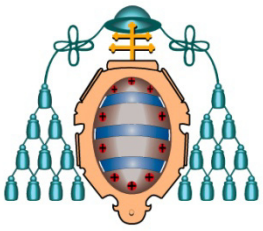
Coding in ConfDB

Javier Fernandez

TOP Trigger Tutorial
2nd October 2013



UNIVERSIDAD DE OVIEDO

What's confDB?

- The HLT configuration in CMS is stored in a dedicated database system, **ConfDB**
  https://twiki.cern.ch/twiki/bin/view/CMS/EvfConfDBDatabase
  - ConfDB models the CMSSW job configuration grammar into a relational database schema
    - Only Oracle is supported (MySQL is not anymore!!)
  - CMSSW components (classes/plugins) are entered in the DB for a given (pre-)release as "Templates" by the ConfDB Parser
    - Administration task
  - Configurations can be entered/edited in the DB with the ConfDB GUI
    - based on a release and its templates
- Existing configurations can be retrieved from the Database as a formatted document (ASCII/Python/HTML) via the ConfDB Converter
  - Run Control Component to deploy HLT configuration to Filter Farm online
  - Web Interface to serve offline requests of any kind & GUI
- Use the dedicated GUI to create, manipulate and store trigger path configurations
  https://twiki.cern.ch/twiki/bin/viewauth/CMS/EvfConfDBGUI

This is what this tutorial is about mainly!!

- Use the ConfDB web browser to browse the content of the configuration database and inspect available HLT menus. http://www.cern.ch/confdb/browser/
- The command-line tool edmConfigFromDB allows you to retrieve (complete or partial) configurations as either ascii or python configuration files to be fed to cmsRun
  https://twiki.cern.ch/twiki/bin/view/CMS/EdmConfigFromDB

Configuration Database Design

- The code of the ConfDB project is in the CMSSW cvs repository in the package EventFilter/ConfigDB.
  https://twiki.cern.ch/twiki/bin/view/CMS/EvfConfDBDesign

- The Database-driven configuration system for High Level Trigger (CMSSW) job configurations, *ConfDB*, is based on four components:

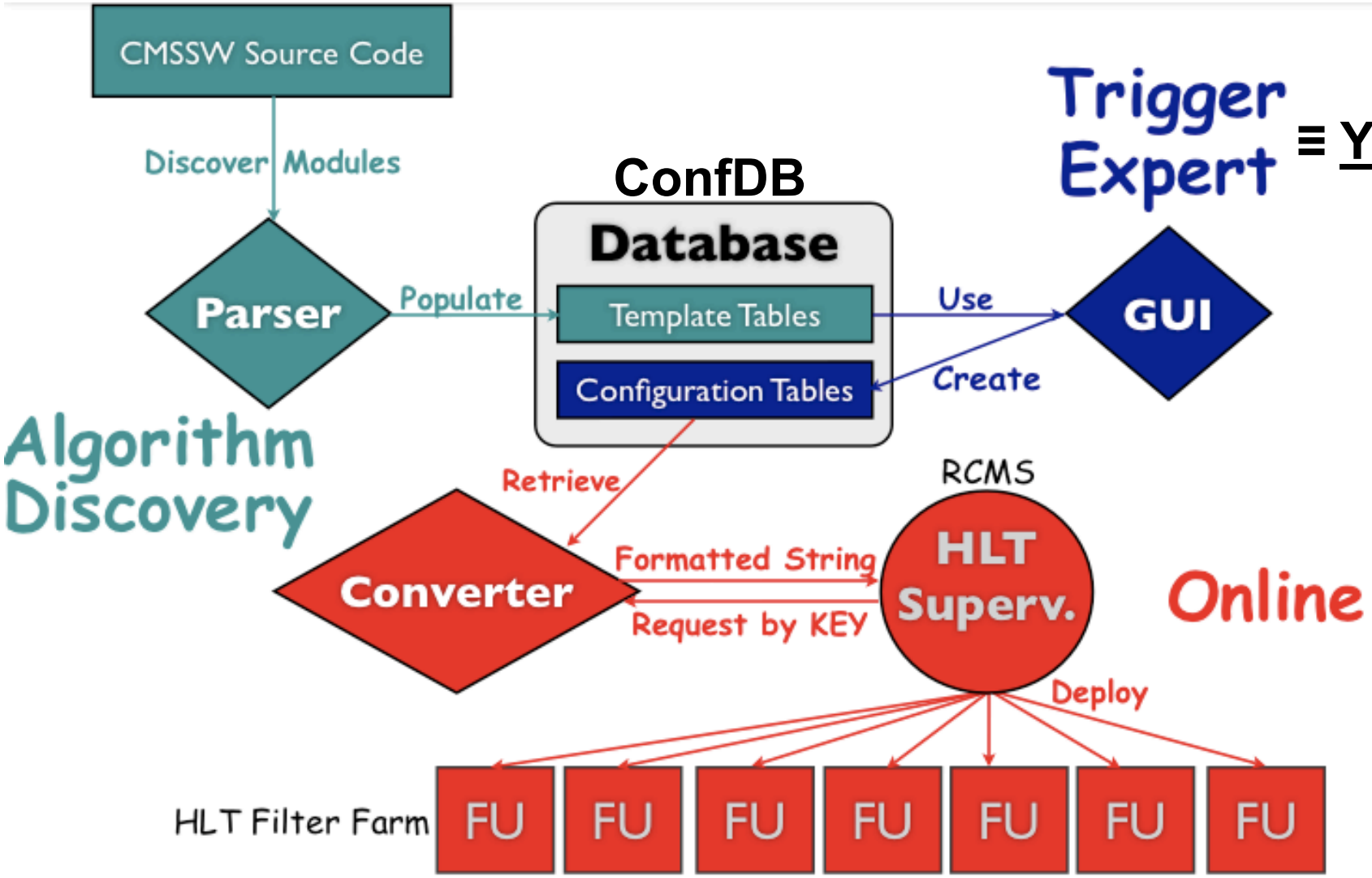
  – Database. Describes the relations between Services, Paths, Sequences, Modules, and Parameters. The schema is provided, tested, and maintained for Oracle.

  – Parser. Each HLT job configuration is related to one specific release of the CMSSW software. The algorithms and modules provided in that release are the building blocks of each configuration. The parser traverses the complete source tree of a chosen CMSSW release and discovers all available modules and, if feasible, their default configuration. The discovered software components are referred to as *templates*; when one of these is used in a configuration, an *instance* of a template is created and related to its mother template.

  – GUI. A graphical user interface is provided to generate an HLT configuration based on a chosen CMSSW release and its templates. The configuration is displayed in a tree like structure, Paths and Modules can be added, and Module Parameters can be changed. The creation of a valid configuration according to the rules of the CMSSW configuration system is enforced.

  – Converter. Once configurations are stored in relational database tables, a component is needed which can access the database and transform the information into other formats: a valid ASCII or Python job configuration file or string to configure cmsRun, a webpage, etc. The Converter is the API for such tasks, with implementations for each useful representation of a configuration.

Graphically



Caveat

- This tutorial will focus on how to develop a new trigger in ConfDB (using the GUI) and how to test it, so that it will be ready to be integrated into the master menu
- You will need JAVA 5.0 or later installed in your computer, see for details of the GUI:

  https://twiki.cern.ch/twiki/bin/viewauth/CMS/EvfConfDBGUI

- It will take you step-by-step through an example,
  - Save your work from time to time (with comments)
  - Try to work with a good connection to CERN…
- We start from /dev/CMSSW_6_2_0/GRun


hands
on

ConfDB GUI

You have two options to start the Java ConfDB GUI:

1. **click here to start the Java Web Start version (recommended!!)**
   - advantages:
     - you are automatically running the latest version
   - disadvantages:
     - **might not work at all!**
       With the recent upgrades of JavaWebStart security restrictions have been tightened on several platforms. You can lift these restrictions but it's probably easier to use method 2 to start the GUI

2. **click here to download an executable jar file which includes all classes needed.**
   - Double-clicking the downloaded jar file should start the GUI on many platforms.
     If not, start the GUI on a command line by entering:

     java -jar confdb-gui-all-inclusive.jar

     in the directory where the jar file can be found

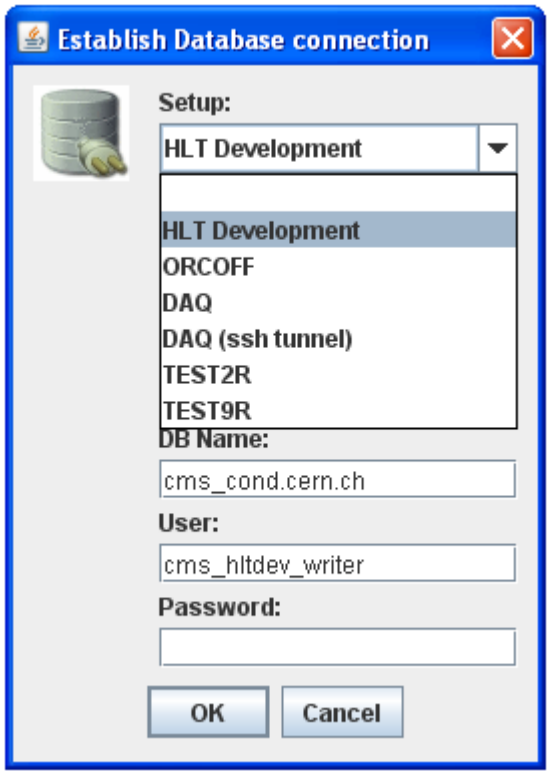
   - advantages:
     - you can run the GUI locally like any other (Java) program
   - disadvantages:
     - it takes longer to start the GUI if you download the jar file each time you want to run the program
     - you might not run the latest version if you do not download the jar file each time you start the GUI

Starting the GUI

- Start the ConfDB GUI

http://www.cern.ch/confdb/gui/start.jnlp

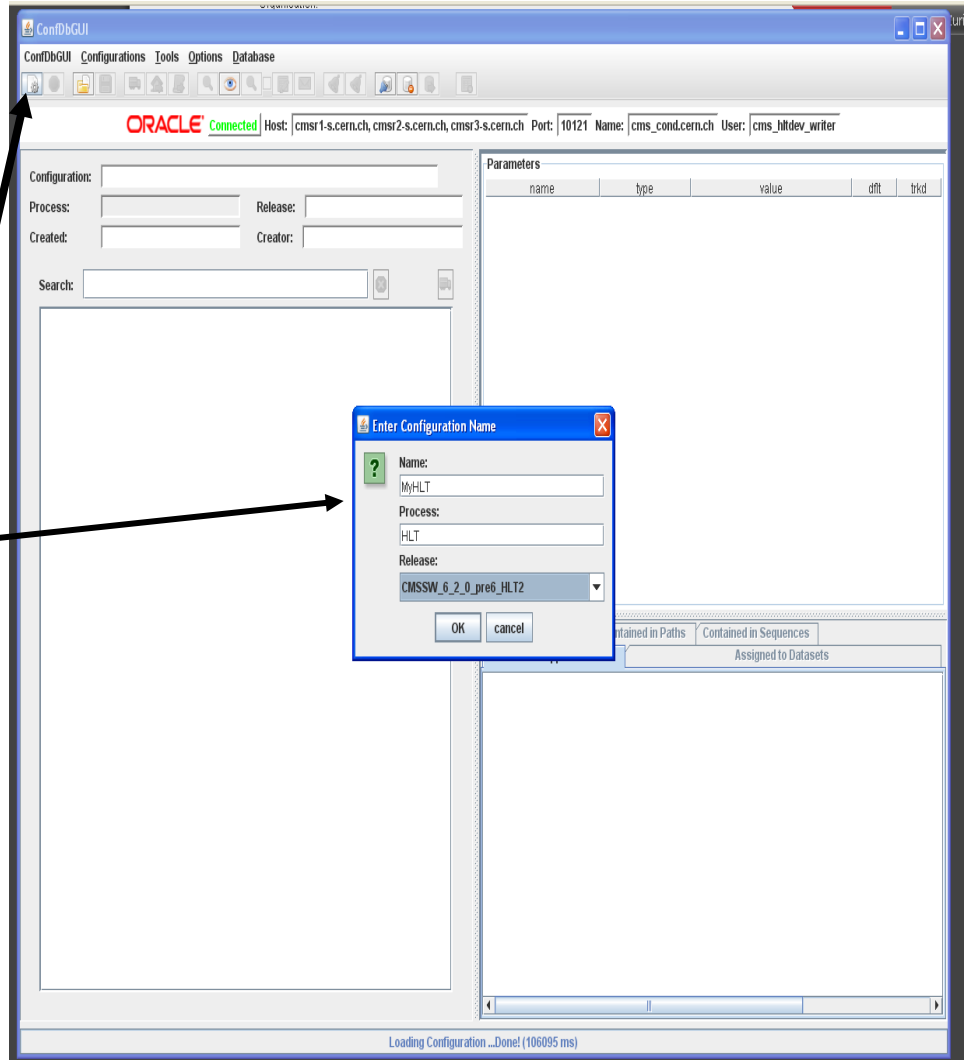
- Starting the GUI
  - Connect to the database: Under Setup, select HLT Development and enter the password …



