TOSCA Test Assertions Version 1.0

Committee Note Draft 01

08 December 2016

Specification URIs

This version:

http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/cnd01/TOSCA-Test-Assertions-v1.0-cnd01.pdf (Authoritative)
http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/cnd01/TOSCA-Test-Assertions-v1.0-cnd01.html
http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/cnd01/TOSCA-Test-Assertions-v1.0-cnd01.docx

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/TOSCA-Test-Assertions-v1.0.pdf (Authoritative)
http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/TOSCA-Test-Assertions-v1.0.html
http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/TOSCA-Test-Assertions-v1.0.docx

Technical Committee:

OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC

Chairs:

Paul Lipton (paul.lipton@ca.com), CA Technologies
John Crandall (jcrandal@brocade.com), Brocade Communications Systems

Editor:

Luc Boutier (luc.boutier@fastconnect.fr), FastConnect

Related work:

This document is related to:

- *Topology and Orchestration Specification for Cloud Applications Version 1.0*. Edited by Derek Palma and Thomas Spatzier. 25 November 2013. OASIS Standard. http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html.
- *TOSCA Simple Profile in YAML Version 1.1*. Edited by Matt Rutkowski and Luc Boutier. Latest version: http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.1/TOSCA-Simple-Profile-YAML-v1.1.html.

Abstract:

This document describes templates and definitions that can be used to test TOSCA tools and orchestrator compliance.

Status:

This document was last revised or approved by the OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee (TC) members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/tosca/.

Citation format:

When referencing this document the following citation format should be used:

[TOSCA-Test-v1.0]

TOSCA Test Assertions Version 1.0. Edited by Luc Boutier. 08 December 2016. OASIS Committee Note Draft 01. http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/cnd01/TOSCA-Test-Assertions-v1.0-cnd01.html. Latest version: http://docs.oasis-open.org/tosca/TOSCA-Test-Assertions/v1.0/TOSCA-Test-Assertions-v1.0.html.

RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1 Conformance

This document aims to specify the requirements for implementations to be considered as TOSCA compliant.

1.1 Conformance Targets

TOSCA simple profile in YAML defines conformance targets and requirements on their conformance in respect to the specification document. This document aims to provide test scenarios that can be used in order to validate tools or artifacts of the TOSCA ecosystem.

As a reminder the following categories are defined as conformance targets in the specification document:

- TOSCA YAML service template
- TOSCA archive
- TOSCA processor
  - Parsers and validator
  - Matching engines
    - Node matcher
    - Locations matcher
  - Orchestrators
  - Auditor
  - Optimizer
  - Catalog/Forge/Registry
- TOSCA generator
  - IDE, Tools etc.
  - Auditor
  - Optimization engines

As specified a TOSCA orchestrator is a specific TOSCA processor.

1.1.1 TOSCA YAML service template or TOSCA archive

A TOSCA yaml service template or TOSCA archive is considered as compliant if it can be validated by a TOSCA compliant parser that successfully qualifies it as a valid TOSCA YAML service template or TOSCA archive.

1.1.2 TOSCA Parser and validator

Tosca parsers are tools that can read and validate TOSCA YAML document and TOSCA Archives. This document details the expected behavior of TOSCA Parsers.

This document defines a set of conformance scenarios that should be processed with correct output by the parser in order to be confirmed as being TOSCA compliant.

Note: This document doesn't provide conformance for TOSCA parsing and validation as separated elements. We consider here tools that Parse and perform validation of the TOSCA templates.

1.1.3 TOSCA Matching engine

A TOSCA matching engine is responsible for matching a TOSCA template with the resources that an underlying cloud system can provide. Most obvious example is the matching of a TOSCA Compute node that specifies a VM and constraints in term of operating system and hardware resources. Advanced matching engines may support policies and also take pricing data in account to provide best choices to their users.

