST-1.1**
Document classification: **PUBLIC**

Abstract:

This document describes the test and evaluation procedure used to verify the CROSSGRID testbed middleware.



Delivery Slip

	Name	Partner	Date	Signature
From	Jorge Gomes	LIP	17 May 2002	
Verified by				
Approved by	Jesus Marco	CSIC	27 May 2002	

Document Log

Version	Date	Summary of changes	Author
1.0-DRAFT_A	10/01/2002	Draft version	Jorge Gomes
1.0-DRAFT_B	14/04/2002	Second draft version	Jorge Gomes
1.1	31/05/2002	After IRB	Jorge Gomes

CONTENTS

1. INTRODUCTION.....	4
1.1. DEFINITIONS ACRONYMS AND ABBREVIATIONS.....	4
1.2. REFERENCES.....	5
2. OBJECTIVES.....	6
3. ORGANIZATION.....	7
3.1. ALPHA PHASE.....	7
3.2. BETA PHASE.....	7
4. INFRASTRUCTURE.....	8
4.1. TEST AND VALIDATION TESTBED.....	8
4.2. INSTALLATION AND CONFIGURATION ISSUES.....	9
4.3. SCHEDULING TESTS	9
5. GLOBUS AND DATAGRID MIDDLEWARE.....	10
6. TEST PROCEDURE.....	11
6.1. ENTITIES ENTITLED TO REQUEST A TEST.....	12
6.2. PROBLEM CLASSIFICATION	12
6.3. REQUIREMENTS TO MIDDLEWARE DEVELOPERS.....	13
6.4. REQUIREMENTS TO TEST REQUESTORS.....	14
6.5. VALIDATION STEPS	14
6.6. TESTING PROCEDURE	14
7. TOOLS.....	16
8. ANNEXES.....	17
8.1. ANNEX 1 - COMPONENT TEST REQUEST FORM.....	17

1. INTRODUCTION

The Grid middleware to be deployed in the CROSSGRID testbed is a complex set of components that will be evolving rapidly. These components are being developed by several teams in many countries, working in the context of several projects.

The Grid middleware is the basis upon which the Grid applications are built and executed therefore the validation and test of the middleware is of extreme importance to build a stable testbed for the development of new Grid applications and services.

Continuous release of new and updated middleware components is expected hence a permanent validation and test activity will be required.

The procedure here detailed shall be applied to all middleware used in the context of the CROSSGRID testbed, which includes both middleware produced internally to the project and middleware developed in other projects but planned to be deployed in CROSSGRID.

An exhaustive in-depth evaluation of the middleware is out of the scope of this procedure. The complexity and large amount of source code prevents this kind of analysis. Instead this procedure will be concentrated in the evaluation of the components functionality and interoperation.

1.1. DEFINITIONS ACRONYMS AND ABBREVIATIONS

Definitions

Gatekeeper	A computer system configured to be an interface between local computing facilities and the Grid. The gatekeeper receives remote requests for job submission through the Grid middleware and submits these requests to a local farm using the farm job scheduling software.
Worker Node	A computer system configured to perform computing operations such as running user programs.
Storage Element	A computer system configured as a data storage server. Each set of Computing Elements most have one or more corresponding Storage Elements from which data is retrieved for processing and stored in the end.
Installation Server	A computer system configured to provide the components to manage and store centrally the site fabric configuration information. The installation server software provides a configuration language and a central repository of configuration specifications and RPMs used to install and upgrade automatically individual Grid nodes.
Replica Catalogue	The replica catalogue server contains a database with the mapping between logical file names and the location of the corresponding physical copies. Associated to the logical name additional attributes are stored as metadata.
Root information index server	The system hosting the root index server of the information tree. This is the point from where the resource broker starts searching for suitable resources.