Creating a New Configuration

Start with a blank configuration and import some path(s) from the master

- It will give you just a few paths you want to edit.
- We will focus on this method in this tutorial
- Create a new blank configuration
  - Enter some Name for your configuration (“MyHLT” in the example), the Process is “HLT”, and in general, you should choose the newest Release in the master (“CMSSW_6_2_0_pre6_HLT2” at the moment)
  - You can check the release on the HLT Global Table twiki page:



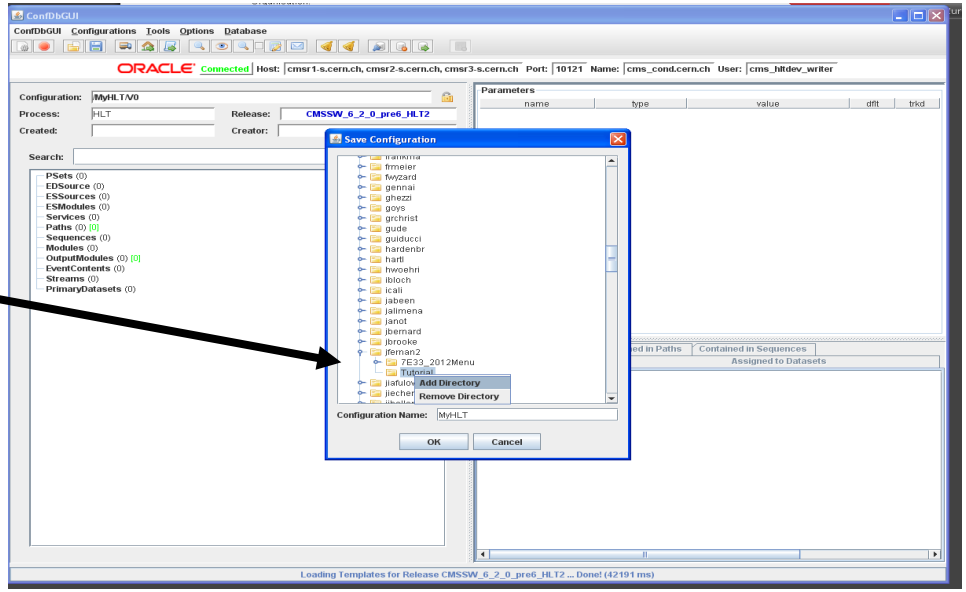
https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideGlobalHLT

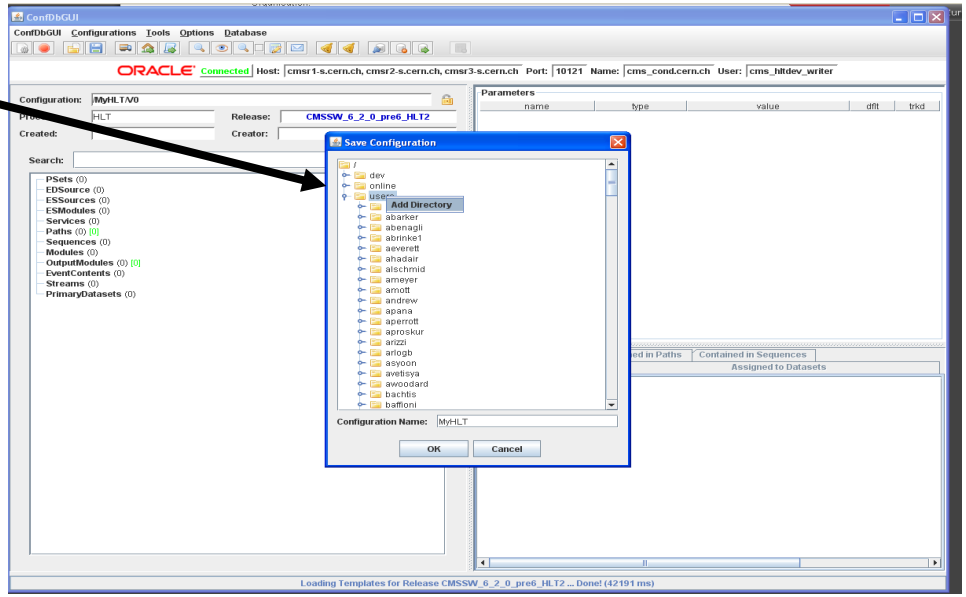
Saving Your Configuration

- You should save your new (and blank, at the moment) configuration in your /user area
  - Configurations →SaveAs →

/users/YourUserName/FolderName/MyHLT

  - You can create your user area if it doesn't exist in /users, and you can create new folders in your user area by right-clicking where you want to create the folder and selecting Add Directory





- Each time you save, your configuration will get a new version number. **Save often**, and **make comments** about each new save, describing the changes.

Mac NOTE: to emulate a right-click in the absence of a second mouse button, hold down the ctrl key and press the mouse button.

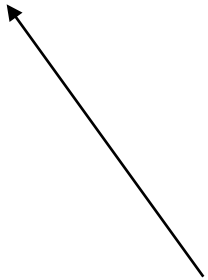
How to develop a new HLT path?

In this example:

- We start from 3 typical defined paths containing the objects/modules we want, and we will mix them:
  - HLT_**IsoMu24_eta2p1**_v16 (Our baseline, i.e. we modify it adding components)
  - HLT_**DiCentralPFNoPUJet50**_PFMETORPFMETNoMu100_v1
  - HLT_DiPFJet80_DiPFJet30_**BTagCSVd07d05**_v6

- We will modify the IsoMuon path and at the end rename it to:
  - HLT_**IsoMu25_eta2p1_CentralPFNoPUJet30_BTagCSVd07**_v1

- To construct a path with 3 basic objects:
  - **Isolated lepton**
  - **CHS (No PU) PFJets**
  - **B-Tagging**



- We will modify thresholds too (so, no typos in the path above ☺)

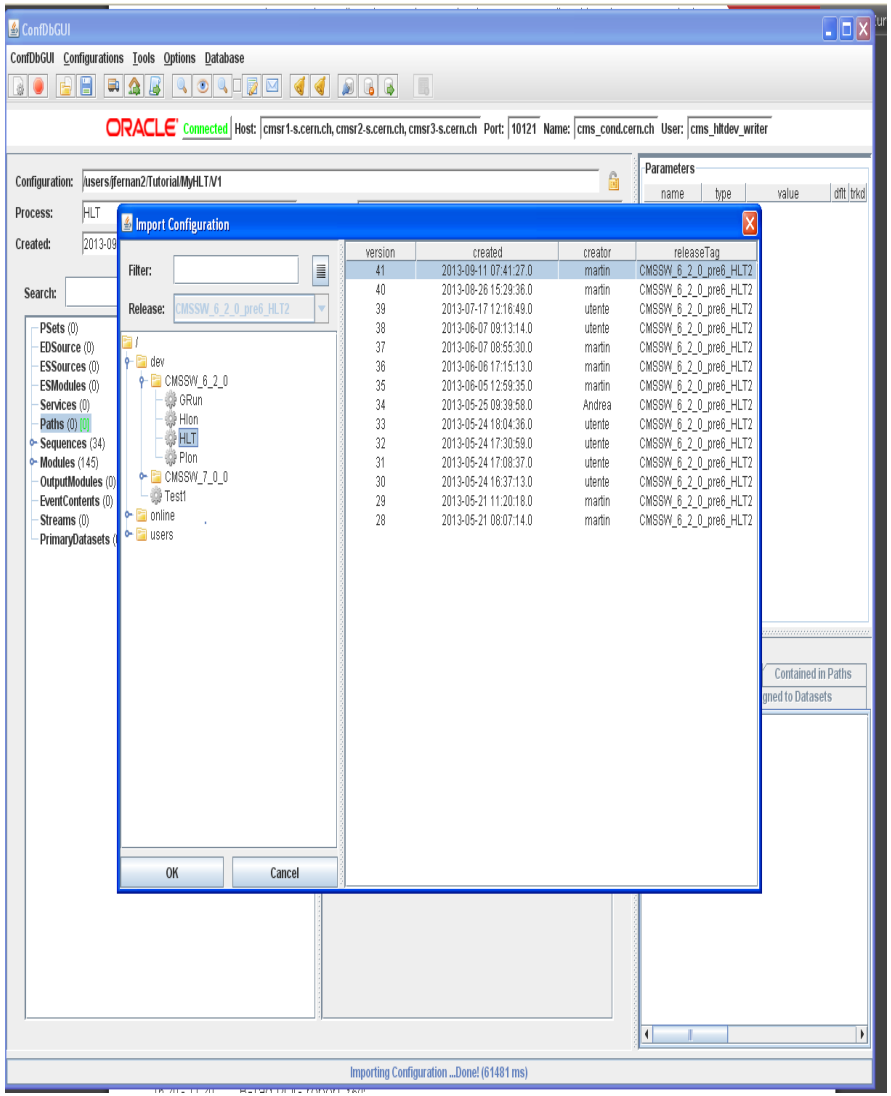
- See all the 2012 TOP HLT paths in: https://twiki.cern.ch/twiki/bin/view/CMS/TopTriggerMenu5E33

Importing paths from the master (I)

- Select Configurations →Import
- The master is kept in /dev,
- In our case:

/dev/CMSSW_6_2_0/HLT

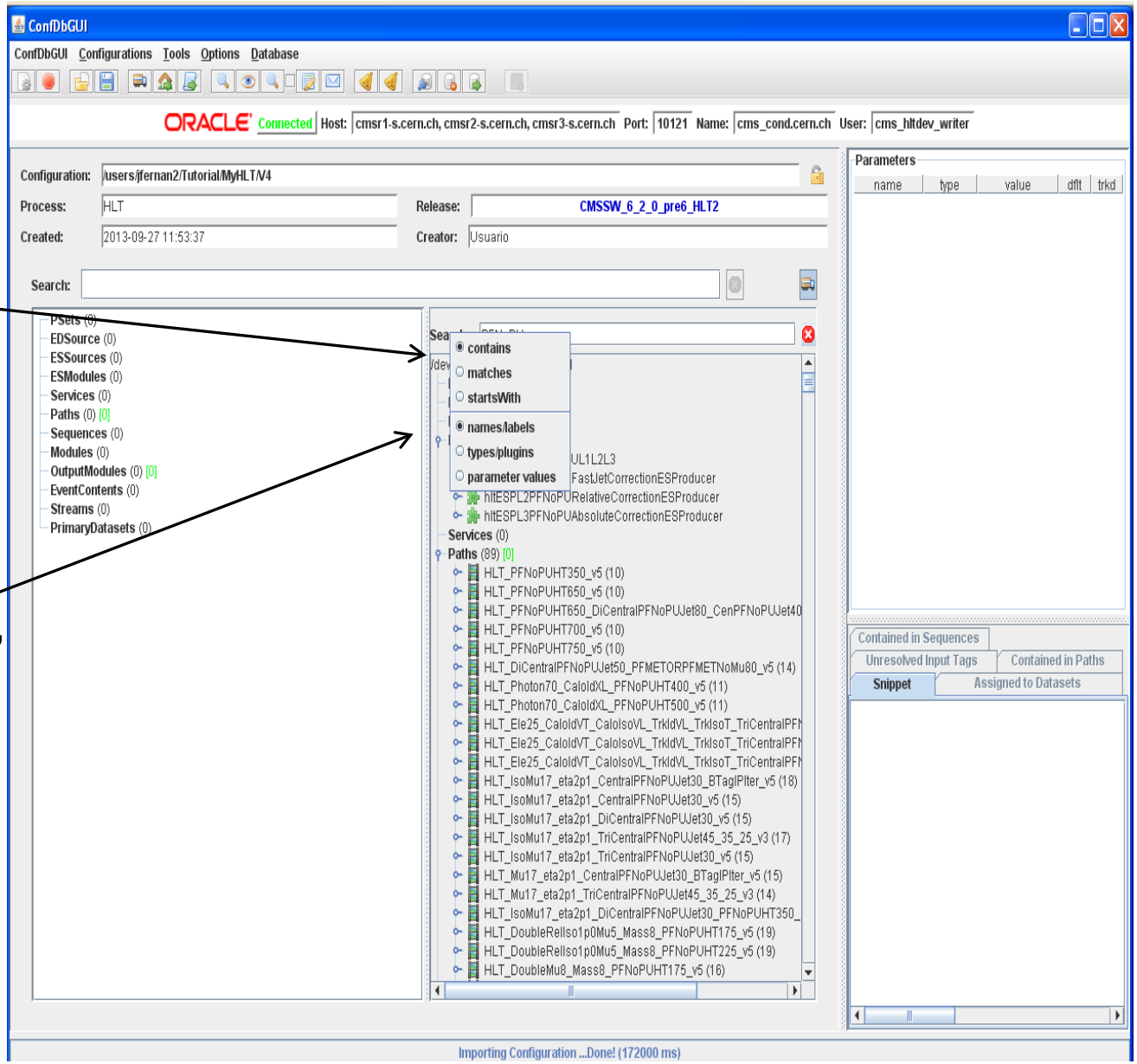
- Select this one and click OK



*__Notice__: You can only import from a configuration that is in the same release as your configuration (i.e. CMSSW_6_2_0).The others won't be available.

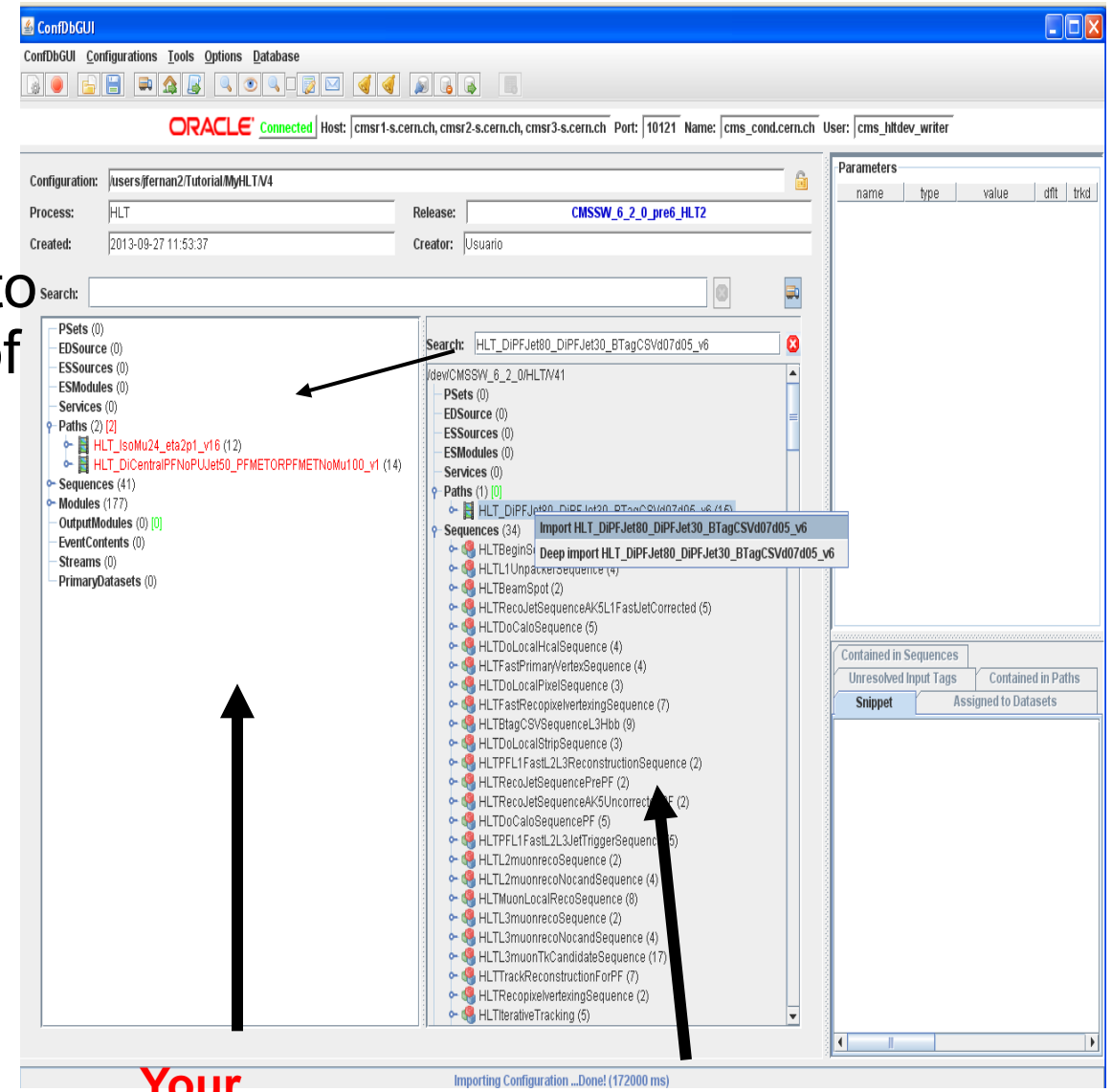
Importing paths from the master (II)

- Searching objects in the GUI
  - Play with the right click!
- Select the type of search:
  - Loose search: “contains”
  - “names/labels”
- Search the paths from previous slide, one by one…



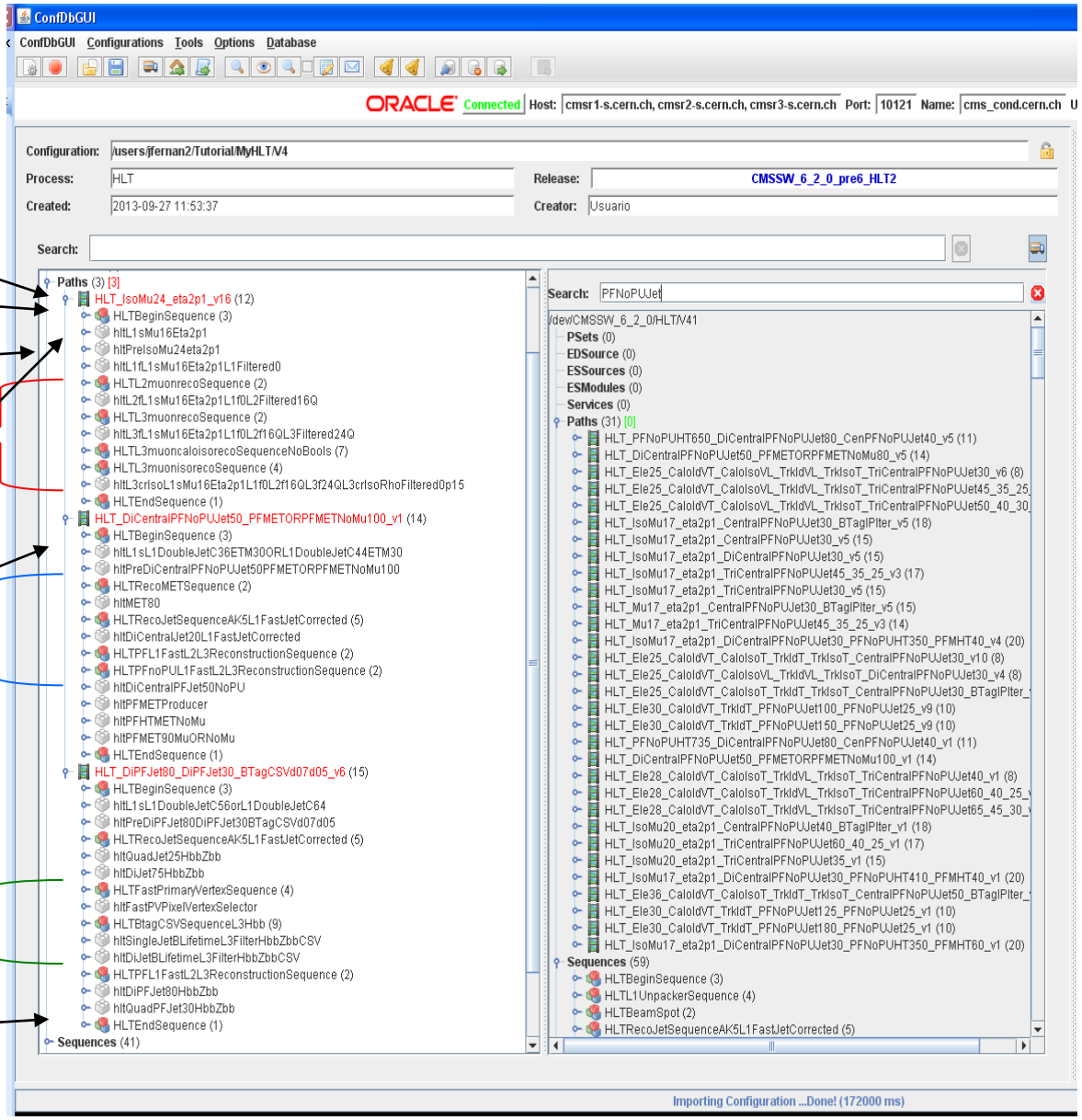
Importing paths from the master (III)

- Go to the paths section
- Use the search tool to restrict the number of objects
- Drag and drop the paths you want to import,
- or right-click and select **Import path_name**



Inspect what you have imported

- Note the different icons:
  - Paths
  - Sequences
  - Modules
- Identify and expand:
  - L1 seeds
  - L2/L3 μ **part**
  - **Jets**
  - **Btag**
  - Final L3 sequence

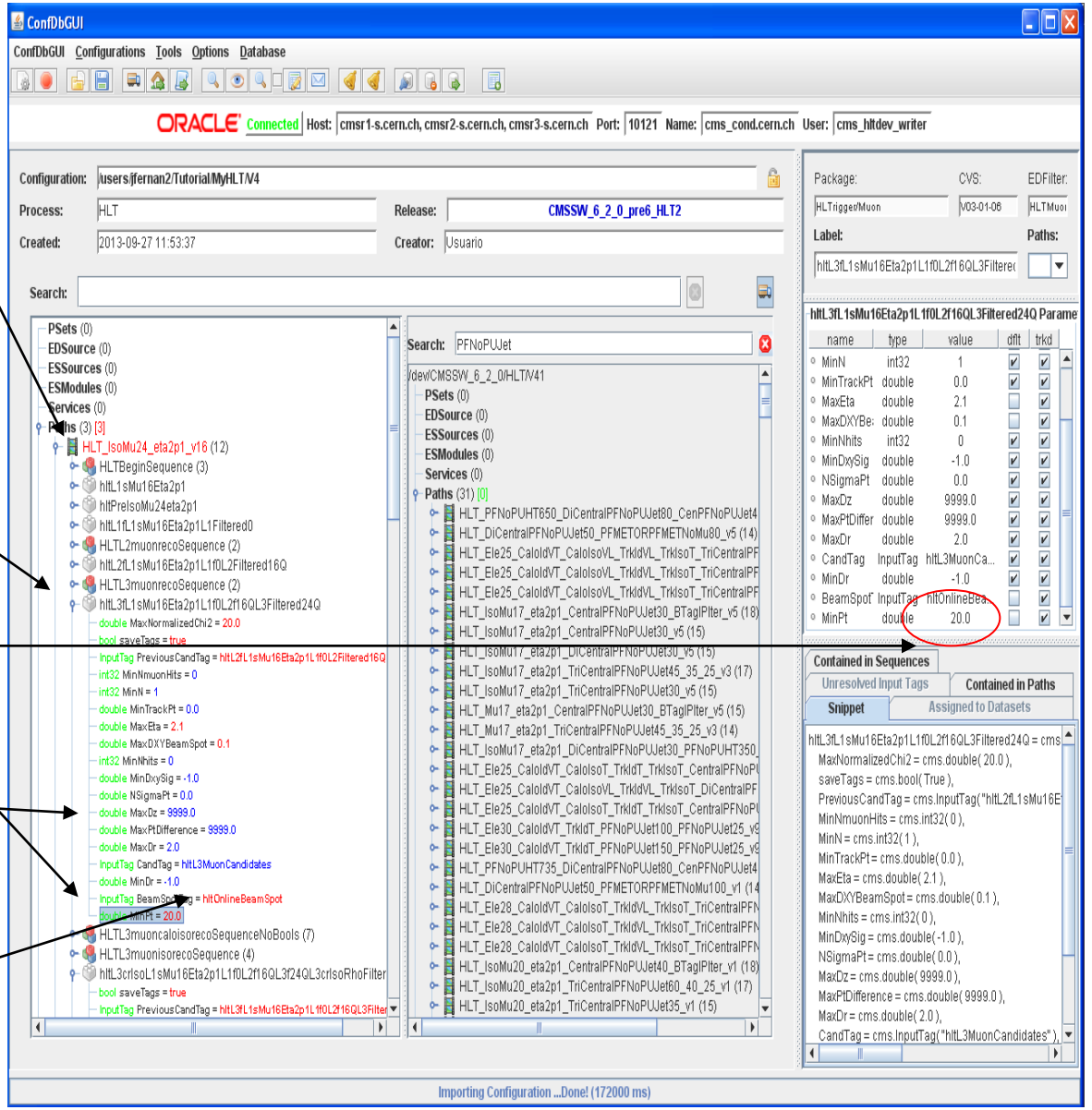


Modifying Paths in Your Configuration (I)

