



SPI

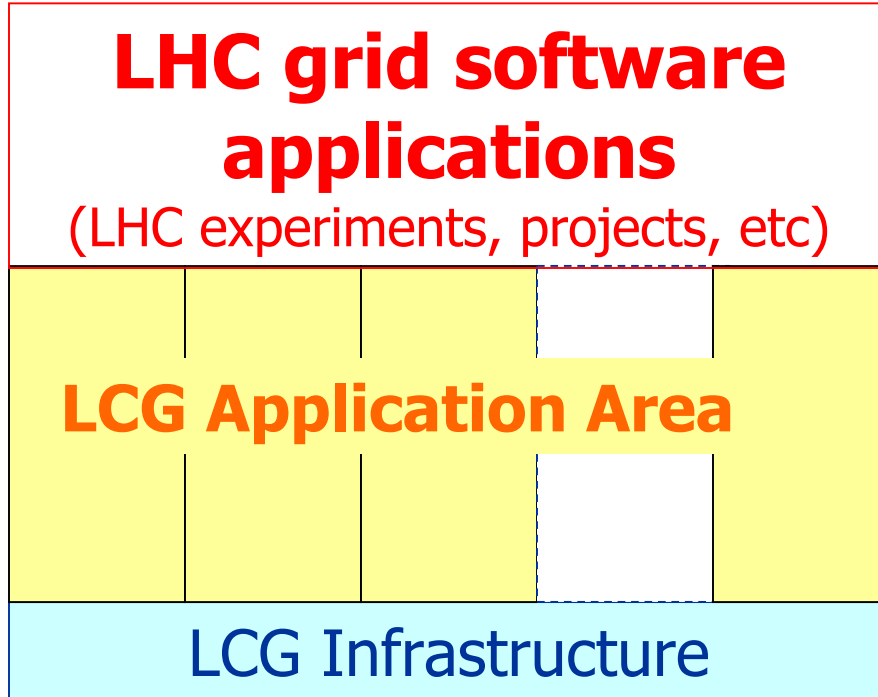
Software Process & Infrastructure for LCG

Project Overview

LHCC Review
24-25 November 2003

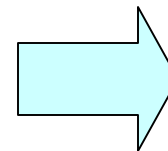
Alberto AIMAR

Project Context of LCG SPI



- Common services
- Similar ways of working (process)
- Tools, templates, training
- General QA, tests, integration, release

- LCG Application Area software projects**
- POOL: Persistency
 - SEAL: Core common software
 - PI: Physics Interfaces
 - SIMU: Simulation
 - ...etc...



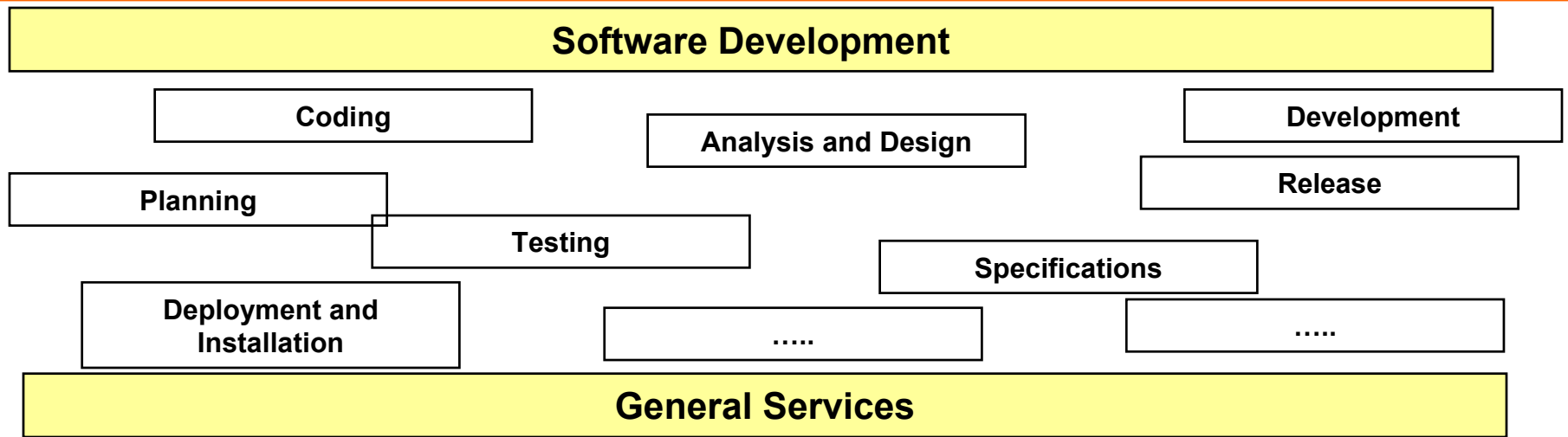
LCG SPI project



Project Context of LCG SPI (2)

- RTAG2: “Software Management RTAG”
 - General recommendations
 - All LCG projects must adopt the same set of tools, standards and procedures
 - Adopt commonly used open-source or commercial software when easily available
 - Avoid “do it yourself solutions”
 - Avoid commercial software, if may give licensing problems
- If each project needs an infrastructure, many projects need it even more...
 - Tools, standards and procedures
 - Try to avoid complexity

Infrastructure Software Development



- a. Provide general services needed by each project
 - CVS repository, Web Site, Software Library
 - Mailing Lists, Bug Reports, Collaborative Facilities
- b. Provide solutions specific to the software phases
 - Tools, Templates, Training, Examples, etc.

