

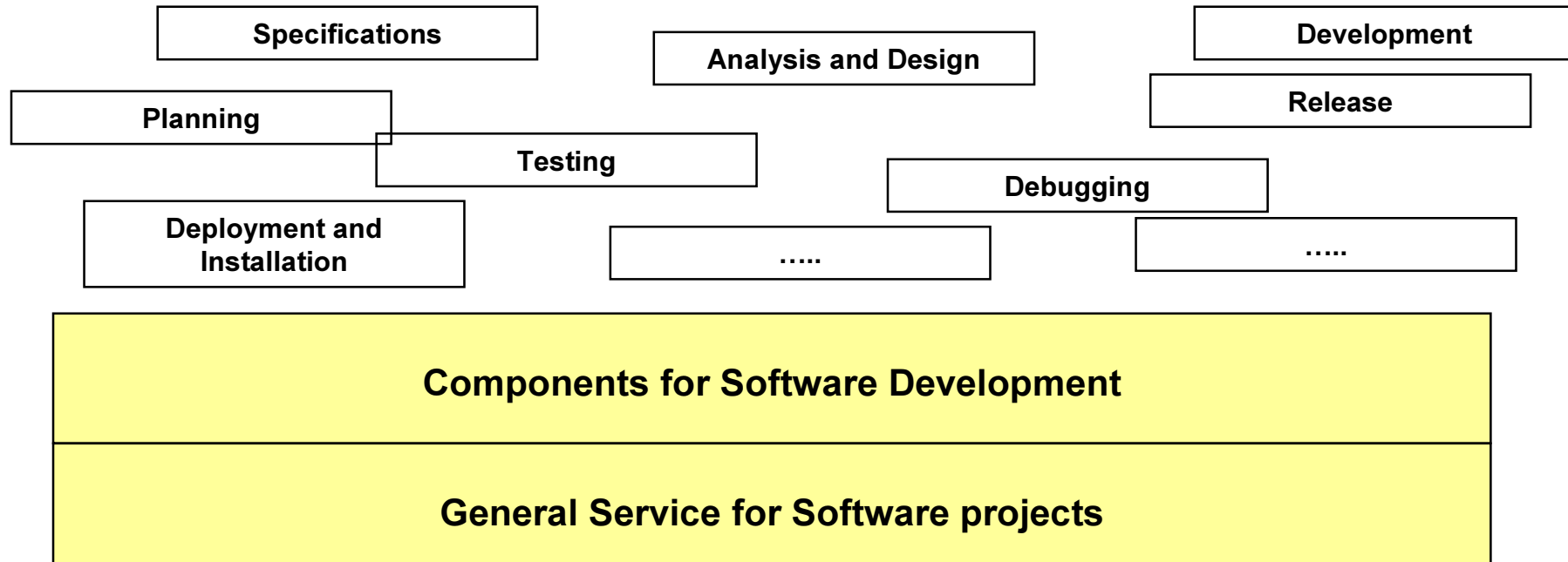


LCG Infrastructure for Software Projects

Overview

**A.Aimar IT/API
CERN**

Infrastructure for Software Development



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



SPI Services

CVS repository and Delivery Areas



CVS repository

- A central CVS repository managed and available to all projects
- Any project just needs to ask for it, and declare its users permissions
- Managing mirroring and backups
- Users access controlled
- Tools for automatic clean up of locks, etc
- Systems to browse the repositories

Delivery areas

- The AFS and DFS delivery areas
 1. an area to install software created by projects in the LCG application area
 2. an area where external and third party software
 3. an area where software is installed which is provided by people for evaluation within a project
- Pre-installed software shall be in directories specifying the package name, the package version and the "OS_Compiler"



:pserver:<anoncvs>@lcgapp.cern

pool

[Query in the check-in database](#)

Current directory: **[POOL]** / pool

File

- [AttributeList/](#)
- [Collection/](#)
- [CollectionFactory/](#)
- [Common/](#)
- [DataSvc/](#)
- [EDGCatalog/](#)
- [FileCatalog/](#)
- [ImplicitRootCollection/](#)
- [MySQLCatalog/](#)
- [MySQLCollection/](#)
- [NaiveAttribList/](#)
- [PersistencySvc/](#)
- [PoolRelease/](#)
- [Reflection/](#)
- [ReflectionBuilder/](#)
- [RootCollection/](#)

Components

[Query in the check-in database](#)

