

	0.7.1 release				refactored model (0.8alpha)			
	Class	Property	Type	Class	Property	Type		
Conversion								
import path	from spdx.uis.import	SPDXNone		from spdx. tools.spdx.model.spdx_none import SpdxNone				
different class names	SPDXNone			SpdxNone				
import path	from spdx.uis.import	NoAssert		from spdx. tools.spdx.model.spdx_no_assertion import SpdxNoAssertion				
different class names	NoAssert			SpdxNoAssertion				
import path	from spdx.document	ExternalDocumentRef		from spdx. tools.spdx.model.document import ExternalDocumentRef				
different parent class & field name	Document	version	str	CreationInfo	spdx_version	str		
different parent class	Document	spdx_id	str	CreationInfo	spdx_id	str		
different parent class	Document	name	str	CreationInfo	name	str		
different parent class	Document	namespace	str	CreationInfo	document_namespace	str		
different field names & parent class	Document	data_license	str	CreationInfo	data_license	str		
need type conversion and different field names	Document	external_document_refs	List[ExternalDocumentRef]	CreationInfo	external_document_refs	List[ExternalDocumentRef]		
different field names & parent class	Document	comment	str	CreationInfo	document_comment	Optional[str]		
import path	from spdx.document	ExternalDocumentRef		from spdx. tools.spdx.model.document_ref import ExternalDocumentRef				
different field names	ExternalDocumentRef	external_document_ref_id	str	ExternalDocumentRef	document_ref_id	str		
different field names	ExternalDocumentRef	spdx_document_uri	str	ExternalDocumentRef	document_uri	str		
need type conversion	ExternalDocumentRef	checksum	Checksum	ExternalDocumentRef	checksum	Checksum		
import path	from spdx.creationinfo	Import		from spdx. tools.spdx.model.document import CreationInfo				
need type conversion	CreationInfo	creators	List[Creator]	CreationInfo	creators	List[Actor]		
adapt import path	CreationInfo	created	datetime	CreationInfo	created	datetime		
different field names	CreationInfo	comment	str	CreationInfo	creator_comment	Optional[str]		
need type conversion	CreationInfo	license_list_version	spdx_version.Version	CreationInfo	license_list_version	Optional[Version]		
import path	from spdx.version	Import		from spdx. tools.spdx.model.version import Version				
adapt import path	Version	major	int	Version	major	int		
adapt import path	Version	minor	int	Version	minor	int		
import path	from spdx.file	File		from spdx. tools.spdx.model.file import File				
adapt import path	File	name	str	File	name	str		
adapt import path	File	spdx_id	str	File	spdx_id	str		
adapt import path	File	comment	str	File	comment	str		
adapt import path	File	types	List[FileType]	File	file_types	List[FileType]		
need type conversion	File	checksums	List[Checksum]	File	checksums	List[Checksum]		
need type conversion	File	conc.lics	Union[License, License, utils.NoAssert, utils.SPDXNone]	File	concluded_license	Optional[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]		
need type conversion	File	licenses in file	Union[List[License], utils.NoAssert, utils.SPDXNone]	File	license_info_in_file	Optional[List[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]]		
adapt import path	File	license_comment	str	File	license_comment	str		
need type conversion	File	copyright	Optional[Union[str, utils.NoAssert, utils.SPDXNone]]	File	copyright_text	Optional[Union[str, SpdxNoAssertion, SpdxNone]]		
adapt import path	File	notice	str	File	notice	str		
adapt import path	File	contributors	List[str]	File	contributors	List[str]		
deprecated, need conversion	File	dependencies	List[str]					
deprecated, need conversion	File	artifact of project name	str					
deprecated, need conversion	File	artifact of project home	str					
deprecated, need conversion	File	artifact of project uri	str					
different field names, convert to List	File	attribution_text	List[str]	File	attribution_texts	List[str]		
import path	from spdx.file	FileType		from spdx. tools.spdx.model.file import FileType				
adapt import path	FileType		1:1 conversion (only different order in Enum)	FileType				
import path	from spdx.checksum	Import		from spdx. tools.spdx.model.checksum import Checksum				
different field names	Checksum	identifier	ChecksumAlgorithm	Checksum	algorithm	ChecksumAlgorithm		
adapt import path	Checksum	value	str	Checksum	value	str		
import path	from spdx.checksum	Import		from spdx. tools.spdx.model.checksum import ChecksumAlgorithm				
adapt import path	ChecksumAlgorithm		1:1 conversion (only different order in Enum)	ChecksumAlgorithm				
