

Configuration of the LAT in Building 33:

- Current situation
- An adiabatic way forward...



The world's first titanium hammer.

Goodbye carpenter's elbow.
This hammer weighs in at only 12 oz. and delivers the same punch as a 21 oz. framing hammer. Call it carpenter's elbow, call it tendonitis or just a sore arm and back; if you swing a hammer all day, you're going to hurt. But Stiletto's titanium hammer will help alleviate repetitive motion injuries and reduce muscle and joint soreness. The secret to this hammer is that you get the same mass at half the weight. Designed with a magnetic nail holder and a straight handle, this hammer is revolutionary!

Titanium Hammer with straight handle
#99943 \$79.99

Powerful magnetic nail slot in the head holds nail securely in place.

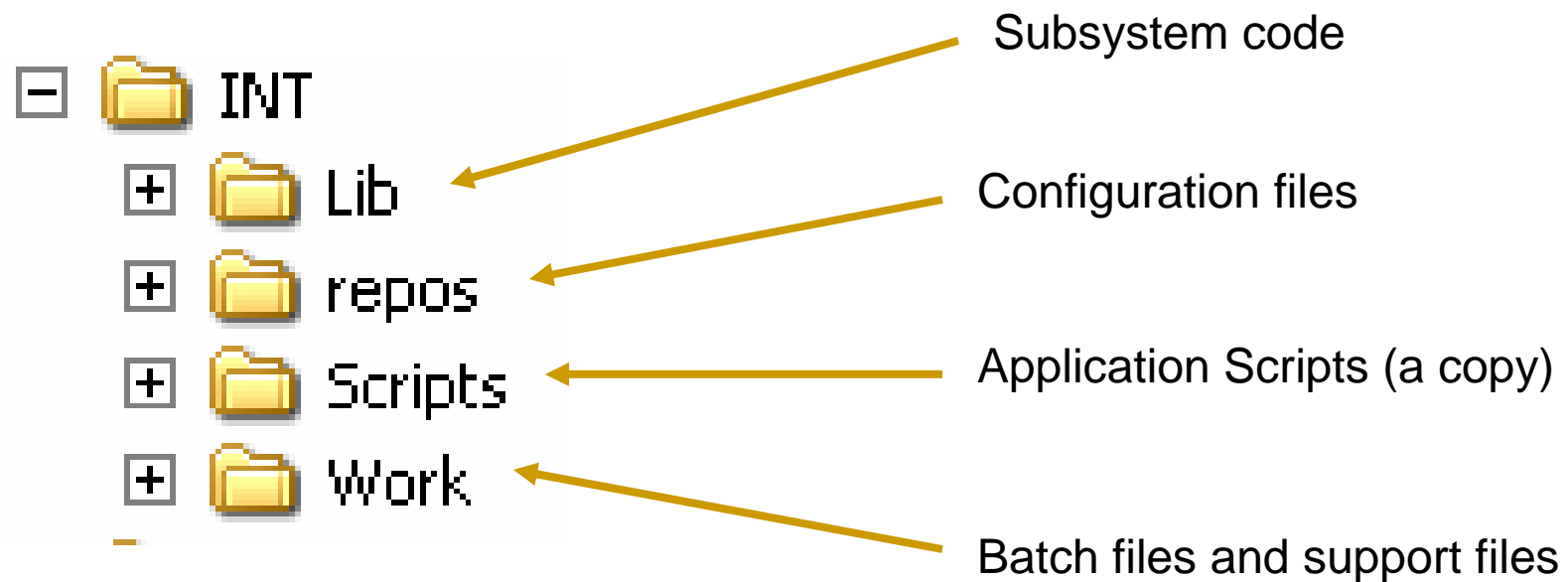
NEW

16 DULUTH TRADING COMPANY

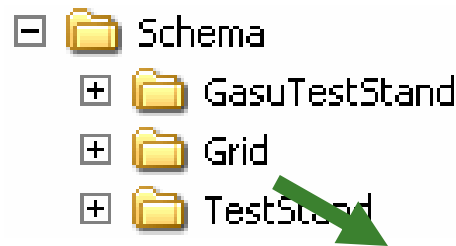
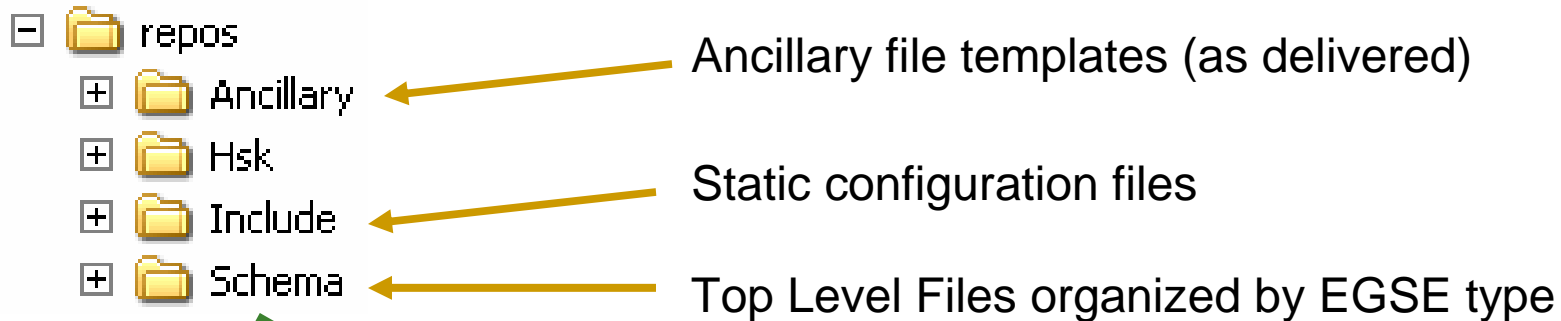
“I gotta get me one of them”

INT releases

- Contain subsystem code
- Responsible for configuration of the LAT
- Building 33 batch files and used Scripts



Configuration is done via Schema files



Schema file name uniquely identifies content of schema

- Grid-0_Tower-0.xml
- Grid-0_TkrFMA.xml
- Grid-0_FM104.xml
- Grid-0-4_Tower-0-4_UutTkrFMB.xml
- Grid-0-4_Tower-0-4_UutTkrFMA.xml
- Grid-0-4_Tower-0-4_Pdu1_Rdnt.xml
- Grid-0-4_Tower-0-4.xml
- Grid-0-4_Tem-0-4_Pdu1_Rdnt.xml
- Grid-0-4_Tem-0-4_Pdu1_Prim.xml

Schema files at top level have 5 sections

Declares existence
Of hardware

```
<!-- Include copyright.xml -->  