Current directory: **[Infrastructure]** / Comp

File

- [BugTracking/](#)
- [CodeDocumentation/](#)
- [CodingDesignConventions/](#)
- [ConfigurationManagement/](#)
- [CvsOrganization/](#)
- [DevelopmentWeb/](#)
- [NightlyBuilds/](#)
- [PlanningMaterial/](#)
- [Profiling/](#)
- [ProjectPortal/](#)
- [SoftwareDocumentation/](#)
- [UnitTesting/](#)

[Download tarball](#)

[SPI support](#)

The image shows two overlapping Windows Explorer windows. The top window is titled 'Z:\.cern.ch\sw\lcg' and displays the directory structure for 'lcg'. The bottom window is titled '\\cern.ch\dfs\Projects\LCG' and displays the directory structure for 'LCG'. Both windows show a 'File' menu, a toolbar with 'Back', 'Search', 'Folders', and 'History', and a list of folders: 'app' and 'contrib'. The status bar at the bottom of the top window shows '3 object(s)' and '6.00 KB'. The status bar at the bottom of the bottom window shows '3 object(s)' and '0 bytes'. The address bar of the top window contains the URL **/afs/cern.ch/sw/lcg** and the address bar of the bottom window contains the URL **\\cern.ch\dfs\projects\lcg**.