SPI Project Guidelines

- Have different and separated services
 - Simple solutions, easy to learn, commonly needed services
 - Leave any process for later
- Establish simple deliverables
- Work with the users
- Develop as little as possible
- Do not re-invent the wheel
 - Everything is done starting from, or using, existing infrastructure
 - Talk to LHC experiments, IT division, Big projects (G4, Root, etc)
- We did not start to provide tools for requirements, design, etc.
- We started from development-related work
 - repository, delivery, releases, testing, bug report, etc

→ The rest of the talk describes SPI services

http://spi.cern.ch



LHC Computing Grid Project > LCG Applications Area > LCG Software Process & Infrastructure

SPI - Software Process & Infrastructure for LCG

Updated: 13-Oct-2003 12:21

SPI Quick Links

- [SPI Home](#)
- [SPI Index Page](#)
- [SPI Workbook](#)

SPI Services Links

- [LCG Workbook](#)
- [Savannah Portal](#)
- [External Software](#)
- [Software Testing](#)
- [Download](#)
- LCG App. Area**
- [Home Page](#)

News

[Older News](#)

- 10 Oct 2003 [SPI material](#) for the LCG App Area Internal Review 2003
- 23 Sep 2003 Added [Software Download](#) section
- 3 Sep 2003 Added [Quality Assurance](#) section
- 1 Sep 2003 [SPI Workbook](#) task-oriented documentation

Search SPI web

[Tips](#)

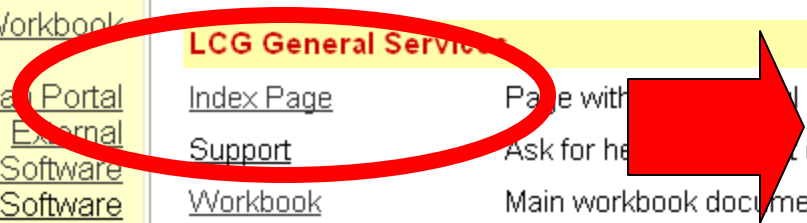
LCG General Services

- [Index Page](#) Page with all links to LCG developers and users
- [Support](#) Ask for help, comments, feedback and bug to SPI
- [Workbook](#) Main workbook documentation from the LCG Projects
- [Savannah Portal](#) Project portal service for LCG Projects

LCG Software Services

- [Software Download](#) Software produced by the LCG, download and instructions
- [External Software](#) External Software installed for the LCG Projects

LCG Development Services





[LHC Computing Grid Project](#) > [LCG Applications Area](#) > [LCG Software Process & Infrastructure](#)

SPI - Software Process & Infrastructure for LCG

Updated: 13-Oct-2003 12:14

SPI Quick Links

[SPI Home](#)
[SPI Index Page](#)

[SPI Workbook](#)

SPI Services Links

[LCG Workbook](#)

[Savannah Portal](#)
[External Software](#)
[Software Testing](#)

[Download](#)

LCG App. Area

[Home Page](#)

Index Page

Infrastructure

CVS Service

[How to use the LCG CVS service](#)

Project Portal

[Projects Portal for LCG](#)
[User registration](#)

Software Library

[LCG External Software service](#)
[LCG Software Distribution](#)
[How to install 3rd party software](#)

Workbook

[Workbook for LCG developers](#)

SPI Templates

Software Development

LCG Policies

- [Policy and Tools Page](#)
- [Setting up environment](#)
- [CVS Directory Policy](#)
- [Build Directory Policy](#)
- [FAQ](#)

Building

[SCRAM configuration, build and release](#)

[NICOS automatic build system](#)

Testing

[Software Testing in LCG App.Area](#)

[Coding conventions](#)

LCG App. Area Projects

Direct access to all projects
[Doxygen](#) | [LXR](#) | [ViewCVS](#)

PI Project

[Home page](#) | [CVS repository](#) | [Bug reports](#)

POOL Project

[Home page](#) | [Project Portal](#) | [CVS repository](#) | [Bug reports](#) | [Mailing Lists](#)

SEAL Project

[Home page](#) | [CVS repository](#) | [Bug reports](#)

