

Apex NT Domain

Introduction

The Apex domain lets you incorporate information from Ada units, Configurations, Views, and Subsystems in your SoDA documents.

Many Apex domain classes are derived from the File System (FileSys) domain classes File and Directory, and therefore inherit the attributes and relationships of those classes.

Classes that represent structure within Ada units are based on the ASIS interface. By default, SoDA opens and closes ASIS libraries as the units with those libraries are accessed. Performance when building a document from a number of large libraries can therefore suffer. Performance can be improved by defining the scope of the project.

Configuration

A configuration file contains a list of views, one per subsystem. Configurations are typically used to specify a set of views that make up a particular version of a system.

Configuration is a subclass of FileSys File.

SubClasses of Configuration

This class has no subclasses.

Properties Specific to Configuration

This class has no properties.

Relationships Specific to Configuration

Name	Kind	Class	Description
AllViews	0..n	View	All the views listed in the configuration file.
AllSubsystems	0..n	Subsystem	The enclosing subsystems of all the views listed in the configuration file.
ImportedViews	0..n	View	All the views imported by any view in the configuration but not themselves listed in the configuration.

Subsystem Class

Subsystem class is an Apex subsystem, that is, a directory ending in .ss and containing zero or more views.

Subsystem is a subclass of FileSys Directory.

SubClasses of Subsystem

This class has no subclasses.

Properties Specific to Subsystem

This class has no properties.

Relationships Specific to Subsystem

Name	Kind	Class	Description
AllViews	0..n	View	All the views contained within this subsystem, that is, directories ending in .wrk or .rel .

View Class

An Apex view, that is, a directory ending in **.wrk** or **.rel** and contained within a Subsystem.

View is a subclass of FileSys Directory.

SubClasses of View

This class has no subclasses.

Properties Specific to View

This class has no properties.

Relationships Specific to View

Name	Kind	Class	Description
EnclosingSubsystem	0..1	Subsystem	The Subsystem containing this view.
ImportedViews	0..n	View	The Views imported by this view.
Subdirectories	0..n	ViewDirectory	The directories immediately contained within this view.
AllSubdirectories	0..n	ViewDirectory	All directories contained within this view or, recursively, within those directories, their subdirectories, and so on.
Files	0..n	File	The files immediately contained in the view. Returns objects of the Apex domain File class, which differ from File System (FileSys) domain files in that Apex CMVC information is available for them.
AllFiles	0..n	CompUnit	All files contained within this view or, recursively, within the subdirectories of the view, their subdirectories, and so on. Returns objects of the Apex domain File class, which differ from File System (FileSys) domain files in that Apex CMVC information is available for them.
AllCompUnits	0..n	CompUnit	All Ada units contained in this view or, recursively, within the subdirectories of the view, their subdirectories, and so on.
UnitSpecs	0..n	UnitSpec	The Ada unit specifications immediately contained in this view, that is, files ending in .1.ad .

Name	Kind	Class	Description
AllUnitSpecs	0..n	UnitSpec	All Ada unit specifications contained in this view or, recursively, within the subdirectories of the view, their subdirectories, and so on.
ImportedUnits	0..n	UnitSpec	All units from other views visible from within this view.
ExportedUnits	0..n	UnitSpec	All units in this view that are visible to other views.

ViewDirectory Class

A directory within an Apex view.

ViewDirectory is a subclass of FileSys Directory.

SubClasses of ViewDirectory

This class has no subclasses.

Properties Specific to ViewDirectory

This class has no properties.

Relationships Specific to ViewDirectory

Name	Kind	Class	Description
EnclosingView	0..1	View	The View containing this directory.
Subdirectories	0..n	ViewDirectory	The ViewDirectories immediately contained within this directory.
AllSubdirectories	0..n	ViewDirectory	All ViewDirectories contained within this directory or, recursively, contained in its directories, their subdirectories, and so on.
Files	0..n	File	The files immediately contained in the directory. Returns objects of the Apex domain File class, which differ from File System (FileSys) domain files in that Apex CMVC information is available for them.
AllFiles	0..n	File	All files contained within this directory or, recursively, within its subdirectories, their subdirectories, and so on. Returns objects of the Apex domain File class, which differ from File System (FileSys) domain files in that Apex CMVC information is available for them.
CompUnits	0..n	CompUnit	The Ada units immediately contained within this directory, that is, files ending in .ada .
AllCompUnits	0..n	CompUnit	All Ada units contained in this directory or, recursively, within its subdirectories, their subdirectories, and so on.

