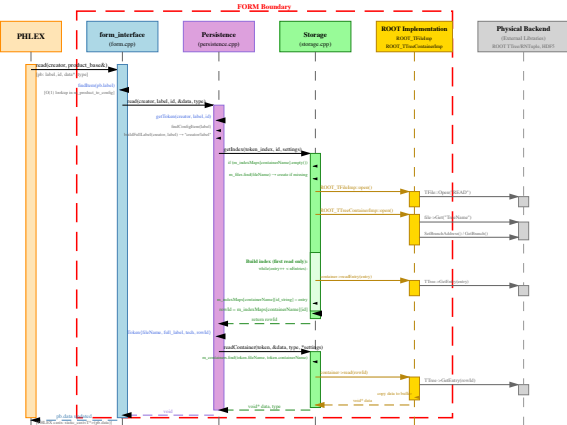


## FORM: Read Operation



#### Storage Layer - Read Details:

- `in_files`: Lazy open on first read
- `in_containers`: Cached after creation
- `in_indexMap[containerName][id_string]` = `iswki`
- Built once by scanning index container
- Cached for subsequent reads
- Mode `T` = input/read mode
- Composites key lookup: `(EltName, containerName)`
- New memory allocated for each read
- `PBEX` responsible for delete
- Type string verified during read
- `XIO::LOCAL::PBEX::utils::id2str` on `id_string` return

### ROOT Implementation Layer (NEW)

- `ROOT_TFileImp` implements `THistogram_File` interface
- Wraps `ROOT's TFile` class
- Manages file opening, mode switching
- `ROOT_TTreeContainerImp` implements `THistogram_FContainer`
- Wraps `ROOT's TTree/TBranch` classes
- Handles `GalaxyIO::SetBranches/AddBranch()`
- `ROOT_TBranchContainerImp` For individual branches
- This layer interfaces FORTRAN source calls to `ROOT API`

### Two-Step Index Lookups

- Step 3: Build index map (first read only)
  - Read all entries from "entries/index" container
  - Each entry contains ID string: "[EVENT-XXXXXXXXXXXX-XXXXXXXXXX]"
  - Map on indexMaps["log\_trunk/index"][id\_string] = entry\_name
- Step 2: Lookup eventID from ID
  - Given semantic ID, get physical entry number
  - Use eventID to read actual data container