1.1.4 TOSCA Orchestrator

Note: TOSCA orchestrators should not try to orchestrate any TOSCA service template or archive that cannot be parsed by a TOSCA parser. This specification will not specify the expected behavior of orchestrators for any sample that should fail at parsing.

Note that a valid TOSCA orchestrator may fail to orchestrate some valid TOSCA Archives in case of errors in user provided scripts or inconsistencies between TOSCA node specifications and artifacts even if the Template or Archive is considered as valid from a TOSCA parsing point of view.

1.1.5 TOSCA generator

TOSCA generators are considered as valid TOSCA generators only if the TOSCA service template or TOSCA archive they generate can be parsed by a TOSCA compliant parser that successfully qualifies it as a valid TOSCA YAML service template or TOSCA archive.

Moreover, the template or archive produced by the processor should be compliant with the generated lifecycle of TOSCA so that the archive can be processed by a compliant TOSCA orchestrator. The created template should provide enough information that enables any TOSCA orchestrator to perform a valid matching of resources that ensures the various implementation artifacts to be process in a valid way.

1.2 Prescription levels

TOSCA plans to support multiple conformance levels in the future so that new tools can support only a subset of the overall specification and improve their support while still being officially recognized as part of the TOSCA ecosystem.

Level	Description
Mandatory	This level is required to be considered a valid TOSCA target.
recommended	This level is recommended to deliver support for some important TOSCA features.
optional	Optional prescription level is used to provide support for TOSCA experimental features or some TOSCA extentions.

2 Conformance coverage

The following table allows to validate that the various definitions and constraints of the specification document are covered by tests.

Note: some specification element may be covered in more than one test but the one(s) considered as most significant(s) and requiring the lower subset of overall coverage are referenced here.

Spec. Ref.	Conformance document references (or assignee)	Target(s)	Note
3			
3.1			
3.1.1			Ensure no people can define a TOSCA alias for their namespace.
3.1.2		Parser	
3.1.3			Namespace collision tests for every type, template and properties etc.
3.2			Parameters and property types support should be managed with property and constraint tests.
3.2.1			YAML types
3.2.2			Version
3.2.3			Range type
3.2.4			List type
3.2.5			Map type
3.2.6			Scalar-unit types
3.3			Normative values
3.3.1			Node States
3.3.2			Relationship states
3.3.3			Directives
3.3.4			Network name aliases
3.4	N.A		TOSCA Metamodel
3.4.1	N.A		Required Keynames
3.5.1			Description definition
3.5.2			Constraint clause
3.5.3			Property filter definition
3.5.4			Node filter definition
3.5.5			Repository definition
3.5.6			Artifact definition
3.5.7			Import definition
3.5.8			Property definition
3.5.9			Property assignment
3.5.10			Attribute definition
3.5.11			Attribute assignment
3.5.12			Parameter definition
3.5.13			Operation definition
3.5.14			Interface definition
3.6			Type-specific definitions
3.6.1			Capability definition
3.6.2			Requirement definition
3.6.3			Artifact Type
3.6.4			Interface Type
3.6.5			Data Type
3.6.6			Capability Type
3.6.7			Requirement Type
3.6.8			Node Type
3.6.9			Relationship Type
3.6.10			Group Type
3.6.11			Policy Type
3.7.1			Capability assignment
3.7.2			Requirement assignment