Name	Kind	Class	Description
UnitSpecs	0..n	UnitSpec	The Ada unit specifications immediately contained in this directory, that is, files ending in .1.adb .
AllUnitSpecs	0..n	UnitSpec	All Ada unit specifications contained in this directory or, recursively, within its subdirectories, their subdirectories, and so on.

File Class

Any file contained within an Apex View or a subdirectory of that view. In addition to normal File System domain attributes, Apex CMVC information is available for objects of this class.

File is a subclass of FileSys File.

SubClasses of File

CompUnit

Properties Specific to File

Properties	Description
IsControlled	True if the file is under Apex CMVC control, otherwise False .
VersionHistory	The CMVC version history name. Returns a null string if the file is not controlled.
VersionNumber	The CMVC version number. Returns a null string if the file is not controlled.

Relationships Specific to File

Name	Kind	Class	Description
EnclosingView	0..1	View	The View containing the file.

CompUnit Class

An Ada compilation unit, that is, a file ending in **.ada**.

CompUnit is a subclass of Apex File.

SubClasses of CompUnit

UnitSpec, UnitBody, SubunitBody

Properties Specific to CompUnit

Properties	Description
AdaName	The name assigned to the unit in the source code. If the unit is a subunit, also includes the name of the parent(s). For example, the AdaName of <code>package Text_Io</code> is <code>Text_Io</code> . The AdaName of the subunit <code>Integer_Io</code> is <code>Text_Io.Integer_Io</code> .
AttachedComments	Comments preceding or following the unit declaration and not separated from it by a blank line.
PrecedingComments	Comments that come before the unit declaration and are not separated from it by a blank line
FollowingComments	Comments that come after the unit declaration and are not separated from it by a blank line.
IsPrivate	True if the unit is a private library unit; otherwise False .
IsChild	True if the unit is a child library unit; otherwise False .
HasChildren	True if the unit has child library units; otherwise False .

Relationships Specific to CompUnit

Name	Kind	Class	Description
ChildUnits	0..n	CompUnit	The child library units for this compilation unit, if any.
DescendantUnits	0..n	CompUnit	The child library units for this compilation unit, if any.
ParentCompUnit	0..1	CompUnit	If the unit is a child library unit, this relationship will return the parent.
UnitDeclaration	0..1	Declaration	The declaration of the compilation unit.
WithedUnits	0..n	UnitSpec	All unit specs upon which this compilation unit directly depends, that is, the contents of the <i>with</i> clauses of the unit.

UnitSpec Class

A CompUnit that is an Ada specification.

UnitSpec is a subclass of CompUnit.

SubClasses of UnitSpec

This class has no subclasses.

Properties Specific to UnitSpec

Properties	Description
HasBody	True if the unit has a corresponding body; otherwise False .

Relationships Specific to UnitSpec

Name	Kind	Class	Description
MyBody	0..1	UnitBody	The compilation unit body that corresponds to this specification.
ReferencingUnits	0..n	CompUnit	All compilation units that reference (that is, <i>with</i>) this specification. The scope of units considered in evaluating this relationship is controlled by the SODA_APEX_LIBRARY environment variable.

UnitBody Class

A CompUnit that is an Ada body.

UnitBody is a subclass of CompUnit.

SubClasses of UnitBody

This class has no subclasses.

Properties Specific to UnitBody

This class has no properties.

Relationships Specific to UnitBody

Name	Kind	Class	Description
MySpec	0..1	UnitSpec	The compilation unit specification that corresponds to this body.
AllDependencies	0..n	CompUnit	All units upon which this body depends for compilation. Includes units in the <i>with</i> clauses of the specification.
MySubunits	0..n	SubunitBody	All subunits declared within this body.