Grid scheduler	Component of the middleware that has the responsibility of managing the grid resources in such a way that applications are conveniently, efficiently and effectively executed. The scheduler includes the logging and bookkeeping service, resource broker and job submission service.
Virtual organization	A mechanism to group users according with affiliation with a particular organization such as a research institute, experiment or field of research. This eases the granting of permissions to access resources.

Acronyms and Abbreviations

EDG	The EU DataGrid Project IST-2000-25182
WP	Work Package
VO	Virtual organization

1.2. REFERENCES

- [1] Software Engineering Tools, CrossGrid, CG-5.1-DES-0008-1-0-DRAFT-A
- [2] Evaluation of Testbed Operation, DataGrid-06-D6.4-0109-1-11
- [3] Software Release Policy, DataGrid-12-D12.3-331457-2-1

2. OBJECTIVES

The main objective of this procedure is to improve the stability of the CROSSGRID testbed by:

- Reducing the number of middleware problems that would pass unnoticed to developers;
- Preventing the wide deployment of components with critical problems;
- Assuring the conformance of the middleware with the specifications;
- Detecting design errors;
- Identify potential bottlenecks;
- Detecting software security issues;
- Evaluating the middleware documentation;
- Testing installation kits;
- Testing the middleware components;
- Providing feedback to developers;
- Producing recommendations to system managers and users.

3. ORGANIZATION

The test process shall be divided in two phases:

- 1) **ALPHA** – Internal validation and testing by a reduced set of WP4 (Testbed) members;
- 2) **BETA** – Testing by a reduced set of knowledgeable users using applications.

3.1. ALPHA PHASE

In phase ALPHA validation and testing activities will be divided by a group of teams composed of WP4 members. In order to be efficient the ALPHA test group shall be restricted to a small number of knowledgeable members in close contact with the integration teams.

In this phase the delivered middleware and corresponding documentation is evaluated and tested with focus on functionality and interoperation with other existing components.

Test activities will be divided according with the expressions of interest and areas of expertise. The evaluation and testing of the same components can be delegated to more than one team.

Once components are available for ALPHA testing they will be published in the task 4.4 web site (<http://www.lip.pt/crossgrid/task4>), and a request for testers will be released among the ALPHA team members.

3.2. BETA PHASE

The phase BETA testing activities will be divided by a small group of invited knowledgeable users. The collaboration of the WP1 application deployment experts will be very important for the installation of test applications, and to verify compliance of the middleware with the application installation scripts and procedures. Other middleware developers strongly interested in the component may be invited to participate in the BETA test phase.

In this phase the delivered middleware and corresponding documentation is evaluated and tested by end users and other developers.

Once components are available for BETA testing they will be published in the task 4.4 web site, and a request for testers will be released among the BETA team members.

4. INFRASTRUCTURE

4.1. TEST AND VALIDATION TESTBED

A separated testbed infrastructure is required for the test activities, in order not to disturb the “production” and “development” testbeds over which stable middleware is being used and applications are being developed. This implies the deployment of three CrossGrid testbed infrastructures.

The three testbeds are:

- **Production Testbed** – A stable testbed used by the application developers to test, develop and run applications. The middleware deployed in the production testbed should not change very often.
- **Development Testbed** – An infrastructure to develop, test and integrate middleware that later will become the basis of the next release of the production testbed.
- **Test and Validation Testbed** – A small infrastructure used to test and validate new or modified middleware components under a controlled environment before they are deployed in the production testbed.

Each site involved in middleware test activities will need hosts dedicated to these activities in addition to the hosts dedicated to the production or development testbeds. The minimum list of equipment for a test and validation site is as follows:

- A PC to host a “Gatekeeper”;
- A PC to host a “Worker Node”;
- A PC to host a “Storage Element”;
- A PC to host an “Installation Server”.

The equipment above must be fully dedicated to test activities.

In addition shared services required to support this separated testbed such as a virtual organization, Grid scheduler, root information index server and replica catalogue will be deployed. However initially these grid services will be shared with the development testbed thus reducing the setup time, complexity and effort required for the first infrastructure.

