

# Internal Refactoring

## Serialisation

In ecflow 4 series, we used boost Serialisation for the client/server communication.

Using the same boost version meant that different versions of ecflow 4.X.X releases had client/server that could be used interchangeably.

However, this caused problems when we wanted to upgrade to a newer boost version since it broke the client/server API.

This has been addressed in ecflow 5, we now use JSON for the client/server communication.

Hence the future release of ecflow 5 series can be done with different boost releases.

## Stats Command

For the stats command in 4 series version, the server returns a struct for the statistics, the client then formats this.

However, this meant we would break client/server API if the struct was changed between releases. When we added new statistics.

In ecflow 5.0.0 we return a string, the server itself will format the string, adding/removing new fields without effecting client/server compatibility.

We still retain the ability to return the struct for test purposes.

## Checkpoint and caching

Whenever the server checkpoints(i.e. saves the definition state), the full definition is written to disk.

The checkpoint uses ecFlow definition format when writing to disk and reading it back into memory. (ecflow checkpointing is faster than boost or even Json serialisation)

In series 5 the server takes advantage of this by caching this as a string before writing it out as the checkpoint file.

This effectively caches the de-serialisation cost.

If any user requests the full definition, we can return the cached string really fast.

## Synchronizing/GUI

In ecflow 4.0 the status change times were not accurate, especially for short-lived tasks. This was because the server to client sync only passed the state change.

This has been improved in ecflow 5.0, where we now also sync for each state change the duration. This duration is relative to the suite clock.

As a result, the status change time is more accurate.

## Generic Attribute

A new generic attribute has been added to the definition in ecflow 5 series.

This will be used to address any new functionality required, without breaking backward compatibility.

```
generic <key> <value1> <value2> .... # comment
```

```
suite a
  generic a
  generic b c f #
  family b
    generic a
    generic b c f #
    task c
      generic a
      generic b c f #
      generic x #
    endfamily
  endsuite
```