Simulation Project

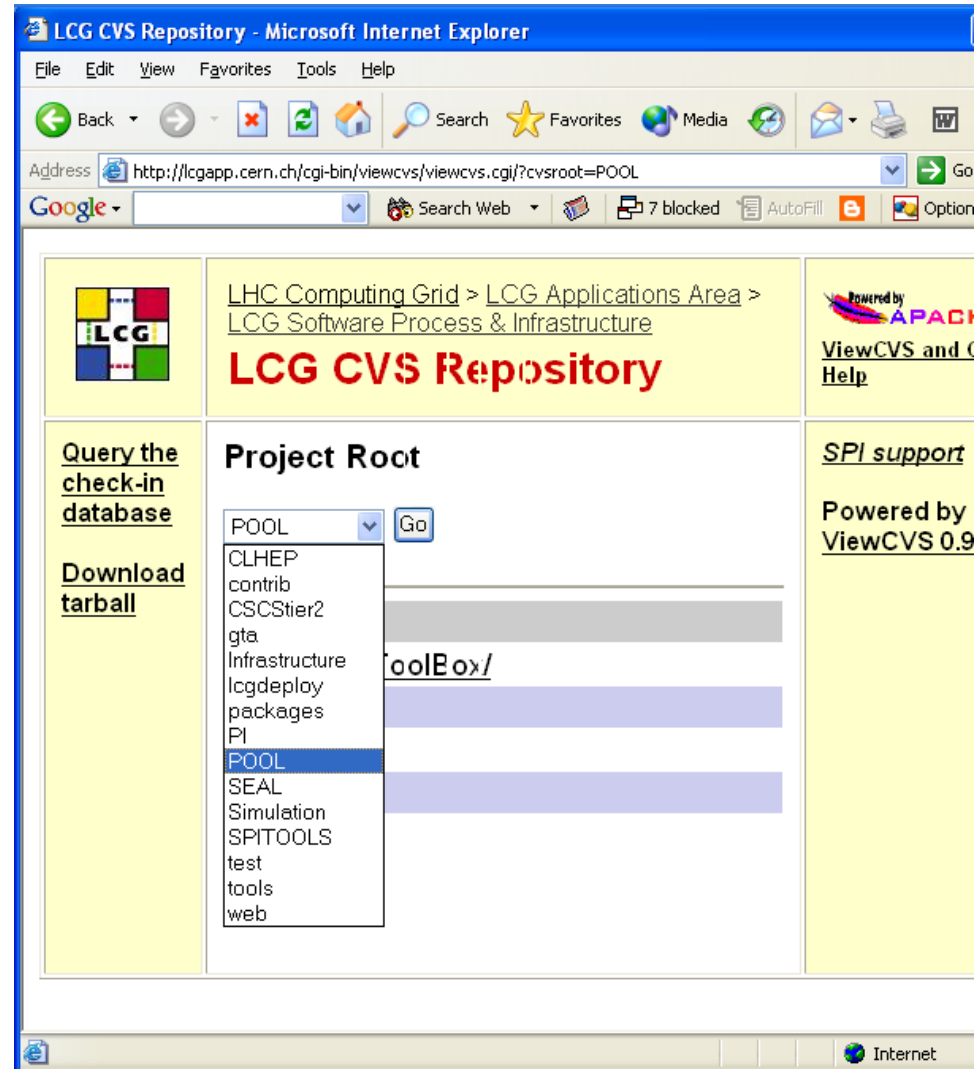
CVS Repository and Delivery Area

CVS repository

- A central CVS repository available to all projects
- Any project just needs to ask for it, and declare its users permissions
- Managing mirroring and backups
- Temporary solution when the IT CVS service was not ready

Delivery areas

- AFS area
- an area to install software created by projects in the LCG application area (lcg/apps)
- an area for external and third party software (lcg/external)
- an area for software under evaluation (lcg/contrib)
- We started with the most basic services




Code Documentation


- Features of interest
 - Code browsing
 - Code searching
 - Code information
 - Various design/data diagrams
- Any LCG project will have them available as part of the infrastructure
 - Doxygen
 - extracts comments, builds documentation and diagrams
 - LXR
 - connects the source code and allows search in the code
 - ViewCVS
 - allows browsing of the CVS repository from the web



LCG Code Documentation Service - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://lcgapp.cern.ch/doxygen/>

 LHC Computing Grid Project > LCG Application

LCG Code Documentation using Doxygen

Doxygen can help you in three ways:

1. It can generate an on-line documentation from a set of documented source files. The output can be in HTML, PostScript, hyperlinked PDF, compressed HTML, or plain text directly from the sources, which makes it very easy to browse the source code.
2. Doxygen can be configured to extract the code for visualization. This is very useful to quickly find your way in large code bases. Elements can be visualized by means of UML diagrams, which are all generated automatically.
3. You can even 'abuse' doxygen for creating diagrams.

Doxygen is run every night over internal and external versions of the CVS in that moment. For any question, please contact support@cern.ch.

Follow this links to learn more:

1. [Brief How-To generate documentation with Doxygen](#)
2. [Manual for Doxygen 1.2.18 \(local copy\)](#)

Done

Code documentation: Doxygen

LCG Code Documentation Service - Microsoft Internet Explorer

Address http://lcgapp.cern.ch/doxygen/POOL/POOL_1_3_3/doxygen/classEventHeader__dict.html

[Main Page](#) | [Namespace List](#) | [Class Hierarchy](#) | [Alphabetical List](#) | [Compound List](#) | [File List](#) | [Namespace Members](#) | [Compound Members](#) | [File Members](#)

EventHeader_dict Class Reference

[List of all members.](#)

Public Member Functions

```
EventHeader_dict ()  
EventHeader_dict ()
```

Static Public Member Functions

```
void * constructor_1031 (void *, const std::vector< void * > &)  
void * constructor_1032 (void *)  
void destructor (void *)  
void * constructor_1031 (void *, const std::vector< void * > &)  
void * constructor_1032 (void *)  
void destructor (void *)
```

Constructor & Destructor Documentation

EventHeader_dict::EventHeader_dict()
Definition at line 25 of file [src/Examples/Libraries/AthenaExample](#)

References [constructor_1031\(\)](#), [constructor_1032\(\)](#), [destructor](#), [seal::reflect::PUBLIC](#), and [seal::reflect::VIRTUAL](#).

EventHeader_dict::EventHeader_dict()

Done

LCG Code Documentation Service - Microsoft Internet Explorer

Address http://lcgapp.cern.ch/doxygen/POOL/POOL_1_3_3/doxygen/classpool_1_1PersistencySvc_1_1DatabaseHandler.html

pool::PersistencySvc::DatabaseHandler Class Reference

```
#include <DatabaseHandler.h>
```

Collaboration diagram for pool::PersistencySvc::DatabaseHandler:

```
graph TD  
    string -- m_PFN --> DatabaseHandler  
    DatabaseConnection -- m_DBC --> DatabaseHandler  
    Transaction -- m_transaction --> DatabaseHandler  
    FileDescriptor -- m_fileDescriptor --> DatabaseHandler  
    IStorageExplorer -- m_storageExplorer --> DatabaseHandler  
    IStorageSvc -- m_storageSvc --> DatabaseHandler
```

[List of all members.](#)

Public Member Functions

```
DatabaseHandler (IStorageSvc &storageSvc, IStorageExplorer &storageExplorer,  
Session *session, long technology, const std::string &fid, const std::string &pfm, long  
accessmode)  
Constructor. Connects to the database.  
~DatabaseHandler ()  
Destructor. Disconnects from the database.
```

bool startTransaction ()



Configuration and Build System



- The tools selected by LCG was SCRAM
- All projects are currently building with Scram
 - common configuration for projects which tools and versions to use
- SCRAM is used in a different way from project to project
 - Transfer of knowledge to LCG people (in all projects) is difficult
 - it is used in different ways
- The improvements needed are not completely there
 - Speed issues, porting to Windows, improving efficiency, separate configure from make
 - Fixes instead of changes and the result is not very satisfactory
- Work is going on to study other solutions not developed in house



A. Aimar - EP/SFT


LCG - Softw
Infras

http://spi.cern.ch/scram/ - Microsoft Internet Explorer

http://spi.cern.ch/scram

LHC Computing Grid Project > LCG Applications Area

SPI - Software Process & Infrastructure



SPI Quick Links

- [SPI Home](#)
- [SPI Index Page](#)
- [SPI Workbook](#)

SPI Services Links

- [Savannah Portal](#)
- [External Software](#)
- [Software Testing](#)
- [LCG Workbook](#)

LCG App. Area

- [Home Page](#)
- [LCG Agenda](#)
- [PI Project](#)
- [POOL Project](#)
- [SEAL Project](#)
- [Simulation Project](#)

SCRAM

SCRAM is the software configuration, release, management Area projects.

