



# **SPI**

## **Software Process & Infrastructure for LCG**

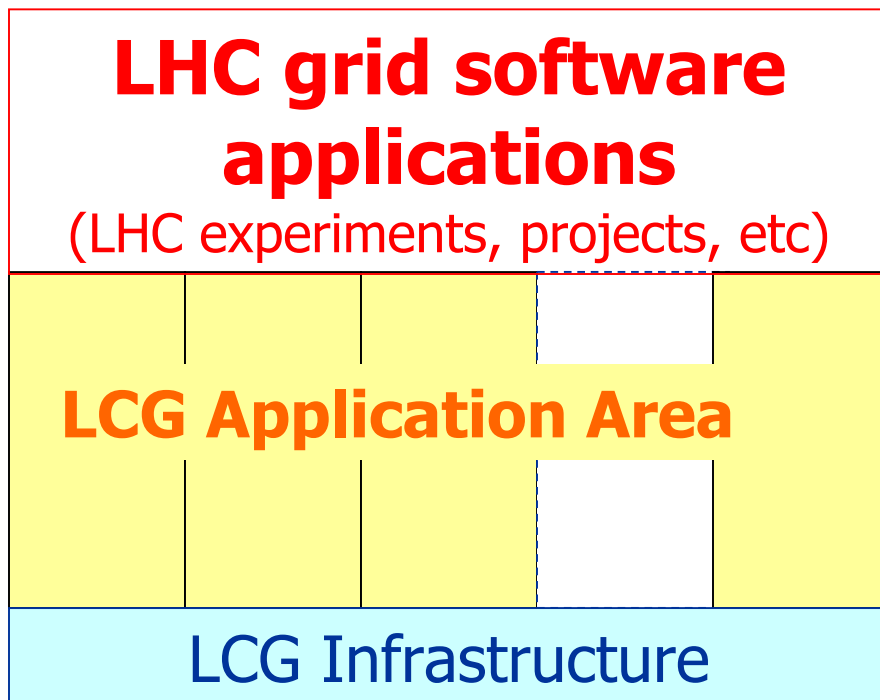
---

### **Project Overview**

LCG Application Area Internal Review  
20-22 October 2003

Alberto AIMAR

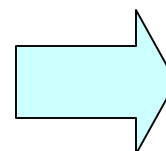
# Project context of LCG SPI



## LCG Application Area software projects

- POOL: Persistency
- SEAL: Core common software
- PI: Physics Interfaces
- SIMU: Simulation
- ...etc...

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

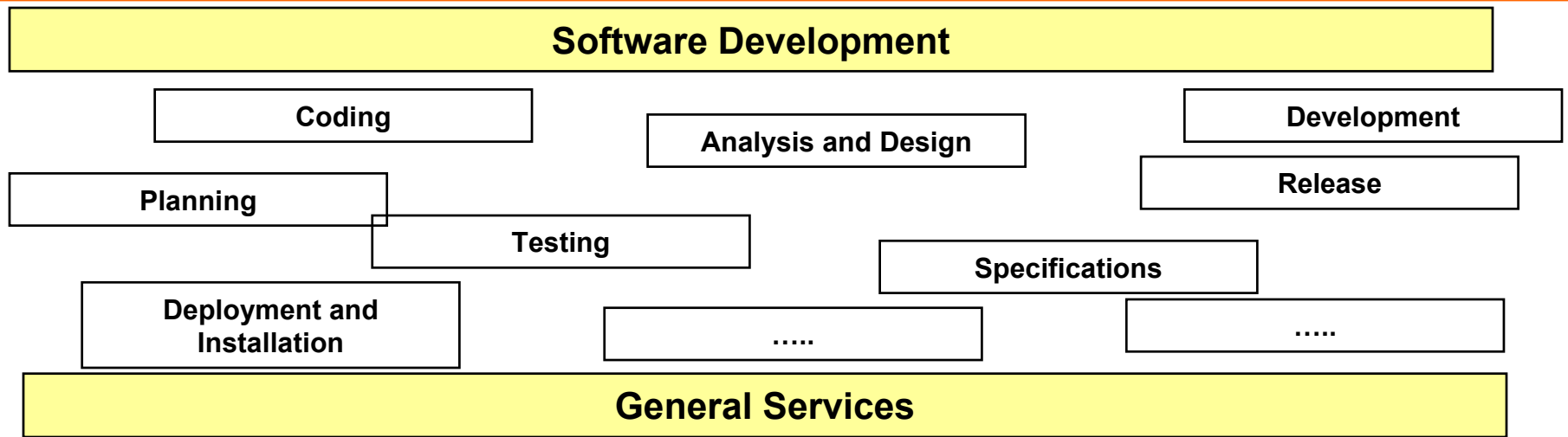


## LCG SPI project

# Project context of the LCG SPI

- “Software Management Process 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.

