IO engine variable passed

```
adios2_adios *adios = adios2_init_config(config_file, cartcomm, 1); // if using ADIOS2 MPI, need
to include debugger.
adios2_io *io = adios2_declare_io(adios, IO_ENGINE); //IO handler declaration
adios2_variable *var_iodata = adios2_define_variable(io, "iodata", adios2_type_double, NDIM,
shape, start, count, adios2_constant_dims_true);

adios2_engine *engine = adios2_open(io, FILENAME, adios2_mode_write);
adios2_step_status status;
adios2_begin_step(engine, adios2_step_mode_update, 10.0, &status);
adios2_put(engine, var_iodata, iodata, adios2_mode_deferred);
adios2_end_step(engine);
adios2_close(engine);
adios2_finalize(adios);
```

```
// if using ADIOS2 MPI, need to include debugger.
adios2_adios *adios = adios2_init_config(config_file, cartcomm, 1);
// IO handler declaration
adios2_io *io = adios2_declare_io(adios, IO_ENGINE);
/*
* constant_dims true variables constant, false variables can change after definition. For every rank this should be true.
* shape is global dimension
* start is local offset
* count is local dimension
*/
adios2_variable *var_iodata = adios2_define_variable(io, "iodata", adios2_type_double, NDIM,
shape, start, count, adios2_constant_dims_true);

adios2_engine *engine = adios2_open(io, FILENAME, adios2_mode_write);
adios2_step_status status;
adios2_begin_step(engine, adios2_step_mode_update, 10.0, &status);
adios2_put(engine, var_iodata, iodata, adios2_mode_deferred);
adios2_end_step(engine);
adios2_close(engine);
adios2_finalize(adios);
```

IO engine string variable passed