All LCG software is compiled using SCRAM, as explained in the [SCRAM User Documentation](#).

SCRAM LCG ToolBox

[What is the SCRAM LCG ToolBox?](#)

Latest configuration: LCG_20 (see [Configuration file](#))

Please check the [LCG configuration release notes](#)

To post a bug about the SCRAM LCG ToolBox use the "ToolBox" category.

SCRAM User Documentation

Check the online manual for the latest release (V0.10.0)

Done

Nightly Build

- Builds periodically the LCG software
- Runs the tests
- Presents the results
- We did not look for external or other tools but currently Nicos is being developed
- Provided by BNL/Atlas (A.Undrus)
- Derived from what is being developed in Atlas
- The author is very motivated to support it for the LCG
- Work is still in progress



A. Aimar - EP/SFT

The screenshot shows a Microsoft Internet Explorer browser window displaying the NICOS project page. The address bar shows the URL: <http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/dist/nightlies/poolwww/>. The page title is "NICOS (Nightly Control System)". Below the title, it says "Project: POOL" and "Location: /afs/cern.ch/sw/lcg/contrib/".

The page content includes a section titled "This page" with a list of bullet points:

- shows the status of nightly builds of POOL project managed by the...
- provides a link to the Project Configuration Page
- P...
- N...

Below this, there is a section titled "The table" with a list of bullet points:

- c...
- in...
- sh...
- in...

At the bottom of the screenshot, a table displays the status of various POOL releases:

release name	status
POOL_RELEASE_5	completed
POOL_RELEASE_4	completed
POOL_RELEASE_3	completed
POOL_RELEASE_1	configuration in pro
POOL_RELEASE_0	completed
POOL_RELEASE_6	configuration in pro
POOL_RELEASE_2	configuration in pro

LCG - S
In

LCG Workbook

- Introduction to new users in the LCG
- Task-oriented
- Web-based
- Inspired by the Babar workbook but we are still far from there



A. Aimar - EP/SFT

SPI Workbook - Home page - Microsoft Internet Explorer

http://spi.cern.ch/workbook

LHC Computing Grid Project > LCG Applications Area > LCG Software Process

SPI Workbook Page 1

SPI Workbook

LCG Workbook

SPI Quick Links

SPI Home
SPI Index Page

SPI Workbook

SPI Services Links

LCG Workbook

Savannah Portal
External Software
Software Testing

Download

LCG App. Area

Purpose of this web site

The purpose of this workbook is to provide help and information to all users of the infrastructure created by the SPI project (Software Process & Infrastructure) for the LCG Application Area projects.

This workbook is part of the LCG Applications Area Workbook that describes all the LCG projects.

If you have feedback or requests please just contact us.

Documentation of the LCG projects

- Reference Documentation (via doxygen)
- CVS Browser (via viewCVS)
- Code Cross Reference (via LXR)

How to ...

- Get started at CERN
- Get started in the LCG projects
- See the **online reference documentation** of the LCG software
- **Send bugs and requests** for help to the LCG projects
- Find what **software is available from the LCG projects**
- **Install LCG software** locally on your machine

LCG Policies

- LCG Policies
 - CVS and Build Directory Policy
 - Software Testing Policies
 - Version Numbers, Tagging and Release Procedure
 - Installation Directory Structure
 - Platform string, binary names, debug flags and more
- They are needed by the LCG
 - They are defined by the LCG projects, collected by SPI
- If everything is different is too difficult to use and to automate
 - compromising on our habits, for project needs
 - tell when they are not followed
- First time that this exists at this extent, and that is checked



http://spi/software_development.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://spi.cern.ch/software_development.html

LHC Computing Grid Project > LCG Applications Area

SPI - Software Process & Infrastructure

SPI Quick Links

- [SPI Home](#)
- [SPI Index Page](#)
- [SPI Workbook](#)

SPI Services Links

- [LCG Workbook](#)
- [Savannah Portal](#)
- [External Software](#)
- [Software Testing](#)
- [Download](#)

LCG App. Area

- [Home Page](#)
- [LCG Agenda](#)
- [PI Project](#)
- [POOL Project](#)

Software Development Tools

- [Setting-up the environment.](#)
- [LCG software is compiled using SCRAM. See SCRAM.](#)
- [Tools for automatic generation of source files and direct](#)

LCG Software Development Policies

LCG Policies Strategy

- [CVS Directory Structure Policy](#)
- [C++ Source Code Policy](#)
- [Build Directory Policy \[lib/bin/include\]](#)
- [Version Numbers, Tagging and Release Procedure](#)
- [Installation Directory Structure](#)



