

# Summary

- n Typical uses for time stamps
  - n Alarms
  - n Logging
  - n Post Mortem
- n GPS
- n Formats for time (text, Unix and Windows, Java, Oracle)
- n How Computers keep time
- n Network Time Protocol
- n PLCs
- n Conclusions

# Timestamping

Presentation to the Third LHC Controls Project Workshop

Alastair Bland, 21<sup>st</sup> March 2002

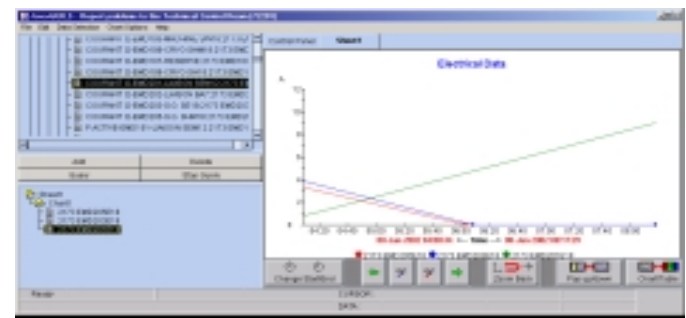
# Timestamps in the Alarm System

- n Currently Alarm System orders alarms with its own timestamps
- n Intention is to allow users to timestamp closer to source so that a more exact sequence of events can be deduced
- n The example below shows a power cut as displayed to the TCR operators in the Alarm screen. Note that there are no fractional seconds.

Date	Time	System Name	Identifier	Problem Description
08/01	08 03 40	SECC	INTEGRATION	SECC INTR-034
08/01	08 03 45	ELEC	BT	MEV ST-PT-04
08/01	08 03 48	ELEC	BT	MEV ST-PS-09
08/01	08 03 45	ELEC	ODDLEARM	BAT281_ST22
08/01	08 03 48	ELEC	SM400V	MEV CAL-1
08/01	08 03 48	ELEC	SM400V	MEV EL-ST 63
08/01	08 03 48	ELEC	SM400V	MEV EL-ST 72
08/01	08 03 48	ELEC	SM400V	MEV EL-ST 71
08/01	08 03 48	ELEC	40V_0-WEARION	BAT151_ST11
08/01	08 03 48	SECC	PRO	MEV OIDS-10008
08/01	08 03 47	ELEC	110V_RECLAIMAGE	SE_BAS1_BAT998_000_029_C1
08/01	08 03 47	ELEC	110V_RECLAIMAGE	SE_BAS10_BAT911_011_029_A8

# Timestamps in Logging Systems

- n In general the logging systems are again using Oracle for storing date and time information, so there are no fractional seconds
- n The JavaGuils example below shows data logged via the Technical Data Server (TDS) in Oracle. Again only second resolution is available.



# Timestamps in Post Mortem Systems

- n Up to now very few real Post Mortem systems because there is no trigger.
- n Many systems do keep a local rotating buffer which the user hopes to catch after an event before it is over written.
- n The timestamps below are from the source (Front End) of the electrical data and include fractional seconds. Normally this data is only looked at by the Equipment Specialist.

```

EBU25/A3 D Mains Failure 8JAN 06:03:44:116 ##### 739 0
EBU24/A3 D Mains Failure 8JAN 06:03:44:506 ##### 740 0
EBU25/A3 A Current limited 8JAN 06:03:44:875 ##### 741 0
EBU24/A3 A Current limited 8JAN 06:03:45:269 ##### 742 0
EBS24/A4 S Rectifier Mains Failure 8JAN 06:03:47:716 ##### 743 0
EBS24/A4 A Rectifier Mains Failure 8JAN 06:03:47:719 ##### 744 0
EBS24/A4 A Reseau2 Alarm 8JAN 06:03:47:721 ##### 745 0
EBS24/A4 A Battery Alarm 8JAN 06:03:47:722 ##### 746 0
EBS24/A4 D Rectifier Mains Failure 8JAN 06:03:47:725 ##### 747 0
EBS24/A4 D Reseau2 Alarm 8JAN 06:03:47:726 ##### 748 0
EBS24/A4 D Battery Alarm 8JAN 06:03:47:727 ##### 749 0
EBS68/B3 S Reseau2 normal Operation 8JAN 06:03:49:405 8JAN 06:03:49:405 750 0
EBS68/B3 A Rectifier Mains Failure 8JAN 06:03:49:409 ##### 751 0

