

STAR Trigger Tier1 Configuration Files
H.J. Crawford, E.G. Judd, J.M. Engelage, C.W. Perkins, J.M. Nelson
August 10, 2018

Tier1 files include binary files that contain the names and values of all registers of all VME boards in the STAR Level-0 Trigger system. They also contain the names of auxiliary files that are used to configure some large memories like look-up tables and DSM FPGA configuration memories. An ASCII Dictionary file is created with every binary file. The Dictionary file contains just register information. It is used by the STAR Run Control system to configure the GUI and when archiving the configuration of each Run in the database. This document describes how to make a new Tier1 file to deal with just software changes. The procedure for dealing with hardware changes (i.e. adding or subtracting an individual board or a whole VME crate) is documented separately at

https://www.star.bnl.gov/public/trg/TSL/Manual/Level0_System_Expansion.pdf

The pieces of the Level-0 VME system that are not configured through the Tier1 file architecture are the definition of which algorithm is loaded into each QT board, and the definition of the QT slew correction tables. The procedure for generating and loading all of those auxiliary files is documented separately at

https://www.star.bnl.gov/public/trg/TSL/Manual/Trigger_Tier1_aux_files.pdf

The Tier1 binary and Dictionary files are generated using the MakeConfig program from a set of ASCII-formatted control files kept on startrg.starp.bnl.gov in the `~trg/cfg/Tier` directory. They should **ONLY be edited by someone logged in to that machine as `trg`.**

Tier1 Control File

- Tier1 Control Files are kept in `~trg/cfg/Tier1`
- Control Files typically have names like `trg_<date>` or `ped_<date>`. “trg” and “ped” specify whether the file is for physics or pedestal data-taking and “date” is an 8-digit number with the format `yymmddvv` specifying year, month, day and instance of the file (version) created on that particular date. Files created purely for test purposes can have names that describe the test, e.g.:
 - `trg_19032900` is the 1st file made on March 29, 2019 for physics data-taking
 - `trg_19032901` is the 2nd file made on March 29, 2019 for physics data-taking
 - `ped_19032900` is for pedestal data-taking
 - `vp101_test` is a test configuration, not intended for production data-taking
- The Control File name is used as the root name for the “wild” file (another required input file for MakeConfig) and the output binary and Dictionary files, e.g.:
 - `trg_19032900` will be used with `trg_19032900.wild` to produce `trg_19032900.bin` and `trg_19032900.dict`
- **NOTES for Production Data Taking:**

- The wild file specifies the value of the TIER1_version variable (see section on page 3), which is copied into the Dictionary file.
- During Production data-taking the STAR Run Control system accesses the Tier1 files through “current” soft links:


```
lrwxrwxrwx. 1 trg  trg  14 Jun  2 11:13 trg_current.bin -> trg_19032900.bin
lrwxrwxrwx. 1 trg  trg  14 Jun  2 11:13 ped_current.bin -> ped_19032900.bin

lrwxrwxrwx. 1 trg  trg  15 Jun  2 11:13 trg_current.dict -> trg_19032900.dict
lrwxrwxrwx. 1 trg  trg  15 Jun  2 11:13 ped_current.dict -> ped_19032900.dict
```
- When Run Control archives the configuration information for a particular Run in the online database it uses the TIER1_version value to reconstruct the name of the underlying Dictionary file. It assumes that file name will always have the form trg_XXXXXXXXX.dict where “XXXXXXXXX” matches the TIER1_version value. It is therefore **essential** that for Production Runs, which use the “current” soft links, the Tier1 control file name has the form trg_XXXXXXXXX (or ped_XXXXXXXXX) and the value of “XXXXXXXXX” matches the TIER1_version value.
- If a test Tier1 file (e.g. vp101_test) is going to be used for Production running the test Control and Wild files should be renamed to use the trg_XXXXXXXXX format, the TIER1_version variable should be updated accordingly, and the binary and Dictionary files should be re-made.
- Each Control File starts with a crate dictionary which is then followed by the list of crate configuration files.
 - The dictionary line must have DICT: as the first 5 characters.
 - The rest of the dictionary line contains a three-letter name for each crate listed in the SAME order as the <crateId>_CONF_NUM variable found in ~trg/trg_soft_dev/include/RC_Config.h
 - The configuration file names may be in any order.
- The safest way to make a new file is to copy an existing Control File then edit it to point to the new Crate Configuration Files.