External Software Service

- We install software needed by LCG projects.
- **Open Source and Public Domain** software (libraries and tools) like:
 - Compilers (icc, ecc)
 - HEP made packages
 - Scientific libraries (GSL)
 - General tools (python)
 - Test tools (cppunit, qmtest)
 - Database software (mysql, mysql++)
 - Documentation generators (lxr, doxygen)
 - XML parsers (XercesC)
- There are currently 50 **different software**, plus others under evaluation. For more than 300 installations
- The LCG projects (SEAL, POOL, PI, Simulation and SPI) propose **what to install and in which version**
- The **platforms**, decided by the Architect Forum
 - Linux RedHat 7.3 with the compilers
 - gcc 3.2 (*rh73_gcc32*)
 - icc 7.1 (*rh73_icc71*)
 - ecc 7.1 (*rh73_ecc71*)
 - Windows
 - Visual Studio .NET 7.1: (*win32_vc7*).
- We also provide configuration for the LCG projects
 - A unique AFS location
 - Standard structure `package_name/version/platform/package_content`

LCG Software[Download Area](#)**External Software**[Alphabetic order](#)[Platforms table](#)[Used in LCG](#)[Projects](#)**SPI Quick Links**[SPI Home](#)[SPI Index](#)[Projects Portal](#)**LCG App. Area**[Home Page](#)[LCG Agenda](#)[PI Project](#)[POOL Project](#)[Simulation Project](#)[SEAL Project](#)**MySQL****Open source relational database.****Description**

The MySQL database server is the world's most popular open source database. Its architecture makes it extremely fast and easy to customize. Extensive reuse of code within the software and a minimalistic approach to producing functionally-rich features has resulted in a database management system unmatched in speed, compactness, stability and ease of deployment. The unique separation of the core server from the storage engine makes it possible to run with strict transaction control or with ultra-fast transactionless disk access, whichever is most appropriate for the situation.

Availability

```
/afs/cern.ch/sw/lcg/external/mysql/4.0.15/win32_vc7/
/afs/cern.ch/sw/lcg/external/mysql/4.0.15/rh73_ecc71/
/afs/cern.ch/sw/lcg/external/mysql/4.0.15/rh73_gcc32/
/afs/cern.ch/sw/lcg/external/mysql/4.0.14b/win32_vc7/
/afs/cern.ch/sw/lcg/external/mysql/4.0.13/rh73_gcc32/
/afs/cern.ch/sw/lcg/external/mysql/4.0.12/rh73_icc71/ (warning: avoid this one, test pb)
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh73_gcc32/ -> rh72_gcc2952/
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh73_gcc2952/ -> rh72_gcc2952/
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh72_gcc2952/
/afs/cern.ch/sw/lcg/external/mysql/4.0.4-beta/rh61_gcc2952/
```

Download

Savannah Project Portal

- The Web portal for LCG SW projects
- Customized from GNU (SourceForge as origin)
- Functionality:
 - Bug tracking
 - Task management
 - Mailing lists, news, faqs
 - Access to CVS repository
 - Download area, etc
- Totally web based
- Single entry point to all projects
- Uniform access to project information
- Set up common web infrastructure for a project without coding

LCG Savannah Page

savannah: Welcome - Microsoft Internet Explorer

<http://savannah.cern.ch>



Login Status:

NOT LOGGED IN

- [Why Log In? >](#)
- [Login via SSL >](#)
- [New User via SSL >](#)

Projects

- [Hosted Projects >](#)
- [Help Wanted >](#)

Portal Help

- [User Docs >](#)
- [Admin Docs >](#)
- [Support >](#)

Search

Software Group

The LCG software development portal

This web site is provided by the [LCG Software Process & Infrastructure \(SPI\)](#) project. It offers facilities for development, distribution and maintenance of LCG software projects and related projects.

If you would like to use these facilities for your project, then go to the **Register new project** menu entry that is displayed after login.

If you have questions about using this site, please consult the **user docs/admin docs** (left menu bar). If you don't find an answer to your problem, use the **support** link and then choose **Submit a Request** in the menu bar which will appear at the top of the page.

The software used to run this site has been developed by the [GNU Savannah](#) project and customized for LCG use by SPI.

Latest News

Savannah was inaccessible

dfeich - 2003-May-08 08:49 - 0 messages

Due to a failed automatic security update, the data base server went down and Savannah was inaccessible from 2h am until 8h am.

SERVER MIGRATION - Fri Apr 25th

dfeich - 2003-Apr-24 15:33 - 0 messages

Savannah will move to <http://savannah.cern.ch>. The service will be unavailable from 9-11h on Fri Apr 25th. After that all access to lcgappdev.cern.ch/savannah will be redirected. Please update your bookmarks!

Server Statistics

Hosted Projects: **76**

- 4 Savannah Project Portal
- 12 LCG Application Area
- 13 CMS
- 8 LHCb
- 3 LCG Grid deployment
- 3 HepPackages
- 24 ATLAS
- 7 CERN IT
- 1 ALICE
- 1 New type
- + 0 Registrations Pending

Registered Users: **421**

Newest Savannah Project Portal Projects

- (10/24) testproject
- (10/23) savcern

[\[all Savannah Project Portal projects\]](#)

Newest LCG Application Area Projects

- (11/05) appwork
- (10/22) GDML
- (10/20) ConditionsDB
- (09/09) DIRAC-2
- (07/04) Simulation
- (02/26) CASTOR
- (02/21) PI
- (02/12) SCRAMToolBox
- (01/20) SEAL
- (12/16) SCRAM




Project Pages

savannah: Modify a Bug - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address https://savannah.cern.ch/bugs/?func=detailbug&bug_id=1785&group_id=5 Go Google Search Web



Development
Portal

aimar's account

- My Personal Page ▾
- My Account Conf ▾
- Bookmark This ▾
- Logout ▾

Projects

- Hosted Projects ▾
- Help Wanted ▾
- Register New Project ▾

Portal Help

- User Docs ▾
- Admin Docs ▾
- Support ▾

Search

SPI - Bugs

Public Areas: [Main](#) | [Homepage](#) | [FAQ](#) | **[Bugs](#)** | [Support](#) | [Patches](#) | [Mailing Lists](#) | [Tasks](#) | [News](#) | [CVS](#) | [Files](#)
>> [Search Bugs](#) | [Submit a Bug](#) | [Browse Open Bugs](#) | [Browse My Bugs](#) | [Reporting](#)