```

# GPS (1)

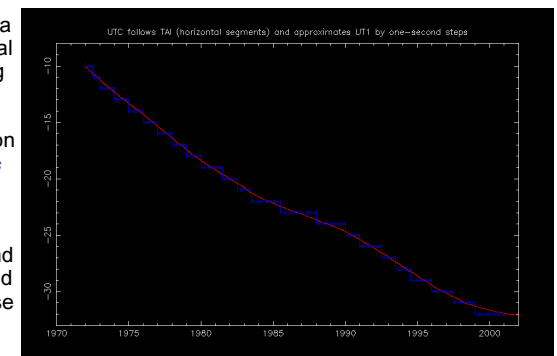
- n The usual way to get accurate time is with a Global Positioning System (GPS) receiver
  - n 1us resolution or better
  - n Lots of hardware solutions
    - n Standalone (serial I/O plus pulse per second), VME bus, PCI bus
    - n With IRIG-B output for distribution over long distances (for example in tunnels)
    - n With freeze input for 100% hardware based time stamping
    - n Pulse per second and programmable pulse outputs
    - n Cable delay correction
  - n Already used in
    - n PS: fed into the Timing System via the Master Timing Generator
    - n SPS: Network Time Protocol Distribution to Work stations and Front Ends
    - n LEP: time stamping Beam Dump requests and RF diagnostics (via IRIG-B)
    - n String2: time distribution over WorldFIP to Power Converters and Magnet Protection systems
- n Lock on to GPS satellites after power on can take many minutes. To speed this up:
  - n Needs on board clock which works when power is off
  - n Needs to be told where it is

# GPS (2)

- n A GPS receiver normally provides
  - n GPS time (does not include leap seconds since 1980)
  - n UTC (Universal Time Coordinated), previously know as GMT
  - n Registers where local time offset can be set, under external software control
- n Unfortunately, not all GPS receivers are designed to be used as time sources. Some will create time glitches when they change which satellites they are using. Some are unable to keep a time lock when fewer than three out the 24 satellites are in view. Others give low priority to time output. Also the crystal for keeping time should be stabilized for temperature variations.
- n There will be one source feeding the SPS / LHC MTGs and the NTP distribution
  - n However there will be other GPS receivers in the outlying buildings to check that the Timing delays have been correctly set
  - n This tends to imply that the time should be sent early to all the outlying buildings so it arrives on time.

# Leap Seconds

- n The decision to introduce a leap second in UTC to deal with the poor time keeping of the earth is the responsibility of the International Earth Rotation Service at the [Observatoire de Paris](#). According to the CCIR Recommendation, first preference is given to the opportunities at the end of December and June, and second preference to those at the end of March and September. Since the system was introduced in 1972 only dates in June and December have been used.



- n No leap second will be introduced in UTC on 30 June 2002
- n All leap seconds so far have been positive



## Time Formats (Text)

- n International Standard ISO 8601 specifies numeric representations of date and time. A long document as usual ...
- n In summary dates should be like:

### YYYY-MM-DD

where YYYY is the year in the usual Gregorian calendar, MM is the month of the year between 01 (January) and 12 (December), and DD is the day of the month between 01 and 31.

- n Times should be:

### hh:mm:ss

where hh is hours since midnight (00-24), mm is minutes since the start of the hour (00-59), and ss is seconds since the start of the minute (00-60). If the hour value is 24, then the minute and second values must be zero. [The value 60 for ss might sometimes be needed during an inserted leap second]



## Time Formats (C library)

- n Time is a signed 32 bit integer giving the number of seconds since the 1<sup>st</sup> of January 1970, not including leap seconds (like our watches). Problems will occur in 2038 when it becomes negative.
- n Unix, Windows2000 and MacOS X natively in their C libraries support the time() system call:

```
#include <time.h>
main() {
    time_t t;

    time(&t);
    printf("%d\n%s", t, ctime(&t));
}
```

- n The output on all our Unix, Windows2000 and Pierre's MacOS X platform will be like:

```
1016645238
Wed Mar 20 18:27:18 2002
```



