

Pathfinder Network

Technical specifications for Use Case 001: PCF Data Exchange

This version:

1.0.0

Authors:

Martin Pompéry (SINE Foundation)
Cecilia Valeri (WBCSD)

Related Documents:

[Pathfinder Framework Version 1](#)

[Greenhouse Gas Protocol](#)

[ISO 14067](#)

[ISO 14044](#)

Powered by

Introduction	6
Scope	6
Intended Audience	6
About PACT and the Pathfinder Network	7
Disclaimer	7
Acknowledgments	7
License	7
Terminology	8
Data recipient	8
Data owner	8
Host System	8
Interoperability	8
GHG - Greenhouse Gas (emissions)	8
Methodology (Pathfinder Framework)	8
Pathfinder Network	8
PCF – Product Carbon Footprint	8
SCA – Supply Chain Actor	9
SP – Solution Provider	9
Conformance	10
Data Model	11
Data Type: ProductFootprint	11
Properties	12
Property “id”	12
Property “specVersion”	12
Property “version”	12
Property “created”	13
Property “updated”	13
Property “companyName”	13
Property “companyIds”	13
Property “productDescription”	13
Property “productIds”	13
Property “productCategoryCpc”	13
Property “productNameCompany”	13
Property “comment”	14
Property “pcf”	14
Data Type: CarbonFootprint	15
Scope of a CarbonFootprint	15

Properties	16
Property “declaredUnit”	16
Property “unitaryProductAmount”	16
Property “reportingPeriodStart”	17
Property “reportingPeriodEnd”	17
Property “geographyCountrySubdivision”	17
Property “geographyCountry”	17
Property “geographyRegionOrSubregion”	17
Property “primaryDataShare”	18
Property “emissionFactorSources”	18
Property “boundaryProcessesDescription”	19
Property “crossSectoralStandardsUsed”	19
Property “productOrSectorSpecificRules”	19
Property “allocationRulesDescription”	19
Property “fossilGhgEmissions”	19
Property “biogenicCarbonContent”	19
Property “biogenicEmissions”	20
Data Type: EmissionFactorDSSet	20
JSON Data Representation	20
Data Type: EmissionFactorDS	20
Properties	20
Property “name”	21
Property “version”	21
JSON Representation	21
Data Type: BiogenicEmissions	21
Properties	21
Property “landUseEmissions”	21
Property “landUseChangeEmissions”	22
Property “otherEmissions”	22
JSON Representation	22
Data Type: CrossSectoralStandard	22
JSON Representation	22
Data Type: CrossSectoralStandardSet	22
JSON Data Representation	22
Data Type: ProductOrSectorSpecificRule	23
Properties	23
Property “operator”	23
Property “ruleNames”	23
Property “otherOperatorName”	23
JSON Representation	23

Data Type: ProductOrSectorSpecificRuleSet	23
JSON Representation	24
Data Type: ProductOrSectorSpecificRuleOperator	24
JSON Representation	24
Data Type: NonEmptyStringVector	24
JSON Representation	24
Data Type: CpcCode	24
JSON Representation	24
Data Type: SpecVersionString	24
JSON Representation	24
Data Type: VersionInteger	25
JSON Data Representation	25
Data Type: DeclaredUnit	25
JSON Data Representation	25
Data Type: NonEmptyString	25
JSON Data Representation	25
Data Type: CompanyIdSet	25
JSON Data Representation	25
Data Type: ProductIdSet	25
JSON Data Representation	26
Data Type: ProductId	26
Data Type URN	26
JSON Data Representation	26
Data Type: String	26
JSON Data Representation	26
Data Type: Percent	26
JSON Data Representation	26
Data Type: StrictlyPositiveDecimal	26
JSON Data Representation	27
Data Type: DateTime	27
JSON Data Representation	27
Data Type: ISO3166CC	27
JSON Data Representation	27
Data Type: PositiveDecimal	27
JSON Data Representation	27
Data Type: Decimal	27
JSON Data Representation	28
Data Type: Boolean	28
JSON Data Representation	28
Data Type: Pfld	28

JSON Representation	28
Product Footprint Lifecycle	29
Introduction	29
Definition of update	29
Data life cycle rules	29
Use Case 001 HTTP REST API Version 1.0.0	30
Introduction	30
Host System	30
Out of scope of Use Case 001	30
A note on data synchronization between Host Systems	31
Authentication and Request Flows	31
Actions reference	31
Action Authenticate	32
Request syntax (HTTP/1.1)	32
Request parameters:	32
Response Syntax	33
Response Parameters:	33
Action ListFootprints	34
Request Syntax (HTTP/1.1)	34
Request parameters:	34
Response syntax	34
Response parameters:	35
Example (HTTP/2)	36
Request	36
Response	36
Action GetFootprint	37
Request Syntax (HTTP/1.1)	37
Request Parameters:	37
Response Syntax	37
Response Parameters:	37
Example (HTTP/2)	39
Request	39
Response	39
Error responses	40
List of error codes	40
Appendix: License	41

Introduction

This section is non-normative.

This document is the first specification in the context of the Pathfinder Network. It contains technical specifications to enable the exchange of standardized greenhouse gas (GHG) data at product level across interoperable technology solutions (i.e., the Use Case 001).

This Use Case 001 document specifies a [data model](#) and an [HTTP REST API](#) for the interoperable exchange of [product footprint](#) information with a special focus on [product carbon footprints](#) (PCF) between independent [solutions](#). It further specifies rules for [conformance of host systems](#), the [digital representation of product footprints](#) and related data [data life cycle rules](#).

The semantic foundation of the data model is the [Pathfinder Framework Version 1](#). Developer resources can be found at the [Use Case 001 repository](#).

Scope

The scope of this document and the feature coverage of the HTTP REST API is *intentionally minimal by design*. This document will be revised and additional use cases specified at a later stage, building on these specifications. The scope may be extended in future versions, for example to include services.

Intended Audience

This document is for

- software developers who want to build software for the exchange of product footprints according to the Pathfinder Framework;
- auditors and sustainability experts who want to understand the data semantics of product footprints or how they are exchanged between partners; and
- anyone that wants to understand more about the technological foundations of the Pathfinder Network.

All parties are encouraged to review the technical specifications, inform the Partnership for Carbon Transparency (PACT), and SINE Foundation of any feedback (pact@wbcsd.org), and implement their own interoperable solutions.

About PACT and the Pathfinder Network

The Pathfinder Network is a concept developed by PACT and powered by the World Business Council for Sustainable Development (WBCSD). PACT is working toward the vision of an open and global network of interoperable solutions for the secure peer-to-peer exchange of accurate, primary and verified product emissions data – across all industries and value chains.

For further information, please refer to the PACT website www.carbon-transparency.com and the [Pathfinder vision paper](#) “Pathfinder Network: Enabling standardized emissions data exchange”.

Disclaimer

This document is published as a version 1, acknowledging that additions or adjustments may be necessary (e.g., following an update of the Pathfinder Framework later this year). We will do our best to ensure updates do not create compatibility concerns, though we cannot exclude this possibility completely. While PACT encourages the implementation of the technical specifications by all entities to start creating a harmonized system, neither PACT, WBCSD, nor any other individuals who contributed to the development of this document assume responsibility for any consequences or damages resulting directly or indirectly from the use of this document.