Administration: [Main](#) | [Bugs](#) | [Support](#) | [Patches](#) | [Mailing Lists](#) | [Tasks](#) | [News](#) | [Files](#)

[Bug #1785] configuration/installation problem: wxPython vs Python

Submitted By:	moscicki	Group:	SPI
Submitted on:	2003-Nov-03 14:43	<input type="button" value="Submit Changes"/>	
Category:	External Software ▾	Severity:	Moderate ▾
Bug Group:	None ▾	Resolution:	None ▾
Assigned to:	None ▾	Status:	Open ▾
Platform Version:	None ▾		
Summary:	configuration/installation problem: wxPython vs Python		
Original Submission:	Date: Mon, 3 Nov 2003 13:40:46 +0100 (CET) From: Veronique Lefebure <Veronique.Lefebure@cern.ch>		

I have created all rpms for POOL using your python installation script.
Now I am doing the exercise of installing them and I get several problems:

[...]
rpm -i Python_2.2.2_LCG_rh73_gcc32-1-1.i386.rpm

Done Internet

Savannah by SPI

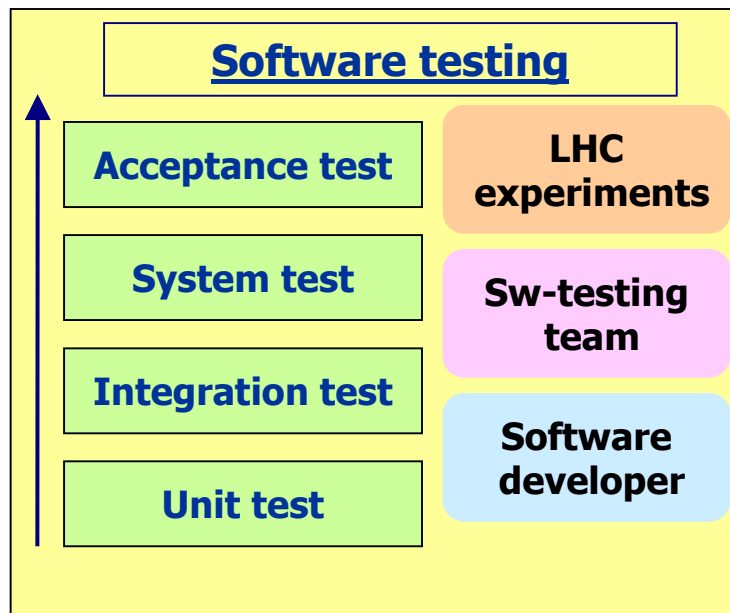
- What SPI changed
 - installation from GNU, general bug fixing and improvements
 - implemented bulk user registration
 - integration with AFS authentication
 - sending these improvements back to GNU
- What SPI does
 - administration (project approval)
 - maintenance (submitted bugs)
 - development (support requests)
 - staying in phase with GNU and keep contact with other developers
- Status
 - 70 hosted projects, 395 registered users
 - major new release in preparation (merge of CERN and GNU branches, common tracker for all services, etc)
 - we work closely with the open source people
 - still a few minor bugs and limited documentation (online help & faqs) ... will be fixed gradually
 - LCG Savannah is at
<http://savannah.cern.ch>

Software Testing Overview

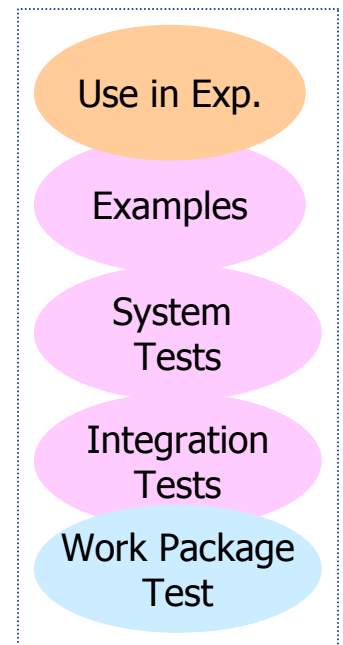
- Software testing should be an integral part of the software development in the LCG App Area
- All level of software testing should be run as part of an automatic process.

SPI provides

- Test frameworks
- Test support
- Test policies
- Test doc



Automated testing



Testing Frameworks

- The goal is to use something that can be run automatically

CppUnit/PyUnit

- The same “assertion style” in different languages, also Java, Perl, etc.
- name of the test case, file, line number where the failure occurred