- <LATdoc name="Grid-0-1-4-5_Tower-0-1-4-5">  
  <!-- The schema -->  
  + <schema>
```

Static configuration

```
    <!-- configurations will be applied in order given  
    + <configuration version="" name="baseConfig">  
    <!-- ancillary includes -->
```

Ancillary configuration

```
    + <configuration version="" name="ancillaryConfig">
```

Eng. Unit
conversions

```
    <!-- EGU blocks -->  
    <include filename="$INCLUDE_ROOT/Egu/egu.xml" />  
    <!-- The FSW flags -->  
    + <flags>  
  </LATdoc>
```

Flight software
configuration

These configurations are applied by LATTE before the LAT is handed to the script. The script takes it from there

Schema files are basically mappings to include files

“Include”s

```
- <configuration version="" name="baseConfig">
+ <declarations>
- <GLAT>
+ <!-- -->
  <include filename="$INCLUDE_ROOT/Gasu/configGEM.xml" />
  <include filename="$INCLUDE_ROOT/Gasu/configAEM.xml" />
- <GTEM ID="0">
  <configuration>0x0</configuration>
  <!-- TKR sources enabled, diagnostic on -->
  <data_masks>0x1000</data_masks>
  <!-- <address>0x0</address> -->
+ <GTIC>
  <!-- serial numbers calInstrument and tkrInstrument used to get Ancillary paths -->
  <serialnos tkr="TkrFMA" calInstrument="FM104" tem="GLAT1749" tps="GLAT1746" temtps="GLAT1752" />
  <include filename="$INCLUDE_ROOT/TkrFMA/configTkrFMA.xml" />
  <include filename="$INCLUDE_ROOT/FM104/configFM104.xml" />
</GTEM>
+ <GTEM ID="1">
+ <GTEM ID="4">
+ <GTEM ID="5">
  <serialnos pdu="GLAT0894" gasu="GLAT1338" />
</GLAT>
</configuration>
```

