Counters

Michael

2017

Data Structure VPP16.09

```
typedef struct vlib_main_t
{
  ...
  /* Error handling. */
  vlib_error_main_t error_main;
  ...
} vlib_main_t;
```

```
typedef struct vlib_node_t
{
  ...
  /* Number of error codes used by this node. */
  u16 n_errors;

  /* Size of scalar and vector arguments in bytes. */
  u16 scalar_size, vector_size;

  /* Handle/index in error heap for this node. */
  u32 error_heap_handle;
  u32 error_heap_index;

  /* Error strings indexed by error code for this node. */
  char **error_strings;
  ...
} vlib_node_t;
```

Example


```
vlib_node_registration_t sample_node;

#define foreach_sample_error \
_(SWAPPED, "Mac swap packets processed")

typedef enum {
#define _(sym,str) SAMPLE_ERROR_##sym,
  foreach_sample_error
#undef _
  SAMPLE_N_ERROR,
} sample_error_t;

static char * sample_error_strings[] = {
#define _(sym,string) string,
  foreach_sample_error
#undef _
};
```

```
VLIB_REGISTER_NODE (sample_node) = {
  .function = sample_node_fn,
  .name = "sample",
  .vector_size = sizeof (u32),
  .format_trace = format_sample_trace,
  .type = VLIB_NODE_TYPE_INTERNAL,

  .n_errors = ARRAY_LEN(sample_error_strings),
  .error_strings = sample_error_strings,

  .n_next_nodes = SAMPLE_N_NEXT,

  /* edit / add dispositions here */
  .next_nodes = {
      [SAMPLE_NEXT_INTERFACE_OUTPUT] = "interface-output",
  },
};
```

```
register_node()
 - vlib_register_errors (vm, n->index, r->n_errors, r->error_strings);
 - node_elog_init (vm, n->index);
```


Example(Cont')


```
vlib_node_increment_counter (vm, sample_node.index,
                             SAMPLE_ERROR_SWAPPED, pkts_swapped)
{

  vlib_node_t *n = vlib_get_node (vm, node_index);
  vlib_error_main_t *em = &vm->error_main;
  u32 node_counter_base_index = n->error_heap_index;
  em->counters[node_counter_base_index + counter_index] += increment;
}
```