

InDiCo & JPSP

InDiCo-JPSP

Volker RW
Schaa

Success Story

To Be Decided

What's
Missing

Demo

InDiCo – JPSP Status & Demo

Volker RW Schaa

GSI Darmstadt
Germany

Team Meeting
Knoxville, TN
October 2007



&



After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

InDiCo-JPSP

Volker RW
Schaa

Success Story

To Be Decided

What's
Missing

Demo

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa

InDiCo-JPSP

Volker RW Schaa

Success Story

To Be Decided

What's Missing

Demo

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa

04.05.07 first JACoW JPSP compatible XML output produced by InDiCo

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa

04.05.07 first JACoW JPSP compatible XML output produced by InDiCo

08.05.07 after slight modifications to my script I lowered the number of error lines for CHEP06 from 3246 to 2382.

InDiCo-JPSP

Volker RW Schaa

Success Story

To Be Decided

What's Missing

Demo

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa

04.05.07 first JACoW JPSP compatible XML output produced by InDiCo

08.05.07 after slight modifications to my script I lowered the number of error lines for CHEP06 from 3246 to 2382.

21.05.07 correction of the InDiCo XML and changes to the script

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa

04.05.07 first JACoW JPSP compatible XML output produced by InDiCo

08.05.07 after slight modifications to my script I lowered the number of error lines for CHEP06 from 3246 to 2382.

21.05.07 correction of the InDiCo XML and changes to the script

23.05.07 The number of error lines are down from 2382 to 3!
The script still runs JACoW conferences :-)

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

- 24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa
- 04.05.07 first JACoW JPSP compatible XML output produced by InDiCo
- 08.05.07 after slight modifications to my script I lowered the number of error lines for CHEP06 from 3246 to 2382.
- 21.05.07 correction of the InDiCo XML and changes to the script
- 23.05.07 The number of error lines are down from 2382 to 3!
The script still runs JACoW conferences :-)
- 29.05.07 Re-synchronization with SPMS on <chairs>

After initial talks with the InDiCo team at the Team Meeting in Frascati in 2005 nothing followed up. At the TM in Trieste March, 2007 John suggested that ERL07 in May should use InDiCo. That decision got the development running.

- 24.04.07 meeting at CERN with John Poole, Thomas Baron, Volker RW Schaa
- 04.05.07 first JACoW JPSP compatible XML output produced by InDiCo
- 08.05.07 after slight modifications to my script I lowered the number of error lines for CHEP06 from 3246 to 2382.
- 21.05.07 correction of the InDiCo XML and changes to the script
- 23.05.07 The number of error lines are down from 2382 to 3!
The script still runs JACoW conferences :-)
- 29.05.07 Re-synchronization with SPMS on <chairs>
- 10.06.07 scripts updated to the final XML

- JPSP Scripts are useable, but our real test case (ERL07) is still not finished
- JPSP's special needs must be documented in InDiCo's conference wiki



Indico software management site

Indico software management site > Indico Doc Wikis > Organisers of a JaCoW conference

Organisers of a JaCoW conference

History | Incoming Links

Organisers of a JACoW (Joint Accelerator Conference Website) conference

Indico can now be used to prepare and manage a JACoW conference. Specifically, Indico is now compatible with the special JACoW scripts which allow to create the final proceedings of a conference on the JACoW web site.

As an organiser, you simply have to create and manage your conference like any other conferences on Indico, you then have to contact JACoW people when the event is over so that they extract the data from your Indico site and create the proceedings pages on JACoW.

This does not prevent you from complying to all JACoW standards.

Last modified at 9/20/2007 3:00 PM by Thomas Baron

Site Hierarchy

- Shared Documents
- site pictures
- Announcements
- Calendar
- Indico Doc Wikis**
- Links
- Tasks
- Team Discussion

- JPSP Scripts are useable, but our real test case (ERL07) is still not finished
- JPSP's special needs must be documented in InDiCo's conference wiki
- InDiCo installation for JACoW conferences locally?
- Use InDiCo's CERN server for all JACoW conferences?
- Or use the infrastructure: CERN, Fermilab, IN2P3, INFN, ...?
- Integration of the JACoW user base in InDiCo?
- But would this be authorised by the JACoW rules?

Author Names

[Home](#) | [Session Index](#) | [Classification Index](#) | [Authors Index](#) | [Keyword Index](#) | [List of Institutes](#)

[? A B C D E F G H I J K L M N O P R S T U V W Y Z](#)

JONES, R.