Acknowledgments

WBCSD would like to thank all PACT members, WBCSD staff, and others who shared their detailed and thoughtful input and contributed actively to the development of this document.

WBCSD would also like to express special thanks to the companies participating in the pilot for testing the interoperable exchange of GHG emissions data across different solutions, as well as to those Solution Providers who have contributed to this document.

License

The license can be found in the [Appendix](#).

Terminology

Data recipient

The Supply Chain Actor ([SCA](#)) requesting and/or receiving [PCF](#) data from another SCA.

Data owner

The SCA sharing/being asked to share PCF data with another SCA.

Host System

See [Host System](#)

Interoperability

Interoperability has technical and non-technical aspects. Throughout this document the following understanding is assumed

1. Technical Interoperability: interoperability at the data and functional level
2. Semantic Interoperability: interoperability at the level of meaning and value exchanged
3. Organizational Interoperability: interoperability at the level of business processes required for data and value exchanged

GHG - Greenhouse Gas (emissions)

Gaseous constituents of the atmosphere, both natural and anthropogenic, that absorb and emit radiation at specific wavelengths within the spectrum of infrared radiation emitted by the Earth's surface, its atmosphere and clouds. GHGs include CDCO₂, Methane (CH₄), Nitrous Oxide(N₂O), Hydrofluoro-Carbons (HFCs), Perfluorocarbons (PFCs) and Sulfur Hexafluoride (SF₆).

Methodology (Pathfinder Framework)

Guidance for the Accounting and Exchange of Product Life Cycle Emissions, building on existing standards and protocols, such as the GHG Protocol Product standard). See the [Pathfinder Framework Version 1](#) for details.

Pathfinder Network

An information network of and for supply chain actors to securely exchange environmental data with each other, with an initial focus on PCF data.

PCF – Product Carbon Footprint

The carbon (equivalent) emissions relating to a product. Products can be any kind of item exchanged between entities, including “pieces”, metric or volumetric quantities of a product, etc.

SCA – Supply Chain Actor

An entity intending on exchanging PCF data with another entity using the technical means specified in this document.

SP – Solution Provider

An entity providing technical solutions to SCAs. In context of this document: software companies implementing the proposed interfaces to enable SCAs' interoperable exchange of information.

Conformance

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words *MAY*, *MUST*, *MUST NOT*, *OPTIONAL*, *RECOMMENDED*, *REQUIRED*, *SHOULD*, and *SHOULD NOT* in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

A conforming host system is any algorithm realized as software and/or hardware that complies with the relevant normative statements in [Use Case 001 HTTP REST API Version 1.0.0](#).

A conforming requesting data recipient is any algorithm realized as software and/or hardware that complies with the relevant normative statements in [Use Case 001 HTTP REST API Version 1.0.0](#).

Data Model

Non-normative: This section specifies a data model for product footprints following the Pathfinder Framework guidelines¹. Additionally, this section specifies the data model representation in JavaScript Object Notation (JSON).

Data Type: ProductFootprint

Normative: Each ProductFootprint represents the carbon footprint of a product with values in accordance with the Pathfinder Framework.

Non-normative: Each ProductFootprint relates to a specific product, identified by 1 or more product identifiers. The data type ProductFootprint is modeled as a multi purpose container for product-specific factors.

1

<https://www.wbcsd.org/Programs/Climate-and-Energy/Climate/SOS-1.5/Resources/Pathfinder-Framework-Guidance-for-the-Accounting-and-Exchange-of-Product-Life-Cycle-Emissions>

Properties

Normative: a ProductFootprint has the following properties:

Property ID	Data Type	Mandatory
id	Pfld	yes
specVersion	SpecVersionString	yes
version	VersionInteger	yes
created	DateTime	yes
updated	DateTime	no
companyName	NonEmptyString	yes
companyIds	CompanyIdSet	yes
productDescription	String	yes
productIds	ProductIdSet	yes
productCategoryCpc	CpcCode	yes
productNameCompany	NonEmptyString	yes
comment	String	yes
pcf	CarbonFootprint	yes

Property “id”

A ProductFootprint MUST include the property `id` with value the product footprint [identifier](#).

Property “specVersion”

A ProductFootprint MUST include the property `specVersion` with value the version of the ProductFootprint data specification. The value of `specVersion` MUST be 1.0.0.

Note: subsequent revisions of the ProductFootprint data specification will update this value according to the rules of [Semantic Versioning 2.0.0](#).

Property “version”

A ProductFootprint MUST include the property `version` with value a [VersionInteger](#).

Property “created”

A ProductFootprint MUST include the property `created` with value the timestamp of the creation of the ProductFootprint.

Property “updated”

A ProductFootprint SHOULD include the property `updated` with value the timestamp of the ProductFootprint update. A ProductFootprint MUST NOT include this property if an update has never been performed. The timestamp MUST be in UTC.

Property “companyName”

A ProductFootprint MUST include the property `companyName` with value the name of the company that is the ProductFootprint Data Owner.

Property “companyIds”

A ProductFootprint MUST include the property `companyIds` with value the non-empty set of Uniform Resource Names (URN)². Each value of this set is supposed to uniquely identify the ProductFootprint Data Owner.

Property “productDescription”

A ProductFootprint MUST include the property `productDescription` with value the free-form description of the product plus other information related to it such as production technology or packaging.

Property “productIds”

A ProductFootprint MUST include the property `productIds` with value the non-empty set of [ProductIds](#). Each of the values in the set is supposed to uniquely identify the product. What constitutes a suitable product identifier depends on the product, the conventions, contracts, and agreements between the Data Owner and a Data Recipient and is out of the scope of this specification.

Property “productCategoryCpc”

A ProductFootprint MUST include the property `productCategoryCpc` with value a UN Product Classification Code (CPC)³ that the given product belongs to.

Property “productNameCompany”

A ProductFootprint MUST include the property `productNameCompany` with value the trade name of the product.

² https://en.wikipedia.org/wiki/Uniform_Resource_Name

³ <https://unstats.un.org/unsd/classifications/Econ/CPC.cshml>

Property “comment”

A ProductFootprint MUST include the property `comment` with value the additional information related to the product footprint.

Whereas the property `productDescription` contains product-level information, the property `comment` SHOULD be used for information and instructions related to the calculation of the footprint, or other information which informs the ability to interpret, to audit or to verify the Product Footprint.

Property “pcf”

A ProductFootprint MUST include the property `pcf` with value conforming to the data type [CarbonFootprint](#).

Data Type: CarbonFootprint

A CarbonFootprint represents the carbon footprint of a product and related data in accordance with the Pathfinder Framework.

Scope of a CarbonFootprint

Each CarbonFootprint is scoped by

1. Time: the time is defined by property `reportingPeriodStart` (including) and property `reportingPeriodEnd` (excluding); see the Pathfinder Framework Section 7.2.1
2. Geography: further set by the properties `geographyRegionOrSubregion`, `geographyCountry`, and `geographyCountrySubdivision`; see the Pathfinder Framework Section 7.2.2.