The test infrastructure should be kept small with up to four sites experienced in the deployment and test of Globus and DataGrid middleware. The test infrastructure should be maintained closed to all end users until each phase ALPHA is finalized. Then it should be open only to authorized BETA test users.

4.2. INSTALLATION AND CONFIGURATION ISSUES

The middleware installation on test sites will be highly dynamic changing according with the requirements of the components to be tested. Different test sites may have different configurations simultaneously depending on nature of the tests being performed at each site. However to perform tests across sites the configurations of the systems involved must be the same. This introduces the necessity for strong coordination between sites to assure that required resources are available when necessary.

Each site will appoint a contact person responsible for local installation and configuration issues.

Different site configurations will be required based on the tests to be conducted, therefore the installation server will play an important role by storing pre-defined configurations which will allow to change the site configuration according to test requirements with a reduced management effort.

4.3. SCHEDULING TESTS

Tests will need to be schedule to assure a smooth workflow. Different tests may require different software versions and configurations therefore tests of different components may not be able to be performed simultaneously.

Scheduling will be based on:

- Priority of the request. The priority is an indication of the necessity of the test results for the requester.
- Number of sites required to perform the tests. A test may require one or more sites.
- Amount of configuration work required to prepare the sites.
- Availability of human and material resources for testing at each site.
- Availability of suitable test programs. Test programs may need to be developed.

5. GLOBUS AND DATAGRID MIDDLEWARE

The Globus grid toolkit developed by the Globus project will be the basis over which CrossGrid new services and applications will be developed. The Globus toolkit is a set of components being used by several ongoing research projects such as DataGrid to develop computing grids. This choice will enable basic compatibility across several testbeds therefore extending the grid coverage in Europe and promoting the deployment of common middleware avoiding the duplication of development efforts.

The DataGrid project is developing middleware that will enable the usage of grid technologies for applications with very large data requirements using Globus as the starting point.

CrossGrid has similar requirements to those of DataGrid however it will extend these technologies to cover the requirements of interactive and parallel applications, therefore both Globus and DataGrid middleware will be used within the testbed constituting the major source of external middleware.

Initially the testbed will be based on middleware produced by these two external sources to which CrossGrid developed components will be added later. This means that middleware produced by other projects will be installed in all CrossGrid testbed sites and therefore needs to be tested jointly with CrossGrid middleware components and applications before deployment.

Feedback on problems found in external components will be provided to the developers whenever possible thus contributing to the reliability and usability the common middleware.

Collaboration with DataGrid on testing of common middleware components will be established.

6. TEST PROCEDURE

The procedure for middleware test and validation is outlined in the picture below that describes the flow of actions from the development to the final deployment. The steps are as follows:

- 1) The WP4 integration team identifies or receives new or updated middleware components.
- 2) A component test request is issued to the task 4.4 verification and quality control team. This can be a request for testing an individual component or a set of integrated components.
- 3) The test is performed in a separated testbed infrastructure.
- 4) Feedback on identified problems is sent to the developers.
- 5) If possible developers will answer with immediate corrections.
- 6) Feedback on integration problems is sent to the integration team. Once the test is complete a full report is also sent to the integration team.
- 7) If problems in the WP4 component integration are found then the integration team may issue immediate corrections.
- 8) The integration team deploys the tested middleware in the production testbed.
- 9) Task 4.4 will monitor the stability of the production testbed through bug reports and monitoring information aiming to identify the need for further tests and to make the test and verification process more efficient.

