# CERN COMPUTER NEWSLETTER

CERN - IT Division
CH-1211 Geneve 23
SWITZERLAND

May – August 2000

CERN-CNL-2000-002
Vol. XXXV, issue no 2

# Contents

**If you need Help** **i**

**Editorial** **iv**

**1 Announcements** **1**
1.1 A New Satellite Help Desk . . . . . 1

**2 Physics Computing** **2**
2.1 CMS Completes Large Scale Objectivity Production . . . . . 2
2.2 AFS File Service, Status and Plans . . . . . 3

**3 Desktop Computing** **5**
3.1 When the Computing Helpdesk Acknowledges your Problem. . . . . . . 5
3.2 Operating System Certification - Present List . . . . . 6
3.3 Range of UNIX Platforms Served by ASIS . . . . . 7

**4 Internet Services and Network** **8**
4.1 Beware of Opening Attachments in E-Mail . . . . . 8

**5 Scientific Applications and Software Engineering** **9**
5.1 Software Development Tools Service . . . . . 9

**6 Desktop Publishing** **10**
6.1 XML Applications at CERN . . . . . 10
6.2 XML document strategies for the Web . . . . . 11

**7 The Learning Zone** **15**
7.1 Computing at CERN, Once Upon a Time... . . . . . 15
7.2 X11 Desktop Environments (KDE and GNOME - Part III) . . . . . 16
7.3 Developing with Plugins and Components . . . . . 22
7.4 Questions and Answers from the Computing Helpdesk . . . . . 27

**8 User Documentation** **32**
8.1 The CNL – 35 years of History at your Fingertips . . . . . 32
8.2 News from the User Support Book Shop . . . . . 32
8.3 User Support Book Catalogue . . . . . 33

**9 Just For Fun . . .** **34**
9.1 A French Story about Consulting . . . . . 34
9.2 Le Monde est Fou... . . . . . 34

# Editorial Information

## Editorial Board



*The opinions expressed in this newsletter are those of the contributors and are not necessarily those of the CERN Management. The Editors reserve the right to edit or omit articles, or hold-over copy due to lack of space.*

## General Information

**The CNL Web home page is at URL:** `http://cern.ch/ref/cnl`

The policy is to have approximatively 4 CNL editions per year (Spring, before and after the Summer, and before Christmas shutdown).

Exceptionally there are only three editions planned this year; so the next scheduled deadline is:

| Issue number | Deadline |
|---|---|
| Christmas edition | Friday 17th of November 2000. |

## Information for contributors

The CNL is processed using XML technology and templates. Article marked up in HTML are preferred, but plain text is also acceptable. A Web form to be used for article submission (contributions) is provided in the CNL "Help&Info" page (`http://cern.ch/ref/cnl`, link "Article Submission form"), or directly at the URL `http://cern.ch/ref/cnl/submit`. Help in using this form is provided at the URL `http://cern.ch/ref/cnl/help/submit`.

## Distribution

- As soon as a new issue of the CNL is available on the Web an electronic mail is sent to the "CNL Announcement Distribution List" at `http://cern.ch/ref/cnl/help/announcement`.
- A paper version is sent to the users who have subscribed to the "CNL Paper Version Registration" list at: `http://cern.ch/ref/cnl/help/register`.

*Subscriptions to our distribution lists* can be made through the CNL "Help&Info" page, at the URL `http://cern.ch/ref/cnl/help`. You can also cancel your subscription in either case by filling in the appropriate form.

The compressed *PostScript file* containing the complete printable version of this CNL is available via the Web at the URL `http://cern.ch/ref/cnl-2000-002/main.ps.gz`. It is possible that, in order to optimize typesetting, the PostScript printable version differs slightly from the HTML master files. In any case the HTML version on the Web must be considered as the reference.

# If you need Help

## FRONT LINE Services

| | Phone | Location | E-mail Address |
|---|---|---|---|
| **Computing Help Desk** | **78888** | **513/R-052** | `Helpdesk@cern.ch` |
| Opening hours: 8:30–17:30 (Monday–Friday) | | | |
| Satellite Helpdesk@B1: Monday–Friday 8:30-12:30 | | | |
| **Administrative Applications (EDH, BHT, etc.)** | **79933** | **5/R-021** | `Ais.Support@cern.ch` |
| Opening hours: 8.30–12:30 and 13:30–17:30 | | | |
| **Central Computer Operators (24/24 hours on call)** | **75011** | **513/R-049** | `Computer.Operations@cern.ch` |
| **General Network Operational Problems** | **74927** | | `Network.Support@cern.ch` |

## Contacts at CERN

The "Computing Help Desk" is the recommended central point for help on all computer-related issues. Following is a non-exhaustive list of individual people responsible or assigned to a given service. If you do not find a pointer to the service you need, then contact the "Computing Help Desk" for submitting your request. For all the services listed below please use the generic electronic mail address whenever possible. These mailboxes will be read even when the usual specialist is absent. It is helpful to use relevant keywords in the subject field of your message. Contact specialists directly only for very urgent cases or for very general consultations. Please note:

- In the *Web version* of this page the "Service Definition" can be a *link* to the Welcome page of this service: we invite you to read carefully all the information given in this page before submitting a question/problem.
- In addition you might find useful to search in the User Support "Questions&Answers" database for possible answers to your query (`http://consult.cern.ch/qas`).
- Many "support" generic electronic mail addresses are just an automatic re-direction to the Computing Helpdesk (this is the case, for instance, for `Mail.Support` or `Www.Support`). Those addresses have been kept mainly for backward compatibility, and also to help the Helpdesk to escalate the query towards the right specialist whenever this is needed.

### USER SUPPORT

| Service Definition | Name | E-mail Address |
|---|---|---|
| User Relations and Service Manager | M. Marquina | `user.relations@cern.ch` |
| Help Desk Manager | R. Woolnough | `user.relations@cern.ch` |
| User Registration | M.C. Perler | `user.registration@cern.ch` |
| Accounting Service | A. Koppanyi | `user.registration@cern.ch` |
| Computing BookShop | J. Megies | `bookshop@cern.ch` |

### CENTRAL COMPUTERS

**Public Services (PLUS batch & interactive services and CSF/PCSF Simulation farms)**
See the complete and up-to-date list of contacts at URL:
`http://http://wwwinfo.cern.ch/pdp/is/main/experiments.html#PUBSERV`

| | | |
|---|---|---|
| SHIFT (Reconstruction & Analysis Clusters) | T. Smith | `tim.smith@cern.ch` |
| CORE Operations | M. Vergari | `mario.vergari@cern.ch` |
| Work Group Servers | T. Smith | `tim.smith@cern.ch` |

**Support for Experiments**
See the complete and up-to-date list of contacts at URL :
`http://http://wwwinfo.cern.ch/pdp/is/main/experiments.html`

| | | |
|---|---|---|
| Distributed File Services (incl. AFS) | | `afs.support@cern.ch` |
| NAP and Parallel Applications | E. Mcintosh | `eric.mcintosh@cern.ch` |
| Central VMS Services | | `vxcern.support@cern.ch` |
| Tape Allocations and Vault Space | R. Minchin | `richard.minchin@cern.ch` |
| Stager, tapes and TMS support | | `tape.support@cern.ch` |
| Central Data Recording (CDR) | | `cdr.support@cern.ch` |
| CASTOR HSM software | | `castor-support@listbox.cern.ch` |
| HPSS Services | | `Hpss.Support@cern.ch` |
| Oracle Databases | N. Segura | `oracle.support@cern.ch` |
| CERN-wide Print Service | J.L. Vosdey | `printer.support@cern.ch` |

**DESKTOP COMPUTING**

| **Service Definition** | **Name** | **E-mail Address** |
|---|---|---|
| NICE (Networked PCs) | F. Hemmer | `nice.support@cern.ch` |
| PC Desktop (Repairs and sales) | C. Ball | `pc.admin@cern.ch` |
| MACintosh Support | F. Ovett | `mac.support@cern.ch` |
| MACintosh Sales | W. Hug | `waltraud.hug@cern.ch` |
| X Terminals Support | | `xterminal.support@cern.ch` |
| ASIS Support | | `asis.support@cern.ch` |

**Operating System Support**

| **Service Definition** | **E-mail Address** |
|---|---|
| HP 700 s/w: | `hp.support@cern.ch` |
| SUN s/w: | `sun.support@cern.ch` |
| Solaris s/w: | `solaris.support@cern.ch` |
| RS/6000 s/w: | `aix.support@cern.ch` |
| Digital Unix s/w: | `dunix.support@cern.ch` |
| Linux: | `linux.support@cern.ch` |

**SCIENTIFIC APPLICATIONS & SOFTWARE ENGINEERING**

| **Service Definition** | **Name** | **E-mail Address** |
|---|---|---|
| PaRC Service (Engineering Cluster) | | `parc.support@cern.ch` |
| CERN Program Library | L. Besson | `heplib.support@cern.ch` |
| PAW Support | O. Couet | `paw.support@cern.ch` |
| LHC++ and GEANT | | `heplib.support@cern.ch` |
| Objectivity DB Service | | `objectivity.support@cern.ch` |
| Software Development Tools | | `sdt.support@cern.ch` |
| Adobe FrameMaker | | `docsys.support@cern.ch` |

**COMPUTING FOR ENGINEERING**

All comments, problem reports, and suggestions can be reported via the "Feedback" Web form accessible from the "IT-CE Home page" (`http://wwwinfo.cern.ch/ce`, link "Feedback" on the left).

**COMMUNICATIONS AND NETWORKS**

A complete and up-to-date list of contacts is at the "Communications Systems Group" Home page at the URL: `http://network.cern.ch`

**INTERNET SERVICES**

| **Service Definition** | **E-mail Address** |
|---|---|
| Web support | `www.support@cern.ch` |
| Mail support | `mail.support@cern.ch` |

In addition to this generic address, the normal support lines are listed at the URL: `http://consult.cern.ch/service/mail/problems/contacts.html`

**COMPUTER SECURITY**

| | |
|---|---|
| Computer Security Information | `computer.security@cern.ch` |
| Security Alert Reports | `cert@cern.ch` |

**Other Newsletters**

*Computing for Engineers* (`http://wwwinfo.cern.ch/ce/newsletter/newsletter.html`)

| | | |
|---|---|---|
| Editor: | L.Pregernig | `ludwig.pregernig@cern.ch` |
| Distribution: | M.Tate-Lavergne | `Monique.Tate-Lavergne@cern.ch` |

## Some Useful Web Pages

| | |
|---|---|
| TvScreen (Systems Status) | `http://tvscreen.cern.ch` |
| Network Connection Request Form | `http://network.cern.ch/register` |
| Remote/Mobile Computing | `http://network.cern.ch/mobile` |
| User-Oriented Catalogue | `http://cern.ch/consult` |
| User-Oriented Service Directory (*) | `http://cern.ch/services` |
| FrameMaker Support | `http://cern.ch/framemaker` |
| Text Processing related documentation | `http://wwwinfo.cern.ch/asdoc/textproc.html` |
| IT Home Page | `http://cern.ch/it` |
| User Support | `http://wwwinfo.cern.ch/support` |
| NICE Home Page | `http://nicewww.cern.ch/` |
| PC Support Service Page | `http://pcwww.cern.ch/` |
| PC Farms Homepage | `http://wwwinfo.cern.ch/pdp/pc/` |
| UNIX Workstation and X Terminals Support | `http://wwwinfo.cern.ch/pdp/ose/support.html` |
| CASTOR Query Report | `http://http://wwwinfo.cern.ch/htbin/pdp/castor_report.pl` |
| Software Development Tools | `http://cern.ch/sdt` |
| Computing Support for Engineering | `http://wwwinfo.cern.ch/ce` |
| Joint Controls Project (LHC Experiments) | `http://itcowww.cern.ch/ITCO/jcop/` |
| Web Services | `http://cern.ch/WebServices` |

(*) *Useful service entry points at this page include the Computing Helpdesk, User Registration, Computing Bookshop, Mail, Netscape, Listbox, etc.*

## Seminars and Presentations

| Designation | Usual Location | Organizer(s) | Tel. | E-mail Address |
|---|---|---|---|---|
| Computing Colloquia | Auditorium / bldg 500 | S. Jarp, IT | 74944 | `sverre.jarp@cern.ch` |
| Computing Seminars | IT Auditorium 31/3-005 | L. Pregernig, IT | 72666 | `computing.seminar@cern.ch` |
| Technical Presentations | IT Auditorium 31/3-005 | S. Cannon, IT | 75036 | `stanley.cannon@cern.ch` |
| Informatics Tutorials | IT Auditorium 31/3-005 | M. Marquina, R. Ramos, IT | | `it-training.service@cern.ch` |

## Computer Resources Allocation and Divisional Contacts

**Computer Resources Allocation Committee (COCOTIME):**

| | | | | |
|---|---|---|---|---|
| Chairman: | H. Hoffmann/DG | 60/5-006 | 75458 | `Hans.Falk.Hoffmann@cern.ch` |
| Secretary: | H. Meinhard/EP | 1/1-028 | 76031 | `Helge.Meinhard@cern.ch` |

**Divisional Representatives for Computing**

For questions about new registration and resource allocation please ask the contact person for your division.

| Div. | Repres. | Div. | Repres. | Div. | Repres. | Div. | Repres. | Div. | Repres. |
|---|---|---|---|---|---|---|---|---|---|
| AS | | AT | P. Heymans | IT | A. Koppanyi | ETT | M. Draper | FI | C. Saitta |
| EST | Ch. Delamare | HR | M. Buttner | EP | A. Bjorkebo | PS | J. Riche | SL | M. Bornand |
| ST | E. Sanchez-Corral | TH | E. Gianolio | TIS | G.R. Stevenson | | | | |

**NICE Divisional Representatives**

See the URL `http://nicewww.cern.ch/homepage/nicerepr.htm`

**Divisional Representatives for Networking**

For questions about new installations, testing, local coordination, etc., please ask the contact person for your division (or, if you are an experimental physicist, the networking contact person within your experimental group).

The complete and up-to-date list of contacts is given in the “Communications Systems Group” Home page, at the URL: `http://network.cern.ch`.

# Editorial

*This second CNL edition of the Year comes at an unusual time, in the middle of the Summer period. In order to have a better planning for the last CNL edition (before the Christmas break) we have reduced the number of issues for the year 2000 to three, instead of the usual four publication. In consequence we have advanced the deadline for the next and last edition of this year, to be:*

*Friday, November 17th.*

*We welcome all your contributions (not restricted to coming from IT division) by that date! This summer edition is quite "light", but we think that the quality is high. We particularly draw your attention to two "dense" contributions that we have received for the* Learning Zone*:*

1. *The third (and last) part of the tutorial series on the new* KDE *and* GNOME *standards. We would like to thank very much* Victor Robles *and* Germán Cancio *for writing this very detailed and interesting tutorial, and sharing with us their own experience.*
2. *A detailed explanation, with many useful references, of the usage of* Plugins and Components*, for application development in a non-monolithic and open approach. We also thank very much* Paula Valença *for providing this tutorial.*

*Either by pure chance, or for some unknown reason, we have to emphasise that these two tutorials were both provided by members of IT/PDP Group! We hope that it will encourage other IT groups to follow the same initiative, and, as stated in the last CNL (`http://cern.ch/ref/cnl/2000/001/EDITORIAL`), help us to* give more weight to the "educational" part.

*With this CNL issue we have also started our new* electronic article check-in *procedure, directly accessible from the* CNL Home Page *(`http://cern.ch/ref/cnl`, link "Article Submission form"), or directly at the URL `http://cern.ch/ref/cnl/submit`.*

*The complete procedure is explained in detailed in the "Submission help Page", at the URL `http://cern.ch/ref/cnl/help/submit`.*

*We should note that this new registration procedure allows any users to see "on-line" the evolution of the draft CNL, before it is officially produced on the Web (and approved by the divisional management).*

*Let us finish this editorial in the hope that you will enjoy reading this NewsLetter issue.*

*Nicole Crémel,*
*CNL Editor, IT/User Support.*

# 1. Announcements

*This chapter contains announcements concrete in time and/or of informative nature from the division to the User Community.*

## 1.1 A New Satellite Help Desk

*Nicole Crémel, Miguel Marquina, Roger Woolnough, IT/User Support*

**Abstract**

*News of the opening of the new satellite helpdesk in Building 1 and the closure of the Building 40 desk.*

As of September 1st, the satellite Helpdesk in Building 40 will close, and a new office, in building 1, will open. Although the idea of a satellite desk in Building 40 appeared to exploit an ideal situation, it has never really been used to full capacity; hence the new desk aims to capture a larger local audience. EP Division has kindly found some office space for us next to the main Library in building 1; the area has been refurbished by ST division and funded fifty-fifty between EP and IT. Our warm thanks go to both EP and ST for their efforts.