If a CarbonFootprint

- a) Has geographical granularity `Global`, then the properties `geographyCountry` and `geographyRegionOrSubregion` and `geographyCountrySubdivision` **MUST** be undefined;
- b) Has a regional or sub-regional geographical granularity, then the property `geographyRegionOrSubregion` **MUST** be defined and the properties `geographyCountry` and `geographyCountrySubdivision` **MUST** be undefined;
- c) Has a country-specific geographical granularity, then property `geographyCountry` **MUST** be defined AND the properties `geographyRegionOrSubregion` and `geographyCountrySubdivision` **MUST** be undefined;
- d) Has a country subdivision-specific geographical granularity, then property `geographyCountrySubdivision` **MUST** be defined AND the properties `geographyRegionOrSubregion` and `geographyCountry` **MUST** be undefined.

A tabular overview of the relationship between the geographic scope and the definedness or undefinedness of properties is given in the following table:

Geographical Granularity / Level of aggregation	Property <code>geographyRegionOrSubregion</code>	Property <code>geographyCountry</code>	Property <code>geographyCountrySubdivision</code>
Global	undefined	undefined	undefined
Regional or Subregional	<u>defined</u>	undefined	undefined
Country	undefined	<u>defined</u>	undefined

Subdivision	undefined	undefined	<u>defined</u>
-------------	-----------	-----------	----------------

Properties

A CarbonFootprint has the following properties:

Property	Data Type	Mandatory
declaredUnit	DeclaredUnit	yes
unitaryProductAmount	StrictlyPositiveDecimal	yes
fossilGhgEmissions	PositiveDecimal	yes
biogenicEmissions	BiogenicEmissions	no
biogenicCarbonContent	PositiveDecimal	yes
reportingPeriodStart	DateTime	yes
reportingPeriodEnd	DateTime	yes
geographyCountrySubdivision	NonEmptyString	(see Scope)
geographyCountry	ISO3166CC	(see Scope)
geographyRegionOrSubregion	NonEmptyString	(see Scope)
primaryDataShare	Percent	yes
emissionFactorSources	EmissionFactorDSSet	no
boundaryProcessesDescription	String	no
crossSectoralStandardsUsed	CrossSectoralStandardSet	yes
productOrSectorSpecificRules	ProductOrSectorSpecificRuleSet	yes
allocationRulesDescription	String	no

Property “declaredUnit”

A CarbonFootprint MUST include the property `declaredUnit` with value the unit of analysis of the product. See Data Type [DeclaredUnit](#) for further information.

Property “unitaryProductAmount”

A CarbonFootprint MUST include the property `unitaryProductAmount` with value the amount of units contained within the product the PCF is referring to.

Property “reportingPeriodStart”

A CarbonFootprint MUST include the property `reportingPeriodStart` with value the start of the reporting period. See the Pathfinder Framework Section 7.2.1.

For further information on time representation, see the definition of data type [DateTime](#).

Property “reportingPeriodEnd”

A CarbonFootprint MUST include the property `reportingPeriodEnd` with value the end (excluding) of the reporting period. See the Pathfinder Framework Section 7.2.1.

For further information on time representation, see the definition of data type [DateTime](#).

Property “geographyCountrySubdivision”

The property `geographyCountrySubdivision` is optional. If defined, the value MUST be an ISO 3166-2 Subdivision Code. Property [geographyCountrySubdivision](#) MUST NOT be defined if property [geographyRegionOrSubregion](#) is defined or property [geographyCountry](#) is defined.

See the [scope of a CarbonFootprint](#) for further details.

Example values:

- US-NY (the State of New York in the United States of America)
- FR-89 (the department Yonne in France)

Property “geographyCountry”

The property `geographyCountry` is optional. If defined, the value MUST conform to data type [ISO3166CC](#). Property `geographyCountry` MUST NOT be defined if property `geographyRegionOrSubregion` is defined or if property `geographyCountrySubdivision` is defined.

See the [scope of a CarbonFootprint](#) for further details.

Example value: FR (in case the geography scope is France)

Property “geographyRegionOrSubregion”

The property `geographyRegionOrSubregion` is optional. If defined, the value must be one of the following:

- a) Africa for the [UN geographic region Africa](#)
- b) Americas for the UN geographic region Americas

- c) Asia for the UN geographic region Asia
- d) Europe for the UN geographic region Europe
- e) Oceania for the UN geographic region Oceania
- f) Australia and New Zealand for the [UN geographic subregion](#) Australia and New Zealand
- g) Central Asia for the UN geographic subregion Central Asia
- h) Eastern Asia for the UN geographic subregion Eastern Asia
- i) Eastern Europe for the UN geographic subregion Eastern Europe
- j) Latin America and the Caribbean for the UN geographic subregion Latin America and the Caribbean
- k) Melanesia for the UN geographic subregion Melanesia
- l) Micronesia for the UN geographic subregion Micronesia
- m) Northern Africa for the UN geographic subregion Northern Africa
- n) Northern America for the UN geographic subregion Northern America
- o) Northern Europe for the UN geographic subregion Northern Europe
- p) Polynesia for the UN geographic subregion Polynesia
- q) South-eastern Asia for the UN geographic subregion South-eastern Asia
- r) Southern Asia for the UN geographic subregion Southern Asia
- s) Southern Europe for the UN geographic subregion Southern Europe
- t) Sub-Saharan Africa for the UN geographic subregion Sub-Saharan Africa
- u) Western Asia for the UN geographic subregion Western Asia
- v) Western Europe for the UN geographic subregion Western Europe

The property `geographyRegionOrSubregion` MUST NOT be defined if property the `geographyCountry` is defined or if the property `geographyCountrySubdivision` is defined.

See the [scope of a CarbonFootprint](#) for further details. Additionally, see the Pathfinder Framework Section 7.2.2.

Note: The set of region identifiers will likely change in future revisions. It is recommended to account for this when implementing the validation of this property.

Property “primaryDataShare”

A `CarbonFootprint` MUST include the property `primaryDataShare` with value the share of primary data. See the Pathfinder Framework Section 7.2.3.

Property “emissionFactorSources”

The property `emissionFactorSources` is optional. If secondary data was used to calculate the `CarbonFootprint`, then it MUST include the property `emissionFactorSources` with value

the emission factors used for the CarbonFootprint calculation (see the Pathfinder Framework Section 6.2).

If no secondary data is used, this property MUST BE undefined.

Property “boundaryProcessesDescription”

The property `boundaryProcessesDescription` is optional. If present, the value MUST be the processes attributable to each lifecycle stage. If no such description is available or otherwise provided, then this property MUST NOT be included.

Example value: Electricity consumption included as an input in the production phase’

Property “crossSectoralStandardsUsed”

A CarbonFootprint MUST include the property `crossSectoralStandardsUsed` with value conforming to data type [CrossSectoralStandardSet](#). The property `crossSectoralStandardsUsed` discloses the cross-sectoral standards applied for calculating or allocating GHG emissions.