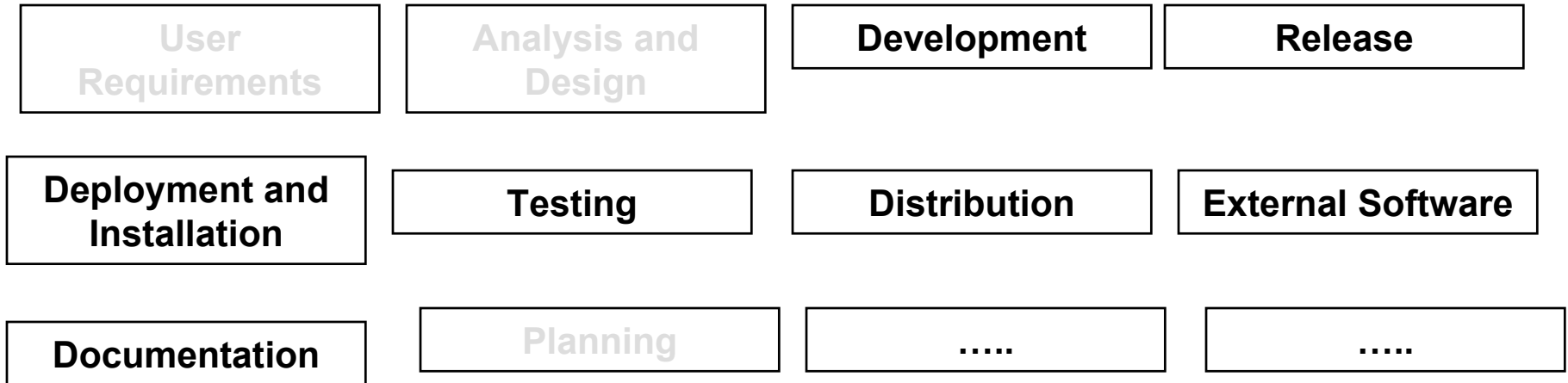
# Project guidelines

- No specific software development
  - The project should avoid any new development, and future maintenance, of any specific software package needed for the LCG software infrastructure.
- Use HEP or free software
  - All software used should be already available, or become available, in the HEP community or in the free software community
- The goal of the project is to define an infrastructure and later a simple process
  - Future maintenance will need separate planning and resources.
- Everything is done in collaboration
  - LCG and LCG projects (Pool, Seal, etc)
  - LHC experiments
  - IT division
  - Big projects (G4, Root, etc)
- Using as the existing IT services
  - The project should use all existing IT services
  - Make sure to match the LCG needs

# SPI project strategy

- Have different and separated services
  - Simple solutions
  - Work with the users
  - Meeting face to face
  - Establish simple deliverables
  - Develop as little as possible
  - Provide common services
- 
- We did not start to provide tools for requirements, design, etc..
  - We started from development-related work (repository, releases, testing, bug report, etc)

# SPI work break-down



- Break the project in “components”
  - Each is a sub-project
  - A responsible person in the LCG SPI
  - Understand and learn the subject
  - Know/find who knows about the subject
  - Provide practical solutions, usable independently

# SPI internal organization

- Each component of the infrastructure has:
  - A responsible person in the project
  - Similar approach
    - Define the goal of the component
    - Standard procedures and documentation
- Standard procedure
  - Survey of possible/existing solutions in HEP and free software
  - Meet the people expert and responsible of the component in real projects (LCG or experiments or big projects)
  - Discuss and agree/decide/verify a solution
  - Present the solution
  - Implement the solutions and make it available
  - Use in the LCG SPI project itself
  - Use it in a real project (LCG or experiment or big project)