SubunitBody Class

A CompUnit that is an Ada subunit.

SubunitBody is a subclass of CompUnit

SubClasses of SubunitBody

This class has no subclasses.

Properties Specific to SubunitBody

This class has no properties.

Relationships Specific to SubunitBody

Name	Kind	Class	Description
EnclosingUnit	0..1	UnitBody	The unit body in which this subunit is declared.
AllDependencies	0..n	CompUnit	All units upon which this body depends for compilation. Includes units in the <i>with</i> clauses of the specification and enclosing unit bodies.
MySubunits	0..n	SubunitBody	All subunits declared within this subunit.

Declaration Class

Any Ada declaration.

SubClasses of Declaration

PackageSpec, PackageBody, SubprogramSpec, SubprogramBody, Parameter, Object, Type, Task, Entry, Exception

Properties Specific to Declaration

Properties	Description
IsGeneric	True if the declaration is an Ada generic, otherwise False .
IsVisible	True if the declaration is visible to other compilation units, otherwise False .
IsAbstract	True if the declaration (type or subprogram) is abstract, otherwise False .
IsAliased	True if the declaration is aliased, otherwise False .
SimpleName	The identifier of the declaration. For example, the SimpleName of the declaration <code>procedure Put (C : Character);</code> is <code>Put</code> .
MyKind	The Apex domain class name of the declaration. For example, MyKind of a package specification returns PackageSpec .
Keyword	The Ada keyword associated with the declaration. For example, the keyword of a package specification (or body) is package . Other possible responses are procedure , function , type , and task . Returns a null string for declarations that do not have a keyword.
Text	The full image of the declaration.
AttachedComments	Comments preceding or following the declaration and not separated from it by a blank line. The comment characters themselves (i.e., "--") are not included. For example, given the declaration: <pre>-- Character output put (C : Character); -- Provides output of a single character</pre> AttachedComments returns: Character output. Provides output of a single character.
PrecedingComments	Comments that come before the declaration and are not separated from it by a blank line.
FollowingComments	Comments that come after the declaration and are not separated from it by a blank line.

Relationships Specific to Declaration

Name	Kind	Class	Description
ParentCompUnit	0..1	CompUnit	The compilation unit in which the declaration is found.
Parent	0..1	Declaration	<p>The declaration, if any, that encloses this declaration. For example, given the code:</p> <pre>package One procedure A (X : Integer); end One;</pre> <p>One is the parent of A, and A is the parent of X.</p>
VisibleDeclarations	0..n	Declaration	The declarations in a package spec that can be referenced by clients of that spec (that is, declarations not in the private part). Returns a null list if the declaration is not a package spec.
PrivateDeclarations	0..n	Declaration	The declarations in a package spec that cannot be referenced by clients of that spec (that is, declarations in the private part). Returns a null list if the declaration is not a package spec.
AllDeclarations	0..n	Declaration	All declarations in the public or private parts of a package spec or within a package body.
GenericParameters	0..n	Declaration	The generic formal parameters of a generic declaration. Returns a null list if the declaration is not generic.

PackageSpec Class

An Ada package specification.

PackageSpec is a subclass of Declaration.

SubClasses of PackageSpec

This class has no subclasses.

Properties Specific to PackageSpec

This class has no properties.

Relationships Specific to PackageSpec

Name	Kind	Class	Description
MyBody	0..1	PackageBody	The package body corresponding to this specification.

PackageBody Class

An Ada package body.

PackageBody is a subclass of Declaration.

SubClasses of PackageBody

This class has no subclasses.

Properties Specific to PackageBody

This class has no properties.

Relationships Specific to PackageBody

Name	Kind	Class	Description
MySpec	0..1	PackageSpec	The package specification corresponding to this body.

SubprogramSpec Class

An Ada procedure or function specification.

SubprogramSpec is a subclass of Declaration

SubClasses of SubprogramSpec

FunctionSpec

Properties Specific to SubprogramSpec

This class has no properties.

Relationships Specific to SubprogramSpec

Name	Kind	Class	Description
MyBody	0..1	SubprogramBody	The subprogram body corresponding to this specification.