Paper Title	Page
86 Storage for the LHC Experiments	395
<ul style="list-style-type: none"> • R. JONES LANCASTER UNIVERSITY <p>Following on from the LHC experiments' computing Technical Design Reports, HEPIX, with the agreement of the LCG, formed a Storage Task Force. This group was to: examine the current LHC experiment computing models; attempt to determine the data volumes, access patterns and required data security for the various classes of data, as a function of Tier and of time; consider the current storage technologies, their prices in various geographical regions and their suitability for various classes of data storage; attempt to map the required storage capacities to suitable technologies; and formulate a plan to implement the required storage in a timely fashion. This group met for several months, and can now report on its findings.</p> <p> Slides</p>	
83 The ATLAS Computing Model	418
<ul style="list-style-type: none"> • R. JONES LANCASTER UNIVERSITY <p>The ATLAS Computing Model is under continuous development. Previous exercises focussed on the Tier-0/Tier-1 interactions, with an emphasis on the resource implications and only a high-level view of the data and workflow. The work presented here attempts to describe in some detail the data and control flow from the High Level Trigger farms all the way through to the physics user. The current focus is on the use of TAG databases for access and the use of streaming at various levels to optimise the access patterns. There has also been detailed consideration of the required bandwidth to tape and disk, which then informs storage technology decisions. The modelling draws from the</p>	

InDiCo-JPSP

Volker RW Schaa

Success Story

To Be Decided

What's Missing

Demo

List of Institutes

The screenshot shows a Mozilla Firefox browser window displaying the 'List of Institutes' page for CHEP 06 - Proceedings Mumbai, India. The page features a navigation menu with links for Home, Session Index, Classification Index, Authors Index, Keyword Index, and List of Institutes. The main content is divided into two columns: 'List of Institutes' and 'BROOKHAVEN NATIONAL LABORATORY'.

List of Institutes

- [ANL](#)
- [ARGONNE NATIONAL LABORATORY](#)
- [ATLAS](#)
- [Aachen University](#)
- [Academia Sinica Grid Computing Centre](#)
- [BARC](#)
- [BNL](#)
- [BROOKHAVEN NATIONAL LAB](#)
- [BROOKHAVEN NATIONAL LABORATORY](#)
- [Bhabha Atomic Research Centre](#)
- [Boston University](#)
- [Brookhaven National Laboratory](#)

BROOKHAVEN NATIONAL LABORATORY

Author

- [BETTS, W.](#)
- [DePhillips, M.](#)
- [FINE, V.](#)
- [HAJDU, L.](#)
- [Katramatos, D.](#)
- [Nevski, P.](#)
- [Nowak, M.](#)
- [POTEKHIN, M.](#)
- [Pinkenburg, C.](#)
- [VAN BUREN, G.](#)
- [Yu, D.](#)
- [Zhao, X.](#)

InDiCo-JPSP

Volker RW Schaa

Success Story

To Be Decided

What's Missing

Demo

Session headings

ERL 2007 - Table of Sessions

Home | [Session Index](#) | [Classification Index](#) | [Authors Index](#) | [Keyword Index](#) | [List of Institutes](#)

ERL07

ERL 2007 - Proceedings
Daresbury, UK

Table of Sessions

- [sess0-1](#) Opening Plenary
- [sess1-1](#) Monday Afternoon Plenary
- [WG1-1-1](#) WG1 Session 1
- [WG2-1-1](#) WG2 Session 1
- [sess4-1](#) WG3 Session 1
- [sess5-1](#) WG4 Session 1
- [sess9-1](#) WG4 Session 2
- [sess6-1](#) WG1 Session 2
- [sess7-1](#) WG2 Session 2
- [sess8-1](#) WG3 Session 2
- [sess10-1](#) ERLP Tour (A)
- [sess11-1](#) WG1 Session 3
- [sess12-1](#) WG2 Session 3
- [sess13-1](#) WG3 Session 3
- [sess14-1](#) WG4 Session 3
- [sess15-1](#) WG1 Session 4
- [sess16-1](#) WG2 Session 4
- [sess17-1](#) WG3 Session 4
- [sess18-1](#) WG4 Session 4
- [sess19-1](#) ERLP Tour (B)

sess0-1 — Opening Plenary (21-May-2007 09:00—12:00)