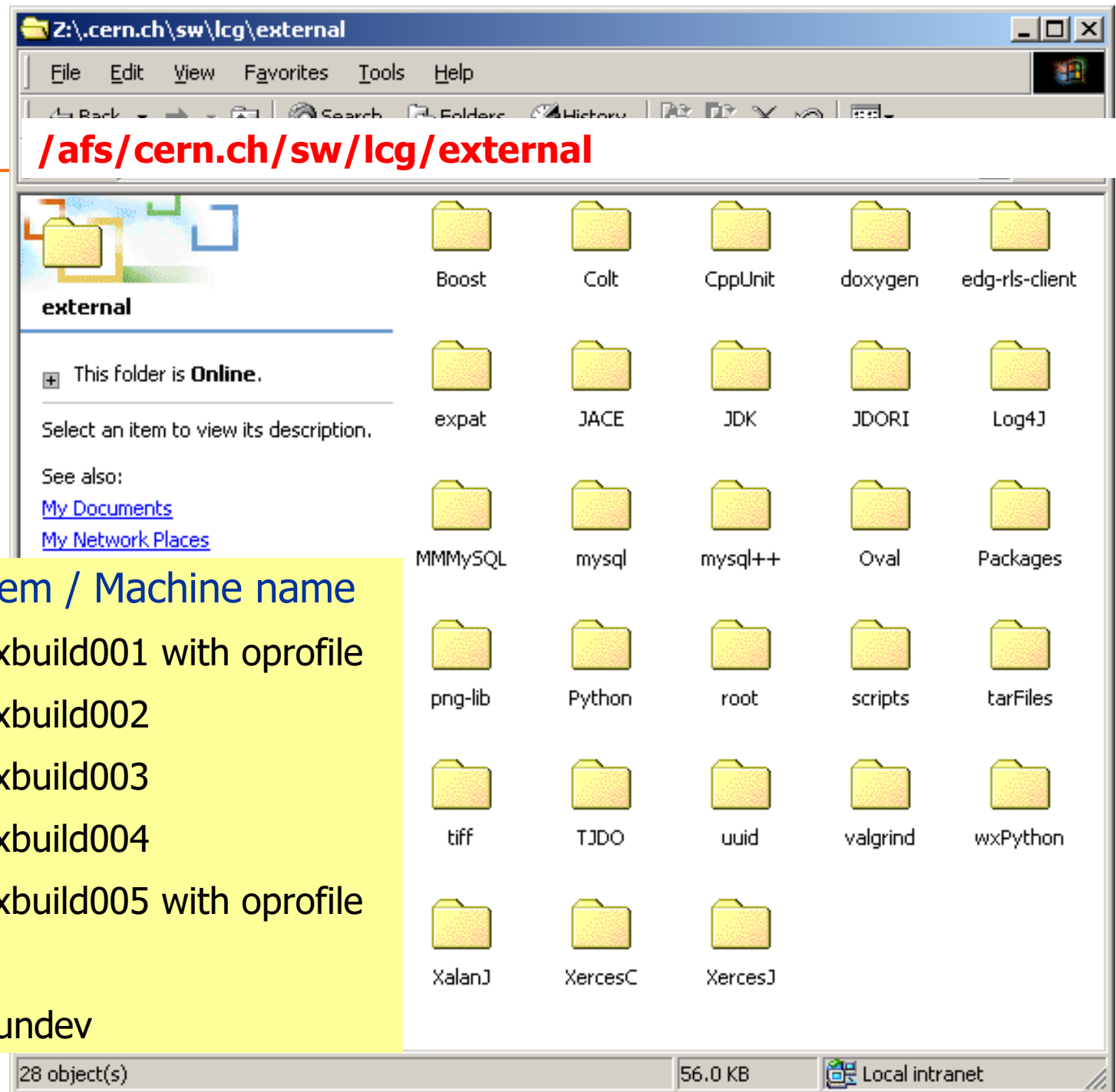
SPI Services

Build systems and Software Library



- Presently the build servers are publicly available, so everybody can login using their AFS account
- Window 2000 and Windows XP are needed too. Currently not available yet.
- All software needed by more than one LCG projects shall be installed in common areas
- Available for several compilers and operating systems
- Maintained centrally with support and help to the user
- External documentation available
- Contact with the authors if the software is developed in HEP or LCG
- Distribution and pre-installation to all LCG and HEP users





- Operating system / Machine name

RedHat 6.1 lxbuid001 with oprofile

RedHat 6.1 lxbuid002

RedHat 7.2 lxbuid003

RedHat 7.2 lxbuid004

RedHat 7.2 lxbuid005 with oprofile

RedHat 7.3

Solaris-7 sundev

SPI Services

Bug Tracker and Project Portal



Project Portal

- The development front-end of every project
- With all features needed, configured and integrated
 - News and Fans
 - Task management
 - Mailing lists access
 - Bug tracking access
 - Download area
 - Access to the CVS repository
- Users can contact the project or also become developers
- We have customized Savannah that is the one from GNU

Bug Tracker

- A central installation so that each project has an available bug tracker system
- Centrally managed but customizable for different needs
- Easier to post bug if projects use similar systems
- No work of maintenance and upgrade on the projects
- Web based

<http://lcgappdev.cern.ch>
(for a few test projects)



<http://lcgappdev.cern.ch/savannah>



Login Status:

NOT LOGGED IN

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

Projects

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

Help

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

Search

- Software/Group
- People
- Require All Words

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 maintainance of LCG software 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 access

dfeich - 2002-Dec-03 19:58 - 1 messages
 LCG Savannah can now be reached from outside of CERN. This is still a testing phase. For support requests use the 'support' link in the left menu bar.

Coordination w.r.t. cvs, file upload/download area, etc.

flob - 2002-Nov-19 17:17 - 1 messages

Server Statistics

Hosted Projects: **14**

- 5 infrastructure
- 1 pool
- 8 CMS

Registered Users: **56**

Newest Infrastructure Projects

- (10/31) SPI
- (10/24) testproject
- (10/23) savcern

[\[all infrastructure projects\]](#)

Newest Pool Projects

- (12/02) POOL

[\[all pool projects\]](#)

Newest CMS Projects

- (12/16) OVAL
- (12/16) SCRAM
- (12/16) OSCAR
- (12/16) ORCA
- (12/16) FAMOS
- (12/16) DDD
- (12/16) COBRA
- (12/06) IGUANA

[\[all CMS projects\]](#)

Help Wanted

- Developer (0)
- Project Manager (0)
- Unix Admin (0)



lcgappdev: Project Info - POOL - Microsoft Internet Explorer provided by CERN

File Edit View Favorites Tools Help

Address <http://lcgappdev.cern.ch/savannah/projects/pool/> Go



Login Status:
[NOT LOGGED IN](#)
[Why Log In? ▾](#)
[Login via SSL ▾](#)
[New User via SSL ▾](#)

Projects
[Hosted Projects ▾](#)
[Help Wanted ▾](#)

Help
[User Docs ▾](#)
[Admin Docs ▾](#)
[Support ▾](#)

Search

 Software/Group
 People
 Require All Words

POOL - Summary

Public Areas: **Main** | [Homepage](#) | [Bugs](#) | [Support](#) | [Patches](#) | [Tasks](#) | [News](#) | [CVS](#)

Project Type: POOL projects

Persistency Framework

License: [GNU Lesser General Public License](#)
Registration Date: Dec 02, 2002
Development Status: 4 - Beta

Developer Info

Project Admins:
[dirkd](#)

Developers:
11 [\[View Members\]](#)

Group id:
9

Public Areas

[Project Homepage](#)

[Bug Tracking](#) (1 open bugs, 9 total)

[Tech Support Manager](#) (0 open requests, 0 total)

[Patch Manager](#) (0 open patches, 0 total)

Latest News

Savannah Project Portal
dfeich - 2002-Dec-03 14:13 - 0 messages

The download area is not yet set up for your project (the viewCVS access is ok, though). You can look at a complete project setup if you select the 'savcern' project in the hosted projects list. This is also where you should submit bugs and support requests for problems with the interface.

