Dark Matter Workshop in UAB - 17th-19th January 2019

Hands-on session for Pointing optimization for IACTs

Joaquim Palacio
(on behalf of J. Rico and D. Navarro-Gironés)

Open-source tool for pointing optimisation of IACTs

C++

JDAstroProfile

- dN/dΩ
- Sky coord.

JDDarkMatter

- JFactor

Inheritance

(...)

- …

JDOptimization

- JD = Joaquim David
- Q (Q_{eff}, Q_{σ}, Q_{Acc}, …)
- θ_{opt} , w_{opt}

Composition

JDInstrument

- IACT
  - FoV
  - PSF
  - Acceptance
- Camera coord.

JDInstrument

Cares about:

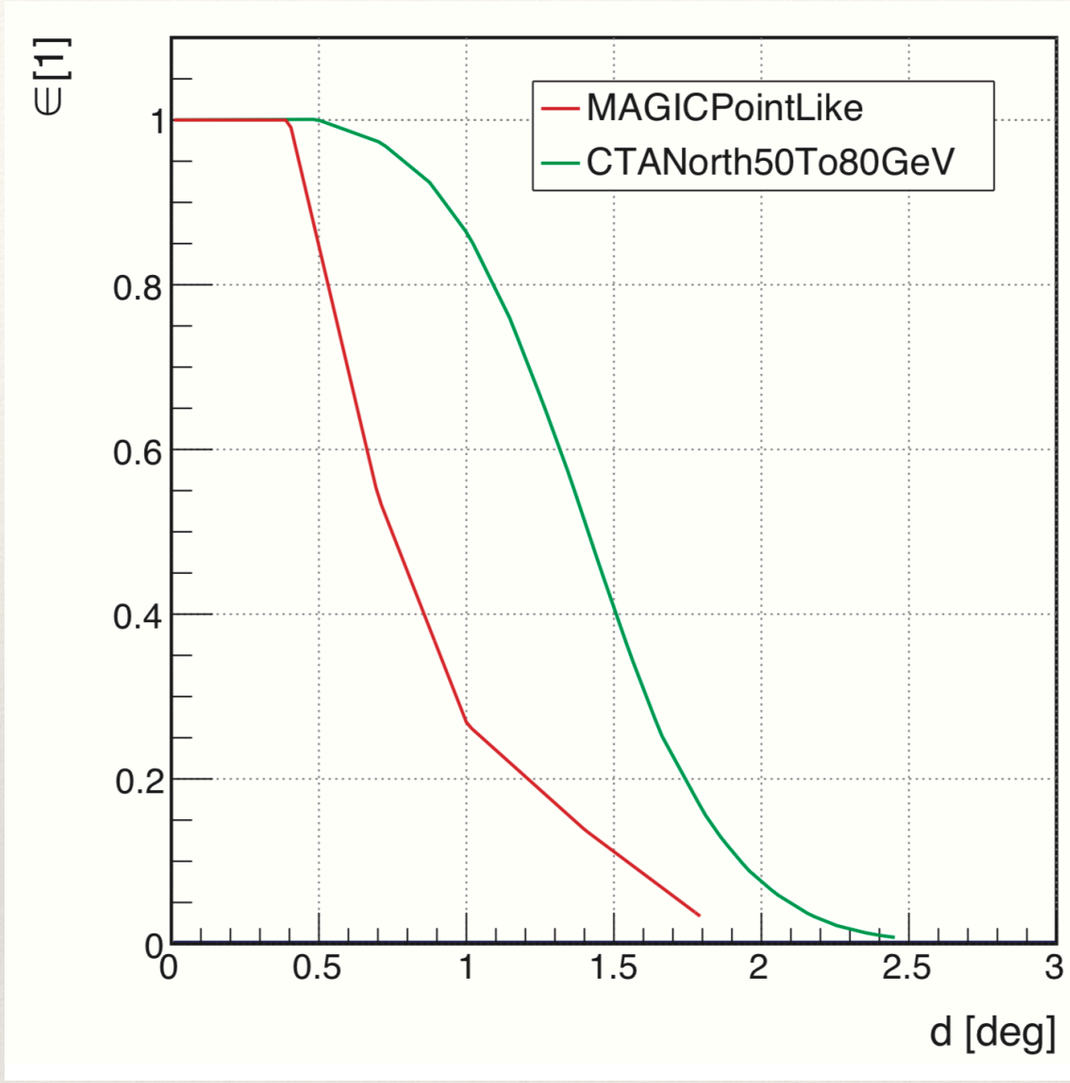
- Acceptance (ε)
- Wobble distance (leakage)