Property “productOrSectorSpecificRules”

A CarbonFootprint MUST include the property `productSpecificRules` with value the [set](#) of values conforming to data type [ProductOrSectorSpecificRule](#). The property `productSpecificRules` discloses the product-specific or sector-specific rules applied for calculating or allocating GHG emissions (e.g., PCRs). If no product or sector specific rules were followed, this set must be empty.

See data type [ProductOrSectorSpecificRuleSet](#) for further information.

See the Pathfinder Framework Section 7.1 and Section 8.3.2.

Property “allocationRulesDescription”

The property `allocationRulesDescription` is optional. If present, the value MUST be the description of the allocation rules applied.

Property “fossilGhgEmissions”

A CarbonFootprint MUST include the property `fossilGhgEmissions` with value the emissions from the combustion of fossil sources. The value of property `fossilGhgEmissions` MUST be calculated per [declared unit](#) with unit kg of CO₂ equivalent per kilogram (kgCO₂e / kg). The emissions value MUST be a decimal equal to or greater than zero.

See the Pathfinder Framework for calculation methodology.

Property “biogenicCarbonContent”

A CarbonFootprint MUST include the property `biogenicCarbonContent` with value the

mass of biogenic carbon per given [unit of exchange](#) (see the Pathfinder Framework Version 1), expressed as a decimal equal to or greater than zero with unit kg CO₂eq per declared unit.

Property “biogenicEmissions”

The property biogenicEmissions is optional. If present, the value MUST be the biogenic emission factors conforming to data type [BiogenicEmissions](#).

Note: This property will be revised once the GHG Protocol FLAG Standard is published.

Data Type: EmissionFactorDSSet

A set of Emission Factor Data Sources ([EmissionFactorDS](#)) of size 1 or larger.

JSON Data Representation

As an array of objects, with each object conforming to the JSON representation of [EmissionFactorDS](#).

Data Type: EmissionFactorDS

An EmissionFactorDS references emission factor databases accepted under Version 1 of the Pathfinder Framework (see the Pathfinder Framework Section 6.2).

Note: Version 2 of the Pathfinder Framework will extend the coverage of emission factors data sources. This specification will reflect upcoming changes in future versions.

Properties

Property	Data Type	Mandatory
name	NonEmptyString	yes
version	NonEmptyString	yes

Property “name”

Each EmissionFactorDS MUST include the property `name` with value the name of the emission factor database.

Property “version”

Each EmissionFactorDS MUST include the property `version` with value the version of the emission factor database used.

JSON Representation

Each EmissionFactorDS MUST be encoded as a JSON object.

Data Type: BiogenicEmissions

BiogenicEmissions contains biogenic emission values according to the GHG Protocol FLAG Standard.

Properties

Property	Data Type	Mandatory
landUseEmissions	Decimal	no
landUseChangeEmissions	Decimal	no
otherEmissions	Decimal	no

If `BiogenicEmissions` is defined, at least one property MUST be defined.

Property “landUseEmissions”

The property `landUseEmissions` is optional. If present, the value MUST be the land use emissions (e.g. cultural practice) as a decimal number equal to, greater or lower than zero.

Property “landUseChangeEmissions”

The property `landUseChangeEmissions` is optional. If present, the value MUST be the land use change emissions (e.g. due to deforestation) as a decimal number equal to, greater or lower than zero. This value must *include direct land use change (dLUC) where available, otherwise statistical land use change (sLUC) can be used. If available, including indirect land use change (iLUC) to remain optional.*

Property “otherEmissions”

The property `otherEmissions` is optional. If present, the value MUST be the other emissions (e.g. biogenic waste treatment) as a decimal number equal to, greater or lower than zero.

JSON Representation

Each `BiogenicEmissions` MUST be encoded as a JSON object.

Data Type: `CrossSectoralStandard`

`CrossSectoralStandard` is the enumeration of accounting standards used for product carbon footprint calculation. Valid values are

1. `GHG Protocol Product standard` for the GHG Protocol Product standard
2. `ISO Standard 14067` for ISO Standard 14067
3. `ISO Standard 14044` for ISO Standard 14044

JSON Representation

Each `CrossSectoralStandard` MUST be encoded as a JSON string.

Data Type: `CrossSectoralStandardSet`

A set of [CrossSectoralStandards](#).

JSON Data Representation

As an array of objects, with each object conforming to the JSON representation of `CrossSectoralStandard`.

Data Type: ProductOrSectorSpecificRule

A ProductOrSectorSpecificRule refers to a set of product or sector specific rules published by a specific operator and applied during product carbon footprint calculation.

Property	Data Type	Mandatory
operator	ProductOrSectorSpecificRuleOperator	yes
ruleNames	NonEmptyStringVector	yes
otherOperatorName	NonEmptyString	no

Properties

Property “operator”

A ProductOrSectorSpecificRule MUST include the property `operator` with the value conforming to data type [ProductOrSectorSpecificRuleOperator](#).

Property “ruleNames”

A ProductOrSectorSpecificRule MUST include the property `ruleNames` with value the non-empty set of rules applied from the specified [operator](#).

Property “otherOperatorName”

If the value of property `operator` is `Other`, a ProductOrSectorSpecificRule MUST include the property `otherOperatorName` with value the name of the operator. In this case, the operator declared MUST NOT be included in the definition of [ProductOrSectorSpecificRuleOperator](#).

If the value of property `operator` is NOT `Other`, the property `otherOperatorName` of the ProductOrSectorSpecificRule MUST be undefined.

JSON Representation

Each ProductOrSectorSpecificRule MUST be encoded as a JSON object.

Data Type: ProductOrSectorSpecificRuleSet

A set of [ProductOrSectorSpecificRule](#) of size 0 or larger.

JSON Representation

Each ProductOrSectorSpecificRuleSet MUST be encoded as an array of JSON objects, with each object conforming to the JSON representation of ProductOrSectorSpecificRule.

Data Type: ProductOrSectorSpecificRuleOperator

A [ProductOrSectorSpecificRuleOperator](#) is the enumeration of PCR operators. Valid values are:

- PEF for EU / [PEF](#) Methodology PCRs
- EPD International for PCRs authored or published by [EPD International](#)
- Other for a PCR *not* published by the operators mentioned above

JSON Representation

Each value is encoded as a JSON String.

Data Type: NonEmptyStringVector

A list of [NonEmptyString](#) of length 1 or greater.

JSON Representation

Each NonEmptyStringVector MUST be encoded as an array of [NonEmptyStrings](#).

Data Type: CpcCode

A CpCode represents a UN CPC Code version 2.1⁴ value.

Example value: 31230 (the CPC code for “wood in chips or particles”)

JSON Representation

Each CpcCode MUST be encoded as a JSON String.

Data Type: SpecVersionString

A String matching a dotted version pattern, given as the regular expression:

```
\d+\.\d+\.\d+
```

Example value: 1.2.3

JSON Representation

Each SpecVersionString MUST be encoded as a JSON String.

⁴ <https://unstats.un.org/unsd/classifications/Econ/CPC.cshtml>

Data Type: VersionInteger

An integer in the inclusive range of $0 \dots 2^{31}-1$.