3.7.3			Node template
3.7.4			Relationship template
3.7.5			Group definition
3.7.6			Policy definition
3.8			Topology template definition
3.9			Service template definition
4			TOSCA functions
4.1			Reserved function keywords
4.2			Environment variable conventions
4.2.1			Reserved environment variable names and usage
4.2.2			Prefixed vs. Unprefixed TARGET names
4.3			Intrinsic functions
4.3.1			concat
4.3.2			token
4.4			Property functions
4.4.1			get_input
4.4.2			get_property
4.5			Attribute functions
4.5.1			get_attribute
4.6			Operation functions
4.6.1			get_operation_output
4.7			Navigation functions
4.7.1			get_nodes_of_type
4.8			Artifact functions
4.8.1			get_artifact
5			TOSCA Normative type definitions
5.2			Data types
5.2.1			tosca.datatypes.Root
5.2.2			tosca.datatypes.Credential
5.2.3			tosca.datatypes.network.NetworkInfo
5.2.4			tosca.datatypes.network.PortInfo
5.2.5			tosca.datatypes.network.PortDef
5.2.6			tosca.datatypes.network.PortSpec
5.3			Artifact types
5.3.1			tosca.artifacts.Root
5.3.2			tosca.artifacts.File
5.3.3			Deployment types
5.3.3.1			tosca.artifacts.Deployment
5.3.3.3			tosca.artifacts.Deployment.Image
5.3.3.4			tosca.artifacts.Deployment.Image.VM
5.3.4			Implementation Types
5.3.4.1			tosca.artifacts.Implementation
5.3.4.3			tosca.artifacts.Implementation.Bash
5.3.4.4			tosca.artifacts.Implementation.Python
5.4			Capability Types
5.4.1			tosca.capabilities.Root
5.4.2			tosca.capabilities.Node
5.4.3			tosca.capabilities.Container
5.4.4			tosca.capabilities.Endpoint
5.4.5			tosca.capabilities.Endpoint.Public
5.4.6			tosca.capabilities.Endpoint.Admin
5.4.7			tosca.capabilities.Endpoint.Database
5.4.8			tosca.capabilities.Attachment
5.4.9			tosca.capabilities.OperatingSystem

5.4.10			tosca.capabilities.Scalable
5.4.11			tosca.capabilities.network.Bindable
5.5			Requirement Types
5.6			Relationship Types
5.6.1			tosca.relationships.Root
5.6.2			tosca.relationships.DependsOn
5.6.3			tosca.relationships.HostedOn
5.6.4			tosca.relationships.ConnectsTo
5.6.5			tosca.relationships.AttachesTo
5.6.6			tosca.relationships.RoutesTo
5.7			Interface Types
5.7.3			tosca.interfaces.Root
5.7.4			tosca.interfaces.node.lifecycle.Standard
5.7.5			tosca.interfaces.relationship.Configure
5.8			Node Types
5.8.1			tosca.nodes.Root
5.8.2	TODO: Parser should parse shorthand Names and Type URL names.	Parser	tosca.nodes.Compute
5.8.3			tosca.nodes.SoftwareComponent
5.8.4			tosca.nodes.WebServer
5.8.5			tosca.nodes.WebApplication
5.8.6			tosca.nodes.DBMS
5.8.7			tosca.nodes.Database
5.8.8			tosca.nodes.ObjectStorage
5.8.9			tosca.nodes.BlockStorage
5.8.10			tosca.nodes.Container.Runtime
5.8.11			tosca.nodes.Container.Application
5.8.12			tosca.nodes.LoadBalancer
5.9			Group Types
5.9.1			tosca.groups.Root
5.10			Policy Types
5.10.1			tosca.policies.Root
5.10.2			tosca.policies.Placement
5.10.3			tosca.policies.Scaling
5.10.4			tosca.policies.Update
5.10.5			tosca.policies.Performance
6			CSAR
7			TOSCA Networking

3 TOSCA Grammar test assertions

This section contains the test assertions for the TOSCA grammar validation (basically parser/validator). It follows the same structure as the Specification document section 3 in order to highlight specification test coverage and to allow implementors to refer to the right section of the specification.

3.1 TOSCA Namespace URI and alias

3.1.1 TOSCA Namespace prefix

3.1.2 TOSCA Namespacing in TOSCA Service Templates

Id: 3.1.2-tosca_definitions_version-01-valid-definition

Prerequisite:

Description: Parsing a document with valid tosca definition version MUST succeed.