[\[Submit News\]](#)
[\[1 News in Archives\]](#)



POOL - Bugs

Public Areas:

[Main](#) | [Homepage](#) | **[Bugs](#)** | [Support](#) | [Patches](#) | [Tasks](#) | [News](#) | [CVS](#)

>> [Search Bugs](#) | [Submit a Bug](#) | [Browse Open Bugs](#) | [Reporting](#)

Login Status:

NOT LOGGED IN

[Why Log In?](#)

[Login via SSL](#)

[New User via SSL](#)

Projects

[Hosted Projects](#)

[Help Wanted](#)

Help

[User Docs](#)

[Admin Docs](#)

[Support](#)

Search

Software/Group

People

Bugs

Require All Words

Browse Bugs using report (or use [Advanced Search](#))

Category (?)

Bug Group (?)

Assigned to (?)

Status (?)

bugs at once.

9 matching bugs

Click a column heading to sort results (up or down), or [Sort by Severity](#) or [Reset sort](#). You can also [Activate multicolumn sort](#). ([Printer version](#))

Items 1 - 9

Bug ID	Summary	Submitted On	Assigned To	Submitted By
49	first bug bug	2002-Dec-02 16:07	dirkd	dirkd
50	bug in cvsrep web page	2002-Dec-02 17:04	None	laman
54	SCRAM HowTo instructions do not work	2002-Dec-05 16:40	dirkd	mato
61	message service prints trailing binary zero ('\0') character	2002-Dec-11 13:43	frankb	dirkd

SPI components available



- All these components are available to LCG projects but still under completion. Release is for February 2003. Can be tried by “friendly” users in LCH experiments.
- Code Documentation Doxygen, LXR, ViewCVS
- Testing Framework CppUnit, Oval
- Memory Leaks Valgrind
- Automatic Builds
- Coding and design guidelines (RuleChecker)
- CVS organization
- Configuration/release mgmt Scram
- Software documentation templates



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

LCG Software Process & Infrastructure

SPI Quick Links

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

[Projects Portal](#)

LCG App. Area

[Home Page](#)
[LCG Agenda](#)

[MATH Project](#)
[PI Project](#)
[POOL Project](#)
[SEAL Project](#)
[SPI Project](#)

External Links

[CERN](#)
[EP Division](#)
[IT Division](#)

NEWS

04/12/2002	Presentation: Project portal, and bug tracking
25/11/2002	New SPI Index Page (please bookmark it)
02/11/2002	Talk on Profiling by Rafi Yaari
20/10/2002	Presentations: Code Documentation and Unit Testing
12/09/2002	Service and Components descriptions
11/09/2002	Presentation: LCG SPI Status

Updated: 18-Dec-2002 10:56

More information

[SPI Index page](#)
All links useful to the users of the SPI project and the developers of LCG Application Area projects.

[SPI Support page](#)
Please write there all you support requests. Or by [email](#).

[SPI project description page](#)
More details on the goals of the project.

[Managing LCG Software](#)
Requirements specified in the document RTAG2 Final Report

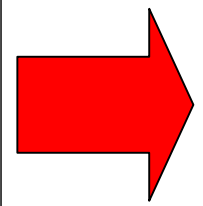
[Thoughts on Software process](#) by T.Wenaus. Thoughts on policy, process methodology, iterative development.

Project Overview

The goal of the project **SPI (Software Process & Infrastructure)** is to provide to the development projects of the LCG:

- basic environment for physics SW development
- general scientific libraries class libraries
- software development tools
- documentation tools and document templates
- compiler expertise
- support activity necessary to ensure that a common grid-enabled environment is available at all grid sites