# LCG decision process

- The SPI project follows what is decided by the LCG management and by the projects and experiment via the Architects Forum
- Actually we always involve as much as possible the real users and developers and value all help and feedback
- But at some point one must decide and we must provide something
- LCG management support is crucial to SPI
  - especially for what concerns QA, policies issues and standardizations of usage of tools

# SPI services

- Following talks will describe
  - External Software Service
  - Savannah Project Portal
  - Software Testing
  - Quality Assurance
  - Software Distribution
- The rest of this talk is devoted to describe the other services and activities within the SPI project

# 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](#)

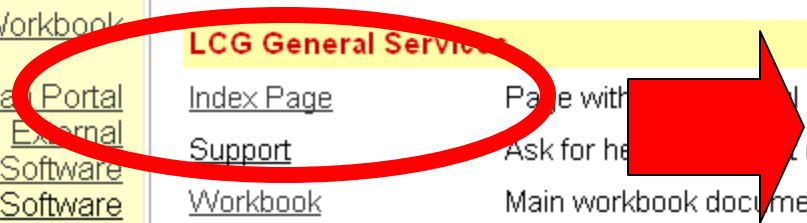
### LCG General Services

- [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 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 is controlled
- Tools for automatic clean up of locks, etc
- Will be moved to use the IT CVS service

## 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 within a project (lcg/contrib)

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search

Address <http://lcgapp.cern.ch/cgi-bin/viewcvs/viewcvs.cgi/>

Google Search Web

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media AutoFill Options

Address <http://lcgapp.cern.ch/cgi-bin/viewcvs/viewcvs.cgi/pool/?cvsroot=POOL>

Google Search Web 7 blocked AutoFill Options



LHC Computing Grid > LCG  
 LCG Software Process & Infrastructure  
**LCG CVS Repository**

[Query the check-in database](#)

[Download tarball](#)

**Project Root**

POOL

- CLHEP
- contrib
- CSCStier2
- gta
- Infrastructure
- lcgdeploy
- packages
- PI
- POOL**
- SEAL
- Simulation
- SPITOOLES
- test
- tools
- web



LHC Computing Grid > LCG Applications Area > LCG Software Process & Infrastructure  
**pool**

[Query the check-in database](#)

[Download tarball](#)

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

Files shown: 1

File	Rev.	Age	Author	Last log entry
<a href="#">AttributeList/</a>				
<a href="#">Collection/</a>				
<a href="#">DataSvc/</a>				
<a href="#">EDGCatalog/</a>				
<a href="#">Examples/</a>				
<a href="#">FileCatalog/</a>				
<a href="#">ImplicitCollection/</a>				
<a href="#">MultiCollection/</a>				

# → 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



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

## LCG Code Documentation using Doxygen

### POOL

- snapshot**
- [POOL\\_1\\_3\\_3](#)
  - [POOL\\_1\\_3\\_2](#)
  - [POOL\\_1\\_3\\_1](#)
  - [POOL\\_1\\_3\\_0](#)
  - [POOL\\_1\\_2\\_1](#)
  - [POOL\\_1\\_2\\_0](#)
  - [POOL\\_1\\_1\\_0](#)
  - [POOL\\_1\\_0\\_0](#)
  - [POOL\\_0\\_5\\_0](#)
  - [POOL\\_0\\_4\\_0](#)
  - [POOL\\_0\\_3\\_1](#)
  - [POOL\\_0\\_3\\_0](#)

### SEAL

- snapshot**
- [SEAL\\_1\\_1\\_0](#)
  - [SEAL\\_1\\_0\\_0](#)
  - [SEAL\\_0\\_3\\_4](#)
  - [SEAL\\_0\\_3\\_3](#)
  - [SEAL\\_0\\_3\\_2](#)
  - [SEAL\\_0\\_3\\_1](#)
  - [SEAL\\_0\\_3\\_0](#)

Doxygen can help you in three ways:

1. It can generate an on-line documentation browser (in HTML) and/or an off-line reference manual (in ) from a set of documented source files. There is also support for generating output in RTF (MS-Word), PostScript, hyperlinked PDF, compressed HTML, and Unix man pages. The documentation is extracted directly from the sources, which makes it much easier to keep the documentation consistent with the source code.
2. Doxygen can be configured to extract the code structure from undocumented source files. This can be very useful to quickly find your way in large source distributions. The relations between the various elements are be visualized by means of include dependency graphs, inheritance diagrams, and collaboration diagrams, which are all generated automatically.
3. You can even `abuse' doxygen for creating normal documentation.