FunctionSpec Class

An Ada function specification.

FunctionSpec is a subclass of SubprogramSpec

SubClasses of FunctionSpec

This class has no subclasses.

Properties Specific to FunctionSpec

This class has no properties.

Relationships Specific to FunctionSpec

Name	Kind	Class	Description
ReturnType	0..1	Type	The declaration of the type returned by the function.

SubprogramBody Class

An Ada procedure or function body.

SubprogramBody is a subclass of Declaration.

SubClasses of SubprogramBody

FunctionBody

Properties Specific to SubprogramBody

This class has no properties.

Relationships Specific to SubprogramBody

Name	Kind	Class	Description
MySpec	0..1	SubprogramSpec	The specification corresponding to this subprogram body.
Parameters	0..n	Parameter	The formal parameters of this subprogram body.
Statements	0..n	Statement	The first-level statements declared in the subprogram body block.

FunctionBody Class

An Ada function body.

FunctionBody is a subclass of SubprogramBody

SubClasses of FunctionBody

This class has no subclasses.

Properties Specific to FunctionBody

This class has no properties.

Relationships Specific to FunctionBody

Name	Kind	Class	Description
ReturnType	0..1	Type	The declaration of the type returned by the function.

Parameter Class

An Ada formal parameter declaration. Can be part of a procedure, function, or task entry declaration.

Parameter is a subclass of Declaration.

SubClasses of Parameter

This class has no subclasses.

Properties Specific to Parameter

Properties	Description
Mode	The mode of the parameter: in , out , or in out .

Relationships Specific to Parameter

Name	Kind	Class	Description
MyType	0..1	Type	The declaration of the parameter's type.

Object Class

An Ada variable or constant declaration.

Object is a subclass of Declaration.

SubClasses of Object

This class has no subclasses.

Properties Specific to Object

This class has no properties.

Relationships Specific to Object

Name	Kind	Class	Description
MyType	0..1	Type	The declaration of the object's type.

Type Class

An Ada type declaration

Type is a subclass of Declaration.

SubClasses of Type

PrimitiveType, CompositeType, ProtectedType, TaskType

Properties Specific to Type

Properties	Description
IsTagged	True if this is a tagged type, otherwise False .

Relationships Specific to Type

This class has no relationships.

PrimitiveType Class

A discrete, real, or fixed type declaration.

PrimitiveType is a subclass of Type.

SubClasses of PrimitiveType

This class has no subclasses.

Properties Specific to Type

This class has no properties.

Relationships Specific to Type

This class has no relationships.

CompositeType Class

An array or record type declaration.

CompositeType is a subclass of Type.

SubClasses of CompositeType

This class has no subclasses.

Properties Specific to CompositeType

This class has no properties.

Relationships Specific to CompositeType

This class has no relationships.

ProtectedType Class

A protected array or record type declaration.

ProtectedType is a subclass of Type.

SubClasses of ProtectedType

This class has no subclasses.

Properties Specific to ProtectedType

This class has no properties.

Relationships Specific to ProtectedType

This class has no relationships.

Task Class

An Ada task object or type.

Task is a subclass of Declaration.

SubClasses of Task

TaskType

Properties Specific to Task

This class has no properties.

Relationships Specific to Task

Name	Kind	Class	Description
EntryPoints	0..n	Entry	The entry points of the task.

TaskType Class

An Ada task type

TaskType is a subclass of Type and a subclass of Task.

SubClasses of TaskType

This class has no subclasses.

Properties Specific to TaskType

This class has no properties.

Relationships Specific to TaskType

This class has no relationships.

Entry Class

An Ada task entry.

Entry is a subclass of Declaration.

SubClasses of Entry

This class has no subclasses.

Properties Specific to Entry

This class has no properties.

Relationships Specific to Entry

Name	Kind	Class	Description
Parameters	0..n	Parameter	The formal parameters of this task entry declaration.

Exception Class

An Ada exception.

Exception is a subclass of Declaration.

SubClasses of Exception

This class has no subclasses.

Properties Specific to Exception

This class has no properties.

Relationships Specific to Exception

This class has no relationships.