The reason for the Software Process & Infrastructure project is to have



LCG SPI - Index Page - Microsoft Internet Explorer provided by CERN

File Edit View Favorites Tools Help

Address <http://spi.cern.ch/indexpage.html> Go

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

LCG Software Process & Infrastructure

Index Page Updated: 16-Dec-2002 18:57

<p>SPI Quick Links</p> <p>SPI Home SPI Index</p> <p>Projects Portal</p> <p>LCG App. Area</p> <p>Home Page LCG Agenda</p> <p>MATH Project PI Project POOL Project SEAL Project SPI Project</p>	<p>Infrastructure</p> <p>CVS Service</p> <p>How to use the LCG CVS service</p> <p>Project Portal</p> <p>Projects Portal for LCG User registration</p> <p>Software Library</p> <p>How to install 3rd party software</p> <p>SPI Templates</p> <p>Work package doc (html) Presentation Test case description Project plan template</p> <p>Other Services</p>	<p>Software Development</p> <p>Coding conventions</p> <p>Coding guidelines Design guidelines Description of the component</p> <p>Code documentation</p> <p>Access to: doxygen LXR ViewCVS</p> <p>How to use the Code Documentation component</p> <p>CVS organization</p> <p>Standard CVS structure How to develop software in the CVS structure</p> <p>SCRAM manual</p>	<p>App. Area Projects</p> <p>POOL Project</p> <p>Home page Project Portal CVS repository Bug reports Mailing Lists</p> <p>SEAL Project</p> <p>Home page CVS repository Bug reports Mailing Lists</p> <p>PI Projects</p> <p>Home page CVS repository Bug reports Mailing Lists</p> <p>Math Libs Project</p> <p>Home page CVS repository Bug reports Mailing Lists</p>
---	---	---	---

<http://spi.cern.ch/Components/CodingDesignConventions/Documentation/LCGdesignGuidelines.html> Internet



Code documentation: Doxygen



The screenshot displays two browser windows. The left window, titled "DataCache.cpp Source File", shows the source code for the `DataCache.cpp` file. The code includes a preprocessor directive for `POOL_OBJECTCACHE_CPP`, an include for `DataCache.h`, and two methods: `getCache()` and `deleteCache()`. The right window, titled "Main Page", shows the main documentation page for the POOL project, version 0_1_0. It features navigation links for `Main Page`, `Packages`, `Namespace List`, `Class Hierarchy`, `Alphabetical List`, `Compound List`, `File List`, `Namespace Members`, and `Compound Members`. The footer of the right window indicates it was generated by Doxygen 1.2.8.1 on October 21, 2002, for the POOL project, written by Dimitri van Heesch.

```
00001 #define POOL_OBJECTCACHE_CPP 1
00002
00003 #include "DataCache.h"
00004
00005 /// Initialize static pointer
00006 //pool::CacheTable* pool::DataCache::theCache = 0;
00007
00008 /// Method to retrieve the (single) Cache instance
00009 pool::CacheTable* pool::DataCache::getCache() {
00010     //if(!theCache){
00011         // theCache = new CacheTable();
00012         CacheTable* theCache = new CacheTable();
00013     //}
00014     return theCache;
00015 }
00016
00017 /// Method to delete the Cache instance
00018 //void pool::DataCache::deleteCache() {
00019 // delete theCache;
00020 // theCache = 0;
00021 //}
00022
```

Generated at Mon Oct 21 10:09:20 2002 for POOL by **doxygen** 1.2.8.1 written by [Dimitri van Heesch](#).

Code documentation : LXR



The image shows two overlapping browser windows. The left window displays the source code for `MethodBuilder.cpp` with a search for `int` highlighted in yellow. The right window shows the search results for `int`, listing various lines from `Geant 4.4.0/Configure` that contain the keyword.

Left Window: Source Code

Address: http://lcgapp.cern.ch/lxr/source/POOL_0_1_0/ReflectionBuilder/src/MethodBuilder.cpp

Navigation: [source navigation] [identifier search] [freetext search] [file search]

Software Cross Reference

Version: [current]

```
File /home/lxr/source/current//POOL_0_1_0/ReflectionBuilder/src/MethodBuilder.cpp
1 // $Id: MethodBuilder.cpp,v 1.4 2002/10/11 18:03:03 roiser Exp $
2
3 // Include files
4 #include "reflect/ClassBuilder.h"
5 #include "reflect/MethodBuilder.h"
6
7 #include "reflect/Method.h"
8 #include "reflect/Class.h"
9 #include "reflect/Namespace.h"
10
11
12
13 //-----
14 Reflect::MethodBuilder::MethodBuilder(const std::string& name,
15                                     const std::string& desc,
16                                     const std::string& returnType,
17                                     const std::vector<std::string>& argumentTypes,
18                                     void* (vfpvs) (void*, std::vector<void*>),
19                                     int modifiers)
20 //-----
21 : ItemBuilder(name, desc, modifiers),
22   m_returnType(returnType),
23   m_argumentTypes(argumentTypes),
24   m_vfpvs(vfpvs),
25   m_vfps(0),
26   m_vfpv(0),
27   m_vfp(0)
```



Code documentation: ViewCVS



- CVSgraph: Displays the tree of revisions and branches graphically
- Enscript: Colorize files in the CVS repository.

The screenshot displays the ViewCVS web interface. On the left, a window shows the source code for `DataTransform.cpp` with a CVS log entry for revision 1.4. The main window displays the revision graph for `ConversionSvc.cpp`. The graph shows a main branch with revisions 1.1 through 1.5, and a branch named `arelease` with revision 1.1.1.1. The graph is titled "Revision graph of pool/PersistenceSvc/src/ConversionSvc.cpp".