Starting point:
IsoMu24_eta2p1

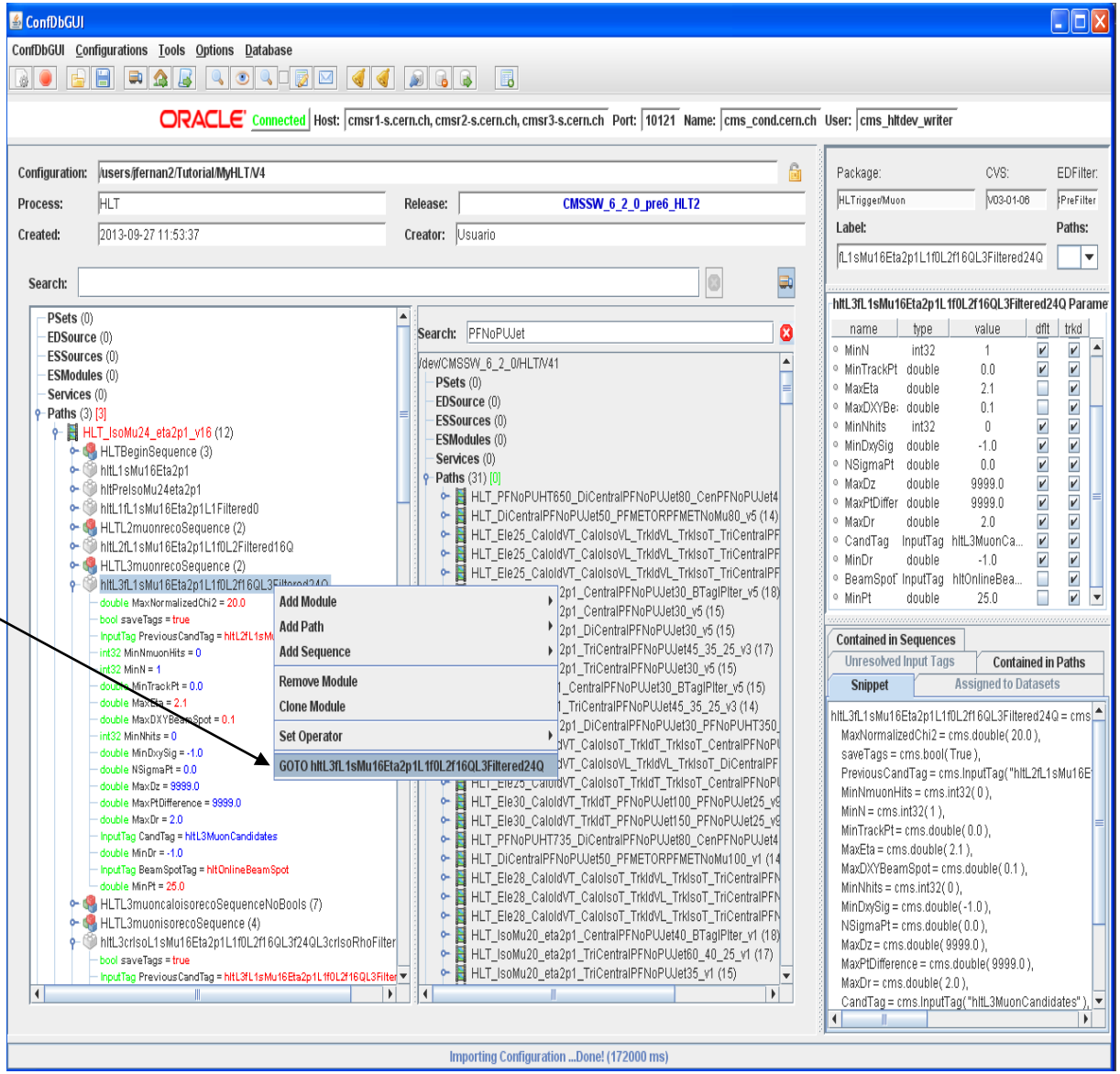
- Change HLT (L3) muon threshold
- Identify and select the right module
- Find the parameter
- Modify in the right hand panel (Double-Click)
- Increase to 25GeV

NB: **blue values** are default/unmodified values wrt the original module. **Red ones** are modified values



Modifying Paths in Your Configuration (II)

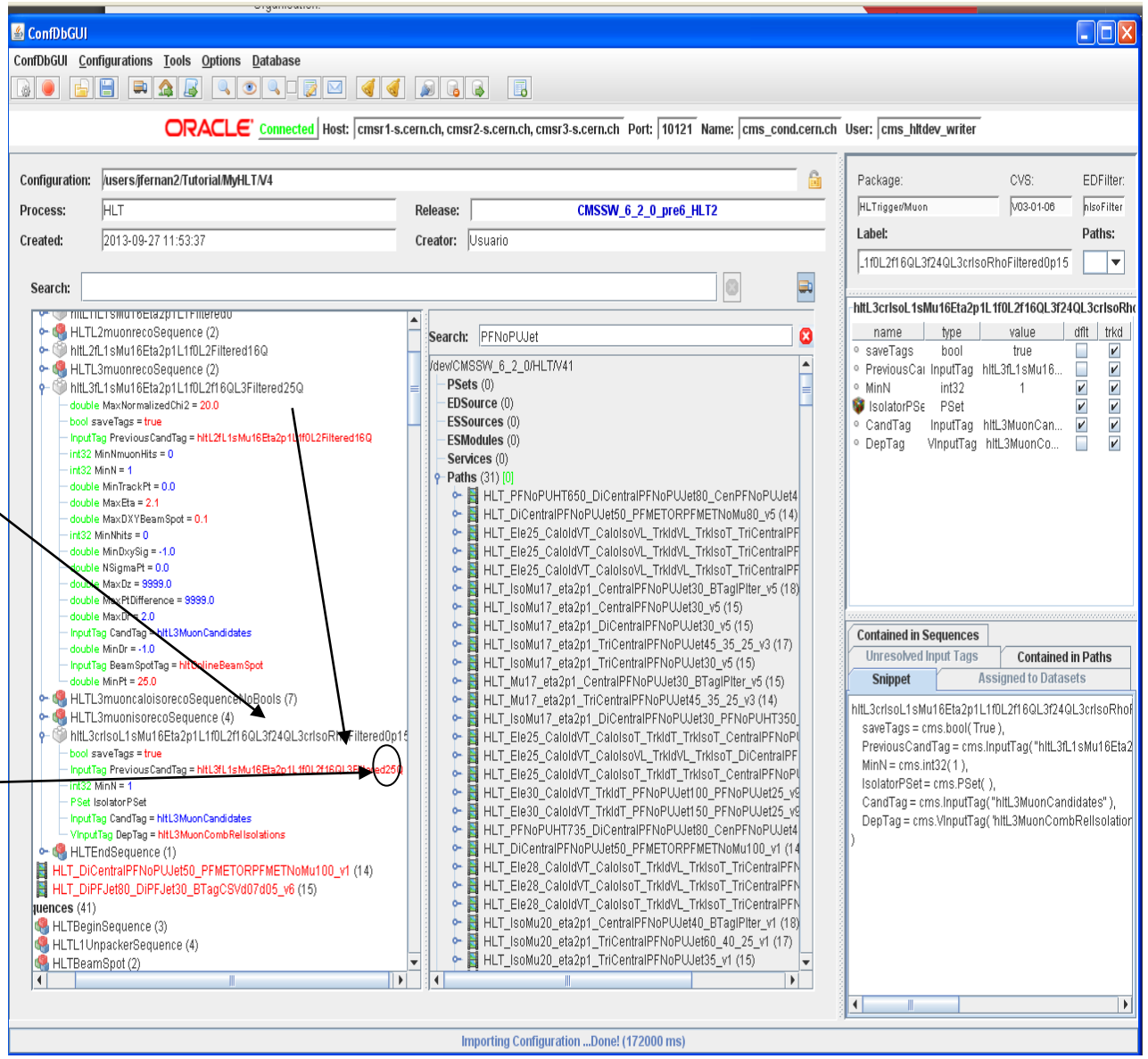
- Change the name of the module from "*Filtered24" to "*Filtered25" :
  - Right-click on the module, select "**GOTO hltL3fL1sMu16Eta2p1L1f0L2f16QL3Filtered24Q**"
  - From the module list, you can right-click on "Rename Module" and change "24" to "25" in the name



Modifying Paths in Your Configuration (III)

Rename any other module and sequence from "*Mu24" "*Mu25" :

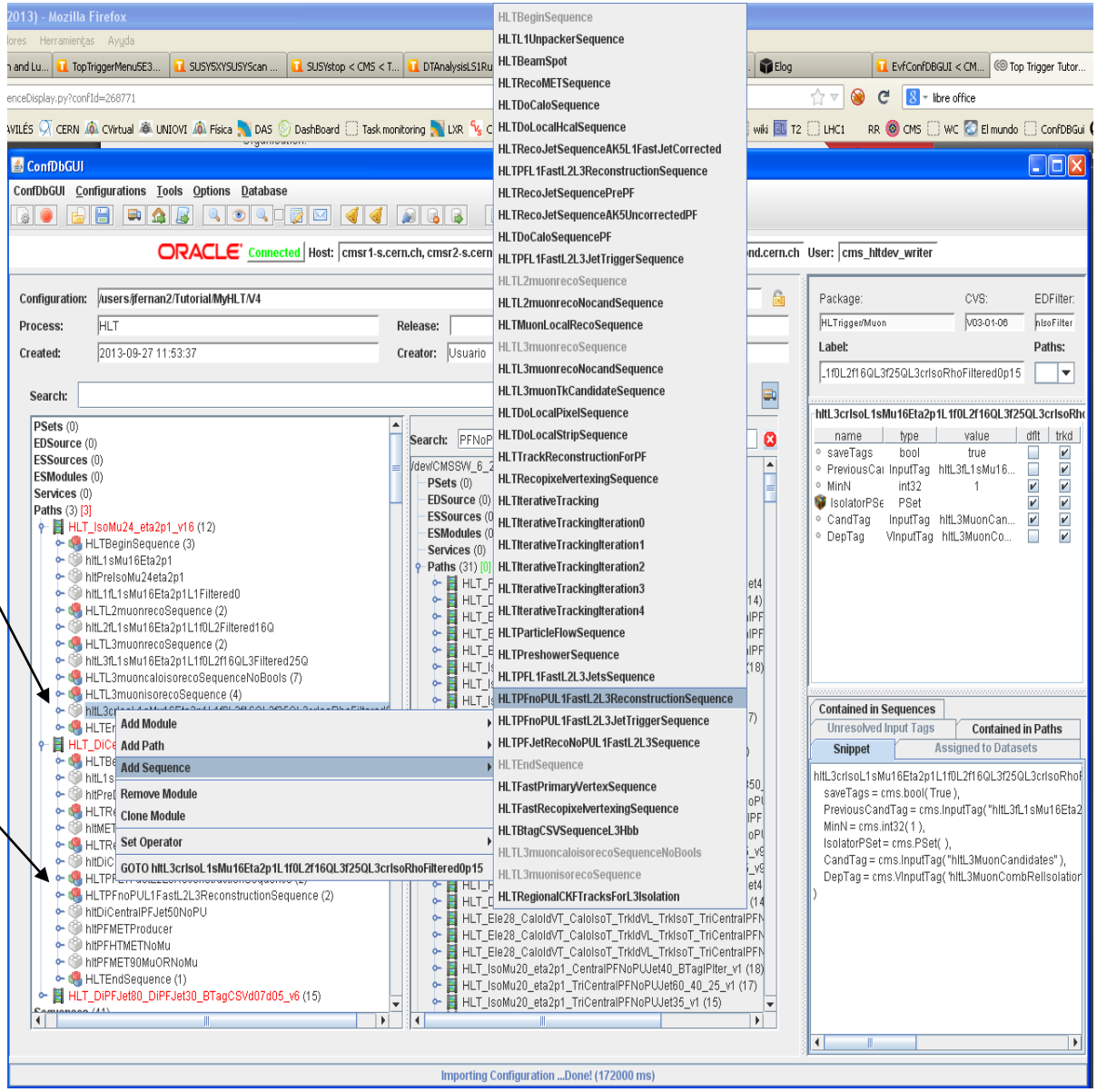
- Right-click on the module/sequence, select "**GOTO whatever**" …
- This one is automatic renamed since it is the **InputTag** from the module we have just renamed in previous slide