Known IACTs:

- Ideal case (ε=1)
- MAGIC PointLike

  Aleksić et al., arxiv.org/abs/1409.5594
- CTA North 50-80GeV

  www.cta-observatory.org



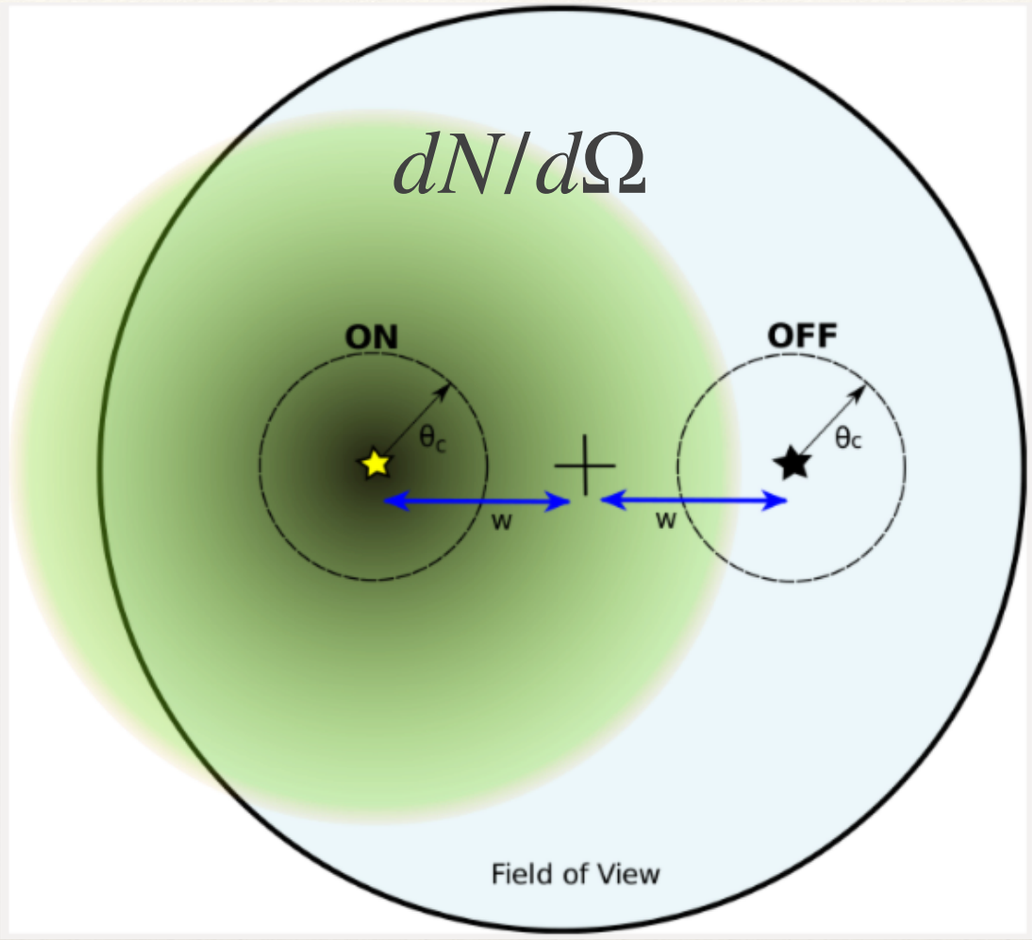
New IACTs through…

- *JDInstrument(TGraph* cameraAcceptance, Double_t wobbleDist);*
- *JDInstrument(TString txtFile, Double_t wobbleDist);*