Return to [DataTransform.cpp CVS log](#) Up to [\[POOL\]](#) / [pool / PersistenceSvc / src](#)

Revision graph of pool/PersistenceSvc/src/ConversionSvc.cpp

/cvs/POOL: pool/PersistenceSvc/src/ConversionSvc.c
Revisions: 6, Branches: 2

```
graph TD
    1["1 MAIN  
04-Sep-2002 17:39:36"] --> 1.1["1.1  
04-Sep-2002 17:39:36"]
    1.1 --> 1.2["1.2  
13-Sep-2002 15:56:35"]
    1.2 --> 1.3["1.3  
13-Sep-2002 16:59:51  
v1r0d20020926"]
    1.3 --> 1.4["1.4  
01-Oct-2002 19:06:10  
v1r1  
POOL_0_1_0  
v1r0d20021002"]
    1.4 --> 1.5["1.5  
11-Oct-2002 11:03:36  
HEAD"]
    1.1 --> 1.1.1.1["1.1.1.1  
aenodor  
04-Sep-2002 17:39:36  
arelease"]
```

```
#include "pool/poolDb.h"
#include "pool/Gui.h"
#include "pool/DataCallBack.h"
#include "pool/DataTransform.h"
#include "pool/DbTypeInfo.h"
#include "reflect/Class.h"
#include <iostream>
#include <algorithm>
#include <map>

namespace pool {
    const std::string typeName(const std::type_info& typ);
}

typedef std::map<pool::Guid, pool::DataCallBack* > Conversion;
typedef std::vector<const pool::DbTypeInfo*> ShapeVector;
typedef std::map<pool::Guid, ShapeVector > Shapes;

namespace { // Don't clutter global namespace
    class RegInfo {
    public:
        pool::Guid defGuid;
        const pool::ObjectType* defType;
        const pool::DbTypeInfo* defShape;
```

[CVS admin](#) Powered by
ViewCVS 0.9.2
WORKS WITH
MySQL



Test frameworks: CppUnit



Similar tools: Junit,
PerlUnit, PyUnit, QtUnit

Output in XML, compiler
or text

Windows version for
MVC++6.0

```
xterm <6>
!!!FAILURES!!!
Test Results:
Run: 19  Failures: 2  Errors: 0

1) test: StringToolsTest::testToStringInt (F) line: 33 StringToolsTest.cpp
equality assertion failed
- Expected: 12345678
- Actual   : 123456789

2) test: ComplexNumberTest::testEquality (F) line: 22 ComplexNumberTest.cpp
assertion failed
- Expression: *m_10_1 == *m_1_1

PASS: test
=====
All 1 tests passed
=====
make[1]: Leaving directory `/home/user/mgallas/CppUnit-examples/newtest'
[pcitapi31] ~/CppUnit-examples/newtest >
```

A simple test

1. Subclass the TestCase CppUnit class
2. Override the method runTest().
3. call CPPUNIT_ASSERT



Test frameworks: Oval



Oval:

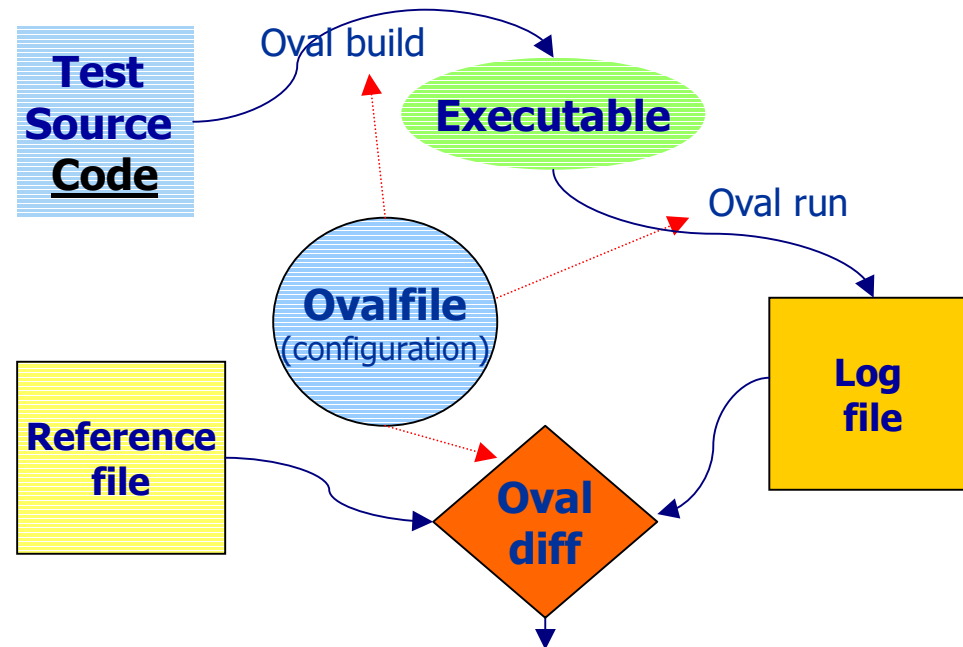
- validation
- regression

Can be use for Unit-test

It is possible to set different run environments

Can run external scripts and external binaries

Authors modified it to run it with or with any environment



```
xterm <6>
[pcitapi31] ~/OVAL-examples/subtest1 > oval p
Prog1: build, run, diff.
Prog2: build, run, diff (DIFFFS).

xterm <6>
[pcitapi31] ~/OVAL-examples/subtest1 > oval diff Prog2
===== diff =====
(DIFFFS)
=== log #29 !~ ref #65 (>15%)
log: [OVAL] result = 1.2
---
ref: [OVAL] result = 1
=====
[pcitapi31] ~/OVAL-examples/subtest1 > 
```



Coding guidelines

A screenshot of a Microsoft Internet Explorer browser window. The title bar reads "LCG Coding Rules for C++ - Microsoft Internet Explorer provided by CERN". The address bar shows the URL "http://spi/Components/CodingDesignConventions/Documentation/LCGcodingGuidelines.html". The main content area has a red heading "C++ Coding Guidelines" followed by "UNDER CONSTRUCTION" and "Max Sang, IT/API July 2002". A note states: "Note: these rules are derived mainly from the coding guidelines of Anaphe, Atlas[1] and CMS[2]. Justification can be found in these documents and the standard textbooks - Stroustrup[3], Meyers I & II[4,5], Lakos[6], etc." A red heading "First Principles" is followed by a paragraph: "In my opinion, the principal benefit of well-written OO code is increased maintainability, which means it is easier to understand, debug and modify than well-written procedural code. But this can be undermined unless we recognise that:" followed by a numbered list of two principles. The first principle is "Coding is mainly for human readers, not for the compiler. Many of the 'rules' below are to increase communication between writer and maintainer (who may be the same person six months later)." The second principle is "Maintaining any large body of code is like trying to read a map through a drinking straw. Code should be written such that the effect on the whole system of changing any piece of the code in it is maximally local, requires minimal changes elsewhere, and has no hidden traps or side-effects. Programmers should therefore constantly strive for:" followed by three sub-points: "abstraction - almost no-one can hold more than ten things in short-term memory (7 +/- 2 is the usual number quoted). Luckily for us, there is very little that cannot be expressed in less than ten statements/loc with enough abstraction.", "encapsulation - insulate client code from implementation. This cannot be emphasised enough.", and "localisation - try to eliminate duplicate code. Sooner or later the copies will start to diverge." Below this is a paragraph: "The most important rule of all is the KISS principle: **Keep It Simple, Stupid!** - clever tricks are bad style. Finally, don't re-invent the wheel. Take time to learn the STL, and also good third-party libraries like Boost, Loki, etc." A red heading "General style" is followed by a bulleted list of five items: "Use a simple style in class declarations", "Declare the contents of a class in the correct order", "Avoid unnecessary visual noise - stick to whitespace", "No extra-long lines, e.g. > 100 characters", and "Use consistent and meaningful indentation (using spaces, not tabs)." The browser status bar at the bottom shows "Done" and "Local intranet".

Rule
Checker

Human
Reviews



CVS organization



```
LCG CVS Directory Structure - Microsoft Internet Explorer provided by CERN
File Edit View Favorites Tools Help
Address http://spi.cern.ch/Components/CvsOrganization/Documentation/cvsDir-proposal1.html
Go

project/
  Package1/
    Package1/ -- public interfaces of the package (header files)
    src/      -- .c, .icc, .cpp and private header files
              sub-directories are not recommended.
    doc/      -- Release notes, README, documentation for this package,
              design models etc.
    tests/    -- tests with reference data for regression tests
              testX/
              testY/
    examples/
      exampleX/
                exampleX.cpp
      exampleY/

  [data/]     -- optional initialization data for the package
  [scripts/] -- optional script for the package (ex: to start a server, etc)

  [build/]   -- makefiles and other utilities to build the package with well-defined
              common targets (applicable for makefile builds only) in case the
              package can be build also only using make
  [etc/]     -- project specific, free structure inside with anything that are
              not sources, scripts, documentation, data.

Subsystem1/
  doc
  Tests      -- integration tests for this collection
  Examples   -- examples using more than one of the packages in this collection
  Package11/ -- nested packages
  Package12/

doc/
  user/*     -- printed user documentation, manuals etc
  developer/* -- related pages for the project, design models etc
```

Just started...