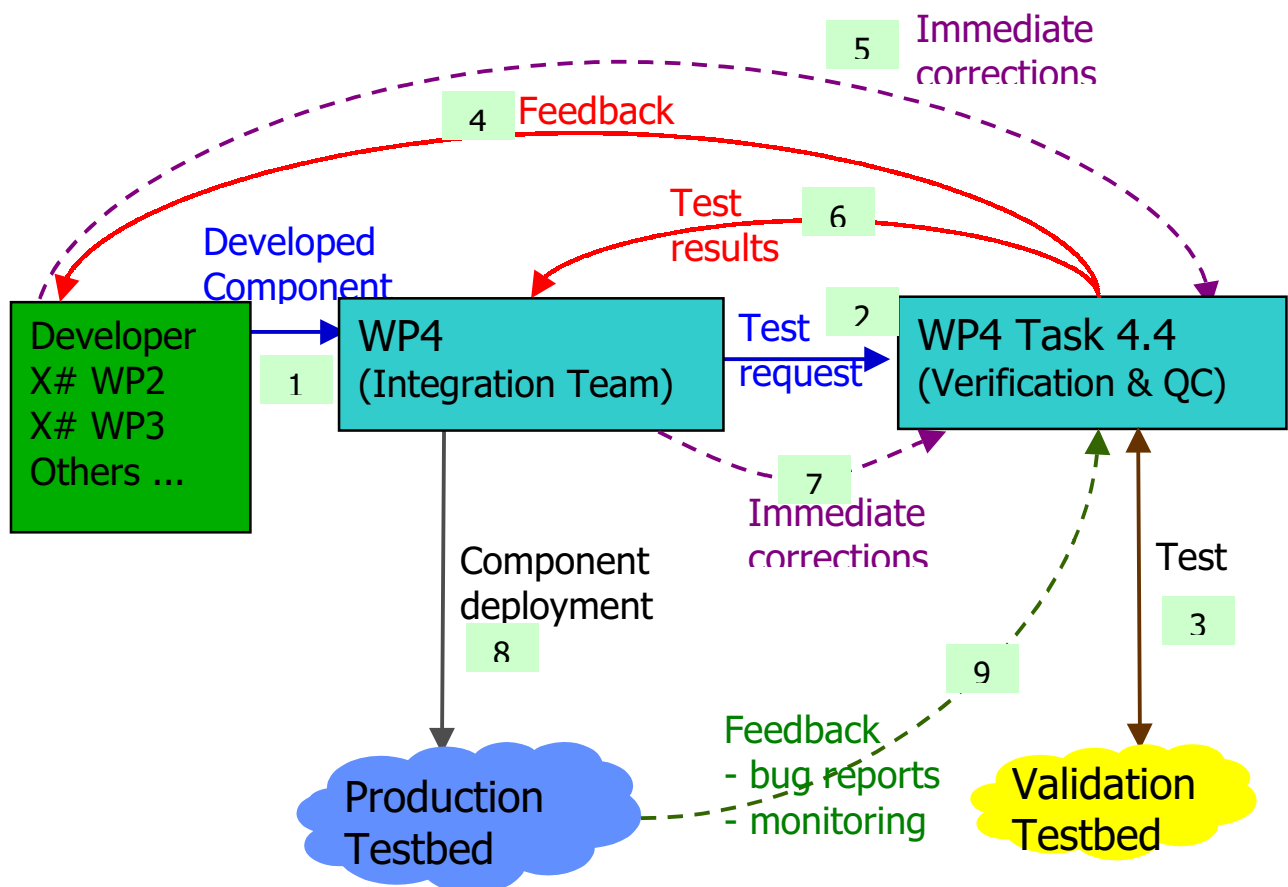


Figure 1 Testing procedure steps.

The WP4 integration team chooses the middleware to be integrated in the CROSSGRID testbed. In order to test and validate each component or fully integrated middleware release the integration team issues a test request to the testbed verification and quality control team. The verification and QC team performs the requested validation tests using a separated testbed infrastructure and reports back the results to the integration team and whenever possible to the developers. Corrections introduced by the developers and the integration team during the test procedure will be accepted for retest whenever possible. Upon receiving the test results the integration teams decides whether to deploy the components in the production testbed.