Doxygen is run every night over internal and external releases of the LCG projects, as well as over the head version of the CVS in that moment. For any question or feedback please e-mail to [project-lcg-peb-spi-support@cern.ch](mailto:project-lcg-peb-spi-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)



# 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](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](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.](#)

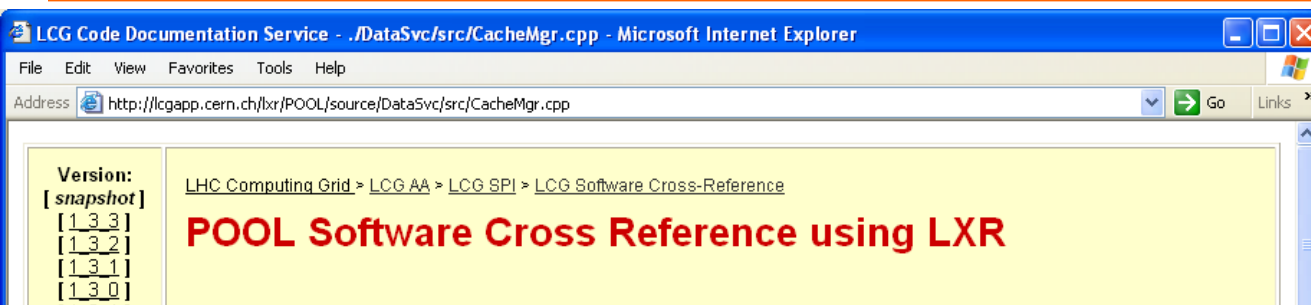
### Public Member Functions

```
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 ()



# Code documentation: LXR



LCG Code Documentation Service - ./DataSvc/src/CacheMgr.cpp - Microsoft Internet Explorer


Address <http://lcgapp.cern.ch/lxr/POOL/source/DataSvc/src/CacheMgr.cpp>

Version: [ snapshot ]

- [ 1\_3\_3 ]
- [ 1\_3\_2 ]
- [ 1\_3\_1 ]
- [ 1\_3\_0 ]
- [ 1\_2\_1 ]
- [ 1\_2\_0 ]

LHC Computing Grid > LCG AA > LCG SPI > LCG Software Cross-Reference

## POOL Software Cross Reference using LXR

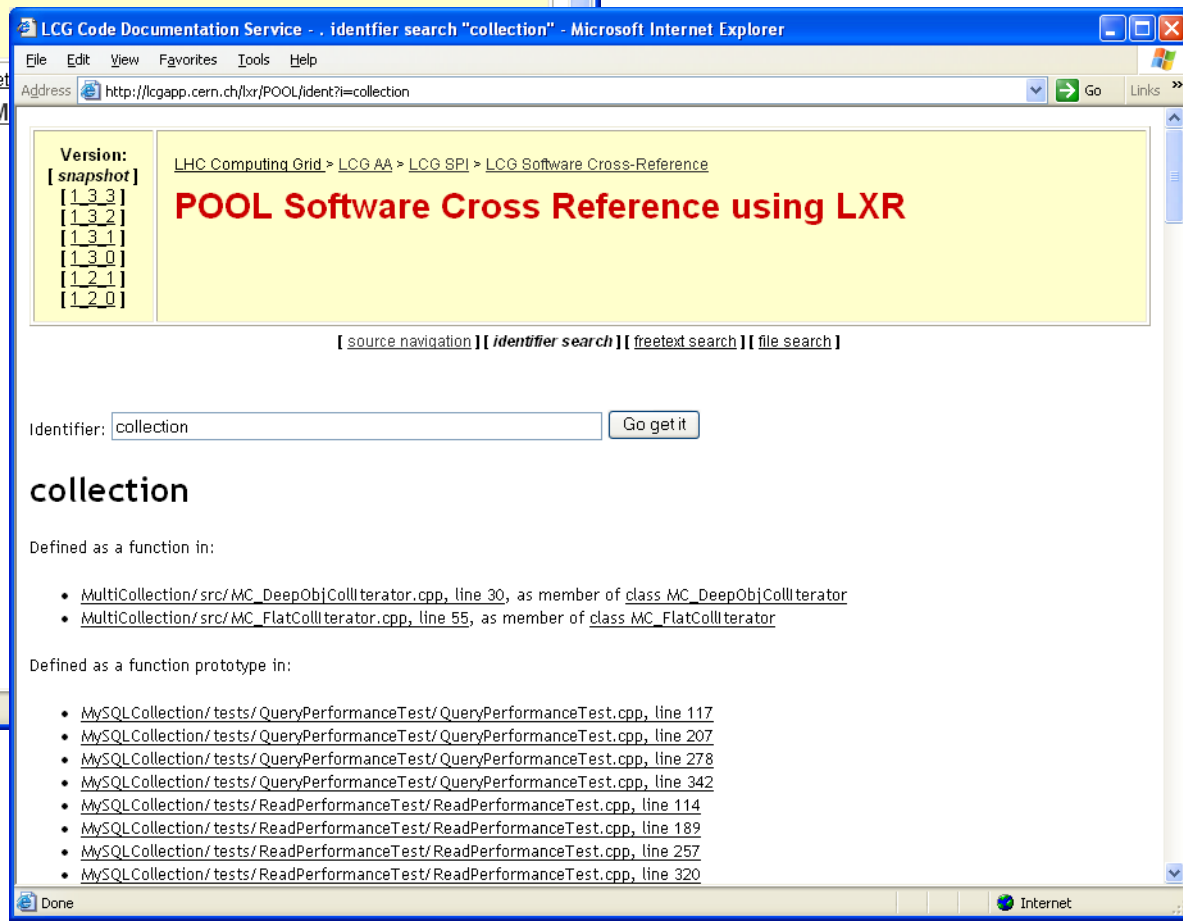


[ source navigation ] [ identifier search ] [ freetext search ] [ file search ]

./DataSvc/src/CacheMgr.cpp