As the premises are small, the documentation and book browsing facilities will be hosted by our colleagues in the Central Library, 20 meters down the corridor. Opening times will be similar to the present desk in building 40, 8.30 to 12.30, Monday to Friday.

Please take advantage of this convenient new location and if well used we will certainly be looking to upgrade the opening hours to full time!

A complete article on Helpdesk strategy (and what users can expect when calling) is planned for the *Weekly Bulletin* in early September. As a foretaste of this please see the CNL article published in the last issue, and available at the URL `http://cern.ch/ref/CNL-2000-001/helpdesk`.

# 2. Physics Computing

*Functional description of all the Central Public Services, specifically oriented to use by the Research sector, either for interactive (central and CPU intensive) or batch processing.*

## 2.1 CMS Completes Large Scale Objectivity Production

*David Stickland, EP*

**Abstract**

*CMS in collaboration with IT has recently completed a massive data processing aimed at validating its Higher Level Trigger (HLT). The CMS off-line reconstruction baseline is to use an Object Database Management System for both Event data and meta-data and this scenario has been used to process 2 million LHC events with Pileup corresponding to a luminosity of* $10^{34}\mathrm{cm}^2\mathrm{s}^{-1}$*. This production required the use of some 200 Linux/Intel CPUs, the interfacing of Objectivity to the CERN Mass Storage System (HPSS) where 5TB of data are stored and the movement of 70TB of Pileup hits. The production was completed in two weeks and is now in use by CMS physicists studying Trigger algorithms and their performance.*

The Higher Level Trigger (HLT) of CMS will be implemented entirely in software and the validation of its rejection power and efficiency is a milestone of CMS. CMS has decided that this validation should be performed using software that is functionally as close as possible to that which will be used in the final experiment. The Object Reconstruction for CMS Analysis program, ORCA (`http://cmsdoc.cern.ch/orca`) has been developed with this as its primary short-term goal. This is an Object-Oriented program implemented in C++ based on the design guidelines laid out in the CMS Computing Technical Proposal and built on the foundations of several years test-beam experience.

The immediate goal of the Spring ORCA/HLT production was to prepare a fully digitized sample of some two million events with full pile-up at a simulated luminosity of $10^{34}\mathrm{cm}^2\mathrm{s}^{-1}$. This is a multi-stage process taking about two million CPU minutes of computing ($2 \times 10^9$ SI95 $\cdot$ s) and involving the disk and network I/O of some 70 TB of pile-up hits.

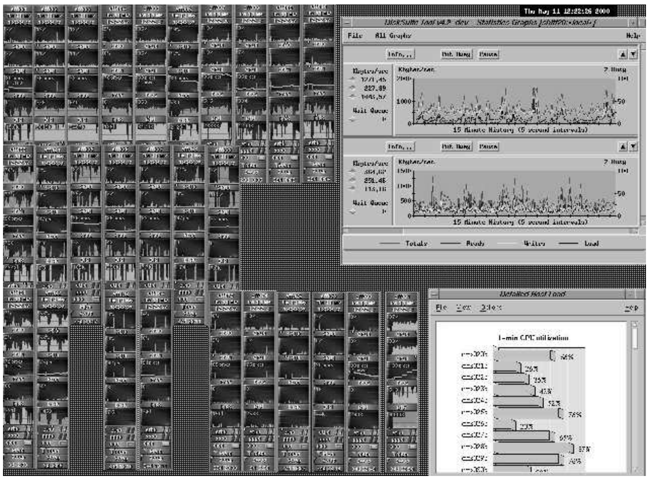
The production starts from some 2 TB GEANT3 files which were converted into Objectivity/DB format and stored in the HPSS (`http://wwwinfo.cern.ch/pdp/hpss`) system. Most of the GEANT3 production was performed off of the CERN site in Finland, Italy, Russia and the USA. Each signal event was then convoluted with $\sim 10$ bunch crossings of minimum-bias events at a rate of $\sim 20$ events/crossing (these are the 70 TB of data that must be transported). The final results are again stored in an Objectivity/DB where they are used in reconstruction and analysis studies.

This was achieved making use of a farm of about 100 dual-cpu high performance PCs running the Linux operating system, part of the Event Filter Farm shared between experiments for such development tests.

Since the total quantity of data is in excess of the disk space available, an interface, developed in collaboration between CERN IT/DB and SLAC, between the Objectivity/DB AMS (Advanced Multi-threaded Server) and the CERN HPSS system was used. Via this interface requests for data objects in files that are no longer on disk are automatically staged in. To make maximum usage of the resources available this AMS interface was set up on more than 30 nodes (Linux and Sun/Solaris) of which 8 also accessed HPSS.

The Central Data Recording (CDR) scripts, previously used heavily in test-beams, were used to migrate completed database files out of the production system and into HPSS. Parts of this work were going on at the same time as the ALICE data challenge reported in the previous CNL resulting in considerable extra challenges for the personnel in IT/PDP responsible for keeping these systems operational.

Considerable effort went into monitoring the many different systems all required to be fully operational to carry out this production. Information was collected from all nodes in the system using a protocol based on that of the NetLogger (`http://www-didc.lbl.gov/NetLogger/homepage.html`) Project from LBNL. The information was collated and presented in the form of time dependent histograms. This typically enabled us to understand why a breakdown occurred. In the next production, planned for this autumn, we intend to deploy more active control of running jobs so that we can react before some problems develop into catastrophic ones.



The figure shows CPU/Disk/Network activities on each of the nodes running an AMS (see `http://cmsdoc.cern.ch./orca/cnl700/prodscreendump.jpg` for a detailed view).

The pileup operation, whereby we try to simulate the LHC event environment, requires the superposition of 35 MB of "hits" from some 150 minimum bias events in many beam crossings and the hits of each signal event (1 MB) and the subsequent simulation of the detector digitization process to yield digits in roughly the same format as will eventually be recorded by the detector. The Computing time for this process was about 1-1.5 minutes/event and with 140 CPUs working in parallel in the production we achieved a steady state reading rate out of Objectivity/DB of 50 MB/s. Physically this was achieved by distributing the pileup "hits" over 24 PCs each running an AMS. A further 6 PCs with larger disk caches served the signal events and a Sun Server was used to record the output objects into the database files

Features of Objectivity/DB such as the ability to put different types of data into different containers and then to cluster them according to their expected use patterns were invaluable in allowing us to meet this challenge. For example when storing the GEANT3 output into Objectivity/DB we ensured that MC truth information, calorimeter and muon hits and tracker hits went into different files, as for this particular exercise only the calorimeter/muon digits were required in the first production step. Users performing analysis may navigate to the MC truth information. Having performed a selection of events satisfying some criteria users can create either a "shallow copy" (partial copy of the complex event containing links to original data objects) or "deep copy" (new copies of the events), depending on how the collections will be used.

In a subsequent user or production step, the Tracker Digitization and Track Finding can be added to the existing digitizations to allow the further analysis of this interesting subset of events.

This is only the second in a series of such challenges designed to reach the goal of a factor of 1000 rejection of the LHC Level-1 triggers before storing the data. Eventually the software to achieve this must be of unprecedentedly high-quality as it will be removing genuine physics events from the data flow at an earlier stage than has been the case in previous experiments. Performing this work with the computing resources available in 2000, instead of in 2005, requires close collaboration between software and computing experts both in CMS and in IT. Many obstacles had to be overcome, and of course many have still to be discovered as the scale of the productions is increased to reach the final scale of LHC computing. There is however no substitute for meeting these challenges one at a time and gaining the experience that will be required to carry out the analysis of the LHC experiments.

In Autumn 2000, the next massive production will take place. This time CMS intends to make use of some of its worldwide computing resources carrying out Objectivity/DB population and analysis in multiple sites and coordinating them using prototype GRID tools derived from the GLOBUS project by CMS.

**About the author(s):** *David Stickland is the CMS Reconstruction Coordinator.*

---

## 2.2 AFS File Service, Status and Plans

*Bernard Antoine, David Asbury, Olivier Le Moigne, Harry Renshall, Tim Whibley, IT/PDP and IT/TS*

During the last twelve months we have been executing a plan to modernise and homogenise the AFS service both at the hardware and software level. By the end of May we had replaced some dozen old file-servers, a mixture of IBM and Sun having disk spaces ranging from 50 to 150GB and not all on secure RAID, by 5 Sun E220 servers each with 300 GB of RAID disk. In June we replaced the three old volume location data base servers by new Sun Netra servers (which no longer have to function also as file servers) and in July the addition of a sixth Sun E220 allowed us to shut down 14 old PCs running a non-standard version of AFS. We also added a large Sun E450 machine to function as a central backup server using an STK DLT 7000 robot and replacing a number of directly attached DLT stackers. All the new machines have fast networking.

All the remaining or new servers are capable of running the same latest level of the AFS server software namely 3.6 patch level 1. This level is multi threaded, so more efficient, and supports backup and restore of volumes up to 8GB in size (the previous limit was 2GB). We successfully migrated to this level on 31 July. When this has proved stable AFS client machines can be brought to the same level.

The final changes planned for this year will be to replace the last two IBM file servers, which use our deprecated FDDI networking, by two more of our standard new configuration. This should happen during August or early September and should be transparent to end users.

We appreciate the patience of our end users during this evolution which has already led to improved reliability and performance. Changes next year should be much more transparent. There are two older Suns that should be replaced then new capacity to be added according to the COCOTIME evaluations currently underway.

The changes have meant that all end user home directory and backed up project directory files are on secure RAID disks. Two disks out of six can fail without losing access (there is a hot spare mechanism). The non-backed up project directory files are also on modern disks but a single disk failure will lose data. It is very important that groups only use the non-backed up space for genuine scratch data.

You can recognise which type of disk a directory is on by the command `fs lq your-directory-name`. This will return, under the heading `Volume Name`, a name beginning:

`user.` backed up user home directory
`u.` backed up user workspace directory
`p.` backed up project directory
`q.` non-backed up project directory - SCRATCH ONLY

We also take this opportunity to announce the departure from CERN of *Olivier Le Moigne* who is returning to French industry. He has made major contributions to the AFS service during his 3 years at CERN not least in training those of us who formed the new AFS team last year. We thank him sincerely for his hard work and dedication and wish him well in his future career.

# 3. Desktop Computing

*It includes CERN desktop environment centrally provided on UNIX and PCs (NICE95 and NICE-NT), desktop software strategy, etc.*

## 3.1 When the Computing Helpdesk Acknowledges your Problem...

*Nicole Crémel, IT/User Support*

**Abstract**

*This is a description of the different kinds of notification messages sent back from the Computing Helpdesk, as set-up in the Problem Report Management System (PRMS) that they use. This article provides also an explanation why the messages are not all the same for the time being.*

When you send a request to the Computing Helpdesk (`Helpdesk@cern.ch`, or any e-mail addresses which is redirected to it, like, for instance, `Nice.Support@cern.ch` or `PARC.Support@cern.ch`) you should almost immediately receive a notification message (with a ticket number), acknowledging that your problem has been received. Currently the format of the notification message is not always the same, and depends on the "domain" where the problem had been assigned by the Helpdesk. Some of the messages are " reply-able" (it is possible to do a simple `Reply` from any mail agent), but not all of them (it is explicitly written in the body of the message when you cannot reply).

The Helpdesk is using a *Problem Report Management System* (PRMS) in order to treat requests in a "professional" way: all requests are recorded in a database, no request must be lost, an answer or a ticket must be sent back to the user within a delay of two hours, and all requests need a solution to be closed. The PRMS also provides many statistical information (e.g., time delay, or time spent on all interventions) which were absolutely needed since the Helpdesk and many lines of support have been outsourced. A commercial tool is used provided by the company *Remedy Corp.*

For the time being here are, basically, the different kind of messages you may receive from the Helpdesk:

### Notification message sent by User Support

All those messages (with ticket numbers `USxxxx` ) are "reply-able", and they have the form:

```
From helpdesk@cern.ch Tue Apr 11 14:47:47 2000
Date: Tue, 11 Apr 2000 14:47:05 +0200 (MET DST)
From: Computing Helpdesk <helpdesk@cern.ch>
To: Michel.Dupond@cern.ch
Subject: US9846 Received - Problem with mail

Message :

Your query was received and assigned to User Support
staff. You may follow the status of your current problem
(in fifteen minutes from the time you receive this mail),
at the Web location:

        http://cern.ch/consult/problem/9846

Please do not forget to use your ticket number "US9846"
in further communications. In addition you might find
useful to search in the User Support "Questions&Answers"
Database (http://cern.ch/consult/qas) for possible answers
to your query.

Best Regards
+-----------------------------------------------------+
CERN Computing Help Desk            tel :  +41.22.76.78888
IT Division (http://wwwinfo.cern.ch)fax :  +41.22.76.78630
European Organization for Nuclear Research
1211 Geneve 23 - Switzerland      email : helpdesk@cern.ch
+-----------------------------------------------------+
User Subject..: Problem with mail
User Statement:

...
```

### Notification message sent by `"ar admin"`

This concerns queries which are assigned to various parts of the "DSSM service contract" (outsourced support staff) and various support sections in IT (e.g., for system administration, or `PARC.Support`). Those messages (ticket number `CTxxxxxxxxxxxx` ) are *not* "reply-able". But as soon as a "possible solution" is provided, you should find, in the text of the message, the name of the contact person in order to communicate (if needed). Such messages look like:

```
Date: Thu, 17 Aug 2000 16:10:49 +0200 (MET DST)
From: ar admin <ARSystem@sunar01.cern.ch>
To: Michel.Dupond@cern.ch
Subject: ...

Your report has been entered into the IT Problem Report
Management System with the ID:  CT0000000033990.

For details see the url:

http://consult.cern.ch/problem/ct/CT0000000033990

Arrival Time      : 08/17/00 16:10:47
Short Description : ...
```

### Notification message sent by EDMS Support

These concern queries specific to the EDMS, CDD, Euclid and Autocad areas. All those messages (with ticket numbers `NNNNN` ) are "reply-able", and they look like:

```
Date: Thu, 18 May 2000 15:52:00 +0200 (MET DST)
From: EDMS.Support@cern.ch
Subject: Request REGISTERED for Help with problem
        in EDMS (No. 25187)
Apparently-To: <Michel.Dupond@cern.ch>

Your request has been REGISTERED.
```

```
Request No : 25187
Request type : Help with problem
Application  : EDMS
Request status : New

Summary :
 ...

Details :
 ...
You will be informed about the work performed for your
request by e-mail or alternatively on the web at
http://aisremedy.cern.ch/ars/cgi-bin/RequestEDMS?25187

The answer to many of your questions may be found on
the new AIS web site at: http://ais.cern.ch ! Please
take the time for a quick tour of the site and return
your comments to us at: ais.webmaster@cern.ch.
We value your feedback!

This mail was automatically generated by the AIS
Request Management System.
```

However, to facilitate the "follow-up" of any request (whatever domain they belong to), the User Support group has provided a "common" Web interface which can be used to check the status and eventually look at the solution when it has been provided. This interface is available at URL `http://consult.cern.ch/problems/`.

You just have to enter the *Ticket number* (only the number part), the *Domain* (`US, CT,` or `AIS`) and click on *View Problem.* With the ticket number it is also possible to access directly the problem description and status, depending on the domain:

- `http://consult.cern.ch/problem/US`*nnnn* for case 1 (message sent by User Support); `US` can be omitted.
- `http://consult.cern.ch/problem/CT`*nnnn* for case 2 (message sent by contact staff `(CT)/"ar admin"`).
- `http://consult.cern.ch/problem/AIS`*nnnn* for case 3 (message sent by EDMS Support/AIS system).

In each case *nnnn* stands for the ticket "number" (last part of the ticket).

In the future (with a better design of the tool currently used at the Helpdesk) all the notification messages will become "reply-able" (i.e. giving the possibility to the user to do a simple reply, to the appropriate person, with his mail agent). Also, some standardisation across the different domains will be made. It is already foreseen that at least cases 1 and 2 will be merged, probably during the autumn, in a unique and completely re-designed system (CPRMS, *CERN Problem Report Management System*), and that the notification messages will be standardized.