Crate Configuration Files

- Crate Configuration Files are kept in ~trg/cfg/<crateId>
- These files have names like DSM_BBC_trg_<date>.dat and QT_MXQ_ped_<date>.dat
- The safest way to make a new file is to copy an existing Crate Configuration File to a new file name and then edit it to implement any changes. NOTE: The new file name MUST be exactly the same as the name specified in the new Tier1 Control File.

Files in the Dictionary subdirectory

- Dictionary Files are kept in ~trg/cfg/Tier1/Dictionary
- There are 3 types of Files in this directory:

- Wild files specify the value of the TIER1_version variable, point to the appropriate TCU file and can also be used to define global QT registers that are common to many boards
- TCU files define the list of bits that the TCU receives from the QT/DSM tree
- Dictionary files contain labels with register values extracted from the Crate Configuration files and the Wild files. Dictionary files are created along with the Tier1 binary file (see “Binary Files” section below) and should not be edited.
- Wild Files
 - QT registers can be defined in both the Crate Configuration files and the Wild files. Registers that are defined in the Crate Configuration Files apply to a specific QT board (or daughter card) and the definition includes a default value. By contrast, registers that are defined in the Wild files apply to every QT board in a crate (or the entire system) and no default value is specified. Instead the user must specify the Wild value during configuration from the Run Control GUI.
 - If a register is defined in the Wild file, and the user actually specifies a value in the GUI, then that value takes precedence over all other settings for that register in all the QT boards in the specified crate. Otherwise, the value that is specified for that register on each individual QT board will be used.
 - Wild Files have names that are identical to their Tier1 Control File name with the “.wild” suffix, e.g. trg_19032900.wild
 - The safest way to make a new file is to copy an existing Wild File to the new file name and then edit it to implement any changes.
 - Make sure to change the TIER1_version value to the new “yymmddvv” value.
 - If there are changes to the TCU file, then make sure to change the name of the TCU file in the “include” line.
- TCU Bit Definitions
 - TCU Files are named TCU_bit_def_<date>.txt
 - The safest way to make a new file is to copy an existing TCU File to a new file name and then edit it to implement any changes. NOTE: The new file name MUST be exactly the same as the name specified in the new Wild file.

Binary Files

- First create all of the Control and Wild ASCII files as described above.
- Go to the ~trg/cfg/Tier1 directory and run the MakeConfig function (e.g.:
 - MakeConfig trg_yymmddvv
 You will have the opportunity to enter a short (1 line) comment describing the new file.
- A large number of messages will stream past. If MakeConfig is successful, you'll see "Error on decoding: QT4 0" about 16 lines up from the bottom. If you see a number other than "0" then there is an error that will need to be fixed (see Problem Solving section below).

- The output of MakeConfig will be a new binary file (.bin) in the Tier1 directory, and a new dictionary file (.dict) in the Dictionary subdirectory.

Problem Solving

- Occasionally MakeConfig will find a problem when processing a crate's information. The error must be found and fixed before the binary and Dictionary files can be made.
- The best way to look for the error is to use the command ReadConfig, which reads and prints information from the binary Tier1 file. The output can be redirected to a text file, e.g.:
 - ReadConfig trg_19032900 <crateId> > trg_19032900.txt
- If <crateId> is ALL, then ReadConfig will print a short list of all the crate names.
- If <crateId> is one of those crate names, then ReadConfig will print out all the information about that crate from the Tier1 file.
- The output is a txt file that can be searched or easily compared to a txt file made from using ReadConfig on the Tier1 file that was the basis for the new Tier1 file.