```
File /data/lxr/source/POOL/snapshot/source/DataSvc/src/CacheMgr.cpp
1 #define POOL_CACHEMGR_CPP 1
2
3 #include <iostream>
4 #include "CacheMgr.h"
5
6 // default constructor
7 pool::CacheMgr::CacheMgr() {
8 // create a new Cache table
9 m_cache = new CacheTable();
10 }
11
12 // destructor
13 pool::CacheMgr::~CacheMgr() {
14 delete m_cache;
15 }
16
17 //pool::CacheMgr::CacheMgr(DataCacheProxy::CacheLabel aLabel) {
18 // m_cache = DataCacheProxy::get(aLabel);
19 //}
20
21 // method
22 seal::Status pool::CacheMgr::getLocation(const void* anObjPtr,
```

<http://lcgapp.cern.ch/lxr/POOL/find>



LCG Code Documentation Service - identifier search "collection" - Microsoft Internet Explorer

Address <http://lcgapp.cern.ch/lxr/POOL/ident?=collection>

Version: [ snapshot ]

- [ 1\_3\_3 ]
- [ 1\_3\_2 ]
- [ 1\_3\_1 ]
- [ 1\_3\_0 ]
- [ 1\_2\_1 ]
- [ 1\_2\_0 ]

LHC Computing Grid > LCG AA > LCG SPI > LCG Software Cross-Reference

## POOL Software Cross Reference using LXR

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

Identifier:

### collection

Defined as a function in:

- [MultiCollection/src/MC\\_DeepObjCollterator.cpp](#), line 30, as member of class [MC\\_DeepObjCollterator](#)
- [MultiCollection/src/MC\\_FlatCollterator.cpp](#), line 55, as member of class [MC\\_FlatCollterator](#)

Defined as a function prototype in:

- [MySQLCollection/tests/QueryPerformanceTest/QueryPerformanceTest.cpp](#), line 117
- [MySQLCollection/tests/QueryPerformanceTest/QueryPerformanceTest.cpp](#), line 207
- [MySQLCollection/tests/QueryPerformanceTest/QueryPerformanceTest.cpp](#), line 278
- [MySQLCollection/tests/QueryPerformanceTest/QueryPerformanceTest.cpp](#), line 342
- [MySQLCollection/tests/ReadPerformanceTest/ReadPerformanceTest.cpp](#), line 114
- [MySQLCollection/tests/ReadPerformanceTest/ReadPerformanceTest.cpp](#), line 189
- [MySQLCollection/tests/ReadPerformanceTest/ReadPerformanceTest.cpp](#), line 257
- [MySQLCollection/tests/ReadPerformanceTest/ReadPerformanceTest.cpp](#), line 320

Done Internet



# Code documentation: ViewCVS

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

Signal.cpp/Signal.cpp - view - 1.5 - Microsoft Internet Explorer

Address: http://lcgapp.cern.ch/cgi-bin/viewcvs/viewcvs.cgi/seal/Foundation/SealBase/src/Signal.cpp?rev=1.5&cvsroot=SEAL

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

## Signal.cpp/Signal.cpp

File: [SEAL] / seal / Foundation / SealBase / src / Signal.cpp (download)  
Revision: 1.5, Mon Aug 18 20:04:28 2003 UTC (8 weeks, 2 days ago) by lat  
Branch: MAIN  
CVS Tags: SEAL\_1\_2\_0, SEAL\_1\_1\_0, SEAL\_1\_0\_1\_pre1, HEAD, Foundation\_1\_2\_0, Foundation\_1\_0\_1\_pre1  
Changes since 1.4: +26 -0 lines

\* Signal.cpp (dumpContext): Implement for OS X.