Modifying Paths in Your Configuration (IV)

Let's add a jet (from the Jet path) right after the last Muon module:

- We need 1 **sequence** and 2 modules:
  - HLTPFnoPUL1FastL2L3ReconstructionSequence
  - Add **cleaning on muons** (2 modules) afterwards (in next slides)

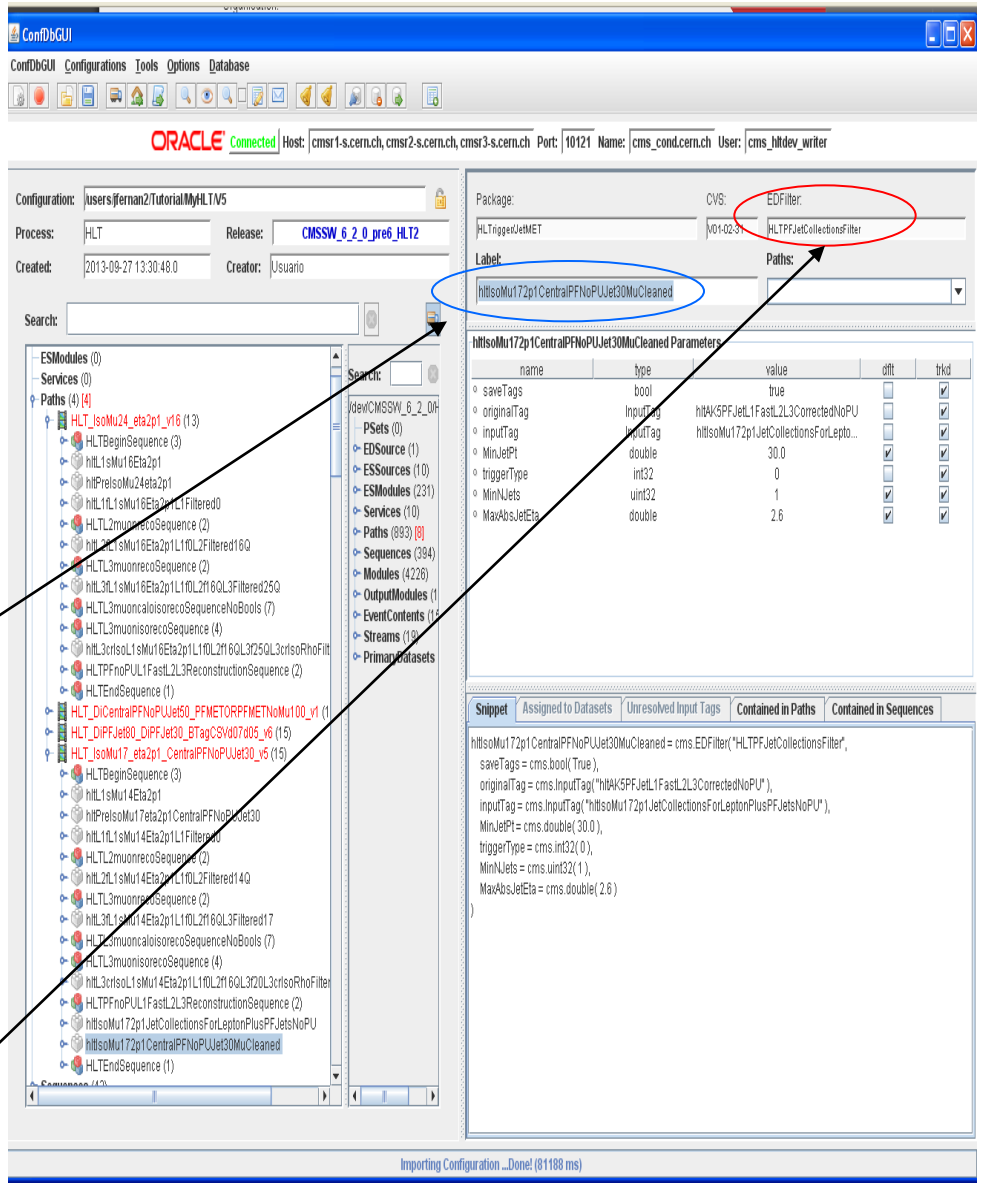


Caveat: To import a new sequence

- If you want to import a sequence not present in your own configuration:
  - Find the sequence in the list of sequences from the mid panel (Master configuration), drag and drop
  - Go back to your path
  - Right-click on the last module before you want to insert it, Add Sequence → YourNewSequence
- Note: **You can only import sequences this way, not modules. If you need a new module,**
  - **you'll have to import a path that uses it, copy or add as a new instance, and rename it or…**
  - **…if you know the type (EDFilter, EDProducer,…) and the class (HLTwhatEver*) of the module, you can add it as a new one (with default modifiable values)**
  - **In both cases :right-click** → Add Module → Type….

Modifying Paths in Your Configuration (V)

- Look for a path which has **LeptonCleaning** modules, e.g.:
  - HLT_IsoMu17_eta2p1_TriCentralPFNoPUJet30_v4
- Inspect how it is done there, 2 key **modules**:
  - hltIsoMu172p1JetCollectionsForLeptonPlusPFJetsNoPU
  - hltIsoMu172p1CentralPFNoPUJet30MuCleaned
- So, actually we need:
  - An **EDProducer**:
    - HLTPFJetCollectionsForLeptonPlusJets
  - An **EDFilter**:
    - HLTPFJetCollectionsFilter

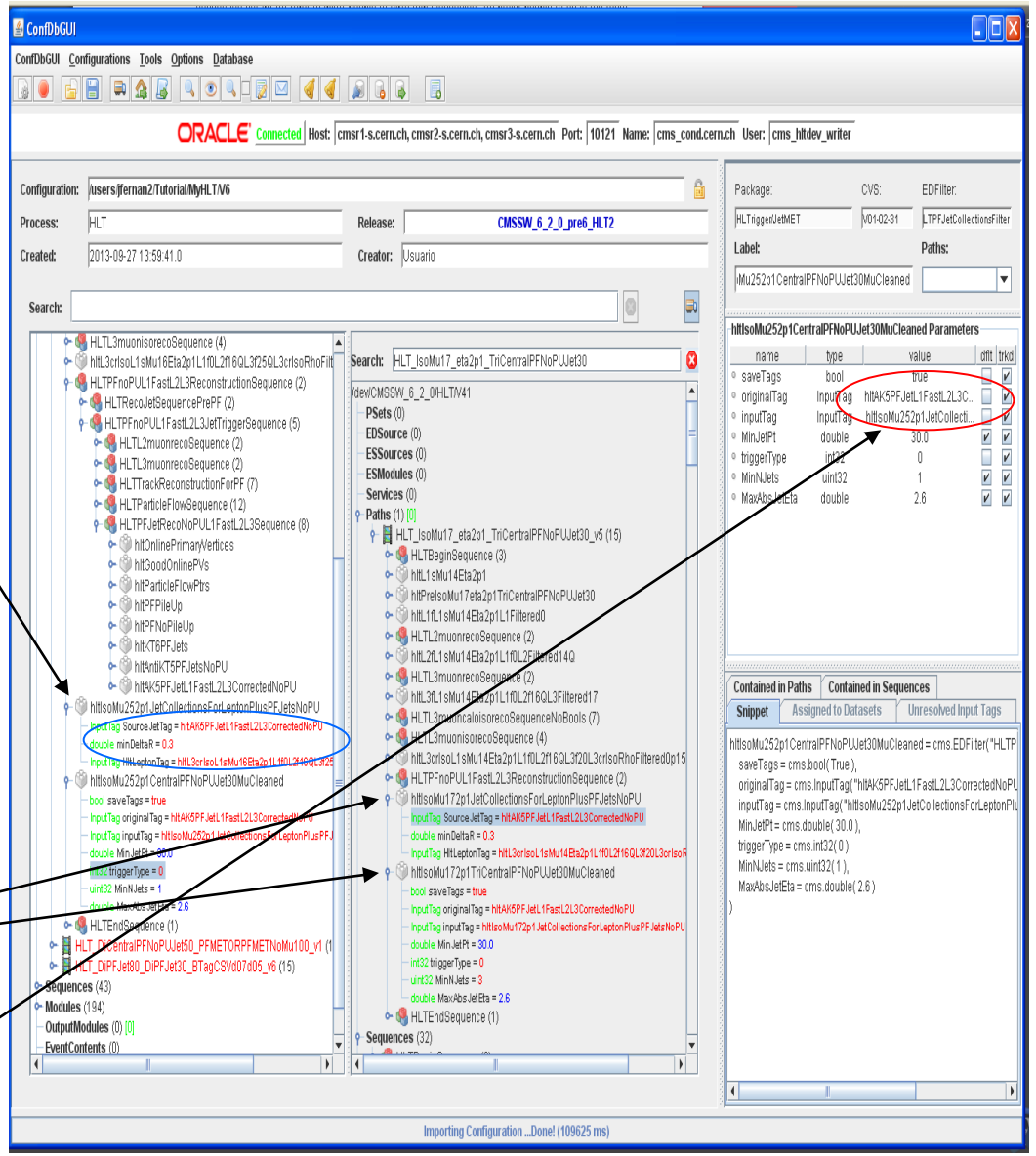


Modifying Paths in Your Configuration (VI)

Add them (order matters) and rename to an unique name, e.g.:

- An **EDProducer**:
  - HLTPFJetCollectionsForLeptonPlusJets → hltIsoMu252p1JetCollectionsForLeptonPlusPFJetsNoPU
  - Change parameters and InputTags
- An **EDFilter**:
  - HLTPFJetCollectionsFilter → hltIsoMu252p1CentralPFNoPUJet30MuCleaned
  - Change parameters and InputTags
  - Modify Jet ET threshold to 30 GeV