Target: a tosca template that has a valid tosca_definitions_version value equals to 'tosca_simple_yaml_1_0'

Predicate: When parsing the template assert 'tosca_definitions_version' value is equal to 'http://docs.oasis-open.org/tosca/ns/simple/yaml/1.0' or 'tosca_simple_yaml_1_0'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.1.2-tosca_definitions_version-02-valid-definition-url

Prerequisite:

Description: Parsing a document with valid tosca definition version defined as an url MUST succeed

Target: a tosca template that has a valid tosca_definitions_version value equals to 'http://docs,oasis-open,org/tosca/ns/simple/yaml/1,0'

Predicate: When parsing the template assert 'tosca_definitions_version' value is equal to 'http://docs.oasis-open.org/tosca/ns/simple/yaml/1.0' or 'tosca_simple_yaml_1_0'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.1.2-tosca_definitions_version-03-invalid

Prerequisite:

Description: Parser SHOULD be failing if tosca_definitions_version is not valid.

Target: a tosca template that has an invalid tosca_definitions_version value equals to 'not_tosca_simple_yaml_1_0'

Predicate: When parsing the template assert raises the error 'InvalidTOSCAVersion'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.1.2-tosca_definitions_version-04-missing

Prerequisite:

Description: Parsing a document without tosca definition version MUST fail.

Target: a tosca template that does not define tosca_definitions_version

Predicate: When parsing the template assert raises the error 'MissingTOSCAVersion'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.1.2-tosca_definitions_version-05-not_first_line

Prerequisite:

Description: Parsing a document where tosca definition is not the first line version SHOULD fail.

Target: a tosca template where description is defined before tosca_definitions_version

Predicate: When parsing the template assert raises the error 'TOSCAVersionMustBeFirstLine'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.1-description-01-valid_single_line

Prerequisite:

Description: Parsing a document with valid description MUST succeed.

Target: a tosca template that has a valid description value equals to 'This is an example of a single line description [no folding],'

Predicate: When parsing the template assert 'description' value is equal to 'This is an example of a single line description (no folding).'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.1-description-02-valid_multi_line

Prerequisite:

Description: Parsing a document with valid multi-line description MUST succeed.

Target: a tosca template that has a valid multi-line description value equals to 'This is an example of a multi-line description using YAML, It permits for line breaks for easier readability,,, if needed, However, [multiple] line breaks are folded into a single space character when processed into a single string value,'

Predicate: When parsing the template assert 'description' value is equal to 'This is an example of a multi-line description using YAML. It permits for line breaks for easier readability... if needed. However, (multiple) line breaks are folded into a single space character when processed into a single string value.'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.1-description-03-invalid

Prerequisite:

Description: Parser MUST fail if description is not a string.

Target: a tosca template where the description value is a complex yaml value [map/list]

Predicate: When parsing the template assert raises the error 'InvalidType'

Prescription level: mandatory

Conformance target: Parser-Validator

3.5.2 Constraint clause

3.5.3 Property Filter definition

3.5.4 Node Filter definition

3.5.5 Repository definition

Id: 3.5.5-repositories-01-valid-definition

Prerequisite:

Description: Parsing a document with valid repository MUST succeed.

Target: a tosca template that defines a valid repository definition

Predicate: When parsing the template Then The repository out of the parsing assert 'description' value is equal to 'A repository' assert 'url' value is equal to 'https://github.com/my-project/' assert 'credential.user' value is equal to 'username' assert 'credential.token' value is equal to 'password'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.5-repositories-02-valid-simple-definition

Prerequisite:

Description: Parsing a document with valid repository MUST succeed.

Target: a tosca template that defines a valid single line repository

Predicate: When parsing the template Then The repository out of the parsing assert 'url' value is equal to 'https://github.com/my-other-project/'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.5-repositories-03-no-url

Prerequisite:

Description: Parsing a document that defines a repository without url SHOULD fail.