Checking the production testbed bug reports and monitoring data will help verify the effectiveness of this test procedure and the need to perform further tests. The monitoring and bug report information will be gathered from the WP4 monitoring and bug tracking services.

Middleware developed in the CROSSGRID context is to be delivered to the WP4 integration team with the basic functionality tests already performed by the corresponding work-package integration task. This will allow WP4 task 4.4 to focus on testing the delivered middleware integrated with other components and verify their correct interoperation.

These activities should be performed in close collaboration with the developers and integration teams from WP2, WP3 and WP4. Direct contact with them will help resolve detected problems and clarify doubts that may appear contributing to reduce the test duration and effort.

6.1. ENTITIES ENTITLED TO REQUEST A TEST

Validation and test of middleware can be requested by the following entities:

- The CROSSGRID integration team;
- The WP4 manager;

6.2. PROBLEM CLASSIFICATION

Problems found while performing the validation tests will be classified regarding their severity and priority, these two attributes define the urgency and impact of the problem and help to determine which issues must be addressed first.

For each severity level described below a guideline action is recommended. The severity classification is as follows:

- **Critical** – When problem causing a failure of the complete software system, subsystem or program within the system is found.
ACTION: The component should not be released. Further tests on the component and components depending on it will not be performed until the problem is corrected.
- **High** – When a problem that does not cause a failure, but causes the system to produce incorrect, incomplete, inconsistent results or impairs the system usability is found.

ACTION: Developers MUST perform the required corrections but the component can pass to the next phase. A deadline will be given to the developers to make the corrections;

- **Medium** – When a problem that does not cause a failure, does not impair usability, and does not interfere in the fluent work of the system and programs has been found.

ACTION: Corrections should be performed but since the component critical functionalities have been successfully tested and the component can be deployed. However the problem must be mentioned in the documentation.

- **Low** – When the problem found is aesthetic is an enhancement or is a result of non-conformance to a standard.

ACTION: The problem does not interfere with the functionality or operation therefore the component can be deployed. However the problem must be mentioned in the documentation.

The priority classification is as follows:

- **Immediate** – When the problem should be resolved immediately.
- **High** – When the problem should be resolved as soon as possible in the normal course of development activity, before the software is released.
- **Medium** – When the problem should be repaired after all other more serious problems have been fixed.
- **Low** – When the problem can be resolved in a future major system revision or not resolved at all.

6.3. REQUIREMENTS TO MIDDLEWARE DEVELOPERS

CROSSGRID components to be tested must be delivered with:

- Software installation kit and download instructions;
- Test software and corresponding source code. The provided test software will be used to perform the first basic tests, and may be changed to perform more comprehensive or intensive tests. The test software must at least cover the basic component functionality and interoperation with other lower level components.
- Documentation:
 - Installation, configuration and test manual.
 - User manual when applicable. User interfaces must be extensively documented.
 - Development manual when applicable. APIs and dependencies on other components must be extensively documented.
 - Developer contact information including name, partner, address, E-mail, phone, fax.

Non-CROSSGRID components can be delivered without the elements described above, although their absence will make the test procedure harder and longer.

6.4. REQUIREMENTS TO TEST REQUESTORS

Test of components based on faulty lower level services should not be requested. However test requests will be accepted if the low level problems do not collide with the features to be tested.

Components to be tested must be delivered with:

- Component test request form (Annex 1) correctly filled.

6.5. VALIDATION STEPS

- Verify the middleware documentation;
- Verify the delivered middleware against the software requirements;
- Compare the delivered middleware interfaces against the architecture documents;
- Test the middleware.

6.6. TESTING PROCEDURE

- Alpha test phase:
 - Test each delivered component alone if possible;

These tests should be performed with test software provided with the component. If needed these tests will be enhanced to perform a thorough evaluation on issues such as reliability or performance.