**About the author(s):**

*Nicole Crémel is working in the User Support group and, with Roger Woolnough, is responsible for the supervision of operations of the Computing Helpdesk.*

---

## 3.2 Operating System Certification - Present List

*Philippe Defert, IT/PDP*

**Abstract**

*This article is a regular reference point for the list of operating systems certified and supported by the UNIX support team.*

As stated in CNL 224 (`http://cern.ch/consult/cnls/224/art_cert.html`), here is the list of various versions of UNIX which we certify as working correctly with the CERN-recommended libraries, tools and utilities as of March, 2000:

- Linux Red Hat 6.1 - certified and recommended
- Solaris 2.6 - certified and recommended
- Solaris 7 - certified but still missing some bits
- HP-UX 10.20 ACE - certified and recommended
- Digital UNIX 4.0.f - certified and recommended
- AIX/6000 4.3.2 - certified and recommended.

For those interested in the certification process, it is described in the Web page: `http://wwwinfo.cern.ch/pdp/ose/certified/`.

The latest news on the certification procedure is always published in the newsgroups associated with each UNIX flavour (for example `cern.linux`, etc.) and summarised in the Web page listed above as well as in each CNL.

**About the author(s):** *Philippe Defert is the UNIX support manager.*

---

## 3.3 Range of UNIX Platforms Served by ASIS

*Philippe Defert, IT/PDP*

**Abstract**

*This article is a regular reference point for the platforms served by the ASIS server and their current level of support.*

The status with the ASIS Application Software repository is presently as follows:

### Supported platforms:

- `i386_redhat61` : Linux Red Hat 6.1
- `sun4x_56` : SUN Solaris 2.6
- `sun4x_57` : SUN Solaris 7 (partial)
- `hp_ux102` : HP-UX 10.20
- `alpha_dux40` : DEC OSF 4.0f
- `rs_aix43` : IBM AIX 4.3.2

### Unsupported platforms:

- `sgi_64` : SGI Irix 6.4 and 6.5 (active)
- `sun4c_411` (SUNOS 4.1.3) (frozen since 1997)
- `hp700_ux90` (HP-UX 9.05) (frozen since 1997)
- `alpha_osf32` (DEC OSF 3.2c) (frozen since 1997)
- `rs_aix41` : IBM AIX 4.1.5 (frozen since 1998)
- `rs_aix42` : IBM AIX 4.2.x (frozen since 1999)
- `sun4x_55` : SUN Solaris 2.5.1 (frozen since 1999)
- `i386_linux2`: Linux Red Hat 4.1 (frozen since 1999)
- `i386_redhat51` : Linux Red Hat 5.1 (frozen in May 2000)
- `i386_redhat60` : Linux Red Hat 6.0 (frozen in February 2000)

A platform is declared *frozen* when all applications are kept in the repository but no new package or version of a package is ever introduced in the repository for this architecture.

A platform is declared *partial* when only some part of the applications is generated for this particular Operating System. Getting access to ASIS applications is to be done with the appropriate tools which will give you some application of this particular platform and the rest from the closest O.S., called the *fail-over*.

**About the author(s):** *Philippe Defert is the service manager of the ASIS software server.*

# 4. Internet Services and Network

*Everything related to Internet in a large sense: i.e., network issues and performances, all technical Web issues, electronic mail, Internet news, home access to CERN facilities, video conferencing and multi-media, network security, etc.*

## 4.1 Beware of Opening Attachments in E-Mail

*Denise Heagerty, IT/IS*

Following the recent e-mail viruses you are asked to be cautious when opening e-mail attachments on your PC. Attachments that are apparently harmless might hide executable content that is activated when the attachment is opened.

On the PC Windows infrastructure, CERN runs software to protect users from known viruses, but new ones can sometimes arrive before our software recognises them. They can infect your system when you open an attachment and can result in data loss or corruption on your PCs local disk and your home directory.

To remove them it may be necessary to re-install the Windows software on your PC.

CERN anti-virus information is available at

`http://www.cern.ch/pcvirus`.

**About the author(s):** *Denise Heagerty is the Computer Security Officer*

# 5. Scientific Applications and Software Engineering

*This includes descriptions of: scientific applications and libraries from CERNLIB and LHC++, mainly used by the Physicists;engineering applications likes the ones on PaRC or those associated to online controls;software development environment.*

## 5.1 Software Development Tools Service

*Eric Poinsignon*

**Abstract**

*Description, content and news from the service.*

In SDT service we actively help software projects in CERN experiments to identify, evaluate, purchase and use industrial software development tools. Network floating licences are used as much as possible, so that the tools can be run remotely. We can also, if you have an AFS account, give you access to our Solaris server in case the binary for your platform is not available. You can find the complete information on what we have and how to use it at URL:

`http://cern.ch/sdt`

Contact e-mail address: `Sdt.Support@cern.ch`

### New tool in production

*CodeWizard 3.1* (`http://cern.ch/sdt/CodeWizard/`) Coding standards enforcement tool that prevents errors and standardises C/C++ code.

It runs on SUN/Solaris, IBM/AIX, HP/HPUX, SGI/Irix, COMPAQ/Tru64, and PC/Linux.

CodeWizard is provided with 70 predefined rules, and you can add your own. A set of CERN rules are defined but not implemented yet. All the details are in the page

`http://cern.ch/sdt/CodeWizard/Rules.html`

### SDT Service has moved to building 513

The e-mail addresses and phone numbers did not change, but if you need paper documentation about a tool, you need to come to Bld. 513, 1-005. Same if you need to borrow one of our CDROMs, except for FrameMaker which is distributed by Mario Ruggier in Bld. 1, R-017.

**About the author(s):** *Eric Poinsignon is working for the SDT Service run by IT/API.*

# 6. Desktop Publishing

*Everything related to "Text Processing": HTML issues, LaTeX, Word, Framemaker, etc.*

## 6.1 XML Applications at CERN

*Michel Goossens, IT/API*

**Abstract**

*Several XML applications (commercial and public domain) are already installed at CERN. They are used in various areas of electronic data handling, such as document production, database queries, electronic data exchange, etc. This article provides an update about the availability of some XML-related free tools.*

More and more XML-related tools are becoming available regularly. I have evaluated a few of them and installed a couple of Java-based interesting offerings on the central Unix systems. Please send your suggestions for adding other packages to the author. Below you find a list of packages that have been updated recently. For each of them there is a brief description. More information is available in the documentation. Please note that the version numbers of the packages can vary if a more recent gets installed. Hence you may have to look into the directory of the package name, e.g., into `/usr/local/doc/JAVA/xerces` for Xerces, etc.

### xerces

Free source validating XML parser distributed by the Apache Project. The Xerces Java Parser 1.1.3 supports XML 1.0 recommendation and contains advanced parser functionality, such as XML Schema, DOM Level 2 version 1.0, and SAX Version 2, in addition to supporting the industry-standard DOM Level 1 and SAX version 1 APIs. Documentation for the current version (presently 1.1.3), is at the URL `http://xml.apache.org/xerces-j/index.html`. At CERN we have only installed this Java version, but work is ongoing at Apache to develop a C++ and Perl version.

At CERN possible command parameters are displayed by typing the `xerces` command. Examples of using `xerces` are

```
xerces -saxcount -v invitationfr.xml
invitationfr.xml: 2003 ms
 (14 elems, 0 attrs, 18 spaces, 384 chars)
xerces -saxcount invitationfr.xml
invitationfr.xml: 1693 ms
 (14 elems, 0 attrs, 18 spaces, 384 chars)
```

The first instance, with the `-v` switch, verifies the file and counts the elements, while the second instance without that switch merely checks the file for well-formedness.

### xalan

Free source XSL parser written in Java distributed by the Apache Project. Documentation for the current version is at the URL `http://xml.apache.org/xalan/overview.html`. The API definition is at `http://xml.apache.org/xalan/apidocs/index.html`. The presently installed version is 1.1.

A transformation of an XML file by an XSL stylesheet is obtained as follows:

```
xalan invitationfr.xml invlat1fr.xsl a.tex

========= Parsing file:invlat1fr.xsl ==========
Parse of file:invlat1fr.xsl took 3851 milliseconds
========= Parsing invitationfr.xml ==========
Parse of invitationfr.xml took 1132 milliseconds
============================
Transforming...
transform took 345 milliseconds
XSLProcessor: done
```

### saxonxsl

Michael Kay's XSL parser. Documentation for the latest version (presently 5.4) is at `http://users.iclway.co.uk/mhkay/saxon/`.

A transformation of an XML file by an XSL stylesheet is obtained as follows:

```
saxonxsl -t invitationfr.xml invlat1fr.xsl > a.tex
SAXON from Michael Kay of ICL
Version 5.4
Elapsed time: 2235 milliseconds
```

### oraxsl

Oracle's XSL parser (`oraxml` is the corresponding XML parser). Information about the current version (presently 2.0.2.9, check for the latest version if needed) is at `http://technet.oracle.com/tech/xml/parser_java2/index.htm`. From that page one can find further documentation, in particular the API definitions for the Java classes. At CERN sample Java code is available in the directory `/usr/local/doc/JAVA/oraxmlxsl/2.0.2.9/sample/`.

A transformation of an XML file by an XSL stylesheet is obtained as follows:

```
oraxsl -v invitationfr.xml invlat1fr.xsl a.tex
1 XML document will be transformed using XSLT
stylesheet specified in invlat1fr.xsl with 1 thread
Parsing file invlat1fr.xsl
Parsing file invitationfr.xml
Transforming XML document specified in invitationfr.xml
```

### fop

Open source XSL formatting object to PDF convertor developed by the Apache Project. Documentation for the current version (presently 0.13.0) is available at `http://xml.apache.org/fop/index.html`. However, most of the time you will just type a command like the following to obtain a PDF file from an XML file using an XSL stylesheet that transform your XML element into XSL FO objects.

```
fop invitation.xml invfo1.xsl  a.pdf
FOP 0.12.2 [dev]
using SAX parser org.apache.xerces.parsers.SAXParser
using renderer org.apache.fop.render.pdf.PDFRenderer
   ...
building formatting object tree
setting up fonts
formatting FOs into areas
 [1]
rendering areas to PDF
writing out PDF
```

The PDF file `a.pdf` can be viewed with `acroread` or `gf` and printed if needed.

### svgtools

A few tools for supporting the Scalable Vector Graphics XML language. The command `svgview-csiro` will open an SVG editor (you need java 1.2.2 or higher to run this program). Samples of SVG sources are (at CERN) in the directory
`/usr/local/doc/JAVA/svgtoolkit/20000712/samples` and its subdirectories. Another utility `svg2jpg-csiro` will translate an SVG file into a JPEG image.

### DocBook

The latest DocBook SGML DTD and associated DSSSL stylesheets, and the corresponding XML DTD version with its associated XSL stylesheets are made available below the directories `/usr/local/share/docbooksgml`,
`/usr/local/share/docbookdsssl`,
`/usr/local/share/docbookxml`, and
`/usr/local/share/docbookxsl`, respectively.

### TEI

The Text Encoding Initiative (TEI) XML DTD version with its associated XSL stylesheets are available below the directories `/usr/local/share/tei`.

### Educational material

A series of four lectures on XML that I gave in the framework of the Academic Training Program in May 2000 is at the URL `http://home.cern.ch/goossens/xml2000.html`.

General XML-related information is available at
`http://web.cern.ch/xml`.

**About the author(s):** *Michel Goossens is a CERN authority on LaTeX, XML and Electronic Document Publishing techniques in general. He has written several articles and books on the subject.*

---

## 6.2 XML document strategies for the Web

*Michel Goossens, IT/API*

**Abstract**

*The use of XML as a central strategy for managing scientific documents on the Web is investigated. We look at the various ways of handling XML in the Web context in the framework of a general document repository. Finally we introduce the TIPS Project that proposes a new approach to scientific information production and dissemination with XML at the heart of its storage paradigm.*

XML[27], a meta-language introduced at the beginning of 1998 for describing structured data on the Web, builds upon experience gained with SGML and HTML during the last decade. XML was developed with the Web in mind and also guarantees a seamless integration with modern programming languages, such as Java, Perl, and Python. XML is based on Unicode[19] (see [11] for an introduction), so that it is well suited for dealing with multi-language documents, especially those containing lots of non-ASCII characters. Thus, XML is an ideal storage format for a central repository containing data in various source formats, languages, and markup schema.

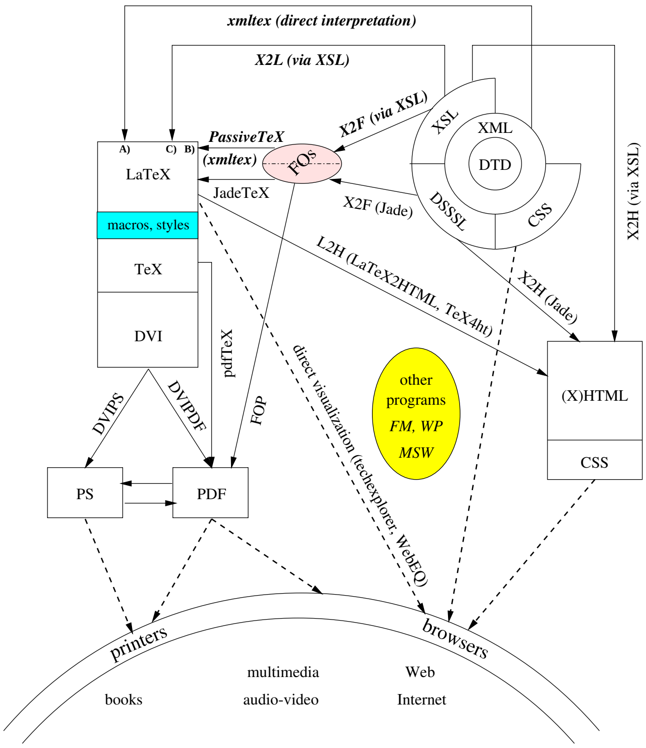
XML is widely supported by all major players in the Internet world, *Open Source* initiatives, as well as commercial vendors. Several free and commercial tools are available for all conceivable operating systems and purposes. In the near future, most Web browsers and visual editors will support XML natively.

### Handling scientific documents

Different ways of using an XML document in a repository are shown schematically in the drawing below.

At the top right we represent the XML document with its defining vocabulary (DTD or XML Schema[24]). This document, which is encoded in Unicode, can be viewed, searched, indexed, edited, validated without problems by any of a series of XML-aware applications all over the world.

The XML document can be typeset using TeX (three methods, labelled A, B, and C, are shown; see [5] and [10]) or its 16-bit Unicode-aware variant Omega[13]. It can also be transformed into HTML for viewing with present-day browsers (X2H via XSL). In the (near) future, once browsers will be able to handle XML directly, we can probably skip the HTML intermediate format and let CSS[21] (possibly via XSLT) style the XML file directly for display on the Web. The figure also contains arrows going from left (TeX) to right (XML/HTML, browsers). They indicate programs to transform existing LaTeX source documents into XML (using one or more standard DTDs) to store the information for archiving purposes. The vertical ellipse in the centre represents other editing tools, such as Adobe's *FrameMaker*[1], Microsoft's *Word*[15], and Corel's *WordPerfect*[7], that allow or are expected to allow import/export of XML documents. Thus, XML genuinely becomes the central element in a global strategy for managing electronic documents by allowing information to be stored, saved, shared, and used by different applications on all computer platforms.




It is not sufficient for XML documents to be available on the Web, but it must also be possible to output them in a typographically optimal way. LaTeX[8] has been one of the pilers of typesetting scientific documents for many years. Therefore, it is important that tools be available to transform an electronic document between XML and LaTeX[9]. PassiveTeX[17] and `xmltex`[4] are two recent developments that ensure that XML sources can be typeset directly or with the help of XSLT [28] stylesheets with TeX. As a added bonus, PassiveTeX supports mathematics marked up in MathML[22] directly, so that an XSLT style sheet can pass MathML's `<math>` source elements and its children through unchanged. This guarantees that mathematical material, involving even complex formulae, will be typeset perfectly. An alternative using DSSSL[14] and Jade[6] is available.

In the other direction, one can translate LaTeX to HTML (or its XML version XHTML[26]) with LaTeX2html[16] or TeX4ht[12]. One can also choose another target language, such as DocBook[20], for computer documentation, or TEI [3], used in the humanities. For both the DocBook and TEI DTDs, XSLT stylesheets exist to transform their source form to HTML or XSL Formatting Objects[29], a generic format that can be translated into PDF by PassiveTeX or FOP[2]

All documents in the XML repository should be accompanied by metadata in the form of RDF[23]. Source files (images, scanned texts or manuscripts, data files) into which it is impossible or impractical to introduce XML markup, should be accompanied by an external XML file containing similar RDF data. This ensures that all documents in the database present a uniform XML interface. This is important for indexing, search, and data mining.

## An example of a scientific document

Part of a source document marked up using the TEI and MathML XML languages follows.