Target: a tosca template that defines a repository definition without url

Predicate: When parsing the template assert raises the error 'MissingRequiredKeyname'

Prescription level: mandatory

Conformance target: Parser-Validator

3.5.6 Artifact definition

3.5.7 Import definition

Id: 3.5.7-imports-01-simple-relative

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with an single line import from a relative file MUST be successful.

Target: a tosca template that defines a valid single line import to a relative file

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-02-relative

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with an import from a relative file MUST be successful.

Target: a tosca template that defines a valid single line import to a relative file

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-03-no-file

Prerequisite:

Description: Parsing a document with an import that does not specifies a file MUST fail.

Target: a tosca template that defines an import without a file key

Predicate: When parsing the template assert raises the error 'MissingRequiredKeyname'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-04-missing-relative-file

Prerequisite:

Description: Parsing a document with an import that points to a local missing file SHOULD fail.

Target: a tosca template that defines an import that points to a local missing file

Predicate: When parsing the template assert raises the error 'MissingImportFile'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-05-simple-url

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with an single line import from a remote file MUST be successful.

Target: a tosca template that defines a valid single line import to a remote file

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-06-missing-remote-file

Prerequisite:

Description: Parsing a document with an single line import from a remote file that does not exists SHOULD fail.

Target: a tosca template that defines a valid single line import to a remote file that does not exists

Predicate: When parsing the template assert raises the error 'MissingImportFile'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-07-repository-remote

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with an single line import from a remote file MUST be successful.

Target: a tosca template that defines an import to a remote file resolved through a repository definition

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-08-missing-repository-remote

Prerequisite:

Description: Parsing a document with an import to a remote file resolved through a missing repository definition SHOULD fail.

Target: a tosca template that defines an import to a remote file resolved through a missing repository definition

Predicate: When parsing the template assert raises the error 'MissingImportFile' assert raises the error 'UnknownRepository'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.5.7-imports-09-unreachable-repository-remote

Prerequisite:

Description: Parsing a document with an import to a missing remote file resolved through a repository definition SHOULD fail.

Target: a tosca template that defines an import to a missing remote file resolved through a repository definition

Predicate: When parsing the template assert raises the error 'MissingImportFile'

Prescription level: mandatory

Conformance target: Parser-Validator

3.5.8 Property definition

3.5.9 Property assignment

3.5.10 Attribute definition

3.5.11 Attribute assignment

3.5.12 Parameter definition

3.5.13 Operation definition

3.5.14 Interface definition

3.6 Type-specific definitions

3.6.1 Capability definition

3.6.2 Requirement definition

3.6.3 Artifact Type

Id: 3.6.3-artifact_type-01-valid_simple

Prerequisite:

Description: This test ensure that the parser is able to parse a TOSCA artifact type. The version keyname is not defined as it's usage is not defined in the specification. The properties keyname is as well not defined as it's usage is also not defined in the specification.

Target: a tosca template that defines a valid artifact_type with all keynames but version and properties

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.3-artifact_type-02-valid_all_keynames

Prerequisite:

Description: This test ensure that the parser is able to parse a TOSCA artifact type where all keynames are defined.

Target: a tosca template that defines a valid artifact_type with all keynames

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.3-artifact_type-03-no_root_inherited

Prerequisite:

Description: This test ensure that the parser is able to parse a TOSCA artifact type where all keynames are defined.

Target: a tosca template that defines a valid artifact_type with only the description

Predicate: When parsing the template assert there is no errors optional assert raises the warning 'WarnNotInheritFromRoot'

Prescription level: preferred

Conformance target: Parser-Validator

Id: 3.6.3-artifact_type-04-unknown_parent_type

Prerequisite:

Description: Parsing an artifact type that derives from an unknown type should fail.

Target: a tosca template that defines an artifact_type with derived_from set to 'tosca,test,UnknownType' and no imports

Predicate: When parsing the template assert raises the error 'MissingArtifactType'

Prescription level: mandatory

Conformance target: Parser-Validator

3.6.4 Interface Type

Id: 3.6.4-interface_type-01-all-keynames

Prerequisite:

Description: Parsing an interface type that defines all keynames should be successful.

