

Table of Contents

Elaborated example of DaVinci algorithm using TriggerTisTos tool.....	1
cpp file.....	1
header file.....	4
Sample output for 1 event in B -> K* mu mu MC.....	5

Elaborated example of DaVinci algorithm using TriggerTisTos tool

THIS EXAMPLE IS OUT OF DATE AND NEEDS UPDATING PLEASE SEE PYTHON EXAMPLE INSTEAD AND DO EQUIVALENT CALLS IN C++ I WILL TRY TO UPDATE THIS PAGE SOON

cpp file

```
// from Gaudi
#include "GaudiKernel/AlgFactory.h"

// local
#include "TisTosDVExample.h"

#include "Event/LODUReport.h"

#include "Event/HltSummary.h"

using namespace LHCB;

//-----
// Implementation file for class : TisTosDVExample
//
// 2007-09-24 : Tomasz Skwarnicki
//-----

// Declaration of the Algorithm Factory
DECLARE_ALGORITHM_FACTORY( TisTosDVExample );

//=====
// Standard constructor, initializes variables
//=====
TisTosDVExample::TisTosDVExample( const std::string& name,
                                 ISvcLocator* pSvcLocator)
    : DVAlgorithm ( name , pSvcLocator )
{
}

//=====
// Destructor
//=====
TisTosDVExample::~TisTosDVExample() {}

//=====
// Initialization
//=====
StatusCode TisTosDVExample::initialize() {
    //=== The following two lines should be commented for DC04 algorithms ! ===
    StatusCode sc = DVAlgorithm::initialize();
    if ( sc.isFailure() ) return sc;

    debug() << "=> Initialize" << endmsg;

    m_TriggerTisTosTool = tool<ITriggerTisTos>( "TriggerTisTos",this);

    return StatusCode::SUCCESS;
}

//=====
// Main execution
//=====
```

TisTosDVExample < LHCb < TWiki

```

StatusCode TisTosDVExample::execute() {

    debug() << "=="> Execute" << endmsg;

    const std::vector<const Particle*> & parts = desktop()->particles();
    if( parts.size() == 0 ) return StatusCode::SUCCESS;

    // L0 trigger
    if( !exist<L0DUReport>( L0DUReportLocation::Default ) ){ return StatusCode::SUCCESS; }
    const L0DUReport* pL0DUReport = get<L0DUReport>( L0DUReportLocation::Default );
    info() << " L0 decision= " << pL0DUReport->decision() << endmsg;

    std::vector<std::string> l0NamesInHlt = m_TriggerTisTosTool->triggerSelectionNames("L0*Decision
                                                                    ITriggerTisT
    info() << " Names of L0 triggers in Hlt="; dumpvts( l0NamesInHlt ); info() << endmsg;

    // Hlt overall
    if( !exist<HltSummary>( LHCb::HltSummaryLocation::Default ) ){ return StatusCode::SUCCESS; }
    const HltSummary* pHltSummary = get<HltSummary>(LHCb::HltSummaryLocation::Default);
    info() << " Hlt final decision= " << pHltSummary->decision() << endmsg;

    // Hlt1 (Alleys)
    bool hlt1Decision, dummyTis, dummyTos;
    m_TriggerTisTosTool->setOfflineInput();
    m_TriggerTisTosTool->triggerTisTos("Hlt1*Decision",hlt1Decision,dummyTis,dummyTos, ITriggerTisT
    // this also works: m_TriggerTisTosTool->triggerTisTos("Hlt1*Decision",hlt1Decision,dummyTis,dummyTos);
    info() << " Hlt1 decision= " << hlt1Decision;
    info() << " from OR between "; dumpvts( m_TriggerTisTosTool->triggerSelectionNames() ); info() <

    // Hlt2 (Selections)
    bool hlt2Decision;
    m_TriggerTisTosTool->triggerTisTos("Hlt2*Decision",hlt2Decision,dummyTis,dummyTos, ITriggerTisT
    info() << " Hlt2 decision= " << hlt2Decision;
    info() << " from OR between "; dumpvts( m_TriggerTisTosTool->triggerSelectionNames() ); info() <

    // see which Hlt1 triggers were on
    //done m_TriggerTisTosTool->setOfflineInput();
    std::vector<std::string> hlt1Pass = m_TriggerTisTosTool->triggerSelectionNames(
    "Hlt1*Decision",ITriggerTisTos::kAllTriggerSelections,
    ITriggerTisTos::kTrueRequired, ITriggerTisTos::kAnything, ITriggerTisTos::kAnything);
    info() << " Hlt1 triggers that fired="; dumpvts( hlt1Pass ); info() << endmsg;

    // see which Hlt2 Selections were on
    //done m_TriggerTisTosTool->setOfflineInput();
    std::vector<std::string> hlt2Pass = m_TriggerTisTosTool->triggerSelectionNames(
    "Hlt2*Decision",ITriggerTisTos::kAllTriggerSelections,
    ITriggerTisTos::kTrueRequired, ITriggerTisTos::kAnything, ITriggerTisTos::kAnything);
    info() << " Hlt2 Selections that succeeded="; dumpvts( hlt2Pass ); info() << endmsg;

    // Tis,Tos with respect to input particles
    for( std::vector<const Particle*>::const_iterator iPart = parts.begin(); iPart!= parts.end(); +

        // Particle Info -----
        const std::vector<const Particle*> finals = descendants()->finalStates(*iPart);
        info() << " Particle " << (*iPart)->particleID().pid() << " Pt " << (*iPart)->pt()
            << " #-of-daughters " << ((*iPart)->daughtersVector()).size()
            << " #-of-final-state-part " << finals.size();
        for( std::vector<const Particle*>::const_iterator pf=finals.begin();pf!=finals.end();++pf){
            info() << " " << (*pf)->particleID().pid() << " Pt " << (*pf)->pt();
        }
        info() << endmsg;

    // L0 Info -----