The result of typesetting this document with `xmltex` and PassiveTeX is shown in Figure 6.1. Although MathML is rather verbose, it is not too difficult to recognize the code for the formulae shown, so that it becomes possible to search all parts of a documents, including the mathematical formulae, and, soon the graphics, once SVG (Scalable Vector Graphics, see [25]) will be more widely supported.

## The TIPS Project

Tim Bray said that *XML is the ASCII for the 21st century*. XML allows documents in all major world languages to be viewed and transmitted in a standard reliable way. Many tens of XML applications and vocabularies exist for XML-encoded tree-structured documents and data.

In order to leverage these XML technologies and to benefit from the interoperability and robustness of widely deployed XML solutions of the Web the TIPS Project[18] was initiated.

TIPS (*Tools for Innovative Publishing in Science*) is a European Union funded IST project between five academic institutes and one commercial company from four different countries. TIPS proposes a new approach to scientific information production and dissemination. The aim is to develop a set of user-friendly and advanced tools and services that are organized in an open system to support research information production, management, access, and use in a coherent manner. As an implementation and to make possible the evaluation of the system, these tools and services will be integrated on a web-based portal for the high-energy physics community.

```
<div1 id="vavref">
<head>Vavilov theory</head>

<p>Vavilov<ptr type="bib" target="bib-VAVI"/> derived a more accurate
straggling distribution by introducing the kinematic limit on the
maximum transferable energy in a single collision, rather than using
<inlinemath><math><msub><mi>E</mi><mrow><mtext>max</mtext></mrow></msub>
<mo>=</mo><mi>&infin;</mi></math></inlinemath>.

Now we can write<ptr type="bib" target="bib-SCH1"/>:

<eqnarray ><subeqn><math><mi>f</mi> <mfenced open='(' close=')'>
<mi>&epsi;</mi><mo>,</mo><mi>&delta;</mi><mi>s</mi></mfenced><mo>=</mo>
<mfrac><mrow><mn>1</mn></mrow><mrow><mi>&xi;</mi></mrow></mfrac>
<msub><mi>&phi;</mi><mrow><mi>v</mi></mrow></msub><mfenced open='(' close=')'>
<msub><mi>&lambda;</mi><mrow><mi>v</mi></mrow></msub><mo>,</mo><mi>&kappa;</mi>
<mo>,</mo><msup><mi>&beta;</mi><mrow><mn>2</mn></mrow></msup></mfenced></math>
</subeqn></eqnarray> where <eqnarray><subeqn><math><msub><mi>&phi;</mi>
<mrow><mi>v</mi></mrow></msub> <mfenced open='(' close=')'><msub><mi>&lambda;</mi>
<mrow><mi>v</mi></mrow></msub><mo>,</mo><mi>&kappa;</mi><mo>,</mo>
<msup><mi>&beta;</mi><mrow><mn>2</mn></mrow></msup></mfenced><mo>=</mo>
<mfrac><mrow><mn>1</mn></mrow><mrow><mn>2</mn><mi>&pi;</mi><mi>i</mi></mrow>
</mfrac><msubsup><mo>&int;</mo><mrow><mi>c</mi><mo>+</mo><mi>i</mi><mi>&infin;</mi></mrow>
<mrow><mi>c</mi><mo>-</mo><mi>i</mi><mi>&infin;</mi></mrow></msubsup><mi>&phi;</mi>
<mfenced open='(' close=')'><mi>s</mi></mfenced><msup><mi>e</mi><mrow>
<mi>&lambda;</mi><mi>s</mi></mrow></msup><mi>d</mi><mi>s</mi>
<mspace width='2cm'/><mi>c</mi><mo>&geq;</mo><mn>0</mn></math></subeqn>

<subeqn><math><mi>&phi;</mi><mfenced open='(' close=')'><mi>s</mi></mfenced>
<mo>=</mo><mo>exp</mo><mfenced open='[' close=']'><mi>&kappa;</mi>
<mrow><mo>(</mo><mn>1</mn><mo>+</mo><msup><mi>&beta;</mi>
      <mrow><mn>2</mn></mrow></msup><mi>&gamma;</mi><mo>)</mo></mrow>
</mfenced><mo>exp</mo><mfenced open='[' close=']'><mi>&psi;</mi>
<mfenced open='(' close=')'><mi>s</mi></mfenced></mfenced><mo>,</mo></math></subeqn>
```

## 3 Vavilov theory

Vavilov[5] derived a more accurate straggling distribution by introducing the kinematic limit on the maximum transferable energy in a single collision, rather than using $E_{\max} = \infty$. Now we can write[2]:

$$f\left(\epsilon, \delta s\right) = \frac{1}{\xi}\phi_v\left(\lambda_v, \kappa, \beta^2\right)$$

where

$$\begin{aligned}
\phi_v\left(\lambda_v, \kappa, \beta^2\right) &= \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}\phi\left(s\right)e^{\lambda s}ds \qquad\qquad c \geq 0 \\
\phi\left(s\right) &= \exp\left[\kappa(1+\beta^2\gamma)\right]\ \exp\left[\psi\left(s\right)\right], \\
\psi\left(s\right) &= s\ln\kappa + (s+\beta^2\kappa)\left[\ln(s/\kappa) + E_1(s/\kappa)\right] - \kappa e^{-s/\kappa},
\end{aligned}$$

and

$$\begin{aligned}
E_1(z) &= \int_z^\infty t^{-1}e^{-t}dt \qquad \text{(the exponential integral)} \\
\lambda_v &= \kappa\left[\frac{\epsilon-\bar{\epsilon}}{\xi} - \gamma' - \beta^2\right]
\end{aligned}$$

The Vavilov parameters are simply related to the Landau parameter by $\lambda_L = \lambda_v/\kappa - \ln\kappa$. It can be shown that as $\kappa \to 0$, the distribution of the variable $\lambda_L$ approaches that of Landau. For $\kappa \leq 0.01$ the two distributions are already practically identical. Contrary to what many textbooks report, the Vavilov distribution *does not* approximate the Landau distribution for small $\kappa$, but rather the distribution of $\lambda_L$ defined above tends to the distribution of the true $\lambda$ from the Landau density function. Thus the routine `GVAVIV` samples the variable $\lambda_L$ rather than $\lambda_v$. For $\kappa \geq 10$ the Vavilov distribution tends to a Gaussian distribution (see next section).

Figure 6.1: XML source (top) and formatted output by LaTeX (bottom)

The proposed system will support the activities of document writing, reviewing, publishing, searching, disseminating and reading, as well as the communication among members of the research community. This approach is suitable for supporting a more productive research community, in which researchers can work in a more effective, inexpensive, and pleasant way: delays and costs due to paper documents can be considerably reduced, multimedia can be added to electronic documents, information access can be improved (and information overload decreased) by using advanced information retrieval and filtering techniques. The full possibilities of XML technologies will be fully exploited wherever possible.

## Bibliography


[1] Adobe. *FrameMaker 6.0.* `http://www.adobe.com/products/framemaker`.

[2] Apache XML Project. *FOP, XSL Formatting Object Processor in Java.* `http://xml.apache.org/fop/`.

[3] Lou Burnard and C.M. Sperberg-McQueen. *TEI Guidelines for Electronic Text Encoding and Interchange.* `http://etext.lib.virginia.edu/TEI.html`.

[4] David Carlisle. *xmltex A non validating (and not 100% conforming) namespace aware XML parser implemented in TEX*. Available on CTAN in the directory `macros/xmltex/`.

[5] David Carlisle, Michel Goossens and Sebastian Rahtz. *De XML à PDF avec xmltex, XSLT et PassiveTEX.* `http://www.gutenberg.eu.org/pub/GUTenberg/publicationsPDF/35-carlisle.pdf`

[6] James Clark. *DSSSL implementation.* `http://www.jclark.com/jade`.

[7] Corel. *WordPerfect Office 2000.* `http://www.corel.com/Office2000` (Microsoft Windows) and `http://linux.corel.com/products/wpo2000_linux` (Linux).

[8] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LATEX Companion.*. Addison-Wesley, Reading, 1994.

[9] Michel Goossens and Sebastian Rahtz. *The LATEX Web Companion.* Addison-Wesley, Reading, 1999.

[10] Michel Goossens and Sebastian Rahtz. *PassiveTEX: XSL processing using TEX.* Presented at TUG2000 in August 2000 at Oxford, U.K. (to be published in the Proceedings).

[11] Tony Graham. *Unicode: A Primer.* M and T Books (IDG), Foster City, 2000.

[12] Eitan Gurari. *TEX4ht: LATEX and TEX for Hypertext.* `http://www.tug.org/applications/tex4ht/mn.html`.

[13] Yannis Haralambous and John Plaice. The latest developments in Omega. *TUGBoat*, 17 (2), pages 181-183, June 1996. (See also `http://www.gutenberg.eu.org/omega/`).

[14] International Organization for Standardization. *Information Technology – Processing Languages – Document Style Semantics and Specification Language (DSSSL). First edition, 1996.* International Standard ISO/IEC 10179:1996, ISO Geneva, 1996.

[15] Microsoft. *Office Tools, Word.* `http://www.microsoft.com/office/word`.

[16] Ross Moore et al. *LATEX2HTML.* `http://saftsack.fs.uni-bayreuth.de/~latex2ht/`.

[17] Sebastian Rahtz. *PassiveTEX.* `http://users.ox.ac.uk/~rahtz/passivetex/`.

[18] Sissa, Udine, City University, Grenoble University, CERN, IOP. *Tools for Innovative Publishing in Science (TIPS).* `http://tips.sissa.it/`

[19] The Unicode Consortium. *The Unicode Standard, Version 3.0.* Addison-Wesley, Reading, 2000. See also `http://www.unicode.org`.

[20] Norman Walsh and Leonard Muelner. *Docbook. The Definitive Guide.* O'Reilly and Associates, Inc., Sebastopol, USA, 1999. `http://nwalsh.com/docbook/index.html`. The URL `http://www.oasis-open.org/docbook/documentation/reference/index.html` gives access to the reference documentation, while DTDs and stylesheets are at `http://nwalsh.com/docbook/index.html`.

[21] World Wide Web Consortium. Håkon Wium Lie, Bert Bos, Chris Lilley and Ian Jacobs (editors). *Cascading Style Sheets, level 2.* `http://www.w3.org/TR/REC-CSS2`.

[22] World Wide Web Consortium. Patrick Ion and Robert Miner (editors). *Mathematical Markup Language (MathML[tm]) 1.01 Specification.* `http://www.w3.org/TR/REC-MathML/`.

[23] World Wide Web Consortium. *Resource Description framework.* `http://www.w3.org/RDF/`.

[24] World Wide Web Consortium, David C. Fallside (editor). *XML Schema Part 0: Primer (W3C Working Draft).* `http://www.w3.org/TR/xmlschema-0`.
World Wide Web Consortium, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn (editors). *XML Schema Part 1: Structures (W3C Working Draft).* `http://www.w3.org/TR/xmlschema-1`.
World Wide Web Consortium, Paul V. Biron, Ashok Malhotra (editors). *XML Schema Part 2: Datatypes (W3C Working Draft).* `http://www.w3.org/TR/xmlschema-2`.

[25] World Wide Web Consortium, Jon Ferraiolo (editor). *Scalable Vector Graphics (SVG) 1.0 Specification (W3C Working Draft).* `http://www.w3.org/TR/SVG`.

[26] World Wide Web Consortium. *XHTML 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0.* `http://www.w3.org/TR/xhtml1/`.

[27] World Wide Web Consortium. Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen (editors). *Extensible Markup Language (XML) 1.0.* `http://www.w3.org/TR/REC-xml`. An annotated version of the specification is at `http://www.xml.com/axml/axml.html`.

[28] World Wide Web Consortium, James Clark (editor). *XSL Transformations (XSLT), Version 1.0 (W3C Recommendation 16 November 1999).* `http://www.w3.org/TR/xslt`.

[29] World Wide Web Consortium, Stephen Deach (editor). *Extensible Stylesheet Language (XSL), Version 1.0 (W3C Working Draft).* `http://www.w3.org/TR/WD-xsl`.

# 7. The Learning Zone

*All kinds of "educational" computing issues, not necessarily directly CERN-related, such as computing news from the Press and tutorials on various subjects.*

## 7.1 Computing at CERN, Once Upon a Time...

*Miguel Marquina (Editor), IT/User Support*

**Abstract**

*It may be interesting to look backwards into what we published in this CNL several years ago. This starts a series which will try to offer a retrospective of how Computing has been evolving, in particular at CERN, over the past 35 years. I am certain this contribution will evoke memories in more than one!*

**Five years ago...**

*Downsizing and Closure of the CERNVM Service*

(CNL issue 221 "Migration Special" - July-December 1995) The CERN Management Board has now approved the time scale for the end of the CERNVM service. The computing capacity will be reduced by half during the Christmas shutdown at the end of 1995 and the service will be terminated on 30 June 1996. From January a clear warning message of the termination date will be displayed when a user logs in.

> *I could not resist this one. CERNVM marked a before and after in the way Computing infrastructure was deployed at CERN (Ed.)*

**Ten years ago...**

*Introduction of TMS, the Tape Management System*

(CNL issue 200 - July-October 1990) During the summer a Tape Management System, known simply as TMS, went into production on the IBM and Cray systems. TMS was written at the Rutherford Appleton Laboratory in the UK and is also in use at IN2P3 in Lyon.

. . .

The last phase of the TMS development is to provide facilities in the tape vault to allow convenient entering and exiting of tapes in the Experimental Group racks.

> *The title of the article might have hinted TMS would have been the "final answer" to tape management. By today standards the tape vault is becoming symbolic. Tomorrow? (Ed.)*

**Twenty years ago...**

*Guest Article on Enhancements to Fortran*

(CNL issue 153 - August 1980)

Following its heroic action in making blanks significant to aid *syntactic interpretation*, the ANSI Fortran Committee JX3.3 is pioneering an approach to aid *semantic* interpretation. As a first step, an ASSERT statement has been proposed, which can aid the compiler in making decisions during optimisation. Such an assertion takes precedence over the actual code in the event of a conflict, because it represents what the programmer really wants.

> *Those heroic days when coding standards guessed what the programmer really wanted... (Ed.)*

**More than thirty years ago...**

*CDC 6000 Series News*

(CNL issue 53 - 12th August 1970)

... A considerable amount of time was lost on printers due to problems with the ribbon and paper sensing mechanisms. Additional attention is now being given to this rather sensitive elements.

> *Hum, it seems printers have always been the favourite torture seat of our users! (Ed.)*

*3800 and 6400*

(CNL issue 8 - 25th July 1966)

Of all problems considered the installation and operation of the 3800 is the most urgent one. During August the computer will be equipped with 64K of core memory, eight magnetic tape units and one on-line printer, card reader and punch. Estimates of the configurations group indicate that this configuration can carry out the present work of the 3400 (150 hours/week) plus the TC production on a 3600 in Paris (75 hours/month) in about 2.5 shifts.

> *64K? A long way down, and still the ancillary stones for LHC Computing (Ed.)*

## 7.2 X11 Desktop Environments (KDE and GNOME - Part III)

*Victor Robles, Germán Cancio, IT/PDP*

**Abstract**

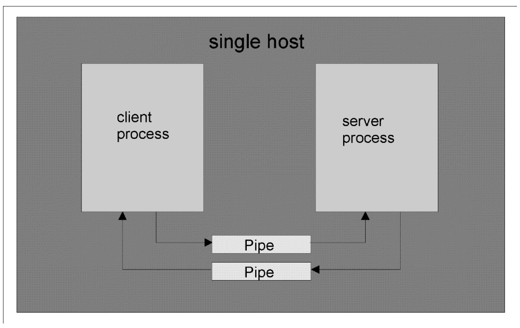
*This is the last in a series of three articles comparing the two emerging desktop standards, KDE and GNOME. Three main subjects will be addressed: interprocess communications and CORBA, API descriptions, and installing the desktops at CERN.*

### Interprocess communications and CORBA

In the last article, we described what information exchange protocols are available for GNOME and KDE, like drag and drop, ICCCM, etc. We are going to have a look now at the means by hwich this information exchange is carried out. In the UNIX world, interprocess communication (IPC) has been traditionally done using *pipes*, sockets and RPCs; CORBA has made its appearance over the last years.