Serial number tags (and include file names)
map hardware to TEM address

All configuration below top level is based on Hardware serial number

Schema sections:

- Schema: declares hardware existence
 - details are included by hardware serial number (“include schemaTkrFMA.xml”)
 - Base configuration: static configuration
 - Default settings of registers of hardware (as delivered by subsystem)
 - Included from configuration controlled area referred to as “INCLUDE_ROOT”
 - Ancillary configuration: dynamic configuration settings
 - Settings which are the results of calibrations (online or offline)
 - Included from non-controlled area
 - Files are seeded from INT release Ancillary area with delivered quantities as hardware is added
 - EGUs: Engineering unit conversions. Single included file
 - Currently static, so controlled
 - Could become calibrated per TEM
 - FSW: flight software flags. No “include”s here. This is top level
 - Specify which PDU
 - Specify primary or redundant GASU
 - Clock edge setting, etc.
-

Ancillary configuration:

```
<include filename="$ANCILLARY_ROOT/topLevel/latest_ext_delay_Tower0.xml" />
- <GTEM ID="0">
  <!-- cal part -->
  <include filename="$ANCILLARY_ROOT/FM104/latest_fle_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_fhe_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_lac_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_uld_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_trg_alignment_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_CAL_TACK_delay_FM104.xml" />
  <!-- tracker part -->
  <include filename="$INCLUDE_ROOT/TkrFMA/TkrSerials_TkrFMA.xml" />
  <include filename="$INCLUDE_ROOT/TkrFMA/TkrHW_TkrFMA.xml" />
  <!-- <include filename = '$INCLUDE_ROOT/TkrFMA/TkrHotStrips_TkrFMA.xml' /> -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_HotStrips.xml" />
  <!-- HT: updated on 1/28/05 -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_DeadStrips_latest.xml" />
  <include filename="$ANCILLARY_ROOT/TkrFMA/latest_trg_alignment_TkrFMA.xml" />
  <include filename="$ANCILLARY_ROOT/TkrFMA/latest_TKR_TACK_delay_TkrFMA.xml" />
  <!-- Ancillary files are commented out since these files do not exist initially. -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_Thresholds.xml" />
  <!-- <include filename = "$ANCILLARY_ROOT/TkrFMA/TkrThresholdCal_latest.xml" /> -->
- <opaque name="TkrChargeScale">
  <!-- <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_050202-074525.xml" /> -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_latest.xml" />
</opaque>
  <!-- opaque blocks -->
  <include filename="$INCLUDE_ROOT/FM104/opaqueFM104.xml" />
</GTEM>
```

Well, lets take this in smaller chunks

Ancillary timing information

```
<include filename="$ANCILLARY_ROOT/topLevel/latest_ext_delay_Tower0.xml" />
</GTEM ID=0 >
<!-- cal part -->
<include filename="$ANCILLARY_ROOT/FM104/latest_fle_FM104.xml" />
<include filename="$ANCILLARY_ROOT/FM104/latest_fhe_FM104.xml" />
<include filename="$ANCILLARY_ROOT/FM104/latest_lac_FM104.xml" />
<include filename="$ANCILLARY_ROOT/FM104/latest_uld_FM104.xml" />
<include filename="$ANCILLARY_ROOT/FM104/latest_trg_alignment_FM104.xml" />
<include filename="$ANCILLARY_ROOT/FM104/latest_CAL_TACK_delay_FM104.xml" />
<!-- tracker part -->
<include filename="$INCLUDE_ROOT/TkrFMA/TkrSerials_TkrFMA.xml" />
<include filename="$INCLUDE_ROOT/TkrFMA/TkrHW_TkrFMA.xml" />
<!-- <include filename = '$INCLUDE_ROOT/TkrFMA/TkrHotStrips_TkrFMA.xml' /> -->
<include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_HotStrips.xml" />
<!-- HT: updated on 1/28/05 -->
<include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_DeadStrips_latest.xml" />
<include filename="$ANCILLARY_ROOT/TkrFMA/latest_trg_alignment_TkrFMA.xml" />
<include filename="$ANCILLARY_ROOT/TkrFMA/latest_TKR_TACK_delay_TkrFMA.xml" />
<!-- Ancillary files are commented out since these files do not exist initially. -->
<include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_Thresholds.xml" />
<!-- <include filename = "$ANCILLARY_ROOT/TkrFMA/TkrThresholdCal_latest.xml" /> -->
- <opaque name="TkrChargeScale">
  <!-- <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_050202-074525.xml" /> -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_latest.xml" />
</opaque>
<!-- opaque blocks -->
<include filename="$INCLUDE_ROOT/FM104/opaqueFM104.xml" />
</GTEM>
```