JSON Data Representation

Each VersionInteger MUST be encoded as a number.

Data Type: DeclaredUnit

DeclaredUnit is the enumeration of accepted declared units with values

1. `liter` for unit liter
2. `kilogram` for unit kilogram
3. `cubic meter` for cubic meter
4. `kilowatt hour` for kilowatt hour
5. `megajoule` for megajoule
6. `ton kilometer` for ton kilometer
7. `square meter` for square meter

JSON Data Representation

The value of each DeclaredUnit MUST be encoded as a JSON String.

Data Type: NonEmptyString

A String with 1 or more characters.

JSON Data Representation

Each NonEmptyString MUST be encoded as a JSON String.

Data Type: CompanyIdSet

A set of [URNs](#) of length 1 or greater.

JSON Data Representation

Each CompanyIdSet MUST be encoded as an array of strings.

Data Type: ProductIdSet

A set of [ProductIds](#) of size 1 or larger.

JSON Data Representation

Each ProductIdSet MUST be encoded as an array of strings.

Data Type: ProductId

Each ProductId MUST be a valid [URN](#).

The namespace of each ProductId SHOULD be included in the [Official IANA Registry of URN Namespaces](#).

Data Type URN

A String conforming to the URN syntax as specified in [RFC 2141](#).

JSON Data Representation

Each URN string MUST be encoded as a JSON String.

Data Type: String

A regular UTF-8 String.

JSON Data Representation

Each String MUST be encoded as a JSON String.

Data Type: Percent

A Decimal number in the range of and including 0 and 100.

Example values:

- 100
- 23.0
- 7.183924
- 0.0

JSON Data Representation

Each Percent MUST be encoded in [IEEE-754 double-precision floating-point format](#) as a JSON number.

Data Type: StrictlyPositiveDecimal

A positive, non-zero Decimal.

Example values:

- 0.123
- 100
- 42.102340

JSON Data Representation

See the JSON Representation of [Decimal](#).

Data Type: DateTime

Each DateTime MUST be a date and time string conforming to ISO 8601. The timezone MUST be UTC.

Example value: 2020-03-01T00:00:00Z
(i.e. beginning of March, the year 2020, UTC)

JSON Data Representation

Each DateTime MUST be encoded as a JSON String.

Data Type: ISO3166CC

An ISO 3166-2 alpha-2 country code.

JSON Data Representation

Each ISO3166CC MUST be encoded as a JSON String.

Data Type: PositiveDecimal

A [Decimal](#) that is equal to or greater than zero.

Example values:

- 0
- 42.12
- 3

JSON Data Representation

See the JSON representation of [Decimal](#).

Data Type: Decimal

A dotted-decimal number.

Example values:

- 10

- 42.12
- -182.84

JSON Data Representation

Each Decimal MUST be encoded as a JSON String.

Data Type: Boolean

A boolean value (`true`, `false`).

JSON Data Representation

Each Boolean MUST be encoded as a JSON boolean.

Data Type: Pfid

A Pfid is a ProductFootprint identifier. It MUST be a UUID v4 as specified in [RFC 4122](#).

JSON Representation

Each Pfid MUST be encoded as a JSON String.

Product Footprint Lifecycle

Introduction

This section is non-normative.

A [Product Footprint](#) can be updated, for instance to incorporate a change in upstream carbon emission factors or to correct errors.

The purpose of this section is to clarify what constitutes an occasion for an update ([Definition of Update](#)) and to specify which kinds of changes ([Life cycle rules](#)) to a [Product Footprint](#) must be applied in such a case.

Definition of update

A Product Footprint update SHOULD be limited to correct reported values.

A Product Footprint update SHOULD be done whenever the value of 1 or more properties of the Product Footprint change which could directly or indirectly affect the Product Footprint calculation or interpretation by data recipients of the Product Footprint.

A Product Footprint update MUST NOT occur to change the [reporting period](#). For this case, a data owner MUST create a Product Footprint with a new Product Footprint identifier.

A Product Footprint update MUST NOT change the Product Footprint identifier.

Data life cycle rules

A Product Footprint CAN be updated by a data owner.

A Product Footprint update MUST include 1 or more changes to a property, excluding changes to properties [version](#) and [updated](#). A change to a property includes a transition to or from being undefined.

Whenever a data owner or a host system updates a Product Footprint according to the [Definition of update](#),

- a) it MUST set the value of property `version` to be by *strictly greater* than the value of property `version` of all preceding footprints, and
- b) it SHOULD set the value of property `updated` to the timestamp of the update. The value SHOULD be *strictly greater* than the value of property `updated` of all preceding Product Footprints.

Use Case 001 HTTP REST API Version 1.0.0

Introduction

This section is non-normative.

Use Case 001 defines foundations for the interoperable exchange of Product Footprint data between host systems.

This version 1.0.0 release is the first publication towards this goal. *Intentionally, the scope of the specifications is minimal* in order to collect market feedback and to incorporate this in future revisions.

Host System

A host system serves the needs of a single or multiple data owners. Additionally, a host system can also serve the needs of data recipients if it retrieves data from host systems by calling the [Use Case 001 HTTP REST API](#).

Interoperable data exchange between a data owner and a data recipient can be achieved by

- a) the data owner offering ProductFootprint data through a host system that [conforms](#) to the Use Case 001 HTTP REST API and
- b) the data recipient making [authenticated calls](#) to retrieve ProductFootprint data; e.g. by calling the [Action ListFootprints](#).

Out of scope of Use Case 001

This section is non-normative.

This standard focuses on the necessary definitions to enable interoperable data exchange between data owners and data recipients. This is mediated through a host system which implements the HTTP REST API defined in this document.

Within the PACT Project, [conforming](#) host systems are called *solutions*.

Solutions add further functionality on top of this standard in order to enable meaningful and interoperable data exchanges.

The following section briefly describes some of the additional functionality which is *beyond* the scope of this document:

- a) Footprint calculation according to the Pathfinder Framework

- b) Authentication and access management: the act of deciding and setting which product footprint may be accessed by each data recipient
- c) Credentials management: the overall functionality to generate access credentials for data recipients, to exchange these credentials with data recipients, to rotate or revoke such credentials, etc.
- d) Logging: creation and storage of access logs and audit trails related to data exchange, authentication processes, etc.

A note on data synchronization between Host Systems

This section is non-normative.

The data flow specified in this document is directed.

To give an example: in a supplier-vendor relationship, the supplier is the data owner and the customer the data recipient. The data owner has the responsibility to run a host system / solution and the data recipient may then retrieve the data through the data owner's host system.

If a data owner has updated a product footprint, it is assumed that the product footprint is made available through the Use Case 001 HTTP REST API as soon as possible. Additionally, each product footprint returned from a host system has properties to disclose whether a product footprint was updated. These properties are: `created`, `updated`, `version`. By comparing the value of these properties for a fixed product footprint, identified by its `id` (Property `id`), solutions can determine whether and which product footprint has been updated.

Note: in future versions, the functionality for data exchange between host systems and capabilities to synchronize data will be specified and developed further.