```

//<<<<< INCLUDES
#include "SealBase/Signal.h"
#include "SealBase/Debuguids.h"
#include "SealBase/Log.h"
#include "SealBase/SharedLibrary.h"
#include "SealBase/ProcessInfo.h"
#include "SealBase/BitTraits.h"
#include "SealBase/sysapi/Signal.h"

/* http://dunawww.epfl.ch/ebt-bin/npb-dweb/dynaweb/SGI_Developer/
T_IRIX_Prog/@Generic_BookTextView/7525

POSIX          SVR4          BSD 4.2
-----
sigaction(2)   sigset(2)   sigrec(3)
sigsetops(3)   signal(2)   signal(3)

```

Signal.cpp/Signal.cpp - graph - 1.5 - Microsoft Internet Explorer

Address: http://lcgapp.cern.ch/cgi-bin/viewcvs/viewcvs.cgi/seal/Foundation/SealBase/src/Signal.cpp?graph=1.5&cvsroot=SEAL

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

## Signal.cpp/Signal.cpp

/cvs/SEAL: seal/Foundation/SealBase/src/Signal.c  
Revisions: 5, Branches: 1

```

graph TD
    1["1 MAIN  
09-May-2003 20:41:43"] --> 1_1["1.1  
09-May-2003 20:41:43"]
    1_1 --> 1_2["1.2  
10-May-2003 08:14:34"]
    1_2 --> 1_3["1.3  
16-May-2003 18:36:47  
SEAL_0_3_4  
SEAL_0_3_3  
SEAL_0_3_2  
SEAL_0_3_1  
SEAL_0_3_0  
SEAL_0_3_0_pre3  
SEAL_0_3_0_pre2  
Foundation_0_3_0_pre2"]
    1_3 --> 1_4["1.4  
02-Jul-2003 22:48:02  
SEAL_1_0_0  
SEAL_1_0_0_pre1  
Foundation_1_0_0_pre3  
Foundation_1_0_0_pre2"]
    1_4 --> 1_5["1.5  
18-Aug-2003 21:04:28  
SEAL_1_2_0  
Foundation_1_2_0  
SEAL_1_1_0"]

```

Revisions of seal/Found

# Configuration and build system

- The tools selected by LCG was SCRAM (build system saga)
- All projects are currently building with Scram
- SPI is providing a common configuration for all projects
  - In term of 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
- Supporting the tool is complicated
  - it is used in different ways
  - not very good quality software
- The improvements needed are not completely there
  - Speed issues, porting to Windows, improving efficiency, separate configure from make
  - Fixes instead of proper solutions and the result is not very satisfactory
- It was difficult to provide a tutorial and a good support
- Difficulty is for all tools that we develop instead of adapting to existing solutions (open source, public domain)



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

## SPI - Software Process & Infrastructure for LCG

Updated: 14-Oct-2003 16:51

### 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 and build tool chosen for the LCG Application Area projects.

All LCG software is compiled using SCRAM, as established by the [LCG Software Development Policies](#).

## SCRAM LCG ToolBox

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

Latest configuration: LCG\_20 (see [Configuration](#) file and [Tools-cern.conf](#) file for more information)

Please check the [LCG configuration release notes](#) page to see the change of this and previous releases.

To [post a bug about the SCRAM LCG ToolBox](#), use the SPI savannah portal and choose the "SCRAM LCG ToolBox" category.

## SCRAM User Documentation

Check the online manual for the latest release (v0\_20\_0) in [HTML](#), [PostScript](#), [PDF](#)

# Nightly build system

- 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

NICOS project page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/dist/nightlies/poolwww/> Go Links

## NICOS (Nightly Control System) project page

### Project: POOL

Location: [/afs/cern.ch/sw/lcg/contrib/releases/POOL](http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/dist/nightlies/poolwww/)