Target: a tosca template that defines a valid interface type with all keynames

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.4-interface_type-02-only-required-keynames

Prerequisite:

Description: Parsing an interface type with required keynames only should be successful.

Target: a tosca template that defines a valid interface type with no optional keynames

Predicate: When parsing the template assert there is no errors optional assert raises the warning 'WarnNotInheritFromRoot'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.4-interface_type-03-inputs-operation

Prerequisite:

Description: Parsing an interface type with an operation mapped to the inputs keyname should fail

Target: a tosca template that defines interface type with an operation mapped to the inputs keyname

Predicate: When parsing the template assert raises the error 'InvalidSyntax'

Prescription level: mandatory

Conformance target: Parser-Validator

3.6.5 Data Type

Id: 3.6.5-data_type-01-complex_type

Prerequisite:

Description: Parsing a data type that defines a complex type should be successful.

Target: a tosca template that defines a valid 'tosca,example,types,Person' data_type with a description, 2 string properties and 1 integer property

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-02-complex_type_derived

Prerequisite:

Description: Parsing a data type that defines a complex type that derives from another one should be successful.

Target: a tosca template that defines a valid 'tosca,example,types,Person' data_type and a 'tosca,example,types,AdvancedPerson' that derives from the 'tosca,example,types,Person' type and override the a string property to add a min_length constraint and add a new string property

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-03-complex_type_derived_unknown

Prerequisite:

Description: Parsing a data type that defines a complex type that derives from an unknown complex type should fail.

Target: a tosca template that defines a valid 'tosca,example,types,Person' that derives from an unknown data type

Predicate: When parsing the template assert raises the error 'InvalidParentType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-04-complex_type_complex_property

Prerequisite:

Description: Parsing a data type that defines a complex type using another complex type should be successful.

Target: a tosca template that defines a 'tosca,example,types,Address' data type and a 'tosca,example,types,Person' data type that defines a property of type 'tosca,example,types,Address'

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-04-complex_type_complex_property

Prerequisite:

Description: Parsing a data type that defines a complex type using another unknown complex type should fail.

Target: a tosca template that defines a 'tosca,example,types,Person' data type that defines a property of type 'tosca,test,UnknownType' with no imports

Predicate: When parsing the template assert raises the error 'UnknownDataType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-06-complex_type_list_property_complex

Prerequisite:

Description: Parsing a data type that defines a complex type with a list of another complex type should be successful.

Target: a tosca template that defines a 'tosca,example,types,Address' data type and a 'tosca,example,types,Person' data type that defines a property of type list of 'tosca,example,types,Address'

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-07-complex_type_list_property_type_unknown

Prerequisite:

Description: Parsing a data type that defines a complex type with a list of another unknown complex type should fail.

Target: a tosca template that defines a 'tosca,example,types,Address' data type and a 'tosca,example,types,Person' data type that defines a property of type list of a type that is neither normative or a defined data type

Predicate: When parsing the template assert raises the error 'UnknownDataType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-08-complex_type_map_property_complex

Prerequisite:

Description: Parsing a data type that defines a complex type with a map of another complex type should be successful.

Target: a tosca template that defines a 'tosca,example,types,Address' data type and a 'tosca,example,types,Person' data type that defines a property of type map of 'tosca,example,types,Address'

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-09-complex_type_map_property_type_unknown

Prerequisite:

Description: Parsing a data type that defines a complex type with a map of another unknown complex type should fail.

Target: a tosca template that defines a 'tosca,example,types,Address' data type and a 'tosca,example,types,Person' data type that defines a property of type map of a type that is neither normative or a defined data type

Predicate: When parsing the template assert raises the error 'UnknownDataType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-10-extend_native

Prerequisite:

Description: Parsing a data type that extends a native type with constraints should be successful.