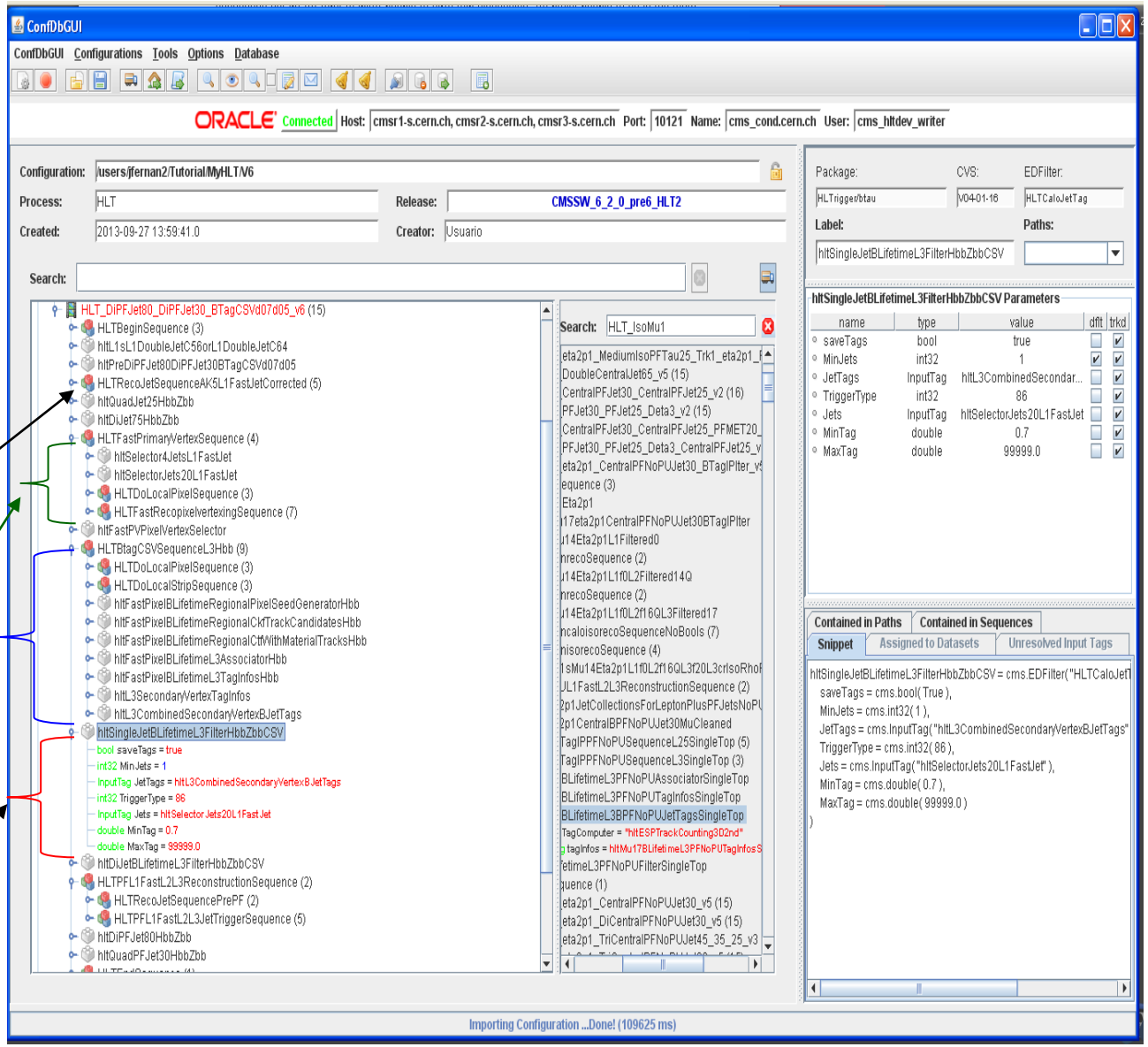
Tip: use Top Trigger as reference and copy&paste label names from your specific modules



Modifying Paths in Your Configuration (VII)

Let's add b-tagging now :

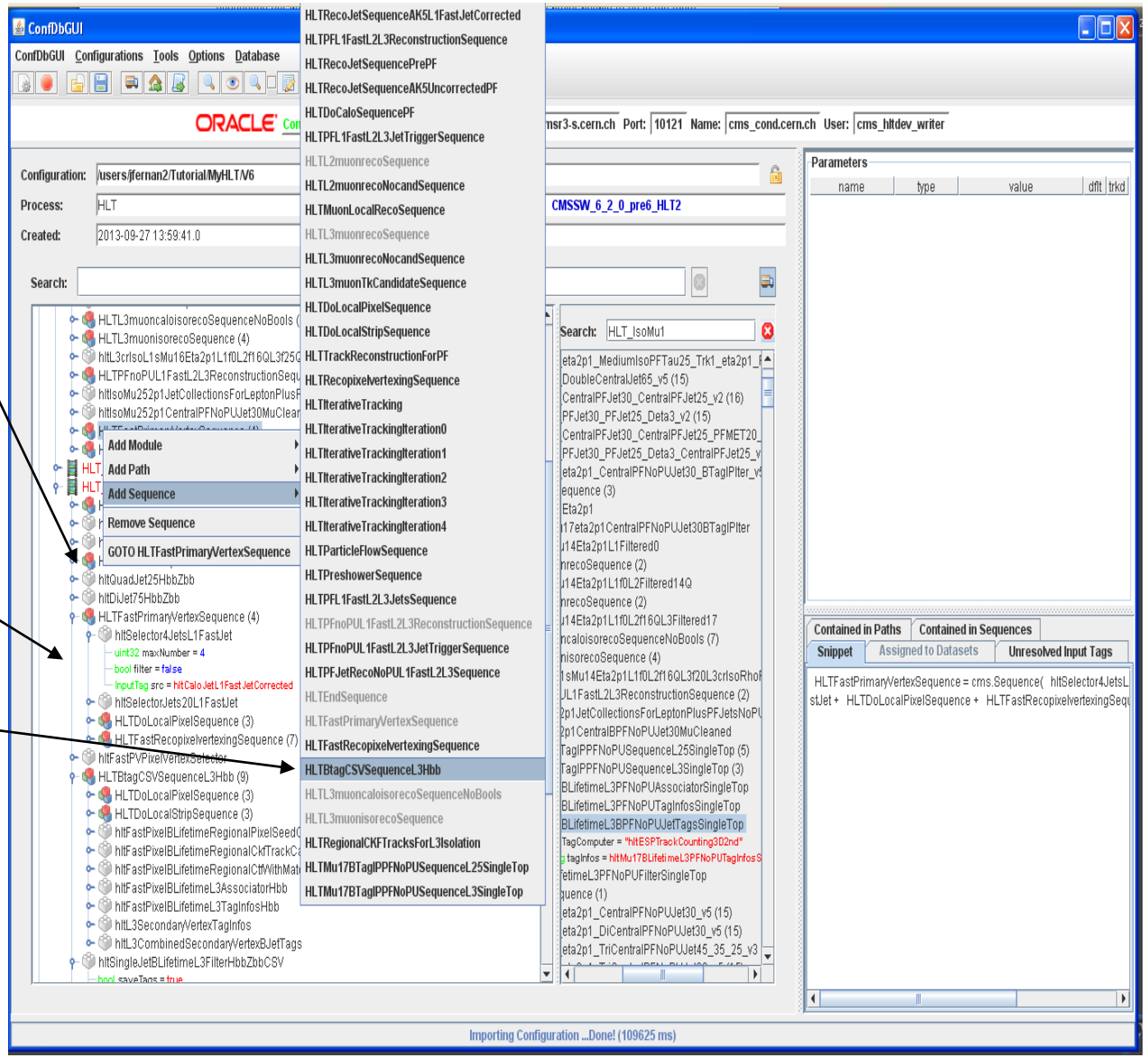
- Look and inspect the b-tagging path we imported
- We need:
  - Do the sequence with CaloJets
  - Fast PrimV
  - Calculates CSV per jet
  - The final filter: 1 b-tagged jet



Modifying Paths in Your Configuration (VIII)

Add at the end of your path (but before HLTEndSequence) clones of:

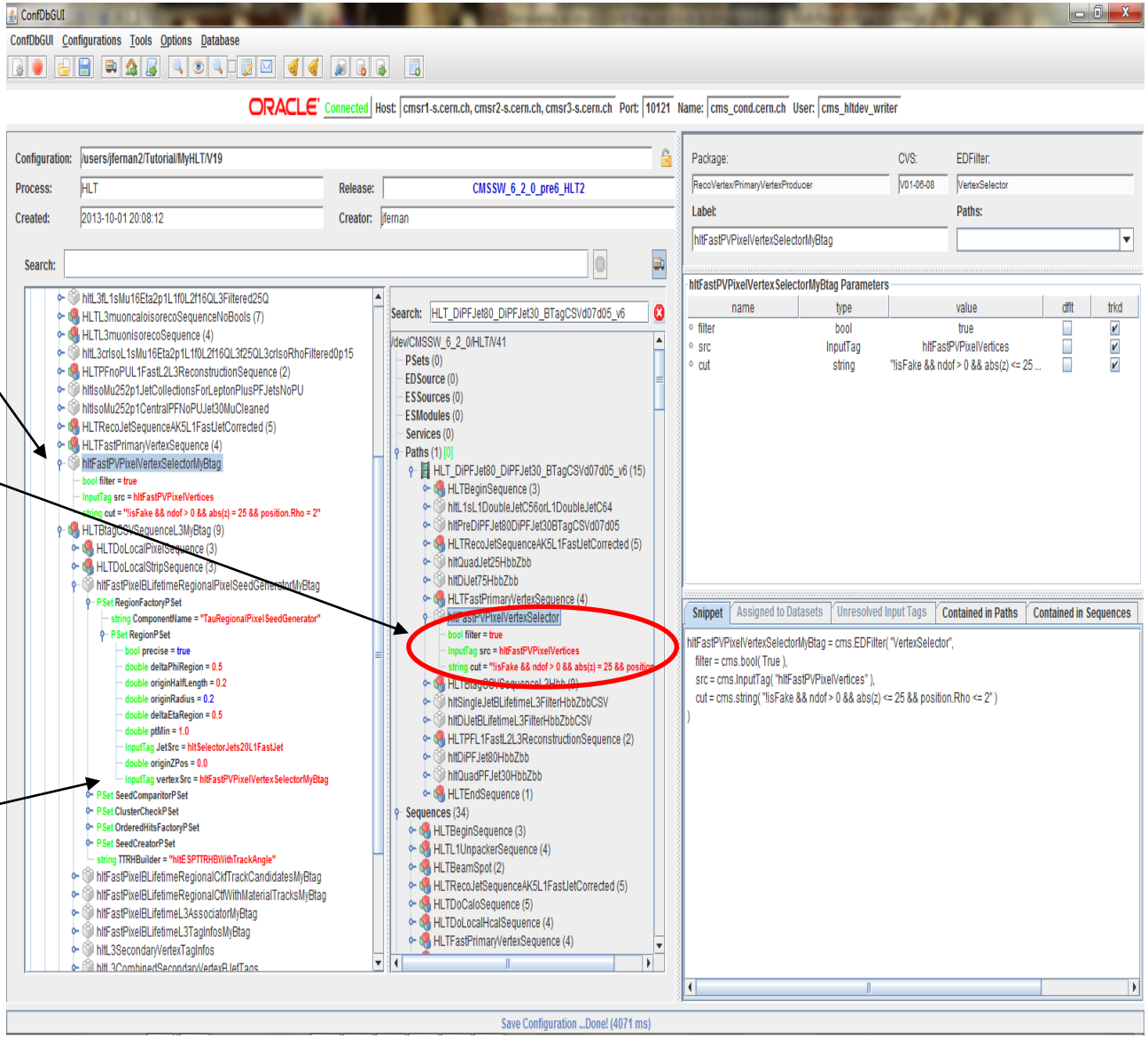
- HLTRecoJetSequenceAK5L1FastJetCorrected
  - It creates CaloJets
- HLTFastPrimaryVertexSequence
  - It performs PV in a fast way: relies on jets in order to select hits in the pixel systems
- HLTBtagCSVSequenceL3Hbb
  - Rename “Hbb” sequence/modules to “MyBtag” (e.g. HLTBtagCSVSequenceL3MyBtag)
  - 1 Sequence & 5 modules to rename
  - 2 final modules to append the MyBtag sufix



Modifying Paths in Your Configuration (IX)

Add module VertexSelector to select a PV (EDFilter):