For non-CROSSGRID components basic tests will have to be developed when unavailable.
 - Test each component locally integrated with other components;

The objective of these tests is to verify the functionality and interoperation of the component with other deployed components from CROSSGRID and external sources.

Programs will have to be developed to test the middleware features available to user applications. These programs will be submitted as a succession of real user jobs with increased complexity using intensively the middleware features in several combinations.

The middleware performance will be evaluated using benchmark software developed by CROSSGRID WP3 as soon as available.
 - Test each component across test sites;

In these tests the component is deployed in several test sites. Jobs of increased complexity are submitted between sites in order to test the component behaviour in situations as similar to real production as possible.
 - Problems found are immediately reported back to the developers for correction and a deadline will be given for the submission of required corrections after which the report will be sent to the WP4 integration team.

- Once the Alpha test phase is finished:
 - A report is sent to the test requester;
 - The Beta phase is started only when the Alpha phase is finished successfully or with low or medium severity problems;
 - The Beta phase might be started for components with high problems at request of an entitled entity when properly justified.

- Beta test phase:
 - Knowledgeable users and other middleware developers are invited to use the testing infrastructure and perform their own tests using preferably real applications.
 - Testers should receive:
 - Information on what new components are being tested;
 - All middleware documentation available;
 - Information on which sites can be used for testing;
 - Recommendations and results from the ALPHA phase.
 - Users must report back immediately any problems found.

- Once the Beta test phase is finished:
 - A final report is sent to the test requester and developer.

7. TOOLS

Task 4.4 will provide the following tools for test and validation activities.

- Web site for testbed and activity coordination;
- Mailing list to exchange information between testers;
- Bug reporting system for testers;
- Test software:
 - General test suite;
 - Specific tests for specific functionalities.

8. ANNEXES

8.1. ANNEX 1 - COMPONENT TEST REQUEST FORM

A component test request form must be filled by the requester for each component to be tested. The description form can be found in the following pages.

Description of the form fields:

Component name:	Name of the component to be tested.
Date:	Request date.
Priority:	Indicative priority from 1 to 5, being 5 the maximum priority.
Requester:	Person requesting the test.
Name:	Name of the person;
Address:	Requester address;
WP:	Work package (eg. WP4);
Task:	Number of the task within the work package;
Partner:	Partner identifier (eg. LIP)
E-mail:	E-mail address;
Telephone:	Telephone number;
Fax:	Fax number;
Developer:	Responsible for the development of the component to be tested.
Name:	Name of the person;
Address:	Developer address;
Project:	CROSSGRID, DATAGRID, Globus etc;
WP:	Work package (eg. WP4);
Task:	Number of the task within the work package;
Partner:	Partner identifier (eg. LIP)
E-mail:	E-mail address;
Telephone:	Telephone number;
Fax:	Fax number;
Download URL:	URL where the component kit can be downloaded;

Description form fields (continuation):

Delivered with:

Test programs:	Test programs delivered with the component;
Installation manual:	Installation manual delivered (Y/N);
User manual:	User manual delivered (Y/N);
Development manual:	Development manual delivered (Y/N);

Requirements document:	URL of the corresponding software requirements document.
Component short description:	Short functionality description.
Component dependencies:	Dependencies on other software components, libraries etc.
Features to be tested:	Description of software features to be tested.
Features not to be tested:	Description of software features that should not be tested.

CROSSGRID Component Test Request Form**(1 of 2)**Component name: Date of request: / / Request priority (1-5):

Requester:

Name: Address: WP: Partner: Task: E-mail: Telephone: Fax:

Developer:

Name: Address: Project: WP: Partner: Task: E-mail: Telephone: Fax: Download URL:

Delivered with:

Test programs:

Installation manual (Y/N): User manual (Y/N): Development manual (Y/N):

Software requirements document URL:

Component short description:

CROSSGRID Component Test Request Form

(2 of 2)

Component dependencies:



Features to be tested:



Features NOT to be tested:

