Trestle upgrade to OSCAL v1.1.2 - STATUS

- The pull request used to deliver updates towards fixing issues encountered during trestle upgrade from OSCAL 1.0.4 to OSCAL 1.1.2 is here: https://github.com/oscal-compass/compliance-trestle/pull/1533
- OSCAL schemas: https://github.com/usnistgov/OSCAL/releases/tag/v1.1.2

Review items

- completed: schema_preprocess.patch_poam_item + Valid Values - explain
    - This code makes a small change in the poam schema only where in the description text “poam-item" is changed to “finding” so that the RelatedObservation becomes a single class.
- completed: schema profile select-control-by-id changed to select-control to avoid conflict
    - This seems to have no adverse affects, since no field name is changed.
- completed: fix warning due to import imp (deferred) - explain
    - requires breaking change to semantic versioning from v7 to v8 or v9
    - simply remove python-semantic-release==7.33.2 from setup.cfg
- completed: pydantic upgrade from 1.x to 2.x
    - choice 1: upgrade version but use v1 interface - fairly simple
    - choice 2: handle v2 interface changes - will take time to understand and make code changes
- completed: fix sonar complaints
    - completed: 4 issues, this PR
    - **code coverage: 96.8%** Estimated after merge
- completed: go to 3.9, 10, 11 - workflow on 3.10

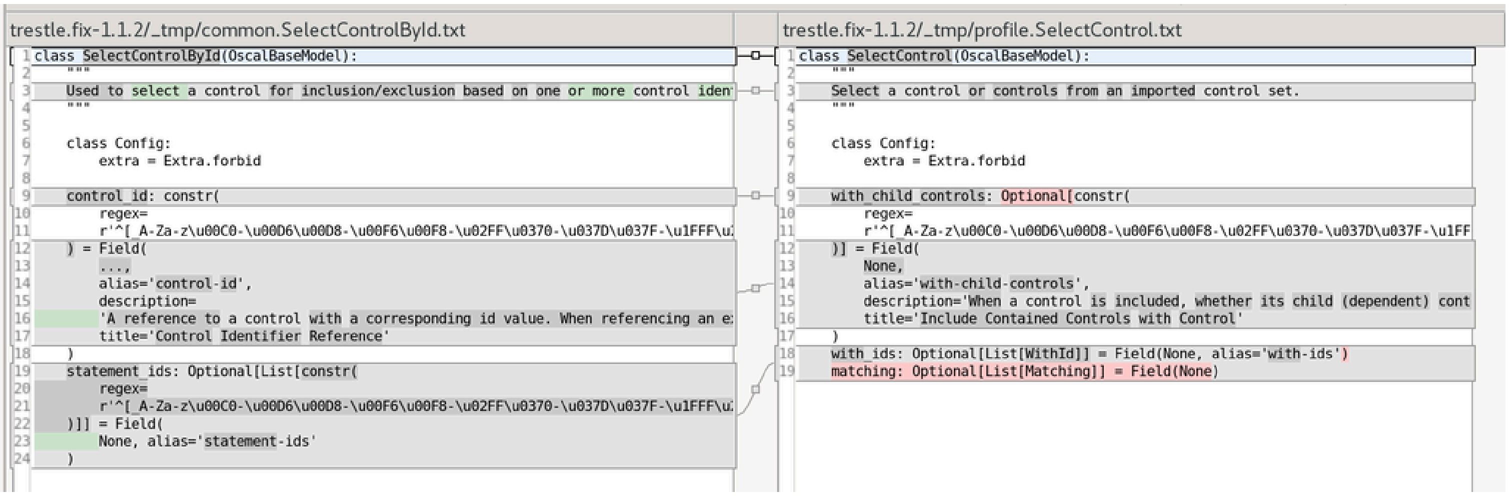
=====

Select-Control - the case for naming in Profile

1. There are 2 SelectByControlId specifications, which are different:
    a. profile
    b. ap, ar, poam

The two kinds are different and therefore to avoid confusion having different names seems appropriate.

Side by side comparison:



trestle.fix-1.1.2/_tmp/common.SelectControlById.txt

```
class SelectControlById(OscalBaseModel):
    """
    Used to select a control for inclusion/exclusion based on one or more control iden
    """

    class Config:
        extra = Extra.forbid

    control_id: constr(
        regex=
        r'^[_A-Za-z\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u02FF\u0370-\u037D\u037F-\u1FFF\u2
    ) = Field(
        ...,
        alias='control-id',
        description=
        'A reference to a control with a corresponding id value. When referencing an e
        title='Control Identifier Reference'
    )
    statement_ids: Optional[List[constr(
        regex=
        r'^[_A-Za-z\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u02FF\u0370-\u037D\u037F-\u1FFF\u2
    )]] = Field(
        None, alias='statement-ids'
    )
```

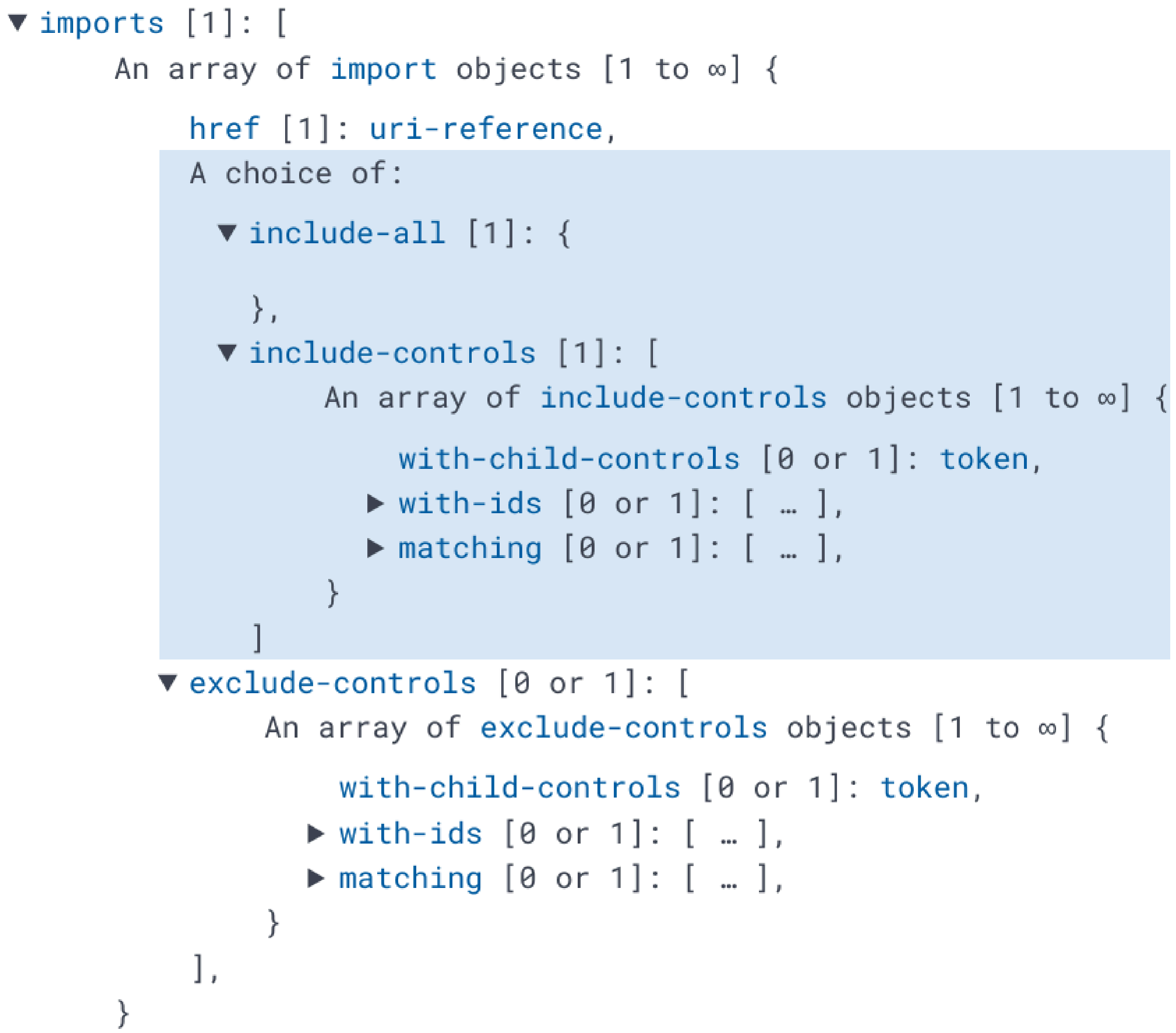
trestle.fix-1.1.2/_tmp/profile.SelectControl.txt

```
class SelectControl(OscalBaseModel):
    """
    Select a control or controls from an imported control set.
    """

    class Config:
        extra = Extra.forbid

    with_child_controls: Optional[constr(
        regex=
        r'^[_A-Za-z\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u02FF\u0370-\u037D\u037F-\u1FF
    )] = Field(
        None,
        alias='with-child-controls',
        description='When a control is included, whether its child (dependent) cont
        title='Include Contained Controls with Control'
    )
    with_ids: Optional[List[WithId]] = Field(None, alias='with-ids')
    matching: Optional[List[Matching]] = Field(None)
```

