

JETDEF WITH THE STRAW MAN MENU

P SHERWOOD

INTRODUCTION

JetDef is code used to configure the jet slice trigger chains. This note describes the version of JetDef which runs with the straw man menu.

The input to the program is a dictionary of data issued by the central trigger menu code (CTM). The output is either a ChainDef or an ErrorChainDef instance. These instances convey jet trigger configuration data for a single trigger chain to the CTM.

Each ChainDef object is identified by a unique string label or chain name.

Under normal running conditions, the sequence information carried by ChainDef objects include instances of python classes which configure ATLAS C++ algorithms. At the time of writing, not all of these C++ configuration classes exist. JetDef is organized such that the algorithms are specified by JetDef Algorithms for most of the processing. These have natural types as attributes. JetDef uses the JetDef Algorithms to optionally instantiate the C++ configuration Algorithms after determining the chain configuration. Thus, handling the ATLAS C++ configuration objects is handled independently from determining the jet slice configuration information.

For development purposes, and in the absence of the existence of the necessary configuration classes, the JetDef Algorithm instances are not converted to the C++ configuration instances.

The ChainDef instances produced by JetDef contain sequences that perform the following tasks:

- unpack the trigger towers and run a clustering algorithm. Optionally, run a hypo.
- run L2 reconstruction algorithms and hypo
- run the EF reconstruction algorithms and hypo

Date: 20 February 2014.

- run the 2015 partial scan algorithms
- run the 2015 jet reconstruction and hypo algorithms

The arguments of the algorithms depend on the chain name. Which algorithms are run can depend on the combination of sequences present in the chain. Not all sequence combinations are legal.

1. PROGRAM FLOW

A dictionary of data is received from the caller code. This is re-arranged to reflect the structure of the JetDef calculation. The incoming information provides information only about the last sequence in the chain. Extra information which will determine which other sequences are to be configured is injected into the data flow at this point. This information is packed into a ChainConfig object, which is sent to the JetSequenceAnalyser. This examines the ChainConfig to determine which sequences are needed. The JetSequenceAnalyser builds the sequences using a combination of data present in the ChainConfig object, and hard-coded knowledge.

The sequences are returned. A ChainDef object is instantiated, and is given the sequences. The signatures are added to the ChainDef (a signature is added for each sequence). If the L1 seed has been provided by the central menu software, it is added, otherwise a default L1 seed for the chain name is provided. The ChainDef object is returned to the caller. If an error is encountered during processing, an exception is raised with an explanatory string. JetDef catches the exception and returns an ErrorChainDef which contains the explanatory string