import path	from spdx.package	Import		from spdx. tools.spdx.model.package import PackagePurpose				
adapt import path	PackagePurpose		1:1 conversion (same order)	PackagePurpose				
need type conversion	ExternalPackageRef	category	str	ExternalPackageRef	category	ExternalPackageRefCategory		
different field names	ExternalPackageRef	pkga_ext_ref_type	str	ExternalPackageRef	reference_type	str		
adapt import path	ExternalPackageRef	locator	str	ExternalPackageRef	locator	str		
adapt import path	ExternalPackageRef	comment	str	ExternalPackageRef	comment	str		
import path	from spdx.package	Import		from spdx. tools.spdx.model.package import Package				
adapt import path	Package	name	str	Package	name	str		
adapt import path	Package	spdx_id	str	Package	spdx_id	str		
adapt import path	Package	version	str	Package	version	str		
adapt import path	Package	file name	str	Package	file_name	str		
need type conversion	Package	supplier	Optional[Union[Person, Organization, utils.NoAssert]]	Package	supplier	Union[Actor, SpdxNoAssertion]		
need type conversion	Package	originator	Optional[Union[Person, Organization, utils.NoAssert]]	Package	originator	Union[Actor, SpdxNoAssertion]		
may need type conversion	Package	download location	str (theoretically also allows utils.NoAssert, utils.SPDXNone)	Package	download_location	Union[str, SpdxNoAssertion, SpdxNone]		
adapt import path	Package	files analyzed	bool	Package	files_analyzed	bool		
need type conversion	Package	homepage	Optional[Union[str, utils.NoAssert, utils.None]]	Package	homepage	Union[str, SpdxNoAssertion, SpdxNone]		
need type conversion: fields need to be merged into one PackageVerificationCode object	Package	verif_code	str	Package	verification_code	Optional[PackageVerificationCode]		
need type conversion	Package	verif_exc_files	List[str]	Package	checksums	List[Checksum]		
adapt import path	Package	checksums	List[Checksum]	Package	checksums	List[Checksum]		
adapt import path	Package	source_info	Optional[str]	Package	source_info	Optional[str]		
need type conversion & different field names	Package	conc.lics	Union[License, License, utils.NoAssert, utils.SPDXNone]	Package	license_concluded	Optional[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]		
need type conversion & different field names	Package	license declared	str	Package	license_declared	Optional[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]		
adapt import path	Package	license comment	str	Package	license_comment	Optional[str]		
need type conversion & different field names	Package	namespaces from files	Union[List[License], utils.NoAssert, utils.SPDXNone]	Package	license_info_from_files	Optional[List[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]]		
different field names	Package	cr_text	Union[License, License, utils.NoAssert, utils.SPDXNone]	Package	copyright_text	Optional[Union[str, SpdxNoAssertion, SpdxNone]]		
adapt import path	Package	summary	str	Package	summary	Optional[str]		
adapt import path	Package	description	str	Package	description	Optional[str]		
adapt import path	Package	comment	str	Package	comment	Optional[str]		
different field names	Package	attribution_text	List[str]	Package	attribution_texts	List[str]		
need type conversion	Package	pkg_ext_refs	List[ExternalPackageRef]	Package	external_references	List[ExternalPackageRef]		
adapt import path	Package	primary package pur	Optional[PackagePurpose]	Package	primary_package_purpose	Optional[PackagePurpose]		
adapt import path	Package	release_date	Optional[datetime]	Package	release_date	Optional[datetime]		
adapt import path	Package	built_date	Optional[datetime]	Package	built_date	Optional[datetime]		
adapt import path	Package	valid_until_date	Optional[datetime]	Package	valid_until_date	Optional[datetime]		
import path	from spdx.snippet	Import		from spdx. tools.spdx.model.snippet import Snippet				
adapt import path	Snippet	spdx_id	str	Snippet	spdx_id	str		
adapt import path	Snippet	name	str	Snippet	name	Optional[str]		
adapt import path	Snippet	comment	str	Snippet	comment	Optional[str]		
different field names	Snippet	copyright	str	Snippet	copyright_text	Optional[Union[str, SpdxNoAssertion, SpdxNone]]		
adapt import path	Snippet	license comment	str	Snippet	license_comment	Optional[str]		
need type conversion & different field names	Snippet	attribution_text	List[str]	Snippet	attribution_texts	List[str]		
different field names	Snippet	snip_from_file_spdxid	str	Snippet	file_spdx_id	str		