Paper	Title	Page
5	Welcome and Goals of Workshop	
	<ul style="list-style-type: none"> M. Poole ASTeC, STFC Daresbury Laboratory S. Chattopadhyay The Cockcroft Institute 	
	no abstract available	
6	Operating ERL-Based FELs and Future Upgrades	
	<ul style="list-style-type: none"> L. Merminga Thomas Jefferson National Accelerator Facility (Jefferson Lab) 	
	no abstract available	
7	Future ERL-Based FELs	
	<ul style="list-style-type: none"> J. Clarke ASTeC, STFC Daresbury Laboratory 	
	no abstract available	
8	ERLs as Hard X-Ray Sources	
	<ul style="list-style-type: none"> G. Hoffstaetter Cornell University 	

InDiCo-JPSP

Volker RW Schaa

Success Story

To Be Decided

What's Missing

Demo

Papercode naming

INDICO-JPSP

Volker RW Schaa

Success Story

To Be Decided

What's Missing

Demo

The screenshot shows a PDF viewer interface with a table of contents on the left and a main document area on the right. The table of contents lists pages 374, 375, 376, and 377. The main document area displays the title 'GENERAL STATUS OF ROOT GUI' and the authors 'I. Antcheva, B. Bellec, R. Brun, F. Eusebi, CERN, Geneva, Switzerland' and 'V. Chuchin, IHEP, Protvino, Russia'. The document text includes an abstract, basic features, cross-platform user interface, and signal/slot communication mechanisms. A diagram at the bottom right of the page shows a flow from 'root [0] x example C' to a GUI window.

Options × Hide

187

Proceedings of CHEP 06, Mumbai, India

GENERAL STATUS OF ROOT GUI

I. Antcheva, B. Bellec, R. Brun, F. Eusebi, CERN, Geneva, Switzerland
V. Chuchin, IHEP, Protvino, Russia

Abstract
ROOT [1, 2] is a scientific data analysis framework providing a large selection of data presentation objects and utilities. The graphical capabilities of ROOT range from 2D canvases to various plots, histograms, and 3D graphical objects. The object-oriented design of ROOT offers considerable benefits for developing object-oriented user interfaces.

The basic features and the progress made with the recent user interface developments in ROOT are presented in this paper.

BASIC FEATURES
The ROOT Graphical User Interface (GUI) classes support an extensive and rich set of user interface elements with a common look and feel. A GUI element (known also as a widget) displays information or provides a specific way for users to interact with the application or the operating system. Widgets include icons, pull-down menus, buttons, progress bars, scroll bars, list trees, toggle buttons, and many other devices for displaying information and for the learning, accepting, and responding to user actions.

The object-oriented GUI library of ROOT offers a full set of application design utilities to programmers. A rich and complete set of widgets can be arranged into container frames that use ROOT container classes for fast look-up.

Cross-Platform User Interface
The ROOT framework offers considerable benefits for developing a fully cross-platform object-oriented user interface.

The widget classes interface to the platform-dependent low-level graphics system, via the abstract class `TFrameUI`. Concrete versions of this abstract class have been implemented for X11, Win32, and Qt [3]. Porting the user interface to any new platform only requires the new implementation of `TFrameUI`.

The benefit of applications running on more than one kind of computer is obvious — it increases the program's robustness, makes maintenance easier and improves the readability of the code.

Signals/Slots Communication Mechanism
The object-oriented, event-driven programming model supports the modern signal/slot communication mechanism as an advanced object communication concept. It largely replaces the old callback functions for handling actions in GUIs. Signals and slots are just like any object-oriented method implemented in C++.

The signal/slot communication mechanism is integrated into the ROOT code. It uses dictionary information and the CINT interpreter to connect signal methods to slot methods. It facilitates programming since it allows a total independence of the interacting classes.

On construction, widgets send out various signals (event handlers). Public object methods can be used as slots (event receivers). Signals and slots can be connected together enabling different components to interact simultaneously like Lego blocks, leaving possibilities for adding new pieces in the future.

Saving User Interface in a Macro
Any ROOT GUI class has `SavePrimitive` method that allows a C++ code generation in a macro file. Users can archive these methods and use any application layout or any dialog layout by using `Canv+V` shortcut. The generated macros can be modified and then executed via the CINT interpreter.

Executing the macro restores the complete original application layout and executes the action handling based on all existing signal/slot connections in a global way.

374

Software Tools and Information Systems

374 (1 of 4)

InDiCo-JPSP

Volker RW
Schaa

Success Story

To Be Decided

What's
Missing

Demo

Demo