2. By naming the two kinds differently, we get the benefit of having the 3 that are the same (ap, ar, poam) in common. Currently in 1.0.4, they are not in the common module, even though identical. Since these are the same for poam+ar+ap having different classes, one for each of these 3 models, is less than ideal.
3. There are no actual structural fields in the model documents called select-by-control-id. Thus, changing the name has no impact other than on the produced python code. From OSCAL model reference:



```
▼ imports [1]: [
      An array of import objects [1 to ∞] {
        href [1]: uri-reference,
        A choice of:
          ▼ include-all [1]: {

            },
          ▼ include-controls [1]: [
                An array of include-controls objects [1 to ∞] {
                  with-child-controls [0 or 1]: token,
                ▶ with-ids [0 or 1]: [ … ],
                ▶ matching [0 or 1]: [ … ],
                }
            ]
      ▼ exclude-controls [0 or 1]: [
              An array of exclude-controls objects [1 to ∞] {
                with-child-controls [0 or 1]: token,
              ▶ with-ids [0 or 1]: [ … ],
              ▶ matching [0 or 1]: [ … ],
              }
        ],
      }
```

4. We already make lots of naming changes different from the ones chosen by the **datamodel-codegen** tool, for example **OscalApOscalMetadataProperty** → **Property** in the post-processing oscal_normalize step.
5. We already (now with 1.1.2) make pre-processing schema changes before **datamodel-codegen** is invoked to generate python code, so changing before or after is not really an issue.
6. Changing during pre-processing yields the desirable benefits of different class names for SelectByControlId and SelectControl, and puts the former into common.

=====

Daily status:

Apr 30: 1203 passed, 3 skipped

- (LD) pydantic upgrade from 1.x to 2.x
- (LD) remove python-semantic-release==7.33.2 from setup.cfg → no warnings!

