EnfoForge

EF-JS-8380-0001

Starter Class Library - White Paper

1	7/19/2025	IFI - Issue for Information	EnfoForge	EnfoForge	EnfoForge
Revision	**Revision Date**	**Issue Purpose**	**Author**	**Checked By**	**Approved By**

Table of Contents

1 Introduction

CFIHOS 2.0 provides information handover specifications. All though, It is a great start, CFIHOS is not a readily usable class library to turn these specifications into a class library that can be used to populate the asset data management system of choice. Each company is left with going through all the pains of building their own class library. This defeats the purpose of a long-term dream of a standard class library that can be used across the industry.

“CFIHOS is a specification, not a Class Library.”

Moreover, the current CFIHOS specification does not cover all the requirements that some owner-operator companies have. Without going into details, there were only about 50% to 60% requirements covered in CFIHOS.

This class library is an attempt to provide a starter class library that is completely CFIHOS compliant with added customizations and enhancements.

This is not just a class library. It is a framework that provides foundation for Engineering Information Management. This framework includes elements of both RDL and Class Library.

The purpose of this class library framework is to promote standard processes within industry, specifically within companies, for efficient data handling.

This class library framework will provide companies with all that is needed to monitor the completeness and correctness of the data.

2 Starter Class Library

2.1 Process

This is how we created this starter class library from CFIHOS specifications. And how we added customizations to make it a framework.

Key customizations and enhancements are:

- Consolidated CFIHOS specifications into a class library format.
    - Incorporated Tag class core properties.
    - Incorporated Equipment class core properties.
    - Added Model classes based on Equipment classes.
    - Added Document classes based on Discipline and Document Type combinations.

        - Engineering Deliverable Classes.
        - Vendor Document Classes.
    - Added Document properties.
- Removed “class” from tag and equipment classes. E.g., electrical equipment class instantiates “class” objects, not electrical equipment.
- Treated document association requirements for tag/equipment/model part classes as properties.
- Rearranged parent class relationships to organize classes into user friendly structure. E.g., As a user, I want to see all piping specialty item classes under one higher level parent class. Cable classes with its own parent. This allows tools to provide user friendly navigation methods based on class hierarchy.
- Properties used as both tag and equipment properties – Differentiated tag and equipment requirements by adding functional or physical suffix at the end of original property name.
- Added validation rules so that the data can be validated easily.
- Ability to configure default behavior at property level but to override at class property level if needed to treat some properties differently at class level.
- Added Tag Type, Asset Register Table mappings for Tag classes.
- Added Equipment Type, Asset Register Table mappings for Equipment classes.
- Added Engineering Data Templates definitions. The idea is to use these as tag register deliverables like Equipment List, Line List, Instrument Index etc.
- Added Vendor Data Templates definitions.
- Added capability to add CMS class mappings and CMS data requirements.
- Added capability to add CMMS class mappings and CMMS data requirements.
- Added framework for Object Maturity.
- Added framework for assigning the responsible party for the ownership of data meaning who is responsible for producing the data.
- Added Document Group to group document classes.
- Added Property Group to group Properties.
- Carried CFIHOS Unique Id for reference purposes.
- And many more and you will identify while going through the material.

It will also serve as a framework to define the data requirements for the downstream applications like CMS and CMMS so that data can be collected efficiently and automatically export/transfer required data.

It also includes some additional classes and attributes that are useful for EPCs during project execution. The idea is to make this framework useful to all parties.

This is not to start yet another data standards movement. The hope is that the CFIHOS organization will take these ideas into account and turn the current specifications into a fully developed class library to help all member companies with their RDL/Class Library development efforts.

2.2 Content

This Starter Class Library framework consists of the following tables.

Table Name	Revised Table Description
Discipline	Discipline definitions derived from the CFIHOS discipline table.
Document Type	Document type definitions based on the CFIHOS document type table.
Property	List of properties sourced from CFIHOS, along with framework-specific customizations.
Class	System and lookup classes (e.g., Area, Process Unit). Default property settings from this level can be inherited by class properties.
Class Property	Property requirements for general classes, including overrides and customizations applied at the class-property level.
Tag Class	Tag class definitions derived from CFIHOS.
Tag Class Property	Property requirements for tag classes, including customizations and overrides at the tag-class property level.
Equipment Class	Equipment class definitions from CFIHOS.
Equipment Class Property	Property requirements for equipment classes, with support for customization and property-level overrides.
Model Part Class	Model part classes derived from equipment classes based on CFIHOS standards.
Model Part Class Property	Document association rules and metadata for model part classes as specified in CFIHOS.
Engineering Deliverable Class	Engineering document classes derived from discipline and document type combinations per the CFIHOS specification.
Engineering Deliverable Class Property	Property requirements for engineering deliverables, based on CFIHOS and framework-specific enhancements.
Vendor Document Class	Vendor document classes derived from discipline and document types in CFIHOS. Separate classes are created for vendor-specific document requirements.
Vendor Document Class Property	Property requirements for vendor documents based on CFIHOS, with added support for project-specific customizations.
Numbering System	Definitions of numbering systems used in the project, such as document numbering or tag numbering specifications.
Numbering Object Code	Code definitions (e.g., tag codes) as specified in the numbering systems.
Numbering System Object Code	Mappings between numbering systems and their corresponding object codes.
Object Maturity Cycle	List of defined object maturity cycles used for lifecycle tracking.
Object Maturity Status	List of maturity status values assigned to objects.