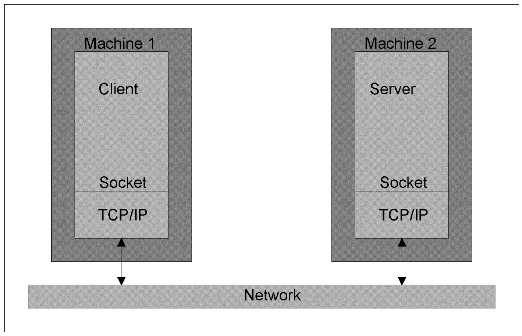
#### Pipes

UNIX offers the possibility of using pipes to share data between processes. A well-known UNIX command sequence like 'ls -l | more', concatenates two processes ('ls' and 'more'), conducting the output of the first process to the input of the second one. Nevertheless, the functionality of a pipe is limited. They are unidirectional, which means that a process can write into a pipe and another one can read out of it, but for a bidirectional communication, two pipes have to be created. It is a very low level protocol, data being read or written word by word. It two applications wish to interchange complex data, they have to agree on a common data structure and format.



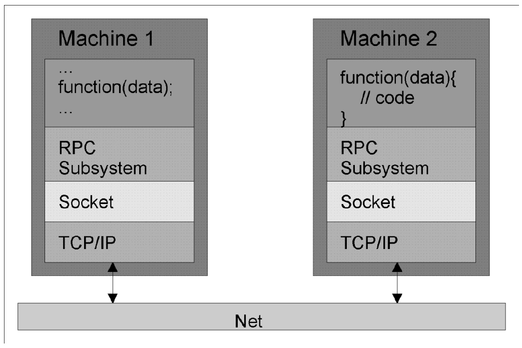

#### Sockets

Like pipes, sockets are low level mechanisms and word oriented. Sockets allow you by using a network protocol (as TCP/IP or IPX) to transparently communicate processes running on different machines. The main difference with pipes is that sockets allow bidirectional communication. Nevertheless, data structure must be defined at application level, as with pipes. Sockets are nowadays a must when using standard internet protocols like FTP, HTTP, etc.



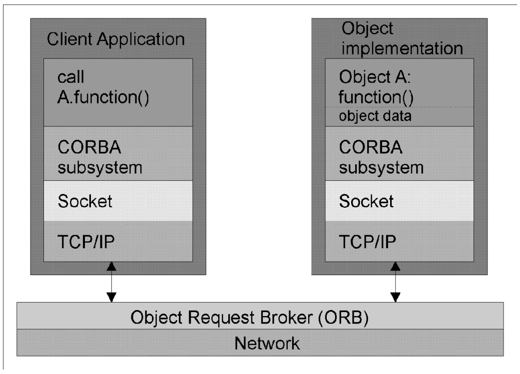

#### RPC (Remote Procedure Calls)

One of the major problems with pipes and sockets is the lack of a standard for interchanging data structures and for remote function activation. Applications should shouldn't have to take care about how, but about what data structures are transmitted. RPC does provide this encapsulation. RPC was initially developed by Sun for the NFS file system. The use of RPC allows calling remote functions as if they were local; programmers don't have to worry about opening connections or converting data, the RPC system takes care of this. If we want to communicate two applications using RPC, we first have to define the functions and data types to be interchanged by client and server using for this a special language.

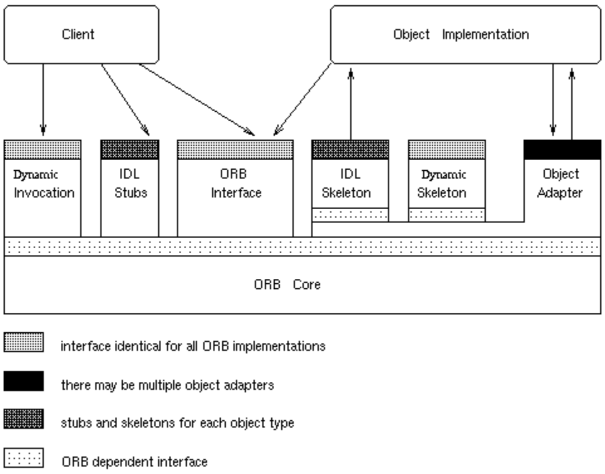



#### CORBA

As object oriented technologies became popular, they were also ported to the IPC world. CORBA, which stands for Common Object Request Broker Architecture, is an IPC standard based on the OO paradigm. In fact, this approach is very similar to the use of RPCs, because like in this, we call a function with certain parameters. Unlike RPC, remote objects may be created which encapsulate their own data, those objects are manipulated via method calls. The object's data resides also remotely and doesn't have to be send over the network, saving network bandwidth and avoiding type conversion overheads. We can think of CORBA like a next generation RPC, just extended to support object-oriented programming.




The CORBA architecture is depicted on the following image. The most important component is the ORB (Object Request Broker), which is responsible for all communications for identifying and locating objects, data transmission and for clients calling the object implementations. The object interface is defined by using a special language called IDL (Interface Definition Language). IDL is used for declaring the object interface, not its implementation. It constitutes the contract between client and server. The IDL is then used for generating the *stubs* for implementing the client and the server *skeleton*. The implementation language of client and server may differ and depends on the ORB in use.




A client communicates with the object by using the stub which links it with the ORB. The communication between the object and the ORB is done via an entity called the Object Adapter (OA). The OA essentially handles activating and deactivating object and method invocations.

Clients and objects may also communicate dynamically on runtime without having to use stubs. The Dynamic Invocation Interface (DII) allows a client to issue requests to objects whose interface was unknown on compilation time. An interface repository is used to store object interfaces which can be looked up when required. A Dynamic Skeleton Interface is also available in case the OA doesn't know an object in compilation time.

More information on CORBA may be found at the URL `http://cgi.omg.org/corba/beginners.html`.

**MICO and ORBIT**

The main component of CORBA is the ORB as we just have seen. There are many free ORB implementations, but most of them are not powerful enough or haven't reached yet maturity. For KDE v1, MICO (`http://www.mico.org`) was chosen as the ORB. The GNOME developers started using MICO, but due to efficiency reasons they decided to write their own ORB, which they call ORBit. MICO has been implemented in C++, while as ORBit was written using C. ORBit claims to be one of the fastest ORB implementations.

Note: recently, version 2 of KDE has been announced, in which the usage of CORBA has been discarded for efficiency reasons. The replacement for CORBA and MICO is called DCOP (Desktop COmmunications Protocol) and is based on X11's ICE. More information on DCOP may be found at the URL `http://developer.kde.org/documentation/library/2.0-api/dcop/HOWTO.html`.

The rest of this article is going to refer to KDE v1.

## CORBA on the desktops

After explaining what CORBA is we are going to analyse how it is used on the desktops. We can define five different uses:

**General IPC mechanism**

CORBA presents a simple and a elegant solution to interprocess communications. For example, the GNOME panel uses CORBA to communicate with panel applets. In addition, some games also use CORBA as communication protocol to play in a network environment.

**Exporting an application's API**

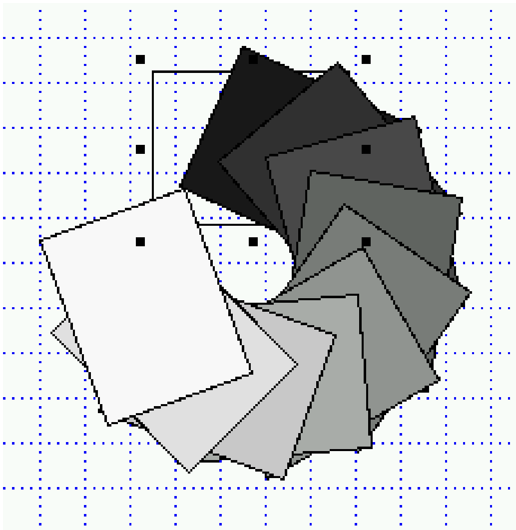
CORBA is used to export the internal motor of the applications to the rest of the system through IDL files. Any CORBA client will be able to use the services provided by this interface. In KDE all KOffice applications export their interface, on the other hand in GNOME we also have some applications doing it, for example, the gmc file manager and the gnumeric spreadsheet. Exporting application APIs is vital if we need a component-based system. This way, for example, an application could use gmc to look for and to select a file within a directory structure.

**Scripting**

In Linux we must often make very common repetitive tasks. Scripting languages, like Perl or Python, allow to automate them. Some KDE and GNOME programs have scripting interfaces, for example killustrator in KDE and gnumeric in GNOME. For example, suppose that every day we need to update a web page with a statistics graph. We could every day read the data from a text file, put it into gnumeric, create a graph using guppi and export the graph to JPG format for the web page. We could make everything through a small program written in Perl. Another example is scripting using KIllustrator. With these few lines:

```
doc = Document ("", ior)
              if doc.checkForSelection ():
              sel = doc.getSelection ()
              x, y = sel[0].getCenter ()
              for i in range(10):
              obj = sel[0].duplicateObject ()
              j = (i + 1) * 25
              obj.rotateAround (j, x, y + 50)
              obj.applyFillColor (j, j, j)
              obj.applyPenColor (0, 0, 0)
```

We will get something like this:



### Define system services

Within GNOME a standard set of interfaces for system services is defined. Services could be created for common tasks like introducing a new user in the machine or sending a mail. Thus, for example, if we want to make a mail client, we do not need to develop all the code, just make a call to the required interface. This mechanism can cause that future applications will be easier to program and to support.

### Compound documents and graph components

KOM/OpenParts in KDE and Bonobo in GNOME are the object models that allow the construction of compound documents and the creation of reusable graphical components. A compound document is a document that is formed by parts of diverse origin. For example, in koffice, a kword document can have a table from kspread or a formula from kformula. Instead of creating an application that implements all the necessary characteristics, we need that this application can interact with other applications or components. For example, instead of adding support to a text editor to publish formulas, we could use a component available elsewhere. KOM/OpenParts and Bonobo are inspired by Microsofts OLE2 (the Windows compound document model).

## Developer's Corner

A comparison between KDE and GNOME without writing about desktop programming would be incomplete. Without pretending to write a tutorial, we quickly show that application design is simple and that you can get good results quickly and easily.

### Desktop Programming

A KDE program is a graphic application builded for Qt and making use of KDE libraries. KDE libraries make possible to have a standard look and feel between all the applications and make the programming easier to developers. Besides, KDE libraries add more widgets to Qt Toolkit. The same definition above is valid for GNOME just replacing Qt with GTK+.

## KDE and GNOME applications.

The best way to know how is desktop programming is to see an example. So we are going to analyze a KDE and a GNOME program.

### KDE application

```
/****************** khello.h ******************/
#include <kapp.h>
#include <ktmainwindow.h>

class KHello : public KTMainWindow
{
   Q_OBJECT

   public:
     KHello (void);

   public slots:
     void slotHello();
     void slotClose();
};

/****************** khello.cc ******************/
#include "khello.moc"
#include <kmsgbox.h>
#include <qpushbutton.h>

KHello::KHello (void)
{
   QPushButton *hello;
   QPushButton *close;

   hello = new QPushButton( "Hello", this );
   hello->setGeometry(10,30,85,30);
   hello->show();
   connect(hello, SIGNAL(clicked()),
           this, SLOT(slotHello()));

   close = new QPushButton( "Close", this );
   close-> settGeometry(10,30,85,30);
   close->show();
   connect(close, SIGNAL(clicked()),
           this, SLOT(slotClose()));
}

void KHello::slotHello()
{
   KMsgBox::message
```

```
        (this, "Important", "Hello KDE!!!");
}

void KHello::slotClose()
{
   kapp->quit();
}


/******************  main.cc  ******************/
#include "khello.h"

int main (int argc, char **argv)
{
        KApplication *app;
        KHello        *hello;

        app=new KApplication
                  (argc, argv, "Hello KDE");
        KHello hello;
        hello.setGeometry(100,100,200,100);
        app->setMainWidget( hello );
        hello.show();
        app->exec();
}
```

The previous listing shows a basic KDE program. This program is developed in C++, like KDE, using object orientation.

In this KDE program we create a KHello class that inherits from KTMainWindow, the basic class of all KDE applications. The program is split in three parts, `khello.h` being the specification class, `khello.cc` is the implementation class and `main.cc` is the main program that calls the created class.

Inside the class KHello we must use `Q_OBJECT` macro which give support for the "signals and slots" object communication mechanism. If we make use of the `Q_OBJECT` macro we must use a precompiler named *moc*. Inside the class implementation we must use the moc output file. Once we have created the KHello class we can make use of it as we want.

The KHello class only has a method, its constructor. In our example this constructor just adds two buttons: Hello and Close. Each button has an associated slot, that is to say, a method. The Hello button slot displays a screen message and the Close button slot end the application.

The main part of the program just creates a KHello class, resizes it and executes it.

As we can appreciate, the construction of KDE programs is not complicated, and the object oriented programming helps to have a readable code. In the developers KDE web page you can find an extensive documentation about KDE classes and how to use them.

### GNOME application

We are going now to build a similar application with GNOME, using the GTK toolkit and C.

```
#include <gnome.h>

static void buttonHello(GtkWidget *button)
{
  GtkWidget *msg;
  msg= gnome_message_box_new
   ("Hello GNOME!!!",GNOME_MESSAGE_BOX_GENERIC,NULL);
  gtk_widget_show(msg);
}


int main(int argc, char **argv)
{
   GtkWidget *app;
   GtkWidget *hello;
   GtkWidget *close;
   GtkWidget *hbox;

   gnome_init ("GNOME example", "1.0", argc, argv);

   app = gnome_app_new ("GNOME example","Hello GNOME");

   /* When the program is closed */
   gtk_signal_connect
           (GTK_OBJECT (app), "delete_event",
            GTK_SIGNAL_FUNC (gtk_main_quit), NULL);

   hbox = gtk_hbox_new (FALSE,5);
   gnome_app_set_contents
            (GNOME_APP (app), hbox);

   hello = gtk_button_new_with_label("Hello");
   gtk_box_pack_start
            (GTK_BOX(hbox), hello, FALSE, FALSE, 0);
   gtk_signal_connect
            (GTK_OBJECT (hello), "clicked",
             GTK_SIGNAL_FUNC (buttonHello),NULL);

   close = gtk_button_new_with_label("Close");
   gtk_box_pack_start
            (GTK_BOX(hbox), close, FALSE, FALSE, 0);
   gtk_signal_connect
            (GTK_OBJECT (close), "clicked",
             GTK_SIGNAL_FUNC (gtk_main_quit),NULL);

   gtk_widget_show_all(app);
   /* GTK main loop */
   gtk_main ();

   return 0;
}
```

The first difference that can be observed in the program is that we do not create a class, everything is made in the main program (as C is not object oriented).

We must initialise all GNOME programs with the gnome_init function. The basic widget for all GNOME applications is gnomeapp, which is initialised calling the function gnome_app_new. We create our main window in which we will be able to introduce menus, a toolbar and a status bar. The program inserts in the main window two buttons, Hello and Close. The button hello is connected with its corresponding slot that shows a screen message, while Close button finalises the execution.

### Comparison

As we can see, the desktop programming between KDE and GNOME presents a lot of similarities. From the beginning we must have a basic window and them insert in it all the different objects we need. In both toolkits, Qt and GTK+, communication between different objects can be made using the “signals and slots” mechanism.

### Adding the application menu and others

Once we have constructed the basic skeleton of our application it is very simple to create menus.

### KDE menu

```
popup = new QPopupMenu();
popup->insertItem("&Hello", this, SLOT(slotHello()));
popup->insertItem("E&xit", this, SLOT(slotQuit()));
help=kapp->getHelpMenu(TRUE, "Hi World!");
menu = new KMenuBar(this);
menu->insertItem("&File", popup);
menu->insertItem("&Help", help);
```

In the KDE application, we are creating a menu with two different parts, File and Help. Inside the File menu we have two options, Hello and Exit, that are communicated with their respective slots, `slotHello` and `slotQuit`.