## Time Formats (Unix)

- n On Unix or MacOS X to get 1 micro second theoretical resolution there is:

```
#include <time.h>
struct timeval t;

gettimeofday(&t,0);
printf("%d.%06d\n", t.tv_sec, t.tv_usec);
```

- n Or on Unix for 1 nano second theoretical resolution:

```
#include <time.h>

struct timespec posix;
clock_gettime(CLOCK_REALTIME, &posix);
printf("%d.%09d\n", posix.tv_sec, posix.tv_nsec);
```



## Time Formats (Windows2000)

- n On Windows2000 it is easy to get 1 millisecond theoretical resolution:

```
#include <sys/timeb.h>
#include <winsock2.h>

struct _timeb timebuffer;
_ftime( &timebuffer );
printf("%d.%03d\n", timebuffer.time, timebuffer.millitm);
```



## Time Formats (Java)

- n Java has a lot of support for dates. The call that gives results consistent with Unix and Windows2000 is in java.lang.System:
 

```
public static long currentTimeMillis()
```
- n It returns the current time in milliseconds in a 64 bit signed long. Dividing by 1000 gives the the number of seconds since the first of January 1970.



## Time Formats (Oracle)

- n Oracle currently uses local time strings to put time and date into the database and to display dates already in the data base. The function TO\_DATE() converts to an Oracle date and TO\_CHAR() converts back to a string.
- n In Winter the the conversion of a Unix time in seconds to Oracle date is:
 

```
TO_DATE('01/01/1970 00:00:00', 'MM/DD/YYYY HH24:MI:SS') + unix_seconds/86400 + 1/24;
```
- n In Summer it is:
 

```
TO_DATE('01/01/1970 00:00:00', 'MM/DD/YYYY HH24:MI:SS') + unix_seconds/86400 + 2/24;
```
- n It is however very difficult to know this 1 or 2 hour offset in the Oracle SQL language.
- n Also there are no fractional seconds
- n The range of an Oracle date is from around 4000 BC to 9999 AD!



## Time Formats (Oracle 9i)

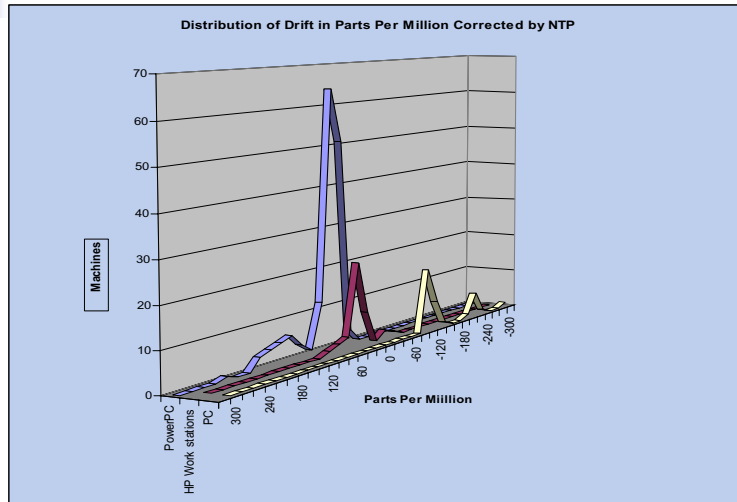
- n DATE:  
Data type which contains the datetime fields YEAR, MONTH, DAY, HOUR, MINUTE, and SECOND. It does not have fractional seconds and no time zone.
- n TIMESTAMP:  
Data type which contains the datetime fields YEAR, MONTH, DAY, HOUR, MINUTE, and SECOND. It has fractional seconds and no time zone.
- n TIMESTAMP WITH TIME ZONE (TSTZ):  
Data type which contains the datetime fields YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE\_HOUR, and TIMEZONE\_MINUTE. It has fractional seconds and an explicit time zone.
- n TIMESTAMP WITH LOCAL TIME ZONE (TSLTZ):  
Data type which has the storage in the same format as TIMESTAMP. The difference from TIMESTAMP is:
  - n 1. Data is normalized to a database time zone when stored in Oracle database.
  - n 2. When the data is selected by users, the value will be adjusted to the session time zone.