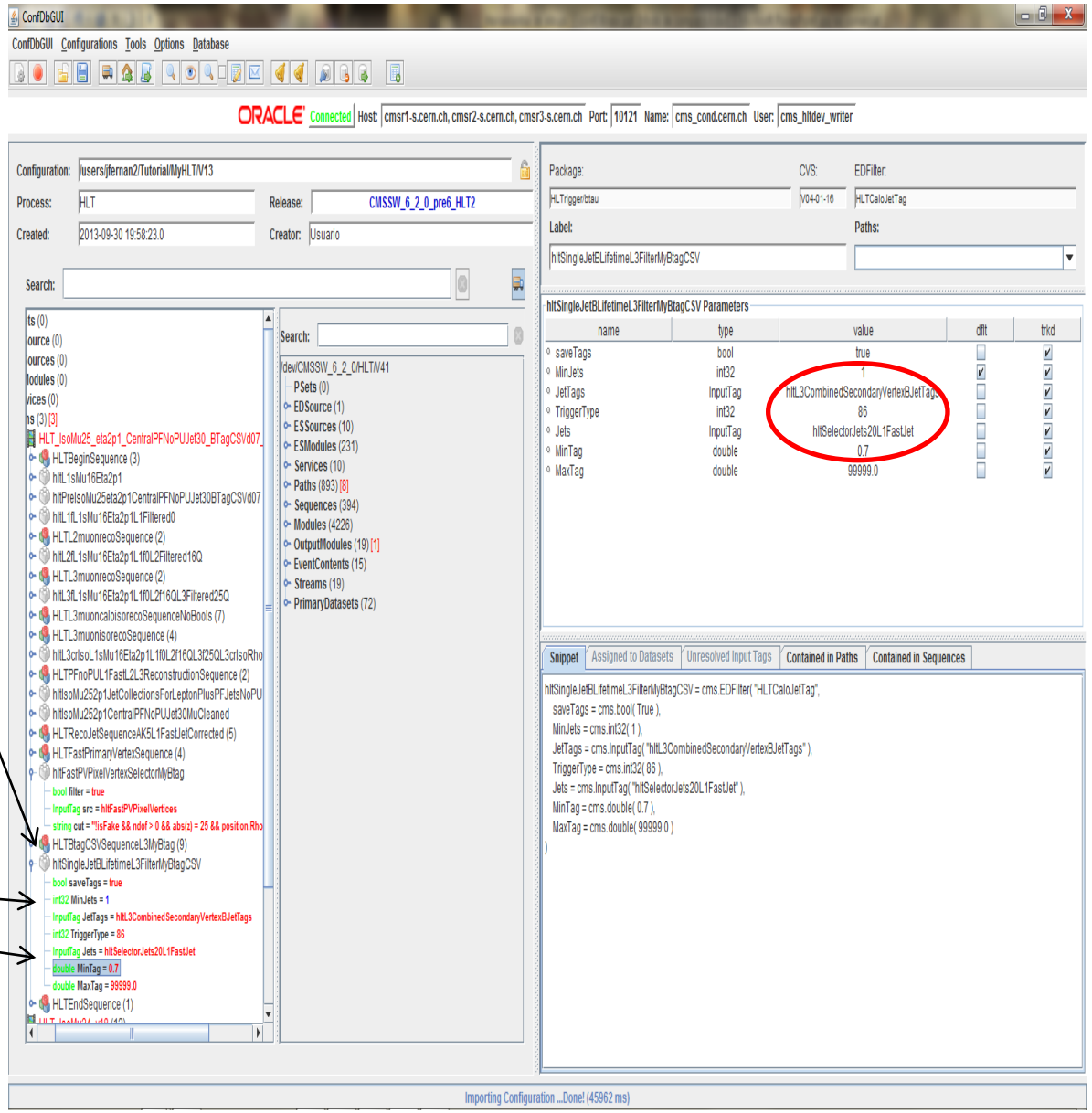
- Between HLTFastPrimaryVertexSequence and HLTBtagCSVSequenceL3MyBtag
- Copy it from hltFastPVPixelVertexSelector
- Rename it to hltFastPVPixelVertexSelectorMyBtag
- Modifiy parameter with your specific InputTag:
  - 4 InputTags (1 hidden under RegionPSet)
- Use the “Unresolved Input Tags” tab to search for undefined references



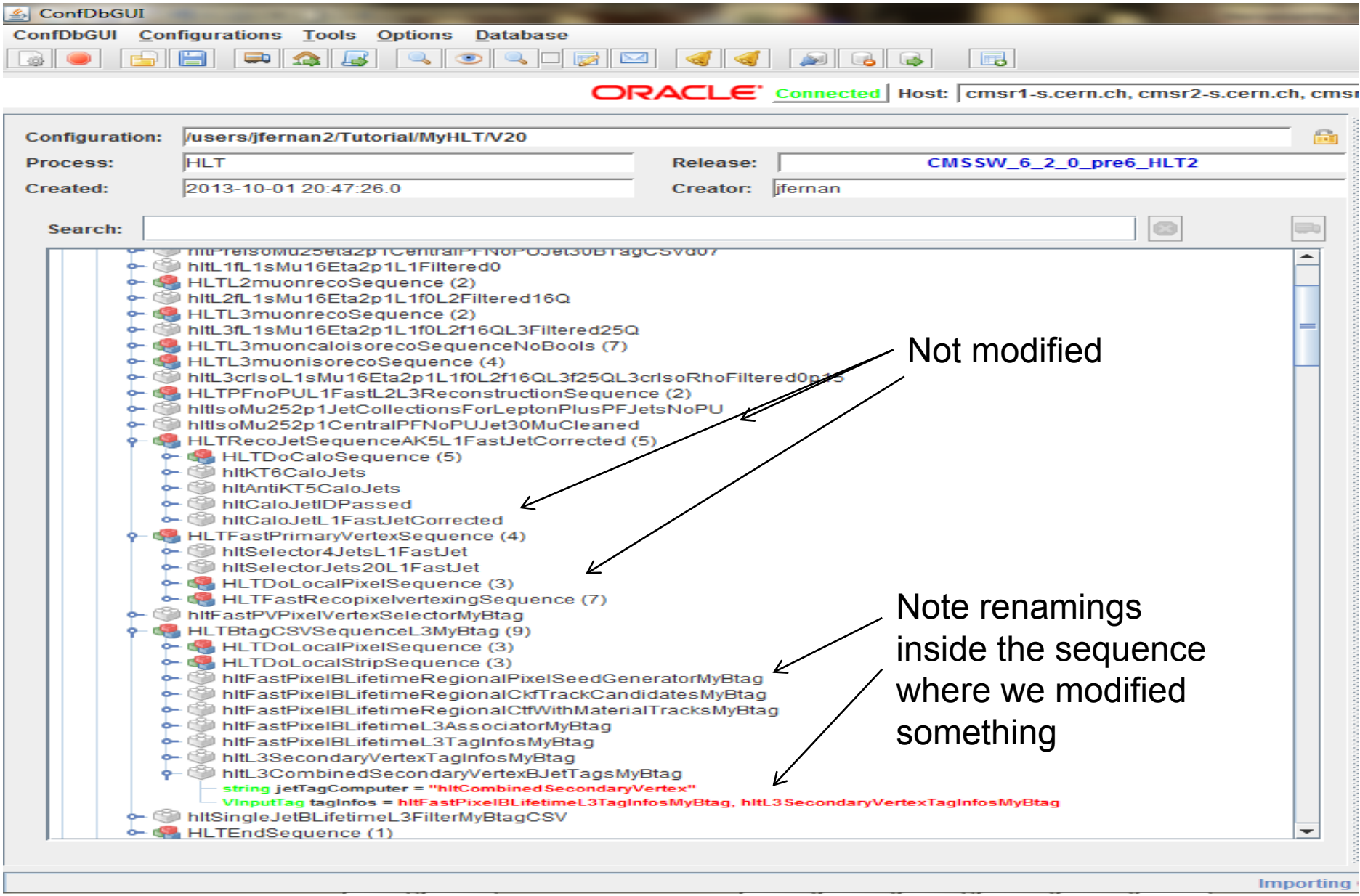
Modifying Paths in Your Configuration (X)

Add the final decission module (EDFilter):

- Right after HLTBtagCSVSequenceL3MyBtag
- HLTPFJetTag
  - Dub it hltSingleJetBLifetimeL3FilterMyBtagCSV
  - Ensure you ask for just 1 jet
  - Set MinTag to 0.7
  - Modify InputTags if needed

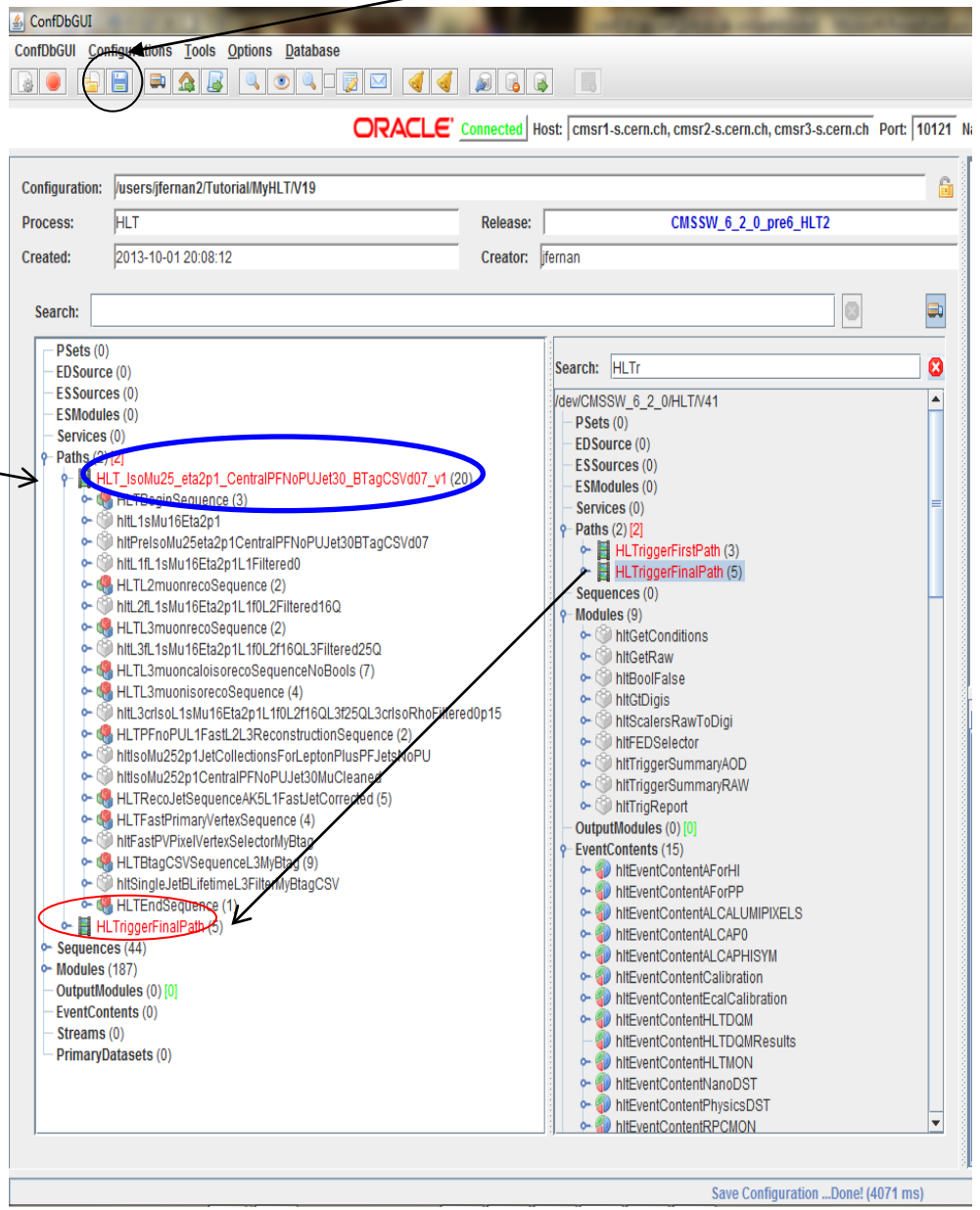


It should look like this



The Final Touches

- Make sure you include the HLTriggerFinalPath from the master table as the last path in your area, just before any Endpath
  - Import HLTriggerFinalPath from the master
- **Rename your Path**
- **Version number**:
  - If the path name is changed (i.e. if this is a really new path), put the version as _v1
  - If the path name didn't change (i.e. you are modifying a path), leave the version number the same as what it is in the master
  - TSG bumps the version number at the last minute, when preparing the new menu
- **Save your config and add a valuable comment!!!**