Apr 29: 1203 passed, 3 skipped, 22 warnings

- (LD) rename trestle.core.commands.author.profile to prof - lint shadow issue
- (LD) use python test matrix 3.9, 3.10. 3.11

Apr 26: 1203 passed, 3 skipped, 17 warnings

- (LD) The other warnings are related to backlevel of semantic-release v7. Going to v8 or 9 would introduce breaking changes. Let's defer for now??? See https://python-semantic-release.readthedocs.io/en/latest/migrating_from_v7.html
- (LD) replace deprecated pkg_resources with importlib_resources
    - see https://importlib-resources.readthedocs.io/en/latest/migration.html

Apr 25: 1203 passed, 3 skipped, 34 warnings

- (LD) simplify pre-process code
- ToDo:
    - ~~Modernize to current pydantic level~~
    - ~~Code simplification~~

Apr 24: 1203 passed, 3 skipped, 34 warnings

- (LD) DefinedComponentType and SystemComponentType valid values
- (LD) ExternalScheme and DocumentScheme valid values
- (LD) TelephoneType and AddressType valid values
- (LD) rectification of property name changes (objective_status, originations)

Apr 23: 1203 passed, 3 skipped, 34 warnings

- (LD) add common valid values integrity check
- (LD) fix direct hack of NIST schema for EmailAddress (handle in “normalizer”)
- (LD) fix direct hack of NIST schema for Selection (HowMany)

Apr 22: 1203 passed, 3 skipped, 34 warnings

- Caveats:
    - ~~replacement of oscal-control-common:parameter-selection in NIST OSCAL schema should not be needed~~
    - ~~replacement of oscal-metadata:email-address in NIST OSCAL schema should not be needed~~
    - ~~checking that schema pre-process common valid values are identical is needed~~
    - rectification of property name changes (in schema pre-process)
    - careful consideration of special cases (in oscal normalize)
    - re-consider added imports
    - ~~conversion of remaining class TypeX to ValidValues in common~~
- Note: 34 warnings due to using deprecated “old” modules, which should be fixed before release
    - import imp (deferred); requires breaking change to semantic versioning from v7 to v8 or v9
    - ~~from pkg_resources import resource_filename~~
- (LD) fix Sonar failures: exclude trestle/oscal/*.py
- (LD) fix discontinue support of python 3.7 (update .github/workflows)
- (LD) fix AssertionError: assert StringDatatype(__root__='1.1.2') == '1.1.2'

Apr 20: 1 failed, 1202 passed, 3 skipped, 40 warnings

- (LD) fix logger.warn deprecation
- (LD) fix :test_profile_alter_props - AttributeError: 'str' object has no attribute 'value'
- (LD) fix TypeError: unhashable type: 'WithId'

Apr 19: 12 failed, 1191 passed, 3 skipped, 37 warnings

- (LD) fix tests/trestle/core/generator_test.py::test_generate_sample_model - OscalVersion

Apr 18: 13 failed, 1190 passed, 3 skipped, 37 warnings

- (LD) fix tests/trestle/tasks/ocp4_cis_profile_to_oscal_cd_test.py
- (LD) fix tests/trestle/tasks/csv_to_oscal_cd_test.py
- (LD) fix ttests/trestle/tasks/xlsx_to_oscal_profile_test.py
- (LD) fix code generation of *URIReferenceDatatype*
- (LD) fix tests/trestle/tasks/oscal_catalog_to_csv_test.py::test_execute
- (LD) fix OSCAL version in data/tasks/xlsx/output/profile.json
- (LD) fix pytest tests/trestle/core/validator_helper_test.py::test_regenerate_uuids_ssp
    - nist ssp example has moved and changed

Apr 17: 82 failed, 1121 passed, 3 skipped, 37 warnings

- (LD) improve pre-process reordering & valid values
- (LD) handle special case (e.g. Base64) code generator

Apr 16: 84 failed, 1119 passed, 3 skipped, 37 warnings

- (LD) ~~investigate~~: tests/trestle/core/generator_test.py::test_get_all_sample_models_optional - pydantic.errors.ConfigError: field "include_controls" not yet prepared so type is still a ForwardRef, you might need to call ControlSelection.update_forward_refs().
- (LD) fix determine common valid values for Task, ThreatId