Source of these is trigger group online scripts
Control of TACK_DELAY, trigger primitive alignment, and
external trigger delay.

CAL Ancillary information

```
<include filename="$ANCILLARY_ROOT/topLevel/latest_ext_delay_Tower0.xml" />
- <GTEM ID="0">
  <!-- real part -->
  <include filename="$ANCILLARY_ROOT/FM104/latest_fle_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_fhe_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_lac_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_uld_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_trg_alignment_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_CAL_TACK_delay_FM104.xml" />
  <!-- tracker part -->
  <include filename="$INCLUDE_ROOT/TkrFMA/TkrSerials_TkrFMA.xml" />
  <include filename="$INCLUDE_ROOT/TkrFMA/TkrHW_TkrFMA.xml" />
  <!-- <include filename = '$INCLUDE_ROOT/TkrFMA/TkrHotStrips_TkrFMA.xml' /> -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_HotStrips.xml" />
  <!-- HT: updated on 1/28/05 -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA/TkrFMA_DeadStrips_latest.xml" />
  <include filename="$ANCILLARY_ROOT/TkrFMA/latest_trg_alignment_TkrFMA.xml" />
  <include filename="$ANCILLARY_ROOT/TkrFMA/latest_TKR_TACK_delay_TkrFMA.xml" />
  <!-- Ancillary files are commented out since these files do not exist initially. -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_Thresholds.xml" />
  <!-- <include filename = "$ANCILLARY_ROOT/TkrFMA/TkrThresholdCal_latest.xml" /> -->
- <opaque name="TkrChargeScale">
  <!-- <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_050202-074525.xml" /> -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_latest.xml" />
</opaque>
  <!-- opaque blocks -->
  <include filename="$INCLUDE_ROOT/FM104/opaqueFM104.xml" />
</GTEM>
```

- Discriminator settings source is INT noise floor algorithm (exception: ULD is delivered default)
- “Opaque” blocks are delivered AFEE serial number information (Note include root)

TKR Ancillary inputs

```
<include filename="$ANCILLARY_ROOT/topLevel/latest_ext_delay_Tower0.xml" />
- <GTEM ID="0">
  <!-- cal part -->
  <include filename="$ANCILLARY_ROOT/FM104/latest_fle_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_fhe_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_lac_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_uld_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_trg_alignment_FM104.xml" />
  <include filename="$ANCILLARY_ROOT/FM104/latest_CAL_TACK_delay_FM104.xml" />
  <!-- tracker part -->
  <include filename="$INCLUDE_ROOT/TkrFMA/TkrSerials_TkrFMA.xml" />
  <include filename="$INCLUDE_ROOT/TkrFMA/TkrHW_TkrFMA.xml" />
  <!-- <include filename = '$INCLUDE_ROOT/TkrFMA/TkrHotStrips_TkrFMA.xml' /> -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_HotStrips.xml" />
  <!-- HT: updated on 1/28/05 -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_DeadStrips_latest.xml" />
  <include filename="$ANCILLARY_ROOT/TkrFMA/latest_trg_alignment_TkrFMA.xml" />
  <include filename="$ANCILLARY_ROOT/TkrFMA/latest_TKR_TACK_delay_TkrFMA.xml" />
  <!-- Ancillary files are commented out since these files do not exist initially. -->
  <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_Thresholds.xml" />
  <!-- <include filename = "$ANCILLARY_ROOT/TkrFMA/TkrThresholdCal_latest.xml" /> -->
  - <opaque name="TkrChargeScale">
    <!-- <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_050202-074525.xml" /> -->
    <include filename="$ANCILLARY_ROOT/TkrFMA/TkrFMA_TkrChargeScale_latest.xml" />
  </opaque>
  <!-- opaque blocks -->
  <include filename="$INCLUDE_ROOT/FM104/opaqueFM104.xml" />
</GTEM>
```