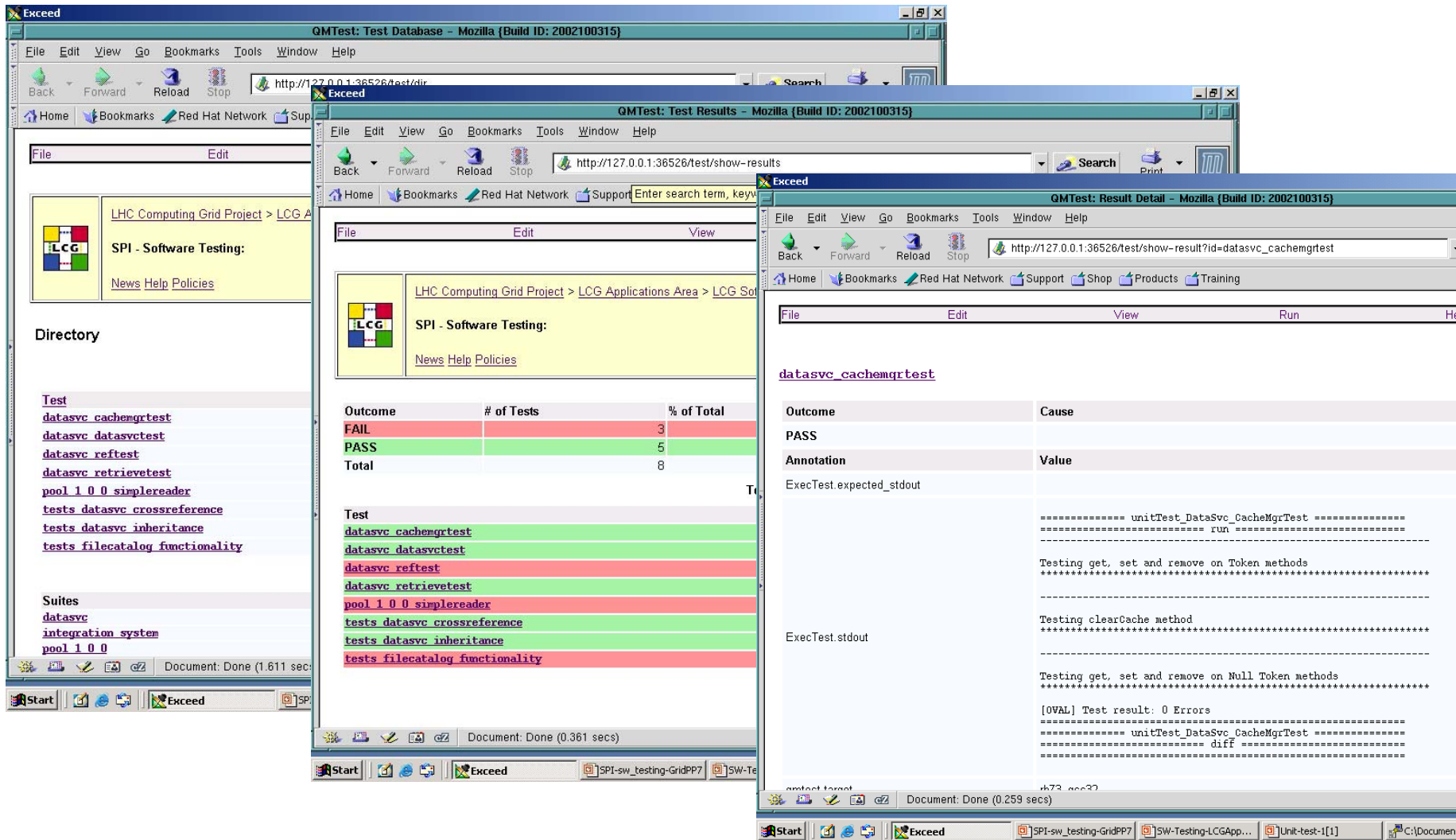
Oval

- Compare the output log file with a given reference file
- Smart comparison of files
- Can run any external scripts and external binaries.

QMTTest

- Uses a graphical interface for creating and running tests
- The configuration files are in XML and can be created from the GUI. We provide also script to do it
- Runs tests in parallel
- Organizes tests hierarchically
- Supports execution of a single test or many at once
- Records dependencies among tests
- Can be run in batch mode -> easy integration with the Nightly-Building systems
- Different platforms/compilers (Linux/Solaris/Windows)

Test Frameworks (2)



The screenshot displays three overlapping browser windows from the 'Exceed' environment. The top window shows the 'QMTest: Test Database' page. The middle window shows 'QMTest: Test Results' with a summary table:

Outcome	# of Tests	% of Total
FAIL	3	
PASS	5	
Total	8	

The bottom window shows 'QMTest: Result Detail' for the test 'datasvc_cachemgrtest'. It displays the test outcome as 'PASS' and includes a detailed log of the test execution, showing '0 Errors'.



Testing Support

Sw-Testing Policies:
Software testing policies within LCG AppArea project

FrameWorks for Sw-Testing:
Currently four test-framework tools are provided as External Software Service. These are: CppUnit and JUnit can cover from unit to validation tests, and QMTest interface to organize and run the tests. A set of scripts is available and can help with this tools.

Sw-Testing HowTo

- [HowTo make SW-Tests](#)
- [HowTo for CppUnit TestFramework](#)
- [HowTo for Oval TestFramework](#)
- [HowTo for PyUnit TestFramework](#)
- [HowTo-TestFramework-QMTest](#)
- [HowTo for Test Execution Framework](#)

Sw-Testing documentation and planning

- [testplan_template.html](#) - The main purpose is to schedule the testing process with all responsibilities. Use it to define software and staff responsibilities. Use it to define (Examples: PQQL)
- [testcase_template.html](#) or test case scenario template and data along with the expected results

Support:
If you need to report a bug or ask for support with the [bug](#) report system and [support area](#)

HowTo for CppUnit TestFramework

HowTo for QMTest TestFramework

What is QMTest?
QMTest (<http://www.codesourcery.com/qm/qmtest>) is an open-source, cross-platform software testing tool written in Python. QMTest is a general purpose testing solution that allows an organization to implement a robust, easy-to-use testing program tailored to its needs. QMTest works with most varieties of UNIX, including GNU/Linux, and with Microsoft Windows. See the [SPI](#) supported platforms at [SPI external software service](#).

QMTest uses a graphical interface for creating and running tests, runs tests in parallel, organizes tests hierarchically, supports execution of a single test or many at once and records dependencies between tests together with the results and expected results.

[return to top of page](#)

How to use QMTest in LCG AppArea?