## How Computers keep time

- n When the Operating system is not running the time is kept, typically at 1 second resolution, by a Real Time Clock (RTC) chip driven from a battery. This is used when:
  - n Power is off
  - n The machine is booting
  - n The machine has stopped working!
- n When the operating system starts it read the RTC and uses the time as the starting value for time.
- n From then on this time value is incremented, typically every 10 milliseconds, by a periodic interrupt from a timer/counter chip.
- n For more time resolution there are processor cycle counters in Pentiums (1 nano-second resolution for 1 GHz Pentium) and the Decrementer on a PowerPC.
- n The quartz crystal oscillators used in computers are chosen for their cheapness and do not keep time as well as your watch!

# Measured and Corrected Clock Drift in SL Control System



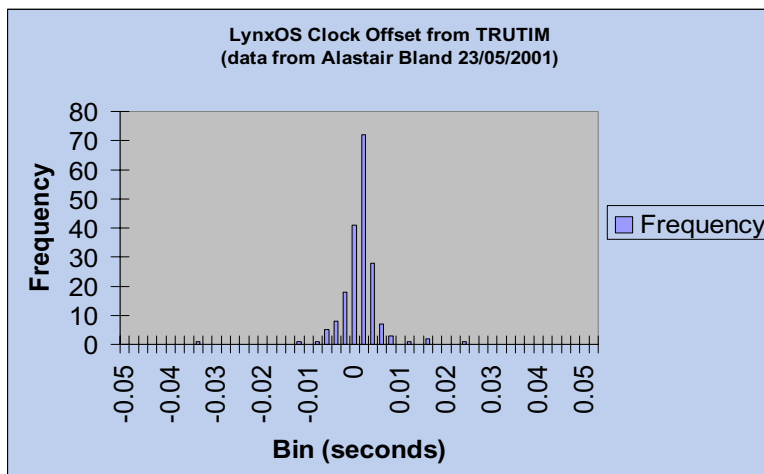
# Network Time Protocol (1)

- n The Network Time Protocol has been used since the early 1990s to keep many CERN computers (including the SL Control System computers) synchronized. The Control System synchronizes from the IT division servers IP-TIME-1 and IP-TIME-2 plus TRUTIM, our own GPS/IRIG-B based server.
- n More recently it has been available on Windows2000, provided by the W32time service which implements the Simple Network Time Protocol. Since a month ago the NICE2000 machines synchronize their time via the Domain Controllers IP-TIME-1 and IP-TIME-2. A great improvement!
- n Internally it is based on a 32 bit unsigned long number of seconds starting 1st of January 1900. It will overflow in 2036. It has 200 picosecond resolution.
- n Theory:
  - n Clock errors are due to variations in network delay and latencies in computer hardware and software (jitter), as well as clock oscillator instability (wander). The time of a client relative to its server can be expressed

$$T(t) = T(t_0) + R(t - t_0) + 1/2 D(t - T_0)^2$$

- n where  $t$  is the current time,  $T$  is the time offset at the last measurement update  $t_0$ ,  $R$  is the frequency offset and  $D$  is the drift due to resonator ageing. All three terms include systematic offsets that can be corrected and random variations that cannot. NTP estimates the first two terms.

# Network Time Protocol (2)



# PLCs

- n Software in the PLCs can read time, normally with fractions of a second. Usually local time has been used.
- n Schneider PLCs can have the time set by MODBUS TCP calls from the outside. They can also synchronize to a Pulse Per Second or Minute hardware input.
- n Siemens PLCs also have facilities for hardware inputs to be used for synchronizing the time with great precision.
- n Work is under way in LHC/IAS to use NTP or SNTP to get the time in Schneider and Siemens PLC from our usual network sources. However the maximum precision of NTP is limited to about half the network round trip time:
  - n 0.5 milliseconds on a fast local network
  - n 5 milliseconds when going a long distance through many routers.



# Conclusions

---

- n A coherent use of time throughout the CERN Control Systems will allow better applications and diagnostics to be made for our Operators and Equipment Specialists.
- n One GPS source (with spare) can be used as the reference. Other GPS receivers must be used to check distribution through the Control System.
- n The recommendation here is to use:
  - n Hardware pulses or Machine Timing for precision finer than a millisecond
  - n NTP for general computer synchronization
- n An obvious choice of format looking from the middle layers of the Control System is time in seconds since the 1<sup>st</sup> of January 1970. However some may disagree and say that for example PLCs should send data directly in Oracle string date format! This has the disadvantage that only Oracle can easily perform calculations on these strings ...