

Fitting

Now that we have learned how to [create functions](#) and to configure the [fitter](#) we are ready to start fitting. The following code shows how to perform a simple chi squared fit to a one dimensional histogram:

```
// The AIDA Factories
IAnalysisFactory af = IAnalysisFactory.create();
ITree = af.createTreeFactory().create();
IFunctionFactory ff = af.createFunctionFactory(tree);
IFitter fitter = af.createFitFactory().createFitter("chi2");

// Fit a gaussian to a given histogram h1
IFitResult result = fitter.fit(h1,"g");
```

Cloning When Fitting

When a function is provided to the fitter the user has to be aware that the AIDA prescription is to leave the input function unchanged, create a clone, perform the fit on the clone and return the clone in the [fit result](#). There advantage to this approach is that the input function can be reused as is for multiple identical fits while the fitted-cloned function is left unchanged in the fit result. To override the above behavior the user can create a fitter by setting the *noClone* option to *true* in the options:

```
// Create a fitter that does not clone the input function
IFitter fitter = af.createFitFactory().createFitter("chi2","minuit","noClone=true");
```

Fit Result

The outcome of a fit is an [IFitResult](#). It contains the fitted function, the error on the parameters, the covariance matrix etc. The advantage of putting the result of the fit into a result object is that additional fits, error analysis, contours or scans can be easily calculated by using the information stored in it.