need type conversion & different field names	Snippet	conc.lics	Union[License, License, utils.NoAssert, utils.SPDXNone]	Snippet	license_concluded	Optional[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]		
need type conversion & different field names	Snippet	licenses in snippet	Union[List[License], utils.NoAssert, utils.SPDXNone]	Snippet	license_info_in_snippet	Optional[List[Union[LicenseExpression, SpdxNoAssertion, SpdxNone]]]		
adapt import path	Snippet	byte range	Optional[Tuple[int, int]]	Snippet	byte_range	Tuple[int, int]		
adapt import path	Snippet	line range	Optional[Tuple[int, int]]	Snippet	line_range	Optional[Tuple[int, int]]		
import path	from spdx.relationship	Import		from spdx.model.relationship import Relationship				
adapt import path	Relationship	spdx element id	str	Relationship	spdx_element_id	str		
adapt import path	Relationship	relationship type	str	Relationship	relationship_type	RelationshipType		
adapt import path	Relationship	related spdx element	str	Relationship	related_spdx_element_id	str		
different field names	Relationship	relationship comment	str	Relationship	comment	Optional[str]		
import path	from spdx.relationship	Import		from spdx.model.relationship import RelationshipType				
adapt import path	RelationshipType		1:1 conversion (only different order in Enum)	RelationshipType				
import path	from spdx.license	Import		from license.expression import LicenseExpression				
adapt import path	License	full name	str	LicenseExpression		License(str).parse(license_expression_str)		
adapt import path	License	identifier	str	LicenseExpression		License(str).parse(license_expression_str)		
import path	from spdx.license	Import		from license.expression import LicenseExpression				
LicenseConjunction	LicenseExpression			LicenseExpression		License(str).parse(license_expression_str1 AND license_expression_str2)		
LicenseDisjunction	LicenseExpression			LicenseExpression		License(str).parse(license_expression_str1 OR license_expression_str2)		
import path	from spdx.license	Import		from spdx. tools.spdx.model.extracted_licensing_info import ExtractedLicensingInfo				
different field names	ExtractedLicense	text	str	ExtractedLicensingInfo	extracted_text	Optional[str]		
different field names	ExtractedLicense	cross_ref	List[str]	ExtractedLicensingInfo	cross_references	List[str]		
adapt import path	ExtractedLicense	comment	str	ExtractedLicensingInfo	comment	Optional[str]		
different field names	ExtractedLicense	full name	Union[str, NoAssert]	ExtractedLicensingInfo	license_name	Optional[Union[str, SpdxNoAssertion]]		
different field names	ExtractedLicense	identifier	str	ExtractedLicensingInfo	license_id	Optional[str]		
import path	from spdx.creationinfo	Import		from spdx. tools.spdx.model.actor import Actor				
adapt import path	Organization	name	str	Actor	name	str		
adapt import path	Organization	email	str	Actor	email	Optional[str]		
additional field needed	Organization			Actor	actor_type	ORGANIZATION		
import path	from spdx.creationinfo	Import		from spdx. tools.spdx.model.actor import Actor				
adapt import path	Person	name	str	Actor	name	str		
adapt import path	Person	email	str	Actor	email	Optional[str]		
additional field needed	Person			Actor	actor_type	PERSON		
import path	from spdx.creationinfo	Import		from spdx. tools.spdx.model.actor import Actor				
adapt import path	Tool	name	str	Actor	name	str		
additional field needed	Tool			Actor	actor_type	TOOL		
import path	from spdx.annotation	Import		from spdx. tools.spdx.model.annotation import Annotation				
need type conversion	Annotation	annotator	Union[Person, Organization, Tool]	Annotation	annotator	Actor		
adapt import path	Annotation	annotation date	datetime	Annotation	annotation_date	datetime		
different field names	Annotation	comment	str	Annotation	annotation_comment	str		
adapt import path	Annotation	annotation type	str	Annotation	annotation_type	AnnotationType		
adapt import path	Annotation	spdx_id	str	Annotation	spdx_id	str		
import path	from spdx.review	Import		from spdx. tools.spdx.model.annotation import Annotation				
need type conversion & different field names	Review	reviewer	Union[Person, Organization, Tool]	Annotation	annotator	Actor		
different field names	Review	review date	datetime	Annotation	annotation_date	datetime		
different field names	Review	comment	str	Annotation	annotation_comment	str		
additional field needed	Review			Annotation	annotation_type	REVIEW		
				Annotation	spdx_id	SPDXREF-DOCUMENT		