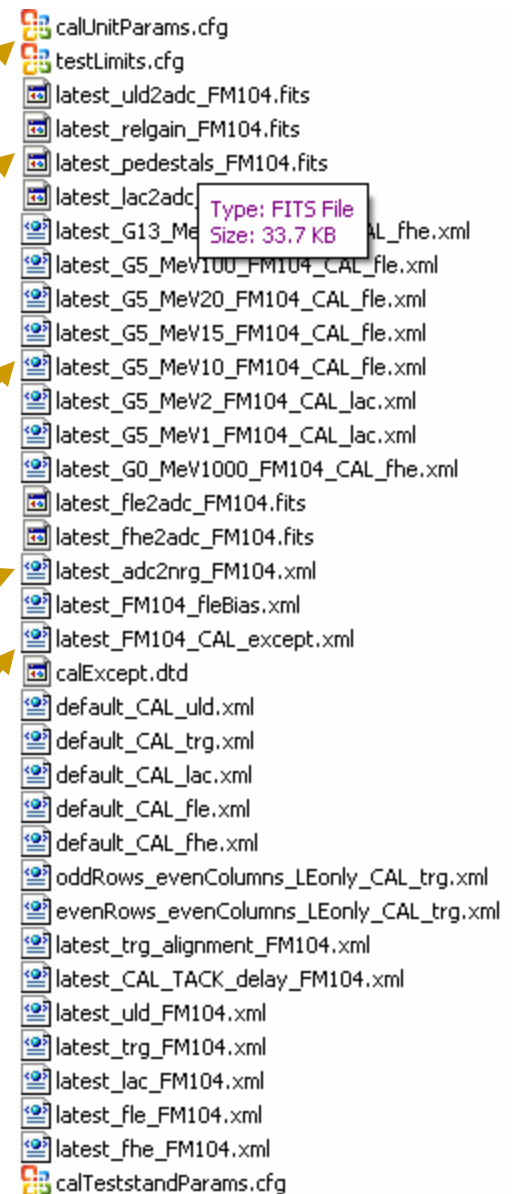
- Serials and HW are delivered (note include root), MCMs, etc.
- Hot strips from online script TkrNoiseOccupancy
- Dead strips from offline and online TkrNoiseAndGain
- Thresholds from online TkrThresholdCal
- ChargeScale from offline (used by algorithms – not LATTE)

But wait, there's more!

Other CAL Ancillary:

- CAL algorithm setup and test definitions: delivered
- Pedestal and gain tables for CAL online algorithms: created online
- Prepared threshold settings by energy for INT scripts: created offline and by CAL scripts dynamically (!)
- Conversion constants for threshold generation by CAL scripts: created online by calibDAC except adc2nrg
- Unit exception list for CAL algorithms: delivered

This means that LAT configuration by scripts is also influenced by the Ancillary area



Weaknesses of this system

- Schema top level files
 - They multiply like rabbits. Releases roll, mistakes are made, time is lost in rolling revs
 - Maintenance is difficult (“I want default flight software flags set to X”: I change 50 files)
 - Ancillary area
 - Control lost because of need for flexibility (ability to pick up new results via overwrite of “latest”)
 - Can’t reconstruct history of what has happened
 - Some attempts here to rectify: all algorithms produce timestamped copy of a file in addition to overwriting “latest”
 - Format changes of files are very difficult to deal with (edit by hand?)
 - Manual modification is tempting and has been used
 - Files not standard (e.g. not all identify themselves by content)
-

Two proposed projects to help

- Top level schema file generator
 - A tool to generate (and regenerate) the top level files
 - Ancillary area file metadatabase
 - A database to track “latest” file identities and changes
-