Authentication and Request Flows

Data recipients **MUST** first retrieve an access token through action `Authenticate`. Only if this has succeeded, they may perform actions such as `ListFootprints` or `GetFootprint`. When executing these actions, they **MUST** provide their access token.

Access tokens **SHOULD** expire. In this case, data recipients **MUST** retrieve a new access token through action `Authenticate`.

Actions reference

A host system **MUST** implement each of the following actions. A host system **MUST** offer the actions under `https` method only.

A host system MAY offer the endpoints under a relative subpath. A host system MUST offer the actions `ListFootprints` and `GetFootprint` under the same Hostname and Subpath.

Example: The host system's DNS domain name is `example.org` and the subpath is `/wbcasd` whereas the ID management system uses a different domain and subpath. The Action URIs would then be:

- for `Authenticate`: `https://id.example.org/auth/token`
- for `ListFootprints`: `https://example.org/wbcasd/0/footprints`
- etc.

Action Authenticate

Request an access token using client credentials.

Host systems MUST implement this action in conformity with [RFC 6749 Section 4.4](#).

Host systems MAY offer this action under a `AuthHostname` and `AuthSubpath` dedicated for the purpose of creating an authentication token, for instance when a company is using a centralized identity management system.

Request syntax (HTTP/1.1)

```
POST AuthSubpath/auth/token HTTP/1.1
host: AuthHostname
accept: application/json
content-type: application/x-www-form-urlencoded
authorization: Basic BasicAuth
content-length: ContentLength
content-type: application/x-www-form-urlencoded
```

Body

Request parameters:

AuthSubpath:

If a host system uses a relative subpath dedicated to the purpose of creating an authentication token, then the requesting data recipient MUST prepend this subpath.

AuthHostname:

The requesting data recipient MUST use the domain name of the host system dedicated for the purpose of creating an authentication token.

BasicAuth:

See RFC 6749 Section 4.4.2

Body:

See RFC 6749 Section 4.4

ContentLength:

The length of the Body. See [RFC 2616](#).

Response Syntax

```
HTTP/1.1 200 OK
content-type: application/json
content-length: ContentLength
```

Body

Response Parameters:

ContentLength:

The length of the Body. See RFC 2616.

Body:

The Body **MUST** be a JSON Object conforming to RFC 6749 Section 4.4. The value of property `access_token` is the access token.

Action ListFootprints

Lists product footprints with optional filtering by property `created` or `updated`.

Host systems SHOULD implement an access management system and only return the product footprints for which the data owner granted access to the requesting data recipient.

Request Syntax (HTTP/1.1)

```
GET Subpath/0/footprints?Filter HTTP/1.1
Host: Hostname
authorization: Bearer BearerToken
```

Request parameters:

Subpath:

If a host system uses a relative subpath, then the requesting data recipient MUST prepend this subpath.

Hostname:

The requesting data recipient MUST use the domain name of the host system.

BearerToken:

The requisition data recipient MUST use as BearerToken the value of property `access_token` of a response body from action `Authenticate`.

Filter:

Filter is an optional request parameter. If defined, it must conform to the `$filter` syntax as defined by the [ODataV4 specification](#). A `$filter` MUST only reference property `created` or property `updated`. The filter MUST specify [filter operation](#) `ge`.

Response syntax

```
HTTP/1.1 HttpStatusCode OK
content-type: application/json
content-length: ContentLength
```

Body

Response parameters:

HttpStatusCode:

If the host system accepts the access token, the HttpStatusCode MUST be 200.

If the host system responds with an [error response](#), the HttpStatusCode MUST match the HTTP Status Code of the respective error response code.

If the host system does not return the list of ProductFootprints, it MUST return an error HTTP Status Code (4xx, 5xx).

ContentLength:

The length of the body. See RFC 2616.

Body:

If the host system accepts the access token, the body MUST be a JSON object with property `data` with value the list of [ProductFootprints](#). The list MUST be encoded as a JSON array. If the list is empty, the host system MUST return an empty JSON array.

Additionally,

- a) if the request parameter Filter is defined against property `updated`, then only ProductFootprints with property `updated` defined and with a value equal to or greater than the timestamp given by the filter SHOULD be included in the Body; or
- b) if the filter is against property `created`, then only ProductFootprints with a creation time equal to or greater than the timestamp given by the filter SHOULD be included in the Body.

Example value of Filter: `$filter=updated ge 2020-03-01T00:00:00Z`

If the host system does not accept the access token, the body MUST be an [error response](#) with code `AccessDenied`.

If the host system does not accept the access token because it expired, the body SHOULD be an [error response](#) with code `TokenExpired`.

In all other cases, for instance in case of a malformed value of the header `authorization`, the body SHOULD be an error response with code `BadRequest`.

Example (HTTP/2)

Request

```
GET /0/footprints HTTP/2
Host: api.pathfinder.sine.dev
authorization: Bearer [BearerToken]
```

Response

```
HTTP/2 200
date: Mon, 23 May 2022 19:33:16 GMT
content-type: application/json
content-length: 1831
server: Pathfinder
```

```
{
  "data": [
    ...
  ]
}
```

Action GetFootprint

Retrieves product footprints.

Host systems SHOULD implement an access management system and only return the product footprints for which the data owner granted access to the requesting data recipient.

Request Syntax (HTTP/1.1)

```
GET Subpath/0/footprints/PfId HTTP/1.1
Host: Hostname
authorization: Bearer BearerToken
```

Request Parameters:

Subpath:

If a host system uses a relative subpath, then the requesting data recipient MUST prepend this subpath.

Hostname:

The requesting data recipient MUST use the domain name of the host system.

BearerToken:

The requesting data recipient MUST use as `BearerToken` the value of property `access_token` of a response body from action `Authenticate`.

PfId:

The requesting data recipient MUST use as `PfId` the [id](#) of a [product footprint](#) it intends to retrieve.

Response Syntax

```
HTTP/1.1 HttpStatusCode OK
content-type: application/json
content-length: ContentLength
```

Body

Response Parameters:

HttpStatusCode:

If the host system accepts the access token, the `HttpStatusCode` MUST be 200.

If the host system responds with an [error response](#), the HttpStatusCode MUST match the HTTP Status Code of the respective error response code.

If the host system does not return a ProductFootprint, the host system MUST return an error HTTP Status Code (4xx, 5xx).

ContentLength:

The length of the body. See RFC 2616.

Body:

If the host system accepts the access token and allows the requesting data recipient to access the ProductFootprint, the body MUST be a JSON object with property `data`. The value of property `data` MUST be the product footprint with footprint identifier **PfId**.

If there were [changes](#) to the ProductFootprint with identifier **PfId**, the host system SHOULD return the latest ProductFootprint identified with identifier **PfId** and the maximum value of property `version`.

Note: If a host system implements the life cycle rules, then the “latest” version of a ProductFootprint is the one with the maximum value of version given a fixed **PfId**.

If the host system does not accept the access token, the body MUST be an [error response](#) with code `AccessDenied`.

If the host system does not accept the access token because it expired, the body SHOULD be an [error response](#) with code `TokenExpired`.