```

TisTosDVExample < LHCb < TWiki

```

bool decisionL0,tisL0,tosL0;
m_TriggerTisTosTool->triggerTisTos(**iPart,"L0*Decision",decisionL0,tisL0,tosL0, ITriggerTisT
if( !decisionL0 )continue;
info() << "          L0          TIS= " << tisL0 << " TOS=" << tosL0 << endmsg;

if( tisL0 ){
    info() << "          L0 TIS selections= ";
    dumpvvs( m_TriggerTisTosTool->triggerSelectionNames( ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kTrueRequired,
                                                         ITriggerTisTos::kAnything ) );

    info() << endmsg;
}
if( tosL0 ){
    info() << "          L0 TOS selections= ";
    dumpvvs( m_TriggerTisTosTool->triggerSelectionNames( ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kTrueRequired ) );

    info() << endmsg;
}

// Hlt1 Info -----

bool decisionHlt1,tisHlt1,tosHlt1;
m_TriggerTisTosTool->triggerTisTos("Hlt1*Decision",decisionHlt1,tisHlt1,tosHlt1, ITriggerTisT
if( !decisionHlt1 )continue;
info() << "          Hlt1          TIS= " << tisHlt1 << " TOS=" << tosHlt1 << endmsg;

if( tisHlt1 ){
    info() << "          Hlt1 TIS selections= ";
    dumpvvs( m_TriggerTisTosTool->triggerSelectionNames( ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kTrueRequired,
                                                         ITriggerTisTos::kAnything ) );

    info() << endmsg;
}
if( tosHlt1 ){
    info() << "          Hlt1 TOS selections= ";
    dumpvvs( m_TriggerTisTosTool->triggerSelectionNames( ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kTrueRequired ) );

    info() << endmsg;
}

// Hlt2 Info -----

bool decisionHlt2,tisHlt2,tosHlt2;
m_TriggerTisTosTool->triggerTisTos("Hlt2*Decision",decisionHlt2,tisHlt2,tosHlt2, ITriggerTisT
if( !decisionHlt2 )continue;
info() << "          Hlt2 TIS= " << tisHlt2 << " TOS=" << tosHlt2 << endmsg;

if( tisHlt2 ){
    info() << "          Hlt2 TIS selections= ";
    dumpvvs( m_TriggerTisTosTool->triggerSelectionNames( ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kTrueRequired,
                                                         ITriggerTisTos::kAnything ) );

    info() << endmsg;
}
if( tosHlt2 ){
    info() << "          Hlt Hlt2 TOS selections= ";
    dumpvvs( m_TriggerTisTosTool->triggerSelectionNames( ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kAnything,
                                                         ITriggerTisTos::kTrueRequired ) );

    info() << endmsg;
}

} // end particle loop

// Mandatory. Set to true if event is accepted.