Apr 15: no improvement

- (LD) fixed (see above): tests/trestle/core/generator_test.py::test_get_all_sample_models - pydantic.errors.ConfigError: field "type" not yet prepared

Apr 14: 85 failed, 1118 passed, 3 skipped, 37 warnings

- (LD) fix base64 issue

Apr 13: 88 failed, 1115 passed, 3 skipped, 36 warnings

- (LD) patch POAM schema to make RelatedObservation same as the other models
- (LD) patch schemas to rename “status” to “objectiveStatus” to avoid conflict
- (LD) fix FAILED tests/trestle/core/commands/import__test.py::test_import_wrong_oscal_version
    - use empty string as bad oscal version
- (LD) fix value is not a valid enumeration member; permitted: 'EXAMINE', 'INTERVIEW', 'TEST', 'UNKNOWN'
    - Add check for type “enum Methods” and return sample accordingly
- (LD) fix ERROR collecting tests/trestle/transforms/transformer_helper_test.py
    - Add Observation to assessment_results from common

Apr 12: 101 failed, 1012 passed, 3 skipped, 36 warnings, 6 errors

- (LD) fix Origin vs. Origin1

    - patch schema to rename “origins” to “originations” in POAM to avoid conflict
- 136 failed, 1067 passed, 3 skipped, 36 warnings
- (LD) fix AttributeError: 'StringDatatype' object has no attribute 'strip'
    - Add special case for 'value: StringDatatype' similar to 'id: TokenDatatype' (below)

Apr 11: 330 failed, 873 passed, 3 skipped, 36 warnings

- (LD) activity resumed
- (LD) ~~investigate: TypeError: unhashable type: 'WithId'~~
    - This looks to be collateral damage from 'StringDatatype' issue, see above
- (LD) fix AttributeError: module 'trestle.oscal.ssp' has no attribute 'Status'
    - include Status (newly moved to common) in ssp
- (LD) fix pydantic.error_wrappers.ValidationError: 1 validation error for Base64
    - Test for type Base64 and use sample by type when detected

Apr 08: 345 failed, 858 passed, 3 skipped, 36 warnings

- (LD) activity suspended to work on AI crosswalks (for BG1)

Apr 05: 345 failed, 858 passed, 3 skipped, 36 warnings

- (LD) revise apply_eligible logic to include lines in which CommonRiskStatus appears
- (LD) use patching of OSCAL meta-schema to handle email-address issue by using schema-substitutes.json (change from hacking in previous solution below, Apr 03)
- (LD) use patching of OSCAL meta-schema to handle how-many issue by using schema-substitutes.json

Apr 04: 349 failed, 854 passed, 3 skipped, 36 warnings

- (LD) handle special case of “id: TokenDatatype” in catalog

Apr 03: 385 failed, 818 passed, 3 skipped, 34 warnings

- (LD) submitted datamodel-codegen tool issue https://github.com/koxudaxi/datamodel-code-generator/issues/1901.
- (LD) add code to schema_preprocess.py for hacking EmailAddressDatatype to get around above issue.

Apr 02: 386 failed, 785 passed, 3 skipped, 34 warnings, 32 errors

---

Changes

- (LD) Update pyproject.toml

Change line length to 200. This allows code bodies generated by datamodel-codegen to have an improved chance of matching when post-processing to gather common elements.

- (LD) New module schema_preprocess.py

The idea behind this module is to pre-process the “raw” OSCAL meta-schemas for standardization before invoking the datamodel-codegen tool to effect more commonality between and amongst the 7 data models when the python code is generated. Each data model is ingested as json, then the keys are ordered in the same fashion in all, for example subject model first (catalog, profile…), then metadata, then common, then implementation. New files are written to a -fixup folder for use by the next step.

- (LD) Automate oscal version file update

Update gen_oscal.py automation to extract OSCAL version from meta-schemas and update trestle/oscal/__init__.py with current version.