### GNOME menu

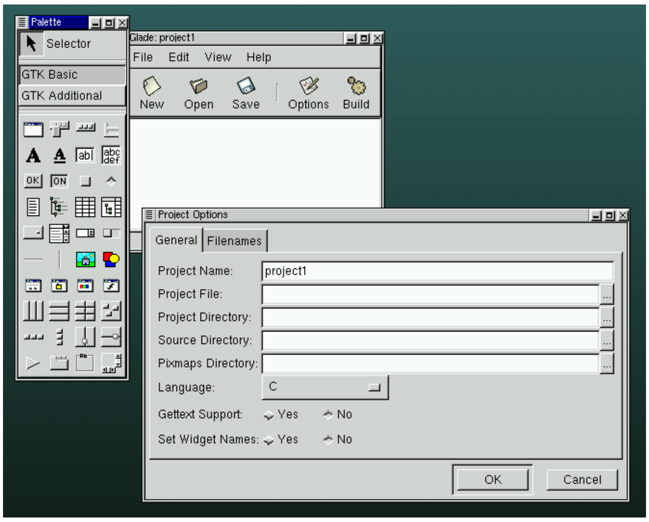
```
GnomeUIInfo popup[] = {
GNOMEUIINFO_ITEM_NONE("_Hello", "", Hello),
GNOMEUIINFO_ITEM_NONE("E_xit", "Quit", Quit),
GNOMEUIINFO_END
};

GnomeUIInfo help[] = {
  GNOMEUIINFO_ITEM_NONE("_Help", "Get Help", Help),
  GNOMEUIINGO_END
};

GnomeUIInfo menubar[] = {
  GNOMEUIINFO_SUBTREE("_File",popup),
  GNOMEUIINFO_SUBTREE("_Help",help),
  GNOMEUIINFO_END
};

gnome_app_create_menus( GNOME_APP(app), menubar);
```

GNOME also provides simple mechanisms for menu creation. In GNOME we have two different ways to create menus, one is directly, in a similar way we made before for KDE, but really more difficult. The other way is filling up a structure and calling to gnome_app_create_menus function (see above).

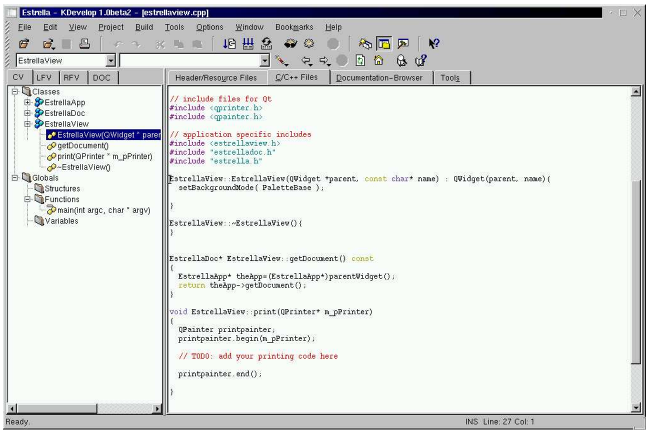
Creation of toolbar, status bars, support for Drag and Drop protocol, Session Management protocol are aspects quite easy to program.

Considering the great amount of facilities that are provided by the desktops is easy to understand the quick evolution of front-ends for the most important X applications. For example we have gtkdiff, an front-end for the well-known diff application, Word Inspector, a GNOME front-end for the dict program, Kgdb, a KDE front-end for the debugging program, etc.

Now we start to have very good RAD applications (RAD= Rapid Application Design). With this development tools build graphic user interfaces is really easy. In GNOME we have glade, in KDE we have KDevelop. But they are different, KDevelop is really a complete IDE (Integrated Design Environment), while glade is only a graphic interface builder.




Glade: GNOME interface builder




KDevelop: the KDE IDE

## KDE programming APIs

But we need more that the programming examples we have seen before. If we want to have a complete idea about the power of KDE libraries we must speak about the KDE programming API.

The *kdecore* library contains several classes that are helpful for creating applications. The main classes of this library are:

- `KAccel`: it handles the hotkeys.
- `KApplication`: controls and provides information to all KDE applications.
- `KConfig` and `KSimpleConfig`: Allow to save and load the application configuration.
- `KDNDDropZone`: Provides support for Drag inside the Drag and Drop Protocol.

- `KDNDWidget`: Provides support for Drop inside the Drag and Drop Protocol.
- `KLocale`: Supports translations and specific formats like dates, numbers, currency and time.
- `KProcess`: Allows KDE applications to execute child processes in a simple and powerful form. In addition it provides support for communications with the child processes.
- `KSocket`: Simple way to handle TCP/IP sockets.
- `KURL`: Handle all type of URL.

The *kdeui library* gives support to most of the graphical elements inside the applications. Some elements are:

- `KToolBar`, `KMenuBar` and `KStatusBar` provide an easy and flexible way for writing KDE applications.
- Some standard dialogues like for example `KColorDialog` to select a colour, `KIconLoader` to select an icon, `KFontDialog` to select a letter type.
- Some extra widgets like `KPanner` to create two redimensionable windows or `KTreeList` to administer tree structures.

There are also some other important *extra libraries*:

- `kfm`: allows any application to download internet files by just providing its URL, completely transparent to the user.
- `kfile`: Supports all the different dialogues for file selection.
- `khtml`: Library that implements all the necessary widgets for HTML editing.
- `kab`: Defines the API to access user data bases created with KDE address Book.

**GNOME programming APIs**

`libgnome` is the utilities library of GNOME. It does not depend on the GTK+ toolkit, so any application may use this library. The main classes are:

- `gnome-config`: easy way to store configuration files.
- `gnome-exec`: program execution from GNOME applications.
- `gnome-util`: diverse utilities, for GNOME and C programming.
- `gnome-mime`: routines for find the MIME format of a given file.
- `gnome-metadata`: allows to store information with the file name.
- `gnome-i18n`: configuration options for internationalisation and localisation.
- `gnome-regexp`: processing of regular expressions.
- `gnome-popt`: parser for command line arguments.
- `gnome-sound`: routines for sound handling.
- `gnome-fileconvert`: routines for file format conversion.

`libgnomeui` is the most used library in the programs, it includes the basic elements of user interface and other aspects relative to GTK+ and X as Session Management and a great amount of extra widgets. The main classes are:

- `gnome-app`: Creates the main window and allows add easily menu bar, state bar and tool bar.
- `gnome-app-helper`: Simplifies the menu and tool bar creation.
- `gnome-stock`: Provides to applications defect icons, such a way that you can keep a standard look and feel.
- `gnome-dialog`: Allows to create our own dialogues.
- `gnome-*entry`: Allows text entry.
- `gnome-pixmap`: widget to visualise and to load images.
- `gnome-client`: Routines for handling GNOME Session Management.
- `gnome-mdi`: Allows applications to handle multiple documents simultaneously.
- `gnome-canvas`: A motor to represent structured graphs.
- `gnome-dns`: Interface for name resolution.

Other additional libraries:

- `libgnorba`: Routines relatives to the basic use of CORBA in GNOME applications and GOAD (GNOME Object Adaptor Directory), that keeps all object implementations in the system and allows to activate them.
- `libart`: handles all GNOME drawing characteristics.

Support for Drag and Drop and file selection dialogs are supported by GTK+, not by GNOME libraries.

**How to install KDE and GNOME**

Currently neither KDE nor GNOME are officially supported at CERN. GNOME is available on ASIS (`http://wwwinfo.cern.ch/pdp/ose/asis`) for Linux (RedHat 6.1), Solaris 2.6/7 and partially for other platforms like HP, AIX, SGI and Alpha.

**GNOME**

As administrator, launch `tkwsm` by running `/afs/cern.ch/asis/adm/bin/asis-wrapper tkwsm`. Open the configuration menu and select:

`'Family=GNU.DESK, Product=<All>, State=Latest, Access=Default'`.

Then press the `Add` button, and finish by clicking on the `Save` button. In the main window click on the `Execution` button. A Window named ASISwsm will be open. Click on the `Run now for all` button (this may take a few minutes, depending on your workstation).

For launching a GNOME session, you have to run `/usr/local/bin/gnome-session` instead of your window manager. You can do that by either creating an executable `.xsession`file in your home directory:

```
#!/bin/sh
/usr/local/bin/gnome-session
```

Or also by changing your HEPiX configuration, by editing the `~/.hepix/xprofile` file to include this line:

```
HX_WM="/usr/local/bin/gnome-session"
```

**KDE** Currently on ASIS, there is no updated KDE version available. On RedHat Linux systems the pre-packaged RPMs are available on AFS.

For RH61, the packages may be found under `/afs/cern.ch/project/linux/redhat/cern/6.1/RedHat/RPMS`.

For the base installation you will need to install:

```
qt1x-1.44-4
kdesupport-1.1.2-3
kdelibs-1.1.2-9
kdebase-1.1.2-10
kdebase-lowcolor-icons-1.1.2-10
```

With this RPMs, the base functionality is available. Optional applications and components are found in:

```
kdeadmin-1.1.2-3
kdegraphics-1.1.2-2
kdemultimedia-1.1.2-3
kdenetwork-1.1.2-4
kdegames-1.1.2-2
kdetoys-1.1.2-2
```

As with GNOME, you may either create a `.xsession` file or modifying your HEPiX configuration for starting up KDE: `~/.xsession`

```
#!/bin/sh
/usr/bin/startkde
```

Or adding to `~/.hepix/xprofile`

```
HX_WM="/usr/bin/startkde"
```

**Conclusion**

In these articles we have tried to provide a general overview on both desktop systems, and also to have a look behind the scenes regarding the used technology. KDE and GNOME do share many common points but do also differ on several aspects, as we have seen. We are not going to make any recommendation yet on which system to use but want to invite you, the users, to install the desktops and work with them and provide us with your experience. We are also looking for KDE maintainers who could compile and configure the KDE environment for Linux and other platforms. Any help and feedback is welcome, please contact `Asis.Support@cern.ch` or directly Germán Cancio (`German.Cancio@cern.ch`).

More information is available at the URL

`http://wwwinfo.cern.ch/pdp/ose/gnome_kde/`.

---

## 7.3 Developing with Plugins and Components

*Paula Valença, IT/PDP*

**Abstract**

*Nowadays, several projects have become so complex and large that they are impossible to develop and maintain as a single monolithic block. On the other hand, the ability for an application to acquire as much functionality as possible, reusing software and without accumulating functionality not needed at that moment and, thus, not wasting resources, is also becoming an important problem.*

*These and other problems caused some new strategies to develop, in particular, the use of* plugins *and* components *which, using some old principles of modularity and implementation facilities like* shared libraries *and* middleware *(e.g., CORBA), allows implementors to overcome these problems in a clean and standard way.*

*This article presents an overview over this technology.*

### Why plugins and components?

One of the biggest problems developers have to handle nowadays is on how to cope with the enormous size and complexity of an increasing number of current projects. It is common knowledge that from a certain point it is impossible to have the application as a monolithic block. It is necessary to divide the project in different parts, as independent as possible, so that designing the different parts to work together is as easy and consistent as possible. This tends to be difficult since different teams are only familiar with their parts and the inter-communication is usually low. Also, it is very common nowadays to have a project spread across different platforms and locations.

On the other hand, it is also usually said that if a certain functionality is needed and it is already available somewhere, one should reuse this implementation and not "reinvent the wheel". This should account also for using/developing products aside especially for a certain task instead of integrating that functionality in the main application. Besides the reusability, this also brings more flexibility and a simpler and clearer main application. For instance, consider an application which captures, evaluates and stores data where in the evaluation stage, some graphics need to be generated. There are a big number of applications that draw graphics and are probably preferred than to add "from scratch" graphics functionality to the application. But the main problem is whether the graphic program is easily integrated which, most of the time, is not the case.

Continuing on the subject of adding/changing functionality and considering the same example, imagine that the data it handles is very heterogeneous and, thus, the way this data is evaluated, for example, is different according to the type of data. When the set of the types handled is small and doesn't change and the way they are handled is also static, the final product is a bit complex but handleable. But when this is not the case several problems appear:

- *Flexibility and consistency*. How to add/change/remove functionality in a flexible manner without compromising the consistency of the application and, preferably, not having to rebuild all the application?
- *Efficiency*. How to do this in a way that doesn't compromise too much the efficiency, e.g., there isn't a waste of resources since most will only be used for a certain period or not even used at all in a session?

Recently, two paradigms have evolved that are especially aimed at this type of problems: *Plugins* and *Components*. These enable us to fulfil the following compilation of objectives in a clever and transparent way. It is easily seen why these two paradigms have become so popular.

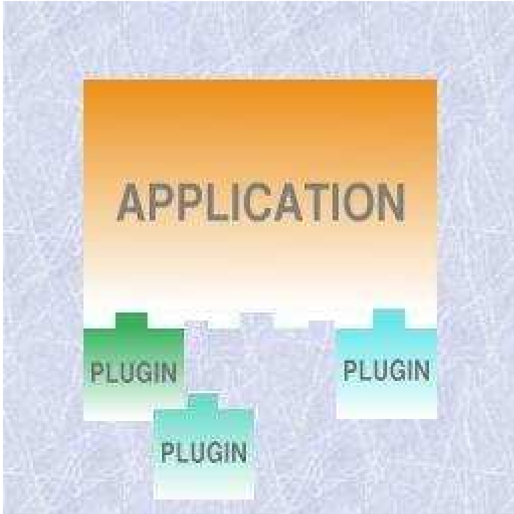
***Modularity*** Modularity is one of the keywords. How to divide an utility in different parts that obey the rules of modularity (independence, encapsulation) and interact with each other in a desired way. A special case would be to divide an utility in its main part (management) and its several branches, thus, separating functionality.

***Dynamic functionality independent of the main part*** In other words, to be able to add/change/remove functionality of an utility without touching the main part. As an add-on, to be able to do that at runtime. This implies that the application shouldn't be rebuilt and, further more, compiled, each time these changes take place.

***Independent development of parts*** The development of each part of a project should be independent from the other ones. This obliges the communication between parts to be transparent and standard. Also, each part should be able to cope with the others being in different platforms and locations. Basically, each part shouldn't have to worry about how others are implemented and located.

***Communication between applications*** This means being able to embed an application A that provides a desired functionality inside an application B thus reusing code and providing a lighter and simpler application B.

## On plugins




An illustration of the plugin-concept: an independent application where units of software, intended to complement the functionality, can be "plugged" to.

A plugin is a unit of software intended to extend the functionality of an independent application at run-time (see *Glossary*). It provides several advantages, making this technology very popular among several programs that process heterogeneous data and would, otherwise, be less flexible and heavier (one of the most famous examples being Netscape that uses plugins to handle and/or view special data like audio or movies):

- The functionality is added/changed/removed at runtime so that the application doesn't have to be rebuilt and, furthermore, not even recompiled.
- The application doesn't know the plugin beforehand (it is the plugin's responsibility to present itself) providing wider flexibility to what functionality can be added.
- It leads to a complete, clean separation from the main application and the various branches with their separate functionality. Thus we get a lighter main application focused on the essentials.
- It is usually very easy for an user to add a plugin to the application.

Plugins are usually implemented as demand-loaded shared libraries. These type of libraries are only loaded at runtime as opposed to static whose code is included in the application at the compilation stage. They can be loaded in two different ways:

- automatically at the start of the process and, thus, only unloaded at the end: the library must be linked at compilation stage;
- by request of the program itself (using, for example, `dlopen()`): the code can be loaded and unloaded in the middle of the process or not even loaded at all; furthermore, the library is not linked at compilation stage.

When the library is loaded, the system checks if it is already in memory, so that, at most, one copy of the library is present in the memory.

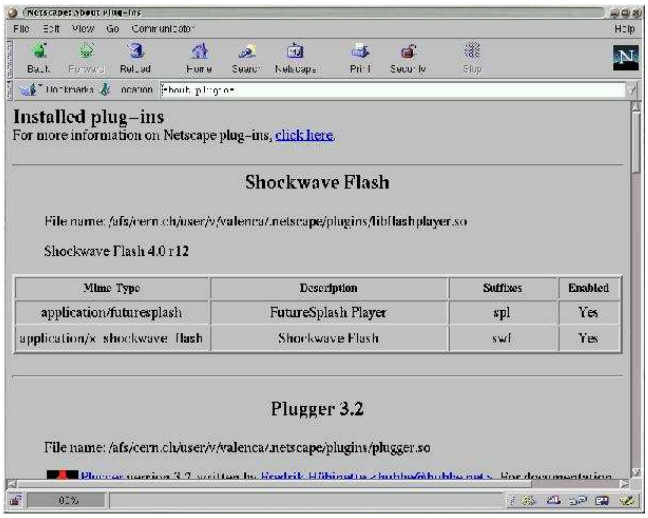
Plugins are typically implemented as the second type since their characteristics are especially adequate for the "plug" concept. But in order to comply with the desired functionality of a plugin, the following list of requirements must be met:

1. The application must be independent from the plugins.
2. There should be a "black-box" view in both directions.
3. The application must not "know" the plugin beforehand; the plugin must introduce itself to the application.
4. The application must be loosely coupled with the plugins; a common interface to be provided by the plugins (whether total or partial) must be defined precisely (Contract).
5. The application should define what services are provided to the plugin via interface definitions.
6. A plugin must be self-contained.
7. A plugin should create a plugin instance to handle each task.

