

# MicroBooNE GENIEGen and PPFX Reweighting Updates

Larsoft Coordination Meeting  
June 27, 2023

H. Greenlee

# Overview

- MicroBooNE has been using ppfx-reweighting for simulating NUMI events for a long time in our MCC9 series of production releases.
- Recently we have been working to migrate support for our MCC9 work flows into the larsoft integration release.
  - This talk is about migrating a set of features related to ppfx-reweighting of genie events.
  - Includes merge/pull requests for larsim and ppfx packages.

# Larsim Updates Summary

- Update GENIEGen module to better support flux type “dk2nu.”
  - This is the only modification of any existing larsim code (see following slide).
  - Nonbreaking.
- Add 13 ppfx event weight calculators.
  - In existing directory larsim/EventWeight/Calculators
  - Calculators work with EventWeight module (larsim). EventWeight module itself does not need to be modified.
- Add source module PPFXFluxReader.
  - New directory larsim/PPFXFluxReader.
- Update various CMakeLists.txt files and product\_deps because of added dependencies on ups products dk2nugenie, dk2nudata, and ppfx.
  - Products dk2nugenie, dk2nudata, and ppfx are already built and included with larsoft integration release, but larsim does not currently depend on them.

# GENIEGen Update

- Producer module GENIEGen updated to better support flux type “dk2nu.”
  - The flux type is a fcl parameter that is passed directly into GENIEHelper (nugen), normally without interpretation.
  - GENIEHelper already supports flux type “dk2nu.”
  - The update to GENIEGen makes flux type “dk2nu” special by adding additional data products into the event.
    - Data product `bsim::Dk2Nu`.
    - Data product `bsim::NuChoice`.
    - Associations.
  - Above data products are needed to run newly added PPFX weight calculators via the EventWeight module.
  - If flux type is anything other than “dk2nu,” no additional data products are added to the event.
  - Full GENIEGen updates in backup.

# Weight Calculators

- For the record, here are the 13 added weight calculators in larsim.
  - PPFXCVWeightCalc.cxx
  - PPFXMIPPKaonWeightCalc.cxx
  - PPFXMIPPPionWeightCalc.cxx
  - PPFXOtherWeightCalc.cxx
  - PPFXTargAttenWeightCalc.cxx
  - PPFXThinKaonWeightCalc.cxx
  - PPFXThinMesonWeightCalc.cxx
  - PPFXThinNeutronPionWeightCalc.cxx
  - PPFXThinNucAWeightCalc.cxx
  - PPFXThinNucWeightCalc.cxx
  - PPFXThinPionWeightCalc.cxx
  - PPFXTotAbsorpWeightCalc.cxx
  - PPFXWeightCalc.cxx
- All of these weight calculators access data product bsim::Dk2Nu (bsim::NuChoice not used).

# PPFXFluxReader

- PPFXXFluxReader is a newly added stand alone art source module.
  - Runs on top of genie and supports flux types “dk2nu” and “simple.”
  - Generates events containing typical generator data products `simb::MCTruth` and `simb::MCFlux`, as well as dk2nu data products `bsim::Dk2Nu` and `bsim::NuChoice`.
  - Not sure what independent significance PPFXXFluxReader has compared to GENIEGen.
  - Not used in standard MicroBooNE production work flows.
    - Not compatible with our way of running overlay MC.

# PPFX Updates

- MicroBooNE made several (minor) updates in ppfx that were never propagated into the integration release.
  - The most recent such update was made in larsoft v08\_05\_00\_20 (ppfx v2\_11\_04).
    - This is the one that we are mainly interested to getting into the integration release now.
    - However there are other updates in v02\_11\_xx versions that were never propagated into the production release that are also included.

# PPFX Package Status

- To my knowledge, larsoft integration release is still building ppfx and dk2nu packages out of redmine github repos.
- The ppfx redmine github repo contains two relevant branches.
  - `lar_v2_11_br`
    - PPFx versions v02\_11\_xx.
    - Used in MicroBooNE MCC9 production releases.
    - Latest version v02\_11\_04 (updated in larsoft v08\_05\_00\_20, Jan., 2023).
  - `lar_v2_16_br`
    - PPFx versions v02\_16\_xx and v02\_17\_xx.
    - Used in recent larsoft integration releases.
    - Latest version v02\_17\_07 (updated in larsoft v09\_75\_02).
- The ppfx development team is using Kordosky's personal github repo as their main repo ([github.com/kordosky/ppfx](https://github.com/kordosky/ppfx)).
  - An  $x_F$  bug was recently found, which is being fixed in K's repo.
  - MicroBooNE hasn't yet decided what we want to do about the  $x_F$  bug (not included in requested updates in this talk).