JDAstroProfile

Cares about:

- dN/dΩ
- Uncertainties in dN/dΩ
- Leakage (w)



WARNING

Ideally, one should be able to define an arbitrary dN/dΩ, however, due to historical reasons, JDAstroProfile was born from JDDarkMatter (and not the other way around). Some inconsistencies may appear in the code…

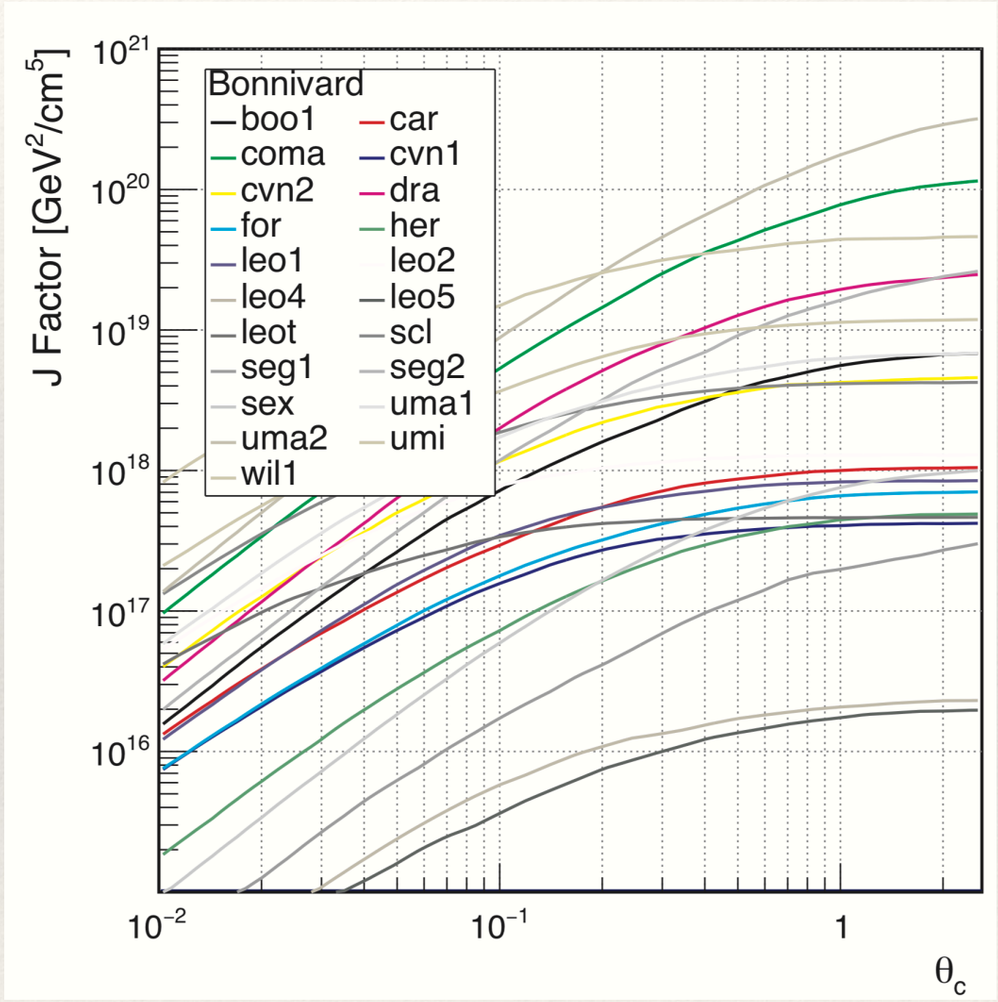
JDDarkMatter

Known dwarfs:

- Bonnivard et al., Mon. Not. Roy. Astron. Soc. 453 (2015) 849-867
- Geringer-Sameth et al., ApJ 801 (2015) 74

New sources:

- *Bool_t SetJFactorFromTGraph()*
- *Bool_t SetJFactorFromTxtFile()*



Knows how to relate **Jfactor** and **dN/dΩ**

$$dN/d\Omega = dN/d\Omega(JFactor)$$

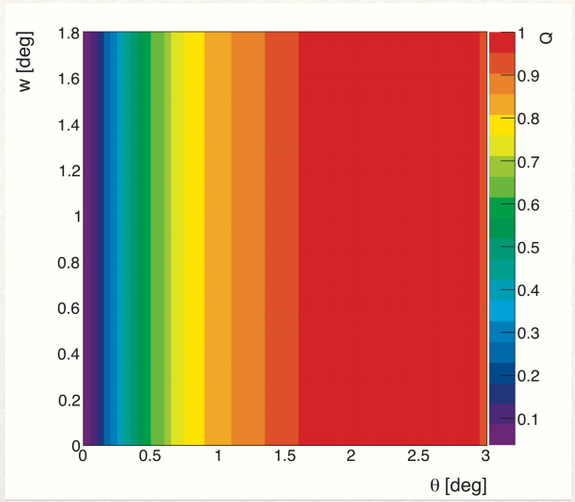
JDOptimization

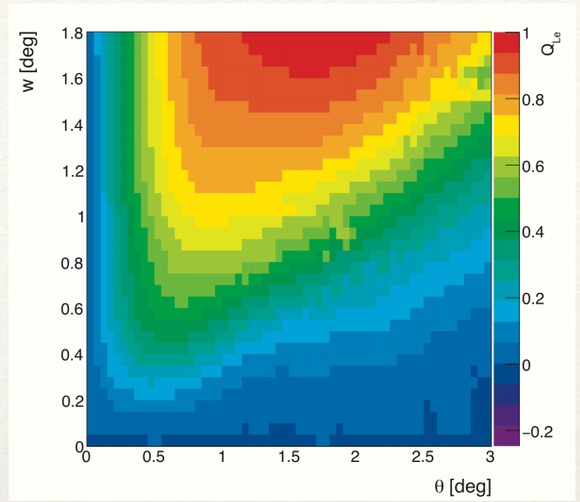
Obtain:

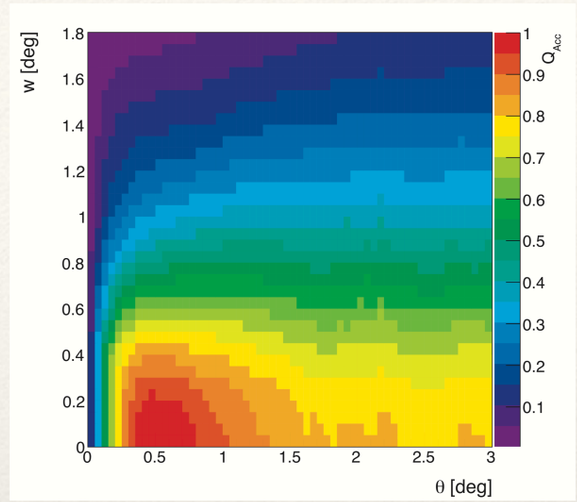
- w_{opt} and θ_{opt}

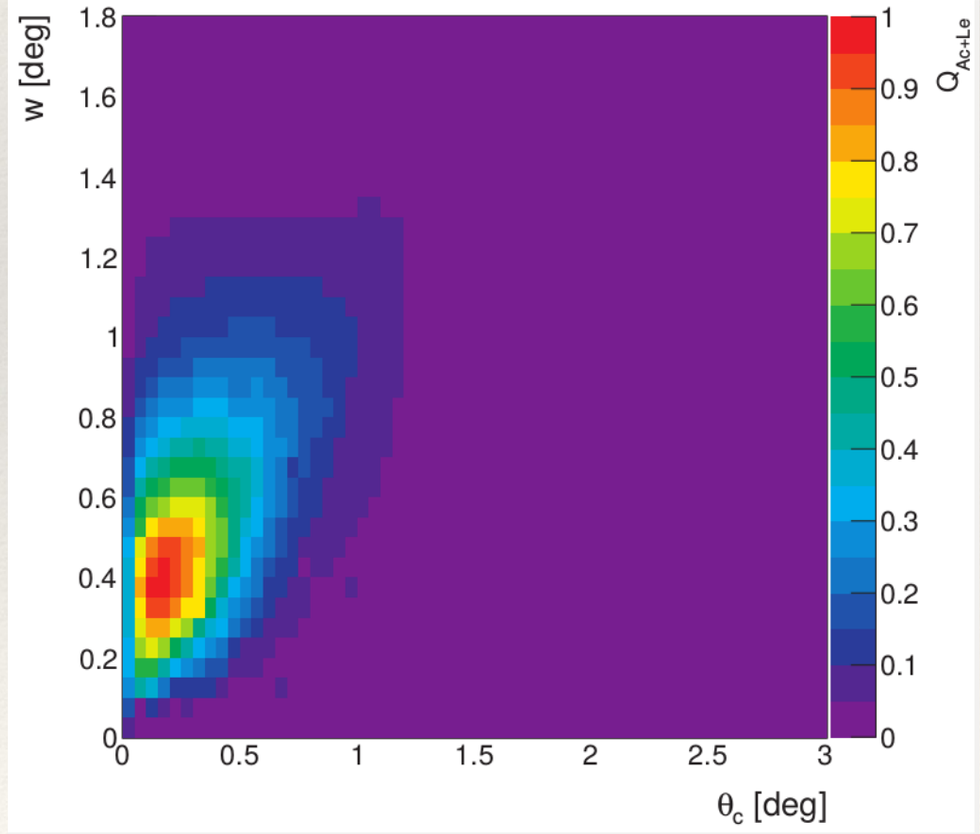
Compute Q:

- J/θ -> $[N_{on}/\sqrt{(N_{off})}]$
  - Instrument Acceptance
  - Leakage effect
  - Instrument PSF
  - ($dN/d\Omega$ uncertainty)
  - **Any of the former combinations**









Main Getters:

- *TF1* GetTF1QFactorVsTheta(…)*
- ***TH2D* GetTH2QFactorVsThetaWobble**(Int_t type=0, …)*
- *void GetOptimalThetaAndWobble(…)*

Hands-on time ...

> *https://github.com/IndirectDarkMatterSearchesIFAE/*
> *git clone https://github.com/IndirectDarkMatterSearchesIFAE/ObservationOptimization.git*
> *git checkout V1.0*

> *ls*
< ***macros plots references source***

> *ls source/**
< ***JDAstroProfile.(cc,h) JDDarkMatter.(cc,h) JDInstrument.(cc,h) JDOptimization.(cc,h)***

> *ls macros/**
< ***exampleJDDarkMatter.cxx exampleJDInstrument.cxx exampleJDOptimization.cxx …***

User friendly environment...

exampleJDDarkMatter.cxx …

-h (from bash)

Bonnivard & Geringer from references

Introduce external JFactors

~dN/dΩ

```
void exampleJDDarkMatter()
{
    PrintListOfPossibilities();

    DrawJFactorFromReferences();
    DrawDifferentialJFactorFromReferences();

    DrawJFactorFromTGraph();
    DrawDifferentialJFactorFromTGraph();
    DrawJFactorFromTxtFile();
}
```

exampleJDInstrument.cxx ...

```
void exampleJDInstrument()
  {
      PrintListOfPossibilities();

      PlotFoVAcceptanceFromMAGIC();
      PlotIntegratedFoVAcceptanceFromMAGIC();

      PlotFoVAcceptanceFromTGraph();
      PlotIntegratedFoVAcceptanceFromTGraph();

      PlotFoVAcceptanceFromTxtFile();
      PlotIntegratedFoVAcceptanceFromTxtFile();
  }
```

MAGIC & CTA from references

-h (…)