The interface common to all plugins is one of the critical points and, although this interface varies between technologies, the plugin should provide:

- information on the type of task it can handle
- a function to be called to handle that type of task (`do_work()`)
- an instance-type strategy including a structure to store all the data of a particular task and a way to know if there are no tasks (instances) still working so that the plugin can be unloaded
- a function to initialize the plugin.

## Some popular examples




Netscape (and other Web browsers) uses plugins to view/handle special data present in some Web pages, for example, audio and video.

The API is very well defined in the documentation: it contains a list of functions provided by Netscape, a list of functions the plugin must and can define and structures to be used. Since the communication is all done by this total interface, both parts act as complete "black-boxes". Another interesting characteristic is the life cycle of a plugin: *Initialize* (when the plugin is first called), *New Instance* (each time it is invoked), *Destroy Instance* (when a task is completed and that instance is no longer needed) and *Shutdown* (when all instances are destroyed).

```
PluginInitResult
init_plugin (CommandContext *context, PluginData *pd)
{
  FunctionCategory *cat;

/* The next line checks if the Gnumeric version being
   used corresponds to the required by the plugin */
  if (plugin_version_mismatch
          (context, pd, GNUMERIC_VERSION))
     return PLUGIN_QUIET_ERROR;
/* +++++++++ registration of new plugin functions
  cat = function_get_category (_("Fact PlugIn"));
  function_add_args (cat, "MyFactorial", "f",
                     "number", NULL, func_fact);
   ++++++++++++++++ */
  pd->can_unload     = can_unload;
  pd->cleanup_plugin = cleanup_plugin;
  pd->title = g_strdup ("My PlugIn Plugin");

  return PLUGIN_OK;
}
```

The program listing above is part of a Gnumeric plugin source code. Gnumeric uses plugins both to handle different formats (e.g., Excel) and add functions not defined (statistical, numerical,...). The API the plugin must define is very concise and partial: the plugin registers the new functions it defines (in lines marked ++++ above, the plugin is registering a function, in this example, the factorial function). The `init_plugin()` function is called at start-up. Besides registering the functions it also provides information on the Plugin (in this case, the title) and functionality to unload the plugin (e.g., check if there are no instances left). Although Gnumeric plugins are not the best example, since the requirements above are not completely met (Gnumeric is completely transparent to the plugin) it provides a very simple example with most of the core ideas.

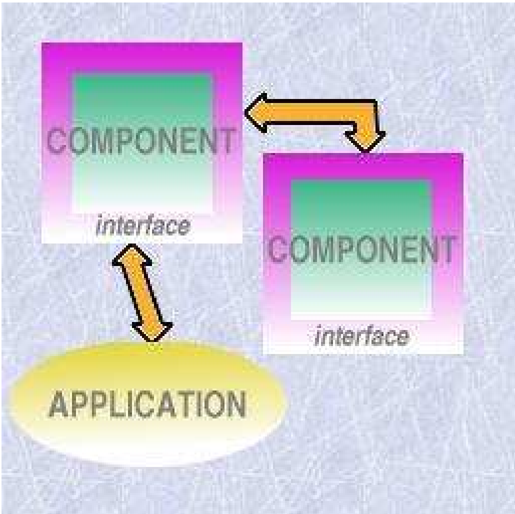
## On components

There is not much consensus on what constitutes a component. Although this is not complete and may seem to general we can say that a component is a unit of software with the following characteristics:

1. *It is an indivisible unit of independent deployment.* The component must be independent from the environment and other components and, so, has to encapsulate its features. It also cannot be partially deployed since it constitutes an whole.

2. *It is a unit of third-party composition*. A party exterior to at least one component must be able to compose them together. The component has to be self-contained, interface hiding and provide a well defined interface
3. *Has no persistent state*. This way, any copy of a component will not differ from the original so that, at most, one copy of the component is present.

An illustration of the use of components is shown below.




A typical use is in big, complex and heterogeneous projects divided in different parts, each with its particular function and interacting with other parts of the project, being developed by different teams and where each part does not want to know how the other parts are implemented or where they are located. Basically, each part constitutes a component. An interface contract for each component is established and interaction is done through these interfaces.

Another example can be found in desktop environments. Imagine a user writing a document or a web page where he wants to integrate a graphic. Putting graphics functionality in the word processor or web page editor would be overloading them (how about audio, movie, barcharts...?). Doing it outside and importing as an image is also not the best solution. But if there is already a mechanism for handling that functionality why not reuse it? Basically, the idea is to "embed" the graphics mechanism in the word processor or web page editor (something like: "when handling this data use this particular mechanism"). But to have a good interaction between mechanisms we need independence, interface contracts and again components appear as a good solution for this.

The key aspects in component-based development are [1]:

- Standardised parts and services (both through the standardisation of components and the environment in which they operate.)
- Minimal or no contact between entities designing components; decoupling of development efforts.
- Access to practical (re)use both internal and external through standardised interoperability, configurability, and deployment.
- Independent variability through documentation of components “seams” (the interfaces used and exposed).
- Source code for custom needs; binary components for standard needs.
- A strict emphasis on component assembly first; manual coding second.
- A robust mechanism for the delivery of architectures, frameworks, and services.

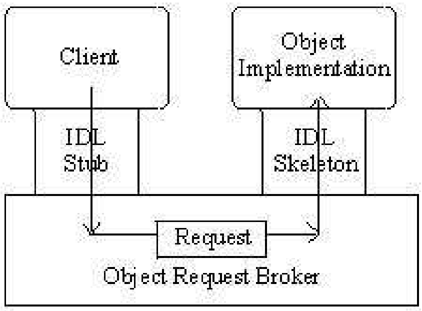
But to conform with this characteristics it is necessary to complement the definition above with a list of requirements that must be met [2]:

1. A component must be loosely coupled with other components via interface definitions and implementation separation.
2. A component must explicitly identify both the interfaces it uses and exposes, and/or the service categories it uses and exposes.
3. A component has an unbreakable contract with all of its clients by virtue of the services its interface promises to deliver (design by contract).
4. A component must be deliverable in binary form (black box).
5. A component must use and expose services through standardised middleware services (DCOM, CORBA, JavaBeans, etc.)
6. A component must be configurable at design-time/run-time through a standardised interface (such as ActiveX and JavaBeans property sheets or standard configuration files).
7. A component must be interoperable with other components across processes, machines, networks, languages, operating systems, transaction environments, and tools.
8. A component should make use of standardised infrastructure services such as security, licensing, versioning, life-cycle management, visual assembly, event notification, configuration, scripting, metadata, introspection, persistence, relationships, administration, management, debugging, and installation.
9. A component should be able to wire itself into a collaboration with other components solely via design-time/run-time configuration.

One of the keywords in component-based technology is *middleware*. Middleware is now a very popular term and is basically used to refer to software "that sits in the middle, connecting diverse (and usually preexisting) programs together". In other words, a separate product that acts as a glue between two applications.

[1] Refer to the documentation mentioned in the Section *Additional remarks*.

[2] The last two items are desirable but not essential. They are also very hard to comply.

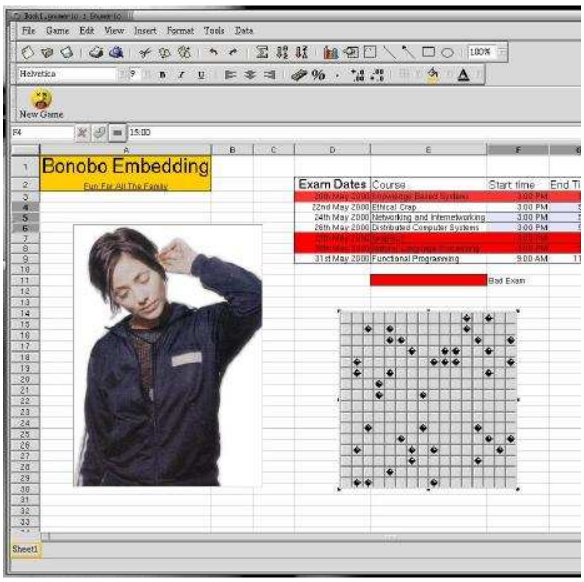



A request passing from client to object implementation (from the OMG documentation)

CORBA is one of the most popular architectures and specifications for creating software components. It provides the functionality required as middleware. CORBA stands for Common Object Request Broker Architecture and is defined by OMG. Its building blocks are:

- *OMG Interface Definition Language, OMG IDL* which defines mappings for a variety of different languages and which purpose is to abstract from the language being used to implement the component so that they all have a common language.
- *The Object Request Broker or ORB* which handles requests from programs to components and between components enabling abstraction of where the program that receives the request and the one sending are.
- *The standard protocol IIOP* which provides a standard communication between different ORBs above TCP/IP.

CORBA provides, with these building blocks, independence of language and independence of platform.




As an example, GNOME implemented BONOBO in order to comply with interoperability between different applications in the desktop (e.g, embed a graphics object in a word processor document). BONOBO is basically a set of CORBA interfaces and an implementation in GNOME/GTK+ of these interfaces that are designed to be simple to use and hide CORBA and between-components protocol used. An application uses these interfaces to export its functionality or integrate other components functionality making it possible to edit a graphic or even play GnoMines in a spreadsheet.

## Additional remarks

When developing either plugins or components, some requirements must be met to obtain all the functionality provided by these two paradigms[3].

It may seem that some of these requirements are, sometimes, dispensable. For example, why use strategies like CORBA when it does not seem to be necessary? The problem is that, if we want to make use of the benefits that this technology enables, not complying with just one of the requirements will make the plugin/component-strategy less robust and flexible. In this example, if later on there would be a need to have different platforms, for example, wrappers would have to be written to do the work this technology is responsible for.

These technologies are very different and each has its advantages and disadvantages as also having different scopes. If it is the right choice to use one of them or which one to use depends on the project. Nevertheless there are some common keywords that pop out:

- Modularity
- Encapsulation
- Interface contracts

This article presents only an overview on this interesting and popular technology. Please refer to `http://wwwinfo.cern.ch/pdp/ose/plcomp00/` for a more detailed study, specially regarding the pros and cons, why the above requirements are needed and more focus on implementation issues.

## Glossary

***Plugins*** A unit of software whose purpose is to extend the functionality of an independent application at runtime. The application has no prior knowledge of the plugin; plugin must act as a "black-box" and present itself by means of a standard interface.

***Components***
1. A component is an indivisible unit of independent deployment.
2. A component is a unit of third-party composition.
3. A component has no persistent state.

***Shared libraries*** a block of code, loaded on request, providing data types and procedures.

***Middleware*** A popular and generic term basically used to refer to software "that sits in the middle, connecting diverse (and usually preexisting) programs together". In other words, a separate product that acts as a glue between two applications.

**About the author(s):** *Paula Valença is a Technical Student at CERN, working in the team of Philippe Defert.*

[3] Some may not agree with the lists presented above, of course.

## 7.4 Questions and Answers from the Computing Helpdesk

*Collected by the IT/User Support Team*

*Nicole Crémel (Editor), IT/User Support*

*This is a collection of Questions & Answers that have been treated by the Computing Helpdesk managed by the User Support group. The number in parentheses refers to its relative numbering in the User Support "Question-Answer" database, at URL:* `http://cern.ch/consult/qa/NNN`*, where* `NNN` *is the problem identifier (number).*

### Windows support

**(Hannes Schwarzbauer, IT/User Support)**

**Question [230] – RealPlayer**

RealPlayer *availability on NICE.*

**Answer**

Since January 2000, the new version of *RealPlayer*, *"RealPlayer v7 Basic"* is available on NICE. You have to reboot to enable the new configuration.

*RealPlayer v7 Basic* replaces the old version, *RealPlayer G2*, which had some installation problems.

The new version can be called from the *Start Menu, More Applications, Shareware, RealPlayer*. The common "look and feel" has not been changed, and transition should be transparent for the users.

**Question [231] –** `nicecopy.log` **cannot be written**

*When I start my PC, at the very beginning (before the system is really ready to be used) I get this error message:*

```
"The log file 'c:\nicecopy.log' cannot be opened.
Is it read protected?
Is the Windows directory writable?"
```

*When I click on "OK" the startup continues and the system seems to behave normally. What is this message for?*

**Answer**

This file is just an internal log, it can be deleted at any time. The reason why it puts itself into "read only" status (although it does not show this under "properties") is yet unknown and seems to be a bug.

**Question [258] – Schedule regular execution of a task on Windows**

*How, on a PC/Windows, can I run a certain programme on a daily (weekly, monthly...) recurrent basis (like a "cron" job in Unix)?*

**Answer**

- W95: There is no supported software for this on NICE 95.
- NT: Use the `"AT"` facility: From the DOS prompt, type `AT` followed by the parameters wanted. To see online help, type `AT /?` to get familiar with this powerful application.

  Note that tle scheduler must be switched on to do this, otherwise `AT` will tell you. To switch it on, go from the control panel into "`services`", highlight "`schedule`" and click "`start`".Its status will now reflect this.
- Other alternatives:
  - install IE5 and use its scheduler task
  - buy or down-load (CERN unsupported) commercial software

**Question [270] – Excel file read only?**

*Excel keeps asking me:*

```
"File name" should be opened as read only unless
you need to save changes to it. Open as read only?
```

*Why and how can I avoid it?*

**Answer**

You might have by mistake your R/O box ticked. This is an Excel security mechanism, controllable by the user. The philosophy is that the file can be modified, but only saved under `"Save as..."`, thus maintaining the original file and file name.

Can you please check this:

- From `"File"` click `"Save as"`
- Now in there, at the bottom, is a line called `"Read only recommended"`. If this one is ticked, just simply un-tick it and the problem should disappear.

**Question [273] – Access control with Exceed**

*I cannot run an X-Window application (like PAW) on my PC with Exceed.*

**Answer**

The first thing to check is that you have enabled host access for the host where you are running your application: i.e., from the Exceed 6 option in Windows:

- choose `"Tools"` (or `Z:\P32\Exceed6\Nice\TOOLS`) Click `"Xconfig"`, and then `"Security"`.
- The Host Access Control list is Disabled by default for any host access: you must *enable* host access (or you can edit the `xhost.txt` file to include the remote workstations).

**Question [279] – Excel chart wizard**

*I was given a task to create some charts within an Excel file. I am used to do this via the* `chart wizard`*, which I cannot find on the toolbar for this particular file.*

**Answer**

Your file is used as a shared workbook. In this mode, there are some restrictions, like the creation of charts. Therefore, Excel does not display the `chart wizard` in the taskbar.

To work around this, go into `"tools"` - `"share workbook"` and un-tick the option `"allow changes...."`

**Question [280] – Print multiple sheets on one page from PC**

*Is it possible to print multiple sheets on one single physical page from a PC?*

**Answer**

It depends on the printer type, i.e. whether the printer is capable of translating the appropriate software or not.

At the moment, there are only a few printers that can do this when addressed from a PC. Some new generation high range printers are capable of doing it, e.g., the HP laser jet 8000 series ones. As an example, one such printer would be *513-hpbis*.

Recipe: once the printer is addressed from the pull-down menu, go into `"properties"`, down into `"printer features"` and `"pages per sheet"`. There you can select `2,4,` etc...up to 16 logical sheets per page to be printed.

## UNIX Support

**(Roger Woolnough and Nicole Crémel, IT/User Support)**

**Question [236] – Access to fonts with ASIS software from outside CERN**

*It looks that there is a problem in the HEPiX X11 login scripts, file* `xprofile`*. At external HEP institutions, it takes ages to login when trying to access the font servers. I found that the problem disappears when modifying the following lines in the file*

*`/usr/local/lib/hepix/X11/site/xprofile`:*

```
case $RANDOM in
 *[0123]) HX_FONT_PATH="tcp/xtsoft1:7100 tcp/xtsoft2:7100";;
 *[456])  HX_FONT_PATH="tcp/xtsoft2:7100 tcp/xtsoft3:7100";;
 *[789])  HX_FONT_PATH="tcp/xtsoft3:7100 tcp/xtsoft1:7100";;
esac
```

*and include* `.cern.ch` *in the font server names (*`xtsoft...`*).*

**Answer**

The file `/usr/local/lib/hepix/X11/site/xprofile` belongs to the ASIS package `HEPiX/scripts-X11-cern` which contains the CERN customisation of the HEPiX X11 scripts (package HEPiX/scripts-X11). It is meant to be used only at CERN.

If you want to use ASIS from outside CERN, you should not take this package and supply your own "site level" HEPiX configuration where you can choose to use the CERN font servers, or your own font servers (or nothing).