1. [QMTest](#) tool and the "lcg-qmtest-config.py" script should be in your path. QMTest path was included in the SCRAM ToolBox and the path to the correct version installed in the [SPI external software](#) is managed by SCRAM. So, if you are in a SCRAM based project you do not need to take care of this. The "lcg-qmtest-config.py" will be available after set up the LCG-SPI environment with:


```
source /afs/cern.ch/sw/lcg/app/spi/tools/latest/setup/lcgspi.csh
```



Quality Assurance Overview

- The main goal of QA activity **help LCG projects**
- **assess and improve the quality of the software**
 - provide tools to collect useful **metrics/statistics** which help assess quality;
 - generate **reports**;
 - verify if project setup is correct with **LCG Policies**.
- Reporting tools
 - analyze project tree in AFS release area
 - time-based analysis (e.g. bugs reports)
 - → generate HTML pages

QA Tools and Focus

- Release process tools
 - include all open/fixed bug reports in the release notes automatically
- Tests/Bugs are central for QA in our environment
 - vague/changing user requirements, no “product specifications”
 - tools/procedures by agreement rather than by decision
 - sophisticated code metrics
- LCG Policies
 - agreed and defined by AF
 - SPI supports them in the tools and procedures and only helps to work them out

Quality Assurance Activities

QA Checklist on each Release

- Build the release
- Run automatic tests
- Statistics
 - Test Inventory
 - Documentation/Examples Inventory
 - Savannah Statistics
 - Code Inventory
 - Rule Checker , Logiscope
- LCG Policies
 - Configuration of a build system
 - CVS directory structure
- Well-defined, transparent, open
 - clear rules and checklist of assessed items
 - anybody at anytime may see statistics
 - create reports themselves
 - anybody may contribute

QA Status

- Manual / semi-automatic reports
 - POOL QA for 0.4.0, 1.0.0, 1.1.0, 1.2.0
 - SEAL QA for 0.3.1, 1.0.0
- Development / integration of automatic tools
 - SEAL_1_1_0
 - tools about to be released / announced
- Evaluation of tools
 - Rule Checker
 - Logiscope, Test coverage
 - SLOC, Valgrind, ignominy

Software Distribution Overview

- Temporary solution
 - local installations (external sites, laptops,...)
 - using simplest approach
 - python downloader + tar format
 - replicate the central AFS tree (in a optimized way)
 - package dependency from SCRAM
- ...until a generic, long-term solution available
- SPI will adopt what LCG Grid Deployment decides to provide
- Simple tool to install
 - successful for users:
 - POOL @ Karlsruhe
 - BNL nightly builds, CMS
 - developers at home, etc
 - very easy to use and reliable
- Different use-cases should have different solutions
 - Our tool is adequate as a temporary solution for LCG Application Area Distribution
 - but long-term solutions must be investigated:
 - pacman, LCFGng
 - GRID WN installations should be supported differently

LCG Distribution

The screenshot shows a Microsoft Internet Explorer browser window with the address bar displaying `http://spi.cern.ch/lcgsoft/POOL_1_3_3_rh73_gcc32-download.html`. The page content is as follows:

LCG Software

Download page for POOL_1_3_3 / rh73_gcc32

[Download Area](#)

External Software

On the machine where you want to install the software run the following command:

```
./lcg-installation-manager.py --project=POOL_1_3_3 --arch=rh73_gcc32 --prefix=/opt/sw/lcg download
```

Python 2.2 or greater is required. Download the script here: [lcg-installation-manager.py](#)

Questions? Problems? [See here.](#)

Manual download

POOL 1 3 3 LCG rh73 gcc32.tar.gz

Required packages:

name	version	download
SCRAM	V0_19_9	SCRAM V0_19_9.tar.gz
uuid	1.32	uuid 1.32 LCG rh73 gcc32.tar.gz
gccxml	0.4.2_patch1	gccxml 0.4.2_patch1 LCG rh73 gcc32.tar.gz
Boost	1.30.2	Boost 1.30.2 LCG rh73 gcc32.tar.gz
mysql++	1.7.9_mysql.4.0.13	mysql++ 1.7.9_mysql.4.0.13 LCG rh73 gcc32.tar.gz
mysql	4.0.13	mysql 4.0.13 LCG rh73 gcc32.tar.gz

The browser window also shows a sidebar with navigation links such as 'LCG Software', 'External Software', 'SPI Quick Links', and 'LCG App. Area'. The status bar at the bottom indicates 'Done' and 'Internet'.

Summary and Conclusions

- The set of services shown is working and fully available
 - Savannah Project Portal
 - Software Testing
 - External Software Service
 - Quality Assurance and Policies
 - Software Distribution
 - ...and many more
- We have followed the strategy defined
 - Work with the users
 - Ask their help
 - Develop as little as possible in order to have little maintenance
 - Provide simple and modular solutions
- We have commitments to the users but also to provide a sustainable service
 - Most people moved to new LCG projects, as it was planned
 - The services are used by LCG projects, and also outside LCG
- Unlike in the past, we behave to match the environment and the way people work (Simple, Pragmatic, Informal)

Summary

- ◆ In general very good support from SPI
 - Some tools are very good (e.g. Savannah, QMTest)
 - Other tools are less good (e.g. SCRAM)
 - Very good collaboration with the SPI people
 - » Very often sitting together in front of the same terminal
- ◆ Some suggestions
 - SPI Software librarian
 - Less policy verification and more practical tools

From SEAL feedback



Summary



- POOL fully relies on many SPI services
 - And actively participates in their definition
 - Service level for POOL is found very adequate
- POOL has followed the evolution of LCG policies maintained and checked by SPI
 - Being the first project is sometimes a disadvantage
- Insuring a consistent/identical build and testing procedure between the LCG AA projects is non-trivial
 - Because of different project requirements
 - The task would be simplified by centralizing the task
 - The load generated by the frequent internal releases in POOL is significant

From POOL feedback

Future plans

- Internal Review Recommendations (are already on the way)
 - Put in place a **software librarian** position to have a central role for building and releasing LCG software
 - Merging our improvements with **Savannah open source**
 - Move to **IT CVS** service as planned from the beginning
 - Continue to back up QA policies, more **QA reporting tools**
 - **Re-asses** the **configuration and build system** and continue the evaluation of a solution simply based on autoconf.
 - Provide **configurations for** the different build systems used in **the experiments**
 - Encourage other LCG areas to use our services
- The current resources are just sufficient to continue what we are doing (~5-6 FTE)
- Collaborate with EGEE that is interested in the SPI services