Last modified 10/16/2003 01:27:39

**This page**

- shows the status of nightly builds of POOL project managed by the [NICOS system](#)
- provides a link to the Project Configuration Page
- P
- N

NICOS project page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/dist/nightlies/poolwww/> Go Links

The table

release name	status	date	list of tags
<a href="#">POOL_RELEASE_5</a>	completed	10/16 01:27	<a href="#">list of tags</a>
<a href="#">POOL_RELEASE_4</a>	completed	10/15 01:57	<a href="#">list of tags</a>
<a href="#">POOL_RELEASE_3</a>	completed	10/14 02:52	<a href="#">list of tags</a>
<a href="#">POOL_RELEASE_1</a>	configuration in progress (1)	10/12 18:13	<a href="#">list of tags</a>
<a href="#">POOL_RELEASE_0</a>	completed	10/05 18:56	<a href="#">list of tags</a>
<a href="#">POOL_RELEASE_6</a>	configuration in progress (1)	10/11 18:13	<a href="#">list of tags</a>
<a href="#">POOL_RELEASE_2</a>	configuration in progress (1)	10/13 18:13	<a href="#">list of tags</a>

# LCG workbook

---

- Provide a central place for the documentation
- Introduction to new users in the LCG
- Task-oriented
- Web-based
- Inspired by the Babar workbook but we are still far from there
- Needs contributions from all projects
- SPI provides the part about the infrastructure





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

## SPI Workbook

Page Updated: 24-Sep-2003 10:20

### SPI Workbook

### LCG Workbook

### SPI Quick Links

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

[SPI Workbook](#)

### SPI Services Links

[LCG Workbook](#)

[Savannah Portal](#)  
[External 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

## Search SPI web

[Tips](#)

## SPI Index Page

All links useful to the users of the SPI project and the developers of LCG are in [this page](#).

## SPI Support page

Please follow [these instructions](#) for all your support requests.

## Related Links

### LCG Workbook

The Workbook of all the LCG project

# LCG Policies

- CVS Directory Structure Policy
  - C++ Source Code Guidelines
  - 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 a needed by the whole LCG, not by SPI
  - They are defined by the LCG projects, not by SPI
  - If everything is different is too difficult to use and to automatize
  - Is just matter of compromising on our habits for project needs
  - We do not stress the policies a lot, but we gather them
  - We have to tell when they are not followed
  - This is a job assigned to SPI, first time that this is done so is sometimes difficult to achieve it "easily"

http://spi/software\_development.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://spi.cern.ch/software\_development.html Go Links >>



[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](#)  
[LCG Agenda](#)

[PI Project](#)  
[POOL Project](#)

### Software Development Tools

[Setting-up the environment.](#)

LCG software is compiled using SCRAM. See [SCRAM Manual](#).

[Tools for automatic generation of source files and directories.](#)

---

### 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](#)



# Next talks

- Some of the SPI services
  - External Software Service
  - Savannah Project Portal
  - Software Testing
  - Quality Assurance and LCG Policies
  - Software Distribution
- User Feedback
  - Seal
  - Pool
- Summary and perspectives