$\epsilon(\Omega)$

$\sim \int \epsilon(\Omega)\, d\Omega$

Introduce external Acceptances

exampleJDOptimization.cxx ....

```
void exampleJDOptimization()
   {
      PlotQ0Factor();          //      J/theta
      PlotQ1Factor();          //      J_on-J_off/theta
      PlotQ2Factor();          //      J_1sm/theta
      PlotQ3Factor();          //      J_eff/theta_eff
      PlotQ4Factor();          //      J'/theta'


      PlotQ12Factor();         //      J_on_1sm/Sqrt{theta^2 + J_off_1sm}
      PlotQ13Factor();         //      J_on_eff/Sqrt{(theta_eff)^2 + J_off_eff}
      PlotQ23Factor();         //      J_1sm_eff/theta_eff
         (…)
      PlotQ1234Factor();       //      J'_on_1sm_eff/Sqrt{(theta'_eff)^2 + J'_off_1sm_eff}
   }
```

loopResults_Q134.cxx ...

```
void loopResults_Q134()
   {
      BonnivardDecayLoop();
      BonnivardAnnihilationLoop();
      GeringerDecayLoop();
      GeringerAnnihilationLoop();
   }
```

(loop)

```
JDOptimization* Q13Factor2D = new JDOptimization(author, source, candidate, mySourcePath,
   myInstrumentPath, instrumentName, distCameraCenter, wobble);
TH2D* Histogram = Q13Factor2D->GetTH2QFactorVsThetaWobble(134);

Q13Factor2D->GetOptimalThetaAndWobble(thetaOpt, thetaOptRangMin, thetaOptRangMax, wobbleOpt,
   wobbleOptRangMin, wobbleOptRangMax, 134);
```

Expert mode…

JDAstroProfile.h ...

```
Double_t TGraphdNdOmegaVsTheta(Double_t* x, Double_t* par);
Double_t TGraphdNdOmegaSigma1VsTheta(Double_t* x, Double_t* par);

Double_t NormdNdOmegaVsTheta(Double_t* x, Double_t* par);
Double_t NormdNdOmegaSigma1VsTheta(Double_t* x, Double_t* par);

Double_t Integrated**NdOmega**ThetaVsTheta(Double_t* x, Double_t* par);
Double_t IntegratedNdOmega**Sigma1**ThetaVsTheta(Double_t* x, Double_t* par);
Double_t Integrated**NdOmegaOff**ThetaVsTheta(Double_t* x, Double_t* par);
Double_t IntegratedNdOmegaSigma1OffThetaVsTheta(Double_t* x, Double_t* par);

Double_t dNdOmegaVsTheta(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1VsTheta(Double_t* x, Double_t* par);
Double_t dNdOmegaOffVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1OffVsThetaPhi(Double_t* x, Double_t* par);

Double_t dNdOmegaThetaVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1ThetaVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaOffThetaVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1OffThetaVsThetaPhi(Double_t* x, Double_t* par);
```

JDInstrument.h ...

```
Double_t EpsilonVsDcc(Double_t* x, Double_t* par);
Double_t EpsilonVsThetaPhi(Double_t* x, Double_t* par);
Double_t EpsilonVsXAndY(Double_t* x, Double_t* par);
Double_t EpsilonThetaVsThetaPhi(Double_t* x, Double_t* par);
Double_t EfficiencyVsTheta(Double_t* x, Double_t* par);

Double_t IntegrateEpsilonThetaVsTheta(Double_t* x, Double_t* par);
```

JDOptimization.h ...

```
// ...VsTheta
Double_t dNdOmega**Epsilon**VsTheta(Double_t* x, Double_t* par);
Double_t dNdOmega**Smeared**VsTheta(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1SmearedVsTheta(Double_t* x, Double_t* par);

// ...VsThetaPhi
Double_t dNdOmegaSmearedVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaEpsilonVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSmearedEpsilonVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1SmearedEpsilonVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSmearedEpsilonOffVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1SmearedEpsilonOffVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSmearedOffVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1EpsilonVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1OffEpsilonVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaSigma1SmearedOffVsThetaPhi(Double_t* x, Double_t* par);
Double_t dNdOmegaOffEpsilonVsThetaPhi(Double_t* x, Double_t* par);

// ...ThetaVsThetaPhi
Double_t dNdOmegaSmearedThetaVsThetaPhi(Double_t* x, Double_t* par);
(...)

// Integrate...ThetaVsTheta
Double_t IntegratedNdOmegaEpsilonThetaVsTheta(Double_t* x, Double_t* par);
(...)
```