**Question [238] – AFS problems on LINUX (reported by the Cron Daemon `/usr/sue/adm/daily.log`)**

*On my Linux machine I have been having problems with AFS which have been reported by the Cron Daemon* `/usr/sue/adm/daily.log`*.*

*Specifically, I cannot use the command* `xprint` *as a segmentation fault occurs. The following statement is repeated again and again in the report:*

```
/usr/local/bin/gnu/df: : command not found
```

*and later on in the report:*

```
/usr/local/bin/gnu/df: syntax error near unexpected token
```

**Answer**

Most probably you have an AFS cache problem, which is a known problem under Linux 5.1. Please follow the steps described at the URL `http://wwwinfo.cern.ch/pdp/ose/linux/tips/afs_cache.html` to solve your problem.

**Question [245] – `find` command on Linux**

*Issuing the identical command*

*`find . -name template -print`*

*gives me dramatically different results when issued on RSPLUS and Linux. The result in Linux gives much less files than on RSPLUS, even if issued by the same AFS user.*

**Answer**

Under Linux, you should use the `-follow` option, because symlinks are not deferred by default. It's not a bug, it's a feature!

**Question [269] – Disk quota on AXCAD cluster**

*How can I check the disk quota on* `axca06` *in the AXCAD cluster, and who can add quota ?*

**Answer**

This depends on whether the user wants to check quota for the AFS (or NFS) file system, or for the local disks:

- To check quota in AFS (and NFS) for a given user (e.g. in the AXCAD cluster) the user only have to type `check_quota`
- To know the available disk space in any local file system on UNIX just type:
  `cd` *`the_directory`*
  `df -k .`

Note that the standard AFS command on central UNIX machine to check AFS quota is `fs listquota`.

User requests for more AFS quota can be handled by the Helpdesk, but request for NFS quota must be sent to `AXCAD.Support@cern.ch`.

## Mail Support

**(Michel Franceschi and Miguel Marquina, IT/User Support)**

**Question [249] – Unreadable Word attachments**

*I have received an attachment via email, apparently a Word document, which Word itself does not understand. It says something like:*

```
im-123456 is a mail merge main document. Word
cannot find its data source, perso.MER
```

*Has the mail system corrupted it?*

**Answer**

No, the attachment is very probably all right. The message you get is due to the way in which the document has been prepared and sent to you, using the Word mail-merge feature.

When you get the "error", you are offered two buttons. Select `"Options"`, then `"Remove all Merge Info"`. If you then save the document you should be able to open it again without any problem.

**Question [278] – Information about e-mail addresses**

*Is it possible to check (or interrogate) which e-mail address will be used by the CERN "fuzzy-matching" program that translates e-mail addresses ?*

**Answer**

Yes. Send an e-mail to `CERNDIR@mail.cern.ch` and put the e-mail address in the subject line.

This can be very useful to know more about a service login. For instance, if you want to interrogate "`Nice.Support`", send a mail:

```
To: cerndir@mail.cern.ch
Subject: Nice.Support
```

And you will receive back the following answer:

```
***** CERN DIRectory Lookup Mail Service (Version 2.1, ...

The following records match exactly your
requested query :  Nice.Support

1. First name(s): NICE  ***** Last name(s): SUPPORT
   Cern phone number:78888  ***** Beep number:
   Division:IT  *****  Office:
   Physical E-mail address: helpdesk@mail.cern.ch
   Generic E-mail address: [NOT FOUND]

2. First name(s): SL NICE  ***** Last name(s): SUPPORT
   Cern phone number:75275  ***** Beep number:13+208
   Division:SL  *****  Office:864/2-B06
   Physical E-mail address: sl-nice-support@listbox.cern.ch
   Generic E-mail address: Sl-Nice.Support@cern.ch

Scanning for search patterns stopped at first blank line
   in message body.

----------------------------------------------------------
           Please take the time to read this
----------------------------------------------------------
...
```

It shows that the e-mail `Nice.Support@cern.ch` is going to the Computing Helpdesk (`helpdesk@mail.cern.ch`), and that there is also a local service for SL division, at e-mail `Sl-Nice.Support@cern.ch`.

## Web Publishing

**(Maria Dimou, IT/User Support)**

**Question [233] – Web site registration problems**

*I have problems with the permissions for the access to the Web site `http://cern.ch/foo`*

*I setup with* Frontpage *browse permissions only from `137.138.*.*` and from another domain but now I cannot access from any computer at CERN.*

**Answer**

When a user has access problems to his Web site (either for access rights and/or IP address restrictions) he can sort this out himself using the form available at:

`http://cern.ch/webredirect/view_database.htm#Edit`

and then select, at the bottom of the form result: *[Set or Recover Access Rights] [Change or View the IP Access restrictions (Internet/Intranet)]*

In this case, open

`http://cern.ch/webredirect/ViewSiteDetailEx.asp?WebName=foo`

and select

*[Change or View the IP Access restrictions (Internet/Intranet)]*

If that site was restricted to an empty list of IP addresses, you can adjust permissions by going to:

`http://cern.ch/WebRegister/ChangeIpSecurity.asp?WebName=foo`

**Question [261] – *JavaScript* syntax**

*Where can I learn more about* javaScript *syntax?*

**Answer**

Below are some pointers:

`http://webservices.web.cern.ch/WebServices/AuthoringDoc/Scripting/Scripting.htm`

`http://developer.netscape.com/docs/manuals/index.html?content=javascript.html`

`http://www.infohiway.com/javascript/indexf.htm`

**Question [242] – Sub-search within my Web site**

*I want to make a search prompt that retrieves pages only from within my site (current directory downwards).*

**Answer**

From the parent document *Documentation for Web Authors* at the URL:

`http://cern.ch/Web/AuthoringDoc/Authoring.htm#AdvancedServices`

the user should click on *Adding sub-searching to your Web*.

**Question [248] – Moving bookmarks between Unix file systems**

*I want to transfer my Netscape bookmarks and preferences from one UNIX machine to another, in order to keep the same environment between these 2 different machines (not AFS based). What are all the files I should transfer?*

**Answer**

Make a tar of the contents of your `$HOME/.netscape` directory, as follows:

```
cd
tar cvf /tmp/netscape .netscape
```

and transfer (e.g., with ftp) the tar file `/tmp/netscape` to the other host. Then untar it:

```
cd
tar xpvf /tmp/netscape
```

Note that if in the destination host you already have a `.netscape` directory, it will be overwritten by the untar. If you want to save it, make a copy using the command: `mv .netscape .netscape-old`

**Question [255] – Viewing Postscript files in Netscape**

*While using Netscape, when I click on a link to a PS file, `ghostview` does not work any more. The error message is*

```
Could not load program /usr/local.bin.X11/gv
Member shr.o not found or file not an archive
Could not load library libXmu.a[shr.o]
Error was: No such file or directory
```

**Answer**

In the Netscape Navigator window:

| | |
|---|---|
| select | `Edit|Preferences|Applications` |
| choose: | `application/postscript` |
| then: | `Edit` |
| and set: | `"Handled by"` |
| to: | `/usr/local/bin/X11/gv %s` |

**Question [276] – French characters in NICE95 Netscape**

*Problem with EDH and Netscape: entering the character ", Netscape crashes with the message:*

```
NETSCAPE caused an invalid page fault in
module MFC42.DLL at 0137:5f401ced.
```

**Answer**

It is a known problem of Netscape on NICE/95, and the solutions are:

1. Do not use "US International" keyboard layout, use the native French one instead if you want to have accented characters.
2. Use NICE/NT instead of NICE/95.

## Miscellaneous

**(Nicole Crémel, IT/User Support)**

**Question [237] – CERNlib 2000 on Linux (`undefined reference to 'yp_...'`)**

*I have the following problem when trying to link a FORTRAN 77 code with CERNlib on Linux Redhat 5.1 (reformatted):*

```
/cern/pro/lib/libpacklib.a(ypgetacctent.o):
   In function 'ypgetacctent':
   ypgetacctent.o(.text+0x11e):
      undefined reference to 'yp_get_default_domain'
   ypgetacctent.o(.text+0x17b):
      undefined reference to 'yp_match'
   ypgetacctent.o(.text+0x1e2):
      undefined reference to 'yp_match'
   ypgetacctent.o(.text+0x26b):
      undefined reference to 'yp_all'
```

**Answer**

This is a known "feature" (since the last release of CERNLIB).

Following is a news which had been posted in cern.computing, on 30 Nov 1999, just after the CERNLIB release for Y2K. The solution is either to use the standard `'cernlib'` command in your link procedure, or to add the flag "`-lnsl`".

```
--------------------
Subject: Unsatisfied yp... on Linux ( in 'ypgetacctent' )
Date: 30 Nov 1999 15:39:19 +0100
From: IT News
Organization: CERN-European Laboratory for Particle Physics
Newsgroups: cern.computing

On the Linux systems there is the following incompatibility
in CERNLIB 2000. The new shift (rfio) software used in this
release needs the nsl library. Application developers using
the latest cernlib command are OK but those with their own
link procedures should add -lnsl. The probable symptoms are
unsatisfied externals yp......

Ian McLaren / Cern Program Library
```

**Question [277] – Multiply defined symbols coming from `libcxxstd.a` on dxplus05**

*It seems I get wrongly some errors (multiply defined symbols coming from `libcxxstd.a` ) when building my C++ code on `dxplus05`.*

**Answer**

This is explained by the following CXX 6.2 compiler release notes:

```
 Compaq_C++_Version_6.2

 Release Notes for Compaq Tru64 UNIX

 [...]

4.3 Restrictions in Version 6.2

 [...]

o  Linker might produce multiply defined symbols from
   Standard Library

   The linker might produce multiply defined symbols from
```

```
the Standard Library when linking against your own
shared library.

If you find multiply defined symbols like the following
coming from libcxxstd.a, you might be creating a shared
library and then using that shared library to create an
executable.
ld:
/usr/lib/libcxxstd.a(typeinfo.o):
std::__vtbl_3std9type_info: multiply defined
/usr/lib/libcxxstd.a(vec_newdel.o):
__vec_new_eh: multiply defined

The problem is that libcxxstd.a is an archive library.
When you create a shared object that references this
library, the symbols are pulled into that shared
object. When you then use the shared library, it
finds the symbol both in your shared object and in
the libcxxstd.a library archive.

To work around this problem, use a command like the
following when you create your shared library objects:

cxx -hidden /usr/lib/cmplrs/cxx/libcxxstd.a \
    -non_hidden -shared foo.cxx

To prevent the generation of multiply defined symbols,
a future release will include a shared object version
of libcxxstd.a.
```

### About the author(s):

*Nicole Crémel is working in the User Support group and, with Roger Woolnough, is responsible for the supervision of operations of the Computing Helpdesk. Editing and revising the "Question and Answers" database is a daily task of most members of the group.*

# 8. User Documentation

*Everything related to "computer documentation": updates to computer documentation distributed by IT (including the User Support reference cards), offerings of the Computing Bookshop, etc.*

## 8.1 The CNL – 35 years of History at your Fingertips

*Julien Gavaggio, and Miguel Marquina, IT/User Support*

**Abstract**

*History is important. It serves our memory when our memories fail. We have been recovering the indexes to the CNL issues published since 1966.*

The CERN Computing Newsletter itself reflects somehow the History of Computing. The first issue saw light on February 15, 1966. It was just eight type-written pages and talked about the CDC 6600, usage policy for an IBM 1401, problems with the card punch… and surprise, surprise, an invitation to users to contact the Program Enquiry Office in case of suspected machine errors (imagine our graffiti with "User Support was here!" :-)).

For this reason we have decided to make available the CNL indexes in electronic form, so that they can be searched for and examined. We did not go as far as to scan the issues themselves, but we will very likely place articles here and there as requesters ask for specific material. After all, we are the privileged owners of a complete CNL series, and we intend to share them for the joy of our readers.

All indexes are available from the CNL Web pages at the URL `http://cern.ch/ref/cnl`.

Your feedback will be much appreciated as to the form in which they are presented.

**About the author(s):** *Julien came as a short term assignment to work for one month among us. With this small project as an excuse, he has learnt perl and gained some insight into the mailing system. Now he has bit the flavours of what programming fun can be. We wish him all the best for the future.*

## 8.2 News from the User Support Book Shop

*Jutta Megies, Roger Woolnough, IT/User Support*

Since the last edition of the CNL the principle change at the bookshop has been the arrival of the *Windows 2000* and *Office 2000* books. In the last edition I indicated we hoped to get an arrangement with *Hachette* but unfortunately this proved too complicated so I await the *Frankfurt book fair* to make arrangements for us to stock French/English dictionaries from another source.

There have not been many new editions since the last report but recently we received *The Java Programming Language* in its third edition, *Thinking Java* second edition and by the time the CNL is published the new edition from *O'Reilly* of *The Linux System Administrators Guide* should be in stock.

The book from *O'Reilly*, *CGI Programming with Perl*, has finally been re-released, having been out of print for a long time. We now have plenty in stock.

Other new books added to the list are :

1. *The Complete Red Hat Linux Training Course, an interactive CD and book*
2. *The XML Companion*
3. *Labview student edition and CD*
4. *The Dreamweaver 3 Bible*
5. *Practical Java*
6. *Fortran 77 for Engineers and Scientists*
7. *Antimatter* (by Gordon Fraser)

For those of you wishing to start with *Windows 2000* we have two specialised books *The Windows 2000 Resource Kit* from *Microsoft Press* and from *Pearson* the *Ultimate Windows 2000 System Administrators Guide*. For the general user the book from *Mc.Graw Hill*, *Windows 2000, the Complete Reference* is a good learning tool and reference guide.

For users of *Office 2000* we have chosen the series we used before, the *Microsoft Press* tutorial series *Step by step* and the reference books *Running*. From this selection we have examples for *Word, Excel, Access, Outlook, Project* and *FrontPage 2000*. We also stock the popular *Using* series as an alternative and here we have the following titles available *Word, Access, Excel, Word and Excel, Outlook, Office* and *FrontPage 2000*. To complete the list we have a new book from *O'Reilly*, *Outlook 2000 in a Nutshell*.

The complete list of books and prices is available at the `http://consult.cern.ch/books/` or you can mail us at `bookshop@cern.ch` where we always welcome suggestions for new titles.

## 8.3 User Support Book Catalogue

*Roger Woolnough and Jutta Megies, IT/User Support*

Situated in Building 513 1-022 (tel. 74050) is the Computing Bookshop (`http://consult.cern.ch/xwho/people/67659`) provided by the *IT / User Support group*, where CERN users can find computer books and CDs at discount prices. The service is open weekdays from 8.30 to 12.30 or contactable by e-mail to `Bookshop@cern.ch`. Books are purchased from some 15 publishing houses and the catalogue offers a selection of documentation aimed at the range of computing utilities available at CERN. The service welcomes suggestions from the user community for new acquisitions. Purchasing may be done internally via EDH or TID or alternatively cash payments via the CERN Bank.

The list, with all the relevant information, is compiled regularly at URL: `http://cern.ch/consult/books`

# 9. Just For Fun . . .

*Some jokes to end this CNL, and make more fun. . . for those who did not read them already, somewhere else!*

## 9.1 A French Story about Consulting

*Unknown author*

### Il était une fois un berger et ses moutons...

Il était une fois un berger et ses moutons au bord de la route. Tout d'un coup surgit une Jeep Cherokee flambant neuve, conduite par un jeune homme en costume YSL, chaussures Cerruti, lunettes Rayban et cravate. La voiture s'arrête et le jeune homme s'adresse au berger :

*– Si je devine combien de moutons vous avez, vous m'en donnez un ?*

Le berger regarde le jeune homme, regarde les moutons qui broutent et dit :

*– Oui.*

Le jeune homme gare la voiture, branche le notebook et le GSM, entre dans un site de la NASA, scrute le terrain a l'aide du GPS, établi une base de données, 60 tableaux Excel pleins d'algorithmes et d'exponentielles, plus un rapport de 150 pages imprimé sur sa mini imprimante HIGH-TECH. Il se tourne vers le berger et dit :

*– Vous avez ici 1586 moutons.*

Le berger répond :

*– C'est tout a fait correct, vous pouvez avoir votre mouton.*

Le jeune homme prend le mouton et le met dans le coffre de la jeep. Le berger lui demande alors:

*– Si je devine votre profession, vous me rendez mon mouton ?*

Le jeune homme répond :

*– Oui.*

Le berger dit tout de suite :

*– Vous êtes consultant.*

*– Comment vous avez deviné ?* Demande le jeune homme.

*– Très facile,* répond le berger :

*1) Vous êtes venu ici sans qu'on vous appelle.*

*2) Vous me prenez un mouton pour me dire ce que je savais déjà.*

*3) Et vous ne comprenez rien à ce que je fais, parce que vous avez pris mon chien !*

## 9.2 Le Monde est Fou...

*Unknown author*