The host system MAY return an error response with code `NoSuchFootprint`.

In all other cases, for instance in case of a malformed value of the header `authorization`, the body SHOULD be an error response with code `BadRequest`.

Example (HTTP/2)

Request

```
GET /0/footprints/7 HTTP/2
Host: api.pathfinder.sine.dev
authorization: Bearer BearerToken
```

Response

```
HTTP/2 200
date: Mon, 23 May 2022 20:23:22 GMT
content-type: application/json
content-length: 961
server: Pathfinder
```

```
{
  "data": {
    ...
  }
}
```

Error responses

When an action error occurs, the header information contains the following:

- Content-Type: application/json
- An appropriate 4xx or 5xx HTTP Status Code

The body of the error response contains information about the error. The host system **MUST** encode error responses as a JSON Object. The JSON object **MUST** have properties `code` and `message`. The table [list of error codes](#) lists the values for the property `code`.

Example error response:

```
{
  "code": "NoSuchFootprint",
  "message": "The footprint you requested does not exist"
}
```

List of error codes

The following table lists the error codes

Error code	Description	HTTP status code
AccessDenied	Access denied	403
BadRequest	Bad Request	400
NoSuchFootprint	The specified footprint does not exist.	404
NotImplemented	The specified Action or header you provided implies functionality that is not implemented	400
TokenExpired	The specified access token has expired	401
InternalServerError	An internal or unexpected error has occurred	500

It is assumed that a host system uses as values for the property `message` the descriptions given in the [list of error codes](#) table. Host system implementations **MAY** choose different messages.

A requesting data recipient **MUST** use the code property and potentially also the HTTP Status Code to differentiate between the different errors.

A host system **SHOULD** conform to [RFC 6750 Section 3](#) when returning the error `TokenExpired`.

Appendix: License

The following terms, conditions, and provisions (this "License") governs any and all use of the Pathfinder Network Technical Specifications for Use Case 001: PCF Data Exchange (the "Licensed Material"). Any person who wants to make any use the Licensed Material or on whose behalf any use of the Licensed Material is made ("User") may make any use of the Licensed Material or any part thereof only subject to and conditioned on User's acceptance and agreement to, and compliance with, all terms, conditions, and provisions of this License. If User makes any user of the Licensed Material, or any part thereof, User will irrevocably be deemed to have accepted and agreed to this License and is obligated to make use of the Licensed Material or any part thereof solely in compliance with this License. Any person not willing to accept and agree to, or comply with, this License shall refrain from making any use of the Licensed Material or any part thereof.

1. Definitions.

1.1 "Database Rights" means, to the extent any content in the Licensed Material is subject to any sui generis database right under any applicable law, the rights to extract, reuse, reproduce, and Share all or a substantial portion of the content subject to such database right.

1.2 "Derivative" means any work or other creation that is a derivative work or otherwise derived from, based on, or including or using the Licensed Material, or any information or content thereof, in whatever form, format, or medium, now known or becoming known in the future.

1.3 "Licensed Rights" means any copyrights (including exclusive rights of reproduction, performance, display, creation of derivative works, and distribution), and any Database Rights, in the Licensed Material anywhere in the world owned by Licensor that Licensor has the right to license. Licensed Rights do not include (i) any intellectual property or other rights other than copyrights and Database Rights (including, without limitation, any rights in, to, or under any patent, patent application, trademark, service mark, trade dress, name, domain name, trade secret, or other intellectual property right anywhere in the world), (ii) any rights of integrity or attribution or other moral rights anywhere in the world, (iii) any rights to assert or enforce any copyrights or Database Rights against any other person, including, without limitation, in case of any infringement or violation, or to assert or collect any remedy therefor, and (iv) any copyrights or Database Rights in any subject matter other than the Licensed Material.

1.4 "Licensor" means World Business Council for Sustainable Development.

1.5 "Modified Version" means any copy or version of the Licensed Material that includes any alteration of, editing of, deletion from, addition or supplement to, or change or redaction of, or any variation of any content or information of, the Licensed Material, in whatever form, format, or medium, now known or becoming known in the future.

1.6 "Non-Commercially" means without the primary intent for or directed towards commercial advantage or monetary compensation.

1.7 "Reproduction" means an exact reproduction and copy of the Licensed Material as a whole or of any Translation released by Licensor under this License, without any alteration, edits, deletions, additions, supplements, changes, or redactions, and including, for the avoidance of doubt, this License. A Modified Version is not a Reproduction.

1.8 "Separate Derivative" means any Derivative that does not consist of or include a Reproduction, any Modified Version, or any Translation, or a part of a Reproduction, Modified Version, or Translation that is by itself the subject of Licensed Rights. An example of a Separate Derivative is a software program, course, written material, or program that uses or implements the information included in the Licensed Material without including, displaying, or performing the Licensed Material, any Modified Version, or any Translation, or any portion thereof.

1.9 "Share" means to provide, by any means or process, including by distribution of a copy, reproduction, public display, public performance, dissemination, communication, or importation, or to make available for access on a website or otherwise, to any other person as a member of the public

1.10 "Translation" means any translation of the Licensed Material as a whole (and not only a part thereof) that into any language other than English that reasonably accurately reflects the content of the Licensed Material in such other language.

1.11 "Use" means the practice or exercise of any of the Licensed Rights

2. License.

2.1 Subject to the terms and conditions of this License, Licensor hereby grants User during the Term (as defined in Section 6.1), solely under the Licensed Rights, a non-exclusive, worldwide, royalty-free, non-sublicensable, non-transferable, irrevocable license to Use the Licensed Material solely to:

- a. reproduce, but not Share (other than as permitted under Section 2.1(c)), the Licensed Material or any part thereof;
- b. create, but not Share (other than as permitted under Section 2.1(d)), any Derivative;

c. Share (subject to compliance with Section 3.1) any Reproduction but only Non-Commercially;

d. Share (subject to compliance with Section 3.2) any Separate Derivative, and create from such Separate Derivative any further Separate Derivatives and Share such further Separate Derivatives, all whether Non-Commercially or commercially.

2.2 If User Shares a Reproduction under Section 2.1(c) to another person, User automatically grants such other person an offer from Licensor to Use such Reproduction under this License, and such other person shall be deemed to accept and be subject to this License (and be a User under this License) once such other person make any Use of the Licensed Material or any part thereof. Other than this License, User may not impose any limitations, restrictions (technical, legal, or otherwise), payment, terms, conditions, or requirement of any kind on any such other person related to the access to or Use of such Reproduction under this License.

2.3 User shall make any User and exercise any of the Licensed Rights solely in accordance and compliance with all applicable law, and User shall not infringe any intellectual property right of any third party and/or misappropriate any information or intellectual property of any third party.

3. Attribution.

3.1 If User Shares any Reproduction under Section 2.1(c), User must retain, and not change, obstruct, or obscure in such Reproduction, the following information:

- (i) identification of Licensor as the owner or rightholder of the Licensed Rights in the Licensed Material, and the name of any creator(s) of the Licensed Material identified herein;
- (ii) the copyright notice in this Licensed Material;
- (iii) the full text of this License.