Testing your configuration (I)

- Find some input events (e.g.):

eoscms ls -l /eos/cms/store/group/comm_trigger/TriggerStudiesGroup/Skims/Top

- Set up your working area and run hltIntegrationTests according to https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideGlobalHLT
  - E.g.: Set up CMSSW_6_2_1 (+ tags, if any…) in tcsh

```
setenv SCRAM_ARCH slc5_amd64_gcc472
cmsrel CMSSW_6_2_1
cd CMSSW_6_2_1/src
cmsenv
git cms-addpkg HLTrigger/Configuration
scram build -j 4
```

- hltIntegrationTests /your/test/menu/with/only/paths \
  -s /dev/CMSSW_6_2_0/HLT -i some/input/data \
  -x "--globaltag auto:hltonline“ > out

- hltIntegrationTests will produce hltintegration/hlt.log and other log files for each path in your configuration
  - If you get errors, fix them!
    - Look at hltintegration/YourPath.log for clues
  - Make sure at least a few events pass all your paths

Testing your configuration (II)

- In my particular case:

```
hltIntegrationTests /users/jfernan2/Tutorial/MyHLT/V21 \
        -s  /dev/CMSSW_6_2_0/HLT   \
-i /store/group/comm_trigger/TriggerStudiesGroup/Skims/Top/SingleMuDS/
    pickevents_SemiMuon10k_1_1_Bbo.root   \
        -x "--globaltag auto:hltonline" > out
```

- Check in the log that (at least) some events passed:

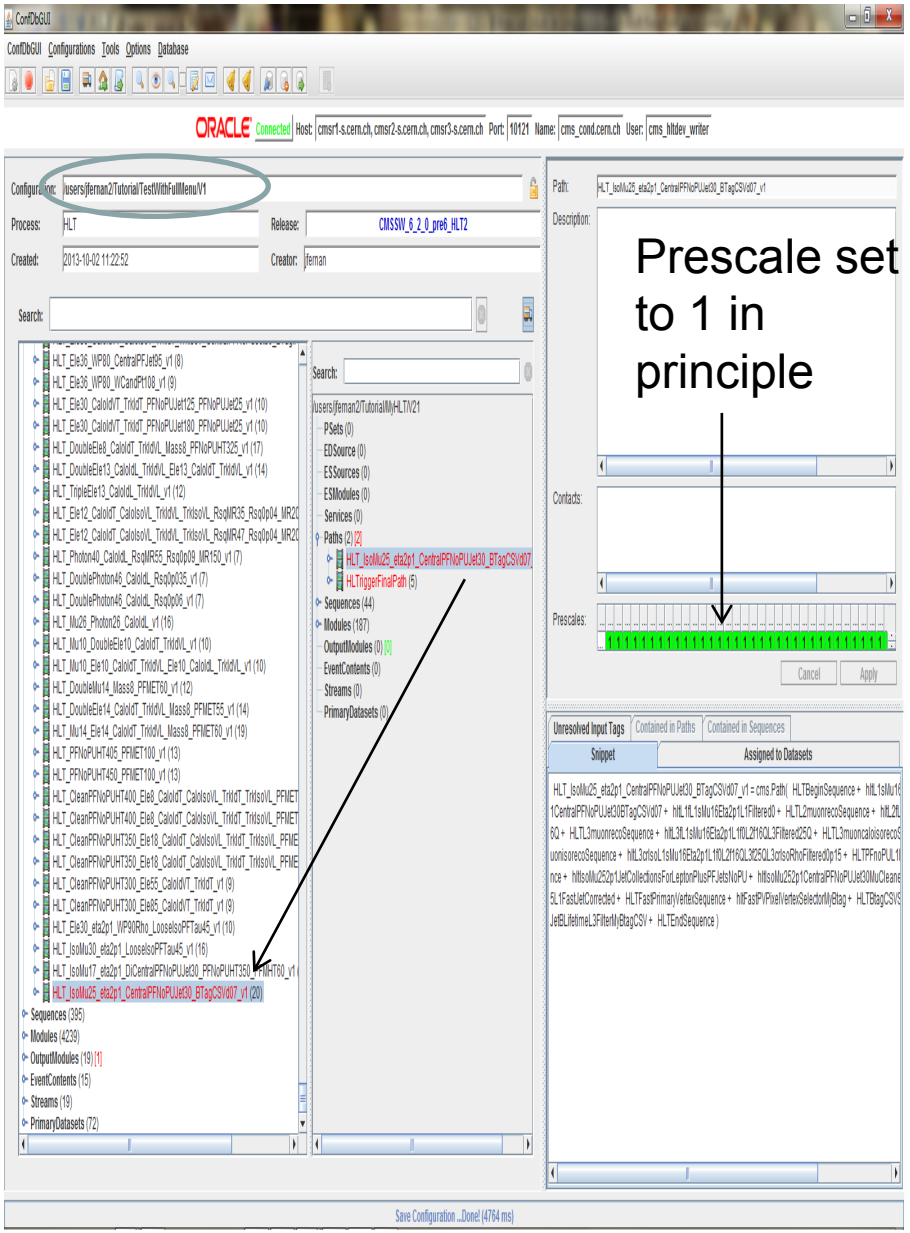
```
TrigReport ---------- Path   Summary ------------
TrigReport  Trig Bit#       Run     Passed     Failed      Error Name
TrigReport    1    0       100       85         15          0 HLT_IsoMu25_eta2p1_CentralPFNoPUJet30_BTagCSVd07_v1
TrigReport    1    1       100       100         0          0 HLTriggerFinalPath
```

A deeper look tell us most of the events are rejected at b-tag level:

```
TrigReport     1    0       100        98         2          0 hltL3crIsoL1sMu16Eta2p1L1f0L2f16QL3f25QL3crIsoRhoFiltered0p15
TrigReport          98        98        98         0          0 hltL3CombinedSecondaryVertexBJetTags
TrigReport          98        98        85         13         0 hltSingleJetBLifetimeL3FilterMyBtagCSV
```

Acid test

- Try to really integrate your path in the full Menu: i.e. do the other way around!
  - Start from a full menu
  - Import your path into the full menu
  - Add it to a stream…
  - Save the configuration **in your users area!! "Save As"**
  - E.g. /users/jfernan2/Tutorial/TestWithFullMenu/V1
- Do the hltIntegrationTests with the new full menu
  - hltIntegrationTests /users/…..



Summary/Questions?

- **Basic idea**:
  - Create a blank configuration
  - Import a similar path(s) from the master and modify to your specifications
  - You can use advanced search-replace tools
  - "Rules" of thumb:
    - **Sequences** (after importing them from Master):
      - If you don't modify them: try to maintain names in Sequence and its modules, they will be shared among other paths in the configuration
      - If you modify them: dub with a name which recalls the path you are constructing
    - **Single modules** (not in sequences) should be always renamed (unique)
    - Whatever you modify, make sure to propagate the changes into the following modules/sequences in terms of InputTags mainly
    - "Usually", a module is fed by previous one, but there are exceptions
  - **Test that the configuration is ready to be integrated into the master**
- Solution is on /users/jfernan2/Tutorial/MyHLT/V21

Additional info

Stolen from latest confDB tutorial (January 2012) by Juliette Alimena

Submitting a JIRA request

- Follow the instructions here: https://twiki.cern.ch/twiki/bin/viewauth/CMS/TriggerTrainModel#Important_notes_for_developers
- Include:
  - Which group of triggers (i.e. Tracker DPG, BTag POG, SUSY PAG, etc.) and when approved
  - Target train (i.e. February 2012)
  - Links to approval presentation[s] in TSG meeting[s] for reference, including expected rates and prescales
  - The "anchor" master table you started out from (i.e. /dev/CMSSW_5_0_0/HLT/V1234)
  - Your user area in ConfDB where to pick up the new paths (i.e. /user/guido/5e33/bestPFMHT/Vxyz)
  - The list of paths to be removed
  - The list of paths changed (do NOT bump the version number)
  - The list of new paths added (put version number _v1) with their dataset assignment including the various monitoring datasets and the expected unprescaled rate for your triggers at 5e33
  - Required changes in the event content for the monitoring streams
  - Any new C++ code required - provide the cvs tags
  - The output of the hltIntegrationTests and the file hlt.log created by it
- How to submit trigger requests on JIRA
  - Log in to JIRA at https://its.cern.ch/jira/browse/CMSHLT/
  - Create a new "Issue" for your request (top-right: click on "+Create Issue"), this pops up a new page to specify the details of the new issue:
    - leave the "Project" as "CMS High Level Trigger"
    - for the "Issue type" choose "Configuration change" in the pull-down list
    - write a concise subject in the "Summary" field
    - as "Version", please pick one of those available for 2013
    - define your request as "HLT configurations" in the "Components" field
    - list the groups (PAGs,POGs, or DPGs, etc.) requesting the issue in the "Environment" field
    - fill the "Description" field with all the information requested above
    - click on "Create" to actually create the issue in the system (or "Cancel" to cancel).

Migrating a configuration

- If you create your configuration based on an older version of the master and in an older release, you might be asked to migrate the configuration to the latest release before being integrated
- Select Migrate from the Configurations menu, and then select the new release to migrate to.
- Once the procedure is complete, a migration report appears, as some modules might have changed their signature (new parameters, removed parameters, change of parameter type, etc.) in a way which can't be resolved automatically.
- *Usually, if a migration shows some conflicts, the easiest solution is to re-import the conflicting modules from the latest master (unless they are meant to be modified).

Compare a Configuration to Another One

- To compare a configuration with another configuration in the same ConfDB instance, choose Compare (Diff) from the Tools menu.
- Choose the configuration you would like to compare to.
- A report of the comparison will be displayed graphically in the Tree tab, and textually in the Text tab. All components which were added, removed, or changed are listed.
- **Tip: When editing a configuration, compare it

with itself - this will show all the changes since the last save.

Search for Components

- Start to type part or all of the component you are looking for in the Search box
- If you right-click on Search, you can choose:
  - Whether the search component label contains, matches, or startsWith the string you are typing
  - Whether you search through names/labels, types/plugins, or parameter values
- While a search is active, the parameters of the displayed modules can be changed freely. However, components can be neither renamed, removed, or added while the search is still active! If you select a node you would manipulate that way and cancel the search, the node will remain visible and selected when the complete configuration is displayed again.

Reorder Modules in a Sequence or Path

- You can reorder modules by **dragging** and **dropping**
- Note: To drag a node, it needs to be selected first! That means first select the node such that its background color turns **blue**, then move the cursor over the node, press the left button, keep it pressed and move the node around until the cursor is above the desired drop target.
- Select the node of the module to be moved, drag it, and drop it either on node of the parent path to move it to the beginning of the path, or to any other path entry, to move it behind that path component.

Locked Configurations

- If a configuration remains LOCKED (can happen if the GUI crashes or looses the network connection or is not closed correctly):

...and it WAS YOU who edited it last: You can unlock it by doing Configurations → Open, right-clicking on the locked configuration, and select Unlock.

...and it WAS NOT YOU who edited it last (and you're sure they're done editing it): Contact administrators

Reference links

- GUI: http://www.cern.ch/confdb/gui/start.jnlp
- Browser: http://www.cern.ch/confdb/browser/

- ConfDB GUI manual: https://twiki.cern.ch/twiki/bin/viewauth/CMS/EvfConfDBGUI

- HLT Global Table twiki: https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideGlobalHLT
  Up-to-date page on how to prepare your paths for each menu and to know the release + tags:

- How to submit JIRA requests: https://twiki.cern.ch/twiki/bin/viewauth/CMS/TriggerTrainModel#Important_notes_for_developers

- Older tutorials
  https://twiki.cern.ch/twiki/bin/view/CMS/EvFConfDBTutorial

Backup

Creating a New Configuration (II)

Two main ways you can make a new configuration:

1. **Save a copy** of the GRun menu (e.g. /dev/CMSSW_5_2_6/GRun latest 2012 pp Menu) in your area, and edit away
   - "Open an existing configuration" button
   - Will give you a full menu, but not necessary for our purpose here.