```

TisTosDVExample < LHCb < TWiki

```
setFilterPassed(true); // Ignore returned status
return StatusCode::SUCCESS;

return StatusCode::SUCCESS;

}

//=====
// Finalize
//=====
StatusCode TisTosDVExample::finalize() {

    debug() << "=> Finalize" << endmsg;

    return DVAlgorithm::finalize();
}

//=====
```

header file

```
#ifndef TISTOSDVEXAMPLE_H
#define TISTOSDVEXAMPLE_H 1

// Include files
// from DaVinci, this is a specialized GaudiAlgorithm
#include "Kernel/DVAlgorithm.h"

#include "Kernel/ITriggerTisTos.h"

/** @class TisTosDVExample TisTosDVExample.h
 *
 *
 * @author Tomasz Skwarnicki
 * @date 2007-09-24
 */
class TisTosDVExample : public DVAlgorithm {
public:
    /// Standard constructor
    TisTosDVExample( const std::string& name, ISvcLocator* pSvcLocator );

    virtual ~TisTosDVExample( ); ///< Destructor

    virtual StatusCode initialize(); ///< Algorithm initialization
    virtual StatusCode execute ( ); ///< Algorithm execution
    virtual StatusCode finalize ( ); ///< Algorithm finalization

    void dumpvs( const std::vector< std::string > & vs )
    {
        for( std::vector< std::string >::const_iterator s=vs.begin();s != vs.end(); ++s)
        {
            info() << " " << *s;
        }
    }

protected:

private:

    ITriggerTisTos* m_TriggerTisTosTool;

};
#endif // TISTOSDVEXAMPLE_H
```

Sample output for 1 event in B -> K* mu mu MC

```
TisTosDVExample      INFO  L0 decision= 1
TisTosDVExample      INFO  Names of L0 triggers in Hlt= L0DiMuonDecision L0
TisTosDVExample      INFO  Hlt final decision= 1
TisTosDVExample      INFO  Hlt1 decision= 1 from OR between  Hlt1ElectronEl
TisTosDVExample      INFO  Hlt2 decision= 1 from OR between  Hlt2SelB2Dplus
TisTosDVExample      INFO  Hlt1 triggers that fired= Hlt1MuTrackDecision HL
TisTosDVExample      INFO  Hlt2 Selections that succeeded= Hlt2SelBd2MuMuKs
TisTosDVExample      INFO  Particle 511 Pt 7568.09 #-of-daughters 2 #-of
TisTosDVExample      INFO  L0                TIS= 0 TOS=1
TisTosDVExample      INFO  L0 TOS selections= L0DiMuonDecision L0
TisTosDVExample      INFO  Hlt1 TIS= 0 TOS=1
TisTosDVExample      INFO  Hlt1 TOS selections= Hlt1MuTrackDecisi
TisTosDVExample      INFO  Hlt2 TIS= 0 TOS=1
TisTosDVExample      INFO  Hlt Hlt2 TOS selections= Hlt2SelBd2MuM
```

-- TomaszSkwarnicki - 01 Oct 2007, 15 Sep 2008

This topic: LHCb > TisTosDVExample

Topic revision: r4 - 2009-12-15 - unknown



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)