3.2 If User Shares a Separate Derivative under Section 2.1(d), User must include a written notice in such Separate Derivative reasonably visible to any user of such Separate Derivative that:

- (i) identifies the Licensed Material and such Separate Derivative as a derivative work or otherwise derived from, based on, or including or using the Licensed Material;
- (ii) identifies Licensor as the owner or rightholder of the Licensed Rights in the Licensed Material, and the name of any creator(s) of the Licensed Materials identified herein;
- (iii) reproduces exactly and in full Section 5 (Disclaimer of Warranties and Limitation of Liability).

3.3 Upon Licensor's request, User shall remove or modify any of the information required by Section 3.1 or Section 3.2, as applicable, with regard to any Reproduction or Separate Derivative Shared after such request as reasonably practicable.

3.4 Nothing in this License constitutes or shall be construed to be, or shall imply, that User is, or User's use of the Licensed Material in any way is, connected with, or sponsored, endorsed, or granted official status by, Licensor or any other person associated with Licensor or to whom attribution is to be given under Section 3.1 or Section 3.2.

4. Ownership.

4.1 In connection with this License, and as between Licensor and User, Licensor is and shall remain the sole owner of all rights, title, and interest in and to any and all of the Licensed Rights and the Licensed Material.

4.2 Licensor does not, and shall not be deemed to, assign, transfer, or convey any right, title, or interest in or to any Licensed Right or any other intellectual property or intellectual property right, or grant any license (other than the express license set forth in Section 2.1), lien, option, or claim in or to any Licensed Right or any other intellectual property or intellectual property right.

4.3 Nothing in this License shall prevent User from owning any intellectual property rights in or to any Derivative created by User under and in accordance with the license in Section 2.1, subject to Licensor's continued ownership and rights in and to the Licensed Rights and Licensed Material. A Derivative does not give, and shall not be deemed to give, User any greater rights in or to any Licensed Rights or Licensed Material, or any right to Use, other than as expressly set forth in Section 2.1.

4.4 Licensor is in no way limited or restricted license, exercise, or exploit any of the Licensed Rights under any other terms or agreements, commercially or Non-Commercially, or to assign, transfer, or convey any of the Licensed Rights, all as decided by Licensor in its sole discretion.

5. Disclaimer of Warranties and Limitation of Liability.

5.1 Licensor offers, provides, makes available, and licenses the Licensed Material, and grants the license and Licensed Rights, "as is" and "as available" without any representations or warranties, express or implied. Licensor hereby disclaims all representations and warranties, express, implied, statutory, or otherwise, under any law, including, without limitation, all representations and warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are limited by applicable law, this disclaimer applies to the greatest extent not limited by such applicable law.

5.2 In no event will Licensor be liable to User or any other person, under contract, negligence, willful misconduct, other tort, property, intellectual property, or other cause or legal theory any legal theory or otherwise, for any direct, indirect, special, incidental, consequential, punitive, exemplary, liquidated, or other losses, costs, expenses, attorneys' or legal fees or costs, or damages of any kind anywhere, under any law, arising or related to this License, any Licensed Right, the Licensed Material, any Derivative, and/or any part thereof or any Use, even if Licensor has been advised of the possibility of such losses, costs, expenses, attorneys' or legal fees or costs, or damages of any kind anywhere. Where the exclusion or limitation of liability is limited by applicable law, this exclusion and limitation of liability applies to the greatest extent not limited by such applicable law.

5.3 The disclaimer of representations and warranties and the exclusion and limitation of liability under Sections 5.1 and 5.2 shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

6. Term and Termination.

6.1 This License and the license and rights hereunder are effective from the time when User accepts and agrees to this License until the expiration or termination of the Licensed Rights or the termination of this Licensed under Section 6.2, whichever occurs earlier (the "Term").

6.2 This License, and the license and rights hereunder, terminate automatically upon User's breach or non-compliance with any of the terms, conditions, or provisions of this License, provided that, if User fully cures and corrects such breach or non-compliance within thirty (30) days after User's first discovery of such breach or non-compliance, this License and the license and rights hereunder are reinstated automatically.

6.3 The provisions of Sections 1, 2.1(d) and 3.2 (with regard to any Separate Derivative created by User prior to the expiration or termination of this License), 4, 5, and 7, and this Section 6.3, shall survive any termination or expiration of this License.

7. Miscellaneous.

7.1 User recognizes that any actual or potential violation, breach, or non-performance of, or default under, any provision in this License may cause irreparable injury to Licensor for which Licensor may have no adequate remedy at law. User agrees that Licensor shall be entitled to seek injunctive relief or specific performance, without need or obligation to post any bond, to enforce any obligation, agreement, covenant, term and condition under this License, in addition to any other rights and remedies available to Licensor, all as Licensor elects in its sole discretion.

7.2 This License does not constitute, and shall not be construed as constituting, a partnership or joint venture between User, Licensor, or any other person. No third party shall be any beneficiary, intended or unintended, under this License.

7.3 User may not assign or transfer this License or any right hereunder or part hereof, or delegate any obligation under this License, without the express prior written consent of Licensor; any such unpermitted transfer, assignment, or delegation shall be null and void. Otherwise, this License shall bind any successor and permitted assign of User. Licensor may transfer or assign this Agreement or any right hereunder, or delegate any obligation hereunder, at any time in its sole discretion, including, without limitation, in connection with any assign, transfer, or conveyance of any Licensed Right

7.4 “Section” refers to any of the numbered sections of this License. Any reference to any law shall be construed as a reference to that law as amended or changed at the relevant time. Where a provision of this License states that a party “shall” or “will” perform or act or not act in some manner, such party is legally obligated to do so in accordance with such provision this License. Section headings and titles are intended solely for the convenience of the parties.

7.5 This License constitutes the entire understanding and agreement between User and Licensor related to the subject matter hereof. No right or remedy of User or Licensor under this License may be waived, discharged, or changed other than by a written instrument expressly identifying such right or remedy and signed by User or Licensor, as applicable. No amendment, modification, or change of this License or any term, condition, or provision of this License will be binding upon User and Licensor unless expressly set forth in a writing signed by each of User and Licensor through its authorized representative. A failure of User or Licensor to exercise any right or remedy under this License shall not be deemed to be a waiver of any right or remedy hereunder.

7.6 This License, any performance hereunder, the interpretation, construction, validity, and enforceability of this License and any term, condition, and provision of this License, and any dispute, and the resolution of any dispute, under this License shall be governed by the law of the State of New York, United States of America, without regard to any conflicts of laws provisions thereof that would result in the application of the law of any other jurisdiction.

7.7 If any term, condition, or provision of this License is held to be invalid or unenforceable, the meaning thereof will be construed, to the extent feasible, so as to render the provision enforceable, and if no feasible interpretation shall save such provision, it will be severed from the remainder of this License, as appropriate, which remainder shall remain in full force and effect unless the severed provision is essential and material to the rights or benefits received by User and Licensor, in which case this License shall be deemed terminated (subject to Section 6.3, as applicable).