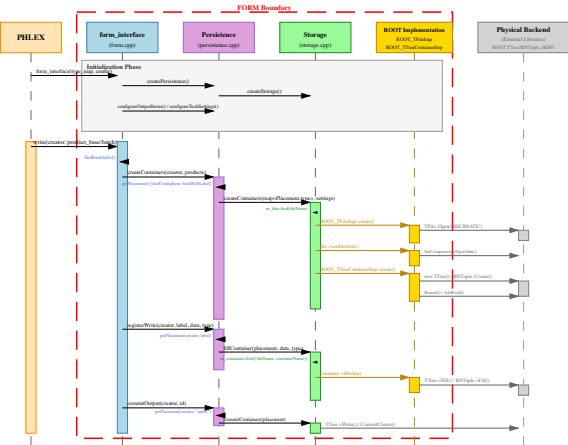


FORM: Write Operation



Storage Layer Details:

- `m_file: map<FileName, Storage_File>`
- `m_containers: map<FileName, containerName>, Storage_Container>`
- Composite key = `(fileName, containerName)`
- Files created once, containers reused
- `setAttribute()` applies code specific settings
- `setWrite()` creates branch/branch columns
- `fill()` maps data, container() writes to disk
- Index container maps ID → entry number
- Mode 'w' = output/write, 'r' = input/read
- NO LOOP: PHLEX calls write() once per event
- FORM processes ONE write request at a time

Placement Structure:

- `FileName: "top-level"`
- `containerName: "Tree_Branch/branchName"`
- `(full_label = creatorLabel)`
- Technology: ROOT_TTree / ROOT_RNTuple / ROOT5

ROOT Implementation Layer (NEW):

- ROOT_TimeContaining Implements Storage_File interface
- Wraps ROOT's TFile class
- Manages file opening, closing, attributes
- ROOT_TimeContaining Implements Storage_Container
- Wraps ROOT's TTree class
- Handles branch creation, filling, committing
- ROOT_Throttle, ContainingMap: For individual branches
- This layer translates FORM storage calls → ROOT API calls

FORM Boundary:

- Not too clever: FORM components
- PHLEX (left) calls FORM, but is separate
- Physical Backend (right) is external ROOT library
- Gold sources → calls to ROOT implementation layer
- Gray sources → ROOT implementation calls external library
- FORM = form_interface + Persistence + Storage