**Mining Contract Events in Javascript using Eris-Contracts**

When you use the javascript library eris-contracts, you have to options to deal with events.

Let’s suppose you have an event,

**fired(uint param1, bytes32 param2)**

In your javascript, you have two ways to mine the event. Suppose you want every instance of the event. Let’s say your contract object in javascript is denoted **myContract**. Let’s say you want to listen for all events, but stop as soon as a condition is met, for example **param1 > 5**. You would do so as follows. There are actually two callback functions you use. The first one is for the continuous mining of events. The second is for the arguments of the event.

var myEvent;

myContract.fired(

function(error,result)

{

myEvent = result;

},

function(error,result)

{

if(result.args.param1 > 5)

{

myEvent.stop()

}

})

Note that in the second function, result is a JSON object. So you need to do result.args to access the parameters the event is supposed to carry.

Now, let’s say you want to capture the next instance of the event (whenever it is call) and stop listening. You don’t need to specify the first function. All you need to do is specify **once** as follows:

myContract.fired.once(function(error,result)

{

//result holds the event information

})

The listener will stop as soon as the next event is done.