Dimensions in the Repository Evaluation Instrument

This document describes how the *low-level dimensions (properties and flags)* in our model affect its *high-level dimensions*. Low-level dimensions describe how an evaluated repository is performing on a narrow aspect, such as its compliance with Creative Commons licenses or the way it allows dataset authors to be associated with ORCID. High-level dimensions describe wide properties such as "Openness" or "Interoperability". Their value is a summary of values in relevant low-level dimensions.

How to Read This Document

Below, we list dimensions names on the left, and their coordinates on the right. For each coordinate, we describe what constraints it imposes on the high-level dimensions that summarizes its dimension. As an example, in the following listing we look at the constraints each coordinate in the `Restrictions` dimension imposes on the evaluated repository's coordinate along the `Open` dimension.

Restrictions

- none ⇒ Open in `partially..fully`
- minimal ⇒ Open is `partially`
- significant ⇒ Open in `not..partially`

Based on the above listing, we see that when an a repository does not restrict the access to its data (i.e. `Restrictions=none`), said repository can be considered *partially open* or *fully open* (`Open=partiallyOpen` or `Open=fullyOpen`, respectively). By way of negation, such repository cannot be considered *closed*.

Conversely, when a repository imposes significant restrictions on accessing data it stores, it cannot be considered *fully open*. However, based on other aspects, it may be considered *partially open*.

Dimensions

Result Dimensions

Used for communicating results outside of the model.

- Open
- FAIR
- Citable

- Trustworthy
- Findable
- Accessible
- Interoperable
- Reusable

Open

Dimensions affecting the `Open` dimension coordinate.

IMPORTANT	All conditions pertaining to the `Open` dimension are conditioned in that the `CCLicenceCompliance` dimension coordinate is above `nonCompliant`.

OpenFlags	• openFormat ⇒ Open in `partially..fully` • platformSupportsDataWork ⇒ Open in `partially..fully` • ccLicenseOK ⇒ Open in `partially..fully` • restrictionsNotJustified ⇒ Result coordinate
Restrictions	• none ⇒ Open in `partially..fully` • minimal ⇒ Open is `partially` • significant ⇒ Open in `not..partially`
CCLicenseCompliance	• nonCompliant ⇒ Open is `not` • none ⇒ Open in `not..partially` • adequate ⇒ Open in `partially..fully` • good ⇒ Open in `partially..fully` • full ⇒ Open in `partially..fully`

Accessible

MetadataPersistence	• no ⇒ Accessible in `not..partially` • by evidence ⇒ Accessible in `partially..fully` • by stated policy ⇒ Accessible in `partially..fully`

AccessibleFlags

- humanAccessible ⇒ Accessible in `partially..fully`
- machineAccessible ⇒ Accessible in `partially..fully`
- persistentMetadata ⇒ Accessible in `partially..fully`
- licenseOK ⇒ Accessible in `partially..fully`
- stdApi ⇒ Accessible in `partially..fully`

Findable

FindableFlags

- internalSearchOK ⇒ Findable in `partially..fully`

PersistentIdentifier

- none ⇒ Findable in `not..partially`
- internalPID ⇒ Findable is `partially`
- externalPID ⇒ Findable in `partially..fully`

MetadataGrade

- minimal ⇒ Findable in `not..partially`
- limited ⇒ Findable is `partially`
- rich ⇒ Findable in `partially..fully`

IdInMetadata

- none ⇒ Findable in `not..partially`
- partial ⇒ Findable is `partially`
- all ⇒ Findable in `partially..fully`

Interoperable

InteroperableFlags

- formalMetadataVocabularyOK ⇒ Interoperable in `partially..fully`
- fairMetadataOK ⇒ Interoperable in `partially..fully`
- qualifiedMetadataReferencesOK ⇒ Interoperable in `partially..fully`
- studyLinkageOK ⇒ Interoperable in `partially..fully`

MetadataFAIRness

Finer-grained `fairMetadataOK`.

- minimal ⇒ Interoperable in `not..partially`
- allowed ⇒ Interoperable in `partially..fully`
- enforced ⇒ Interoperable in `partially..fully`

MetadataReferenceQuality Finer-grained `qualifiedMetadataReferencesOK`.

- freeText ⇒ Interoperable in `not..partially`
- informal ⇒ Interoperable in `partially..fully`
- formal ⇒ Interoperable in `partially..fully`

StudyLinkage Finer-grained `studyLinkageOK`

- none ⇒ Interoperable in `not..partially`
- freeText ⇒ Interoperable in `not..partially`
- textualMetadata ⇒ Interoperable in `partially..fully`
- machineReadableMetadata ⇒ Interoperable in `partially..fully`

Reusable

ReusableFlags

- documentationOK ⇒ Reusable in `partially..fully`
- dkNetMetadataOK ⇒ Reusable in `partially..fully`
- communityStandard ⇒ Reusable in `partially..fully`
- generalMetadata ⇒ Reusable in `partially..fully`
- metadataProvenanceOK ⇒ Reusable in `partially..fully`

DocumentationLevel Finer-grained `documentationOK`

- lacking ⇒ Reusable in `not..partially`
- adequate ⇒ Reusable in `partially..fully`
- good ⇒ Reusable in `partially..fully`
- full ⇒ Reusable in `partially..fully`

ReuseLicense Finer-grained `licenseOK`

- none ⇒ Reusable in `not..partially`
- repositoryLevel ⇒ Reusable in `partially..fully`
- datasetLevel ⇒ Reusable in `partially..fully`

MetadataProvenance Finer-grained `metadataProvenanceOK`.

- unclear ⇒ Reusable in `not..partially`
- adequate ⇒ Reusable in `partially..fully`
- full ⇒ Reusable in `partially..fully`

DkNetMetadataLevel Finer-grained `dkNetMetadataOK`

- none ⇒ Reusable in `not..partially`
- dataset ⇒ Reusable in `partially..fully`
- datasetAndSubject ⇒ Reusable in `partially..fully`

Citable

OrcidAssociation

- none ⇒ Citable in `not..partially`
- supported ⇒ Citable in `partially..fully`
- required ⇒ Citable in `partially..fully`

CitationMetadataLevel

- none ⇒ Citable in `not..partially`
- partial ⇒ Citable is `partially`
- full ⇒ Citable in `partially..fully`

MachineReadableLandingPage

- none ⇒ Citable in `not..partially`
- exists ⇒ Citable is `partially`
- supportsDataCitation ⇒ Citable in `partially..fully`

Trustworthiness

GovernanceTransparency

- opaque ⇒ Trustworthy in `significantConcerns..minorConcerns`
- partial ⇒ Trustworthy is `minorConcerns`
- full ⇒ Trustworthy in `minorConcerns..noConcerns`

StakeholderGovernance

- none ⇒ Trustworthy in `significantConcerns..minorConcerns`
- weak ⇒ Trustworthy is `minorConcerns`
- good ⇒ Trustworthy is `minorConcerns`
- full ⇒ Trustworthy in `minorConcerns..noConcerns`

SourceOpen

- no ⇒ Trustworthy in `significantConcerns..minorConcerns`
- partially ⇒ Trustworthy is `minorConcerns`
- yes ⇒ Trustworthy in `minorConcerns..noConcerns`