# PPFX Updates

- What MicroBooNE is requesting now is to merge revisions from the latest MCC9 version of ppfx into integration release.
  - Main change is adding ability to set base universe number (effectively a random number seed) in class ppfx class MakeReweight.
    - Adding class method MakeReweight::SetBaseSeed(int).
      - Previously base universe / seed was some hard-coded value.
  - Other changes.
    - Add directory \$PPFX\_DIR/xml to framework search path in ppfx table file.
    - Bug fixes where some config files were being read from source directory instead of install directory (mrb build environment vs. run environment bugs).
  - Full updates in backup.

# Larsoft Merge Branches

- Merge branches for larsim and ppx can be found in hgreenlee's personal github ([github.com/hgreenlee/larsim](https://github.com/hgreenlee/larsim) and [github.com/hgreenlee/ppx](https://github.com/hgreenlee/ppx)).
  - **Larsim branch** `greenlee_ppx`
  - **PPFX branch** `greenlee_set_seed`
- Branches are updated and tested through integration release larsoft v09\_75\_02.

# MicroBooNE-Specific Features and Updates

- Package uboonecode merge branch `greenlee_ppfx` adds two unit tests to run genie in “simple” and “dk2nu” modes (including ppfx reweighting in the latter case).
- The MCC9 version of ubsim (branch `v08_00_00_br`) includes two ppfx reweight calculators that are not included in larsim.
  - `UBPPFXCVWeightCalc.cxx`
  - `UBPPFXWeightCalc.cxx`
  - No specific merge branch for integration release yet.

# GENIEGen Updates

# GENIEGen\_module.cc

```
-----  
↓  
↑  
@@ -166,6 +176,14 @@ namespace evgen {  
  
166     produces<art::Assns<simb::MCTruth, simb::GTruth>>();  
167     produces<std::vector<sim::BeamGateInfo>>();  
168  
  
169     std::string beam_type_name = pset.get<std::string>("BeamName");  
170  
171     if (beam_type_name == "numi")  
  
-----  
↓  
↑  
@@ -374,6 +392,17 @@ namespace evgen {  
  
374     std::unique_ptr<std::vector<sim::BeamGateInfo>> gateCollection(  
375         new std::vector<sim::BeamGateInfo>);  
376  
  
377     while (truthcol->size() < 1) {  
378         while (!fGENIEHelp->Stop()) {  
379  
  
-----  
↓  
↑  
@@ -396,6 +425,17 @@ namespace evgen {  
  
176     produces<art::Assns<simb::MCTruth, simb::GTruth>>();  
177     produces<std::vector<sim::BeamGateInfo>>();  
178  
179 +     // dk2nu additions  
180 +     if (pset.get<std::string>("FluxType").find("dk2nu") != std::string::npos) {  
181 +         produces< std::vector<bsim::Dk2Nu> >();  
182 +         produces< std::vector<bsim::NuChoice> >();  
183 +         produces< art::Assns<simb::MCTruth, bsim::Dk2Nu> >();  
184 +         produces< art::Assns<simb::MCTruth, bsim::NuChoice> >();  
185 +     }  
186 +  
187     std::string beam_type_name = pset.get<std::string>("BeamName");  
188  
189     if (beam_type_name == "numi")  
  
392     std::unique_ptr<std::vector<sim::BeamGateInfo>> gateCollection(  
393         new std::vector<sim::BeamGateInfo>);  
394  
395 +     std::unique_ptr< std::vector<bsim::Dk2Nu> >  
396 +         dk2nucol(new std::vector<bsim::Dk2Nu>);  
397 +     std::unique_ptr< std::vector<bsim::NuChoice> >  
398 +         nuchoicecol(new std::vector<bsim::NuChoice>);  
399 +     std::unique_ptr< art::Assns<simb::MCTruth, bsim::Dk2Nu> >  
400 +         dk2nuassn(new art::Assns<simb::MCTruth, bsim::Dk2Nu>);  
401 +     std::unique_ptr< art::Assns<simb::MCTruth, bsim::NuChoice> >  
402 +         nuchoiceassn(new art::Assns<simb::MCTruth, bsim::NuChoice>);  
403 +  
404 +     genie::flux::GDk2NuFlux* dk2nuDriver =  
405 +         dynamic_cast<genie::flux::GDk2NuFlux*>(fGENIEHelp->GetFluxDriver(true));  
406     while (truthcol->size() < 1) {  
407         while (!fGENIEHelp->Stop()) {  
408  

```

# GENIEGen\_module.cc

@@ -396,6 +425,17 @@ namespace evgen {	
396	425
397	426
398	427
	428 +
	429 +
	430 +
	431 +
	432 +
	433 +
	434 +
	435 +
	436 +
	437 +
	438 +
399	439
400	440
401	441
@@ -439,6 +479,14 @@ namespace evgen {	
439	479
440	480
441	481
	482 +
	483 +
	484 +
	485 +
	486 +
	487 +
	488 +
	489 +
442	490
443	491
444	492