Object Maturity Cycle Status	Mappings of object maturity cycles to their associated statuses.
Property Picklist	List of property picklists sourced from CFIHOS, including additional framework-specific customizations.
Property Picklist Value	Values associated with each property picklist. Mostly sourced from CFIHOS with some custom additions.
Property UOM Dimension	Unit of Measure (UOM) dimensions for properties, derived from CFIHOS.
Property UOM Symbol	Unique UOM symbols defined in CFIHOS for property measurement units.
Property UOM Dimension Symbol	Combined list of UOM dimensions and symbols to define measurable property values.
Document Group	Logical groupings of document classes. Supports defaulting and inheritance of values at group level.
Property Group	Logical groupings of properties used to organize property definitions and usage.
Tag Type	Tag type classifications used to group tag classes into common tag lists (e.g., equipment list, instrument index, line list).
Tag Type Property	Properties associated with tag types, supporting further classification and filtering.
Equipment Type	Equipment type definitions used to group equipment classes under common tag lists (e.g., line list, equipment list).
Equipment Type Property	Properties associated with equipment types for classification or data collection.
Asset Register Table	List of handover tables used to define and collect asset register data.
Asset Register Table Column	Column definitions and metadata for the asset register tables.
Engineering Data Template	Templates used to capture discipline-specific engineering data, acting as structured alternatives to tag lists.
Engineering Data Template Column	Column definitions and metadata for the engineering data templates.
Vendor Data Template	Templates used to collect structured data from vendors.
Vendor Data Template Column	Column definitions and metadata for the vendor data templates.

3 Next Steps

This class library is Open to all for use. Contributions are welcome in the form of content recommendations like any new classes and properties to be included or any other concepts that will improve the efficiency of project execution (design, construction, and commissioning etc.) and plant/facility operations phase.

The following are the next steps that need to be considered and incorporated with this starter class library.

- Review classes and properties and make necessary adjustments like adding new classes, new properties, new class properties, or removing some requirements.
  - Tag classes.
  - Equipment classes.
  - Properties.
  - Tag class properties.
  - Equipment class properties.
- Review tag core properties and remove them where not applicable. All core properties are applied to all tag classes in this starter class library.
- Review equipment core properties and remove them where not applicable.
- Finalize list of Model Part classes in case not all Equipment classes have equivalent Model Part classes.
- Define Model Class properties.
- Decide to go with separate tag and equipment classes, or just one class to cover both and differentiate functional and physical properties by adding it at property level.
- Review and update RFI and TQ properties.
- Review and update action item properties.
- Review and update document attributes.
- Review and update document revision attributes.
- Review discipline association and make any necessary changes.
- Add ENS information in the form of
  - Numbering Systems.
  - Tag Codes.
  - Tag Code to Tag Class Mappings.
- Identity how to name equipment objects. Use tag number plus some prefix/suffix?
- There could be physical properties used as functional properties. Review them and make necessary adjustments. E.g., “rated” properties assigned to tag classes.
- Review and update the following mappings for tag and equipment classes.
  - Tag Type
  - Asset Register Table
- Review *ISO 14224 - Petroleum, petrochemical and natural gas industries – Collection and exchange of reliability and maintenance data for equipment* and internal CMMS requirements and incorporate the CMMS data requirements into the class library.
- Add some hierarchy for properties to organize them into groups.
- Develop Data Matrix from all available data from the finalized class library.
- Include Digital Twin data requirement into the framework.
- Templates

  - Review and modify Asset Register Data templates.
  - Review Vendor Data Templates and modify as required.
  - Review Engineering Tag Register templates.
  - CMS Export Registers.
  - CMMS Export Registers.
  - Define Master Registers.
    - Master Tag Register.
    - Master Document Register.
    - Master Vendor Document Register.

4 Final Thoughts

Considering all the work and resources that each company requires to convert CFIHOS specifications into a usable class library and to add custom content, how can we achieve some standard work processes around data management?

The recommended approach is to develop a standard class library (fully functional and ready to go) and all its derivatives. Make it a project. All companies that have class library can share it with the project so that all the requirements from all class libraries can be incorporated into this standard class library.

5 Parking Lot for Further Consideration

1. Need 2 sets of vendor data templates?
   a. One for equipment designed by Engineering contractor.
   b. Other for tag and equipment completely designed by vendors. Like a part of skids.
2. Tag and Equipment class hierarchies are almost similar. Why not make it 100% less the tag classes that do not need equipment class. This way companies can choose to go with just one class hierarchy.
3. What is the correlation between [equipment expected to be installed indicator] and equipment classes? Do we need equipment classes if the equipment is not expected to be installed?
4. Are Tag Area and Document Area same? Or are those 2 different things?
5. Object Maturity Status and Tag Status - Are these the same?
6. Line requires ‘equipment’ class?