Schema file generator

- Schema file names unique to content
 - The only essential function is mapping hardware to TEM Id
- Use a template to create file based on bays requested and FSW flags requested
- It's easy to imagine the interface to do this...
- See LAT-TD-6637



Grid-0-4_Tem-0-4_Pdu1_Rdnt.xml

Schema file generator: migration

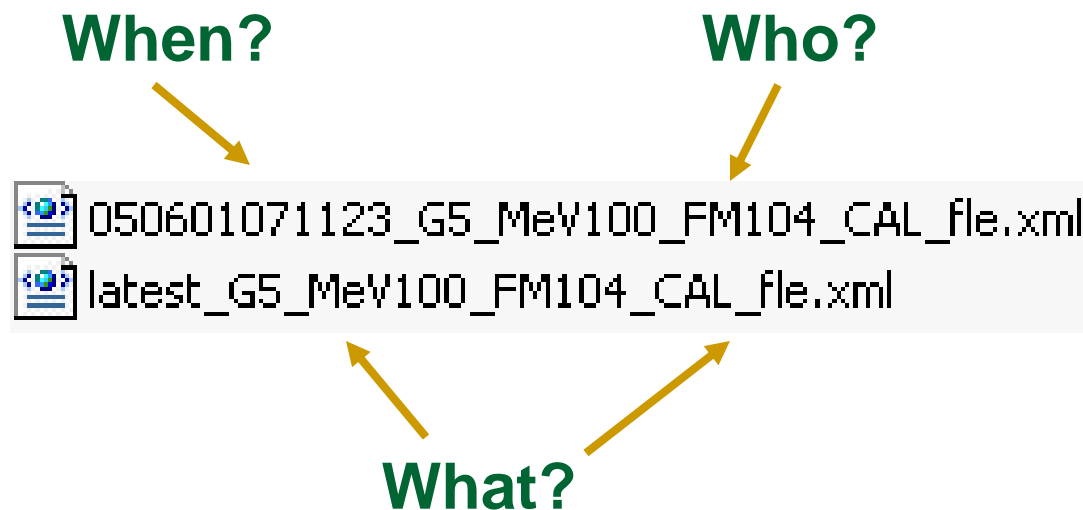
- First: Tool used to manage cvs release files (schema files controlled)
 - Second: Tool used to create files as needed by INT
 - Control migrated from schema files to tool
 - Third: tool migrated to LATTE interface
 - No more top level schema files as we know it
 - A database would hold the mapping of HW serial numbers to bay location
-

Ancillary file metadatabase

- Each ancillary file generically has a standard name to identify the “current” one and a timestamped copy
 - Standard name used by schema file and algorithms to know where to look
 - Timestamped copy is for record keeping, stored in Ancillary directory of unit with the standard files
 - Metadatabase will keep track of mapping of time-stamped files to standard files at any given time
 - Each writing “interaction” to the Ancillary area will create an entry in this database
 - Database tool interfaces to scripts writing to the ancillary area, and has “hand” interface for new unit population and/or special modifications
 - Database can recreate complete set of standard files at any time
-

Ancillary Metadatabase migration

- First: Normalize file naming
 - Timestamp, file type, unit SN
 - (Embarrassingly few conform at this level currently)
 - Also, make all files content-identifying...



Ancillary metadatabase migration

- Second: Tool for interacting with database
 - Interface needs to be flexible enough to deal with unforeseen manual changing capability
 - Third: Tool becomes “owner” of ancillary filesystem
 - Introduces control to area: tool is gatekeeper
 - Once control of ancillary area is established, we can move “static” configuration files there as well
-

Ancillary metadatabase migration

- Fourth: Migrate file functionality to actual database tables
 - These files are effectively database queries, they become the configuration database...



```
050601071123_G5_MeV100_FM104_CAL_file.xml  
latest_G5_MeV100_FM104_CAL_file.xml
```

“**from** thresholds **select** file **where** unit=FM104 and gain=5 and energy=100 **order by** timestamp”

Summary

- The current situation in Building 33 for LAT configuration probably has some useful lessons learned
 - That situation and the view forward presented here is driven by very prosaic, pragmatic concerns as opposed to elegant (or even conscious) planning
-