Target: a tosca template that defines three data types extented from string each adding some constraints to it

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.5-data_type-11-extend_native_add_properties

Prerequisite:

Description: Parsing a data type that adds properties to a native type should fail.

Target: a tosca template that defines a data types extented from string and that add properties

Predicate: When parsing the template assert raises the error 'InvalidNativeTypeExtend'

Prescription level: mandatory

Conformance target: Parser-Validator

3.6.6 Capability Type

Id: 3.6.6-capability_types-01-valid

Prerequisite:

Description: Parsing a capability type with all keynames should be successful.

Target: a tosca template that defines a valid capability type with all keynames

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.6-capability_types-02-valid-required-only

Prerequisite:

Description: Parsing a capability type with required keynames only should be successful.

Target: a tosca template that defines a valid capability type with no optional keynames

Predicate: When parsing the template assert there is no errors optional assert raises the warning 'WarnNotInheritFromRoot'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.6-capability_types-03-unknown-parent-type

Prerequisite:

Description: Parsing a capability type that derives from an unknown capability type should fail.

Target: a tosca template that defines a valid capability type that derives from an unknown capability type

Predicate: When parsing the template assert raises the error 'InvalidParentType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.6.6-capability_types-04-unknown-source-type

Prerequisite:

Description: Parsing a capability type that defines a capability type with an unknown source type should fail.

Target: a tosca template that defines a capability type with an unknown source type

Predicate: When parsing the template assert raises the error 'UnknownCapabilityType'

Prescription level: mandatory

Conformance target: Parser-Validator

3.9.1 Keynames

Id: 3.9.1.1-metadata-01-valid

Prerequisite:

Description: Parsing a valid metadata section MUST succeed.

Target: The 'metadata' value out of the parsing

Predicate: When parsing the template assert 'template_name' value is equal to 'tosca_meta_data_test'

Prescription level: mandatory

Conformance target: Parser-Validator

3.9.2 Grammar

3.9.3 Top-level keyname definitions

Id: 3.9.3.3-metadata-02-complex_template_name_metadata

Prerequisite:

Description: Parser should be fail if metadata is a complex yaml value and template_name type is a string.

Target: a tosca template that has a complex yaml value [map/list] defined for template_name metadata

Predicate: When parsing the template assert raises the error 'InvalidType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.9.3.4-metadata-03-complex_template_author_metadata

Prerequisite:

Description: Parser should be fail if metadata is a complex yaml value and template_author type is a string.

Target: a tosca template that has a complex yaml value [map/list] defined for template_author metadata

Predicate: When parsing the template assert raises the error 'InvalidType'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.9.3.5-metadata-04-version_metadata_type

Prerequisite:

Description: Parser shoud fails when template_version metadata is not a version type

Target: a tosca template that has a non-version value defined for template_version metadata

Predicate: When parsing the template assert raises the error 'ValueTypeMismatch'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.9.3.7-dsl_definitions-01-valid

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with valid dsl_definitions version MUST succeed.

Target: a tosca template that has a valid dsl_definitions

Predicate: When parsing the template assert there is no errors

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.9.3.7-dsl_definitions-02-invalid-value-type

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with valid dsl_definitions that contains wrong values MUST fail.

Target: a tosca template that has a valid dsl_definitions which defines incorrect value types

Predicate: When parsing the template assert raises the error 'ValueTypeMismatch'

Prescription level: mandatory

Conformance target: Parser-Validator

Id: 3.9.3.7-dsl_definitions-03-unknown-definition

Prerequisite: TODO Add reference to the node template parsing test.

Description: Parsing a document with valid dsl_definitions and wrong reference MUST fail.

Target: a tosca template that refers to an unknown dsl_definition

Predicate: When parsing the template assert raises the error 'UnknownDslDefinition'

Prescription level: mandatory

Conformance target: Parser-Validator