- LCG External Software Library
 - Available already because of the kind help of Pool, Seal, Root and other contributors
- Configuration and Build with SCRAM
- LCG Workbook
 - Introduction to newcomers, developers and users of the LCG software.
- LCG QA
 - Using the tools provided by SPI
 - Having the testing tools can help, but is not sufficient
 - Actively help/encourage LCG projects to use them
 - Adapting SPI to the needs and experience of LCG projects



Configuration and Build: SCRAM



Microsoft Excel - Scram First Task List.xls

File Edit View Insert Format Tools Data Window Help Acrobat

D1 = 5.12.2002

SCRAM - task list				5.12.2002
	Task	Comment	Priority	
1	Development	Improve performance	Calculating dependencies takes time even when nothing needs to be done.	1
2		Port to Windows	(1) cigwin (2) ability to run (3) VS.net integration. External help will speed up this activity.	1
3		Bug fixing	No outstanding bugs currently.	1
4		Output standard makefiles	To be able to install without Scram available, is it very difficult? To me does not seems so.	2
5	User support	Help POOL, SEAL, SPI	Help from experinece users of Scram would be very important to help the new users	1
6		Users Tutorial	Needs to be done soon for one day training.	1
7		Review existing documentation	Review and try out the manual completely	2
8		Presentation to end users	As often as possible (AAM, SPI, etc)	2
9				
10	Infrastructure	Move Shaun near the LCG projects	On the way, work in 32-R-C22 for next 6 months	1
11		Project web for user support and bug reports	Use Savannah for bugs user feedback, tasks, etc	2
12				

Sheet1 / Sheet2 / Sheet3

Ready

LCG Software Library



SPI (CERN IT LCG) - Microsoft Internet Explorer provided by CERN

Under construction

LHC Computing Grid Project > Application Area > Software Process & Infrastructure

External Software Service

[Home](#) [News](#) [Contact us](#) [Search](#)

SPI Quick Links

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




[Projects Portal](#)







LCG App. Area

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

The purpose of the **External Software Service** is to provide software tools and libraries to LCG development teams.

External Software alphabetic list

The 3 icons ( for Linux,  for MS Windows and  for Solaris) indicate the currently available platforms in `/afs/cern.ch/sw/lcg/external/` and `\\cern.ch\dfs\Projects\LCG\external\`. If the platform you are interested in is not present, just contact [us](#) and ask for it.

	Boost	Portable C++ source libraries
 	Colt	Open Source Libraries for High Performance Scientific and Technical Computing in Java
	CppUnit	The C++ Unit Test Library
	doxygen	Documentation system for C++, C, Java, IDL (Corba, Microsoft, and KDE-DCOP flavors) and to some extent PHP and C#.
	edg-rs-client	Protocol to access the data in the LRC through a set of secure SOAP-RPC calls, implements the client-server access through SOAP over https and JDBC on the server side.



LCG Workbook




LCG Application Area Workbook - Microsoft Internet Explorer provided by CERN

File Edit View Fav

Address http://lcgapp

http://lcgapp.cern.ch/project/workbook/



Introduction

This is the workbook of

- give you a hands-on tour of the LCG application area
- get you started as a user or contributor
- help you find the documentation and information you need

If you are looking at this workbook for the first time, please read more [about its structure and scope](#).

If you want to contribute to the workbook, please read the [instructions for contributors](#).

General Information

- [Overview of the LCG Application Area](#)
- [What you should do as a user of the LCG Application Area](#)
- [What you should do as a contributor to the LCG Application Area](#)

[SPI Workbook \(Software Process & Infrastructure\)](#)

SEAL Workbook (Core Libraries and Services)

[POOL Workbook \(Persistency Framework\)](#)

PI Workbook (Physics Interfaces)

Math Libraries Workbook

Miscellaneous Information

- [Tools](#)

LCG Application Area Workbook - Microsoft Internet Explorer provided by CERN

Conclusions



- The SPI project is also based on our true effort to work with the user community to:
 - adopt some common infrastructure
 - help in the definition of the solutions
 - contribute to make a successful implementation of the infrastructure
- Homogeneity is important for the LCG and for its projects
- All work is done seeing the experts and work in LHC experiments and projects
- We all need to compromise
- The reference URL is **<http://spi.cern.ch>** the others may change.