# PPFX Updates

# MakeReweight.h

```
8 include/MakeReweight.h

@@ -67,6 +67,11 @@ namespace NeutrinoFluxReweight{
67     //!< Reweigher Drivers for the central value
68     ReweightDriver* cv_rw;
69
70     private:
71     /*!
72     * Inititalize the job and configurethe
73     ReweigherDrivier
74
75     private:
76     /*!
77     * Inititalize the job and configurethe
78     ReweigherDrivier
79
80     + bool AlreadyInitialized() {return init;};
81     +
82     + //!< Override the base universe seed used
83     + void setBaseSeed(int val);
84     +
85
86     @@ -84,6 +89,9 @@ namespace NeutrinoFluxReweight{
84     std::vector<double> vec_wgts;
85     std::map<std::string, std::vector<double> >
86     map_rew_wgts;
87     double cv_wgt;
88
89     + int base_universe = 1000000;
90     +
91     + bool init = false;
92
93     static MakeReweight* instance;
94
95     static MakeReweight* instance;
96
97
98
99
100
```



# MakeReweight.cpp

```
▼ 9 ■■■■■ src/MakeReweight.cpp ...  
↑ ... @@ -62,7 +62,6 @@ namespace NeutrinoFluxReweight{  
62     vec_rws.reserve(Nuniverses);  
63     std::cout<<"Initializing reweight drivers for  
    "<<Nuniverses<<" universes"<<std::endl;  
64  
65 -   const int base_universe=1000000;  
66     // cvPars.reserve(Nuniverses+1);  
67     univPars.reserve(Nuniverses+1);  
68  
↓ ...  
↑ ... @@ -90,6 +89,9 @@ namespace NeutrinoFluxReweight{  
90  
91     std::cout<<"Done configuring universes"  
    <<std::endl;  
92  
93     }  
94  
95     std::vector<double> MakeReweight::GetTotalWeights(){  
↓ ...  
↑ ... @@ -171,4 +173,9 @@ namespace NeutrinoFluxReweight{  
171     delete instance;  
172     instance = 0;  
173     }  
174     }  
↓ ...  
↑ ... @@ -171,4 +173,9 @@ namespace NeutrinoFluxReweight{  
173     delete instance;  
174     instance = 0;  
175     }  
176 +   ///  
177 +   void MakeReweight::setBaseSeed(int val) {  
178 +       base_universe = val;  
179 +       std::cout << "Updated base universe: " <<  
        base_universe << std::endl;  
180 +   }  
181     }  
174     }
```

# Bug fixes

```
scripts/rwgt_job.sh
```

@@ -47,7 +47,7 @@ echo ls -l \$_CONDOR_SCRATCH_DIR/inputs	
47	47
48 for f in `ls \$_CONDOR_SCRATCH_DIR/inputs/*.root`; do	48 for f in `ls \$_CONDOR_SCRATCH_DIR/inputs/*.root`; do
49 FOUT="out_`\${basename \$f}`	49 FOUT="out_`\${basename \$f}`
50 - \$PPFX_DIR/bin/doReweight_dk2nu \$f \$FOUT	50 + \$PPFX_DIR/bin/doReweight_dk2nu \$f \$FOUT
\$PPFX_DIR/scripts/inputs_default.xml	\$PPFX_DIR/xml/inputs_default.xml
51 done	51 done
52	52
53 echo "ls -lh on working directory:"	53 echo "ls -lh on working directory:"

```
src/FillIMapHists.cpp
```

@@ -37,7 +37,7 @@ double FillIMapHists(TChain* tdk2nu, TChain* tdkmeta, HistList* hist	
37	37
38 const char* ppfxDir = getenv("PPFX_DIR");	38 const char* ppfxDir = getenv("PPFX_DIR");
39 MakeReweight* makerew = MakeReweight::getInstance();	39 MakeReweight* makerew = MakeReweight::getInstance();
40 - makerew->SetOptions(Form("%s/scripts	40 + makerew->SetOptions(Form("%s/xml
/inputs_imap.xml", ppfxDir));	/inputs_imap.xml", ppfxDir));
41	41
42 FillIMapHistsReweighters reweighters;	42 FillIMapHistsReweighters reweighters;
43 reweighters.NumiPions =	43 reweighters.NumiPions =
(makerew->cv_rw)->MIPP_NUMI_PION_Universe;	(makerew->cv_rw)->MIPP_NUMI_PION_Universe;

# Framework Search Path

```

ups/product_deps
@@ -176,6 +176,10 @@ defaultqual      e20
176  bindir fq_dir      bin
177  incdir product_dir
178  libdir fq_dir      lib
179  + table_fragment_begin
180  +   pathPrepend(FW_SEARCH_PATH, ${PPFX_DIR}/xml)
181  + table_fragment_end
182  +
183  #####
184
185  #####

```