Statement Class

An Ada statement.

SubClasses of Statement

This class has no subclasses.

Properties Specific to Statement

Properties	Description
MyKind	The statement kind, as defined by ASIS. Examples: AN_IF_STATEMENT, A_CASE_STATEMENT, A_BLOCK_STATEMENT.
Text	The full image of the statement.
CommentsWithin	All comments that appear after the first line of the statement, but before the last line.
PrecedingComments	Comments that come before the statement and are not separated from it by a blank line.
FollowingComments	Comments that come after the statement and are not separated from it by a blank line.

Relationships Specific to Statement

Name	Kind	Class	Description
ParentDeclaration	0..1	Declaration	The subprogram body declaration that encloses this statement.

Subsystem Structure for Apex NT Templates

SoDA templates relating to Rational Apex NT are located in the following directory:

```
<install drive>\Program Files\Rational\SodAWord\template\Apex
```

Note: SoDA's Apex templates are designed specifically for use in a particular hierarchy of Apex subsystems and view.

Apex NT Subsystem Structure

SoDA's Apex 498 templates assume a certain structure for the subsystems in which they reside and from which documents will be generated. This default structure is described below. If you want, you can alter the templates to use a structure of your choosing.

```
a_system\
. interface_views.cfg
. a_system.ss\view.wrk\
... irs\
... idd\
... interface_diagram.*
. cscis\a_csci\
... a_csci.ss\view.wrk\
... srs\                      #
... sdd\                      #
... system_architecture.*
... csci_architecture.*
... states_and_modes.*
... all_views.cfg
... exported_views.cfg
... local_views.cfg
... source_subsystem.ss\view.wrk\
... sdd\                      #
... .preliminary\             #
... .architecture.*
... .scenarios\               #
... .scenario_n.*
... .detailed\                #
... .architecture.*
... .instance_diagrams\      #
... .instance_n.*
... .scenarios\               #
... .scenario_n.*
```

Note: The directories marked with a (#) can be empty, but they must exist for the documents that reference them to generate without errors.

Apex NT Subsystem Directories

SoDA System Directory

A 498 system is represented with a directory containing a subdirectory named **cscis** and an Apex configuration named **interface_views.cfg**. The **cscis** directory contains all the CSCI directories for the system. The **interface_views.cfg** configuration contains the views that are external interfaces of the system.

SoDA CSCI Directories

A 498 CSCI is represented with a directory containing three Apex configuration files **all_views.cfg**, **exported_views.cfg**, and **local_views.cfg**. Each of these files contains a list of Apex views, one per subsystem.

SoDA Document Subsystems

The SoDA templates assume that documents will be located within Apex subsystem views with a specific relationship to their information sources.

When you use the standard SoDA templates, you will create one additional subsystem to contain the system level documents and an additional subsystem for each CSCI in the system.

System-Level Documents

You create a subsystem in the root directory of the project to hold the project-wide documents (the IRS and IDD). This subsystem can have any name, but a good convention is to give it the same name as the project root directory but with the **.ss** extension. For example:

```
a_system\a_system.ss\view.wrk
```

The IRS and IDD documents are located in subdirectories within this subsystem's views.

CSCI-Level Documents

A directory is used to represent a CSCI. You create a subsystem within that directory to hold the CSCI-level documents (the SRS and SDD). This subsystem can have any name but a good convention is to give it the same name as the CSCI's directory but with the **.ss** extension. For example:

```
a_system\cscis\a_csci\a_csci.ss\view.wrk
```

The SRS and SDD documents are located in subdirectories within this subsystem's views.

SoDA Document Directories

Each of SoDA's 498 templates is, by convention, located in its own subdirectory. In some cases, the templates expect these directories to contain additional subdirectories and files.

Requirements Documents

For requirements, SoDA supports a project-wide IRS and CSCI-level SRSs.

Interface Requirements Specification (IRS)

The IRS template is located in an **irs** subdirectory of the project-wide document subsystem view. For example:

```
a_system\a_system.ss\view.wrk\irs
```

The IRS template is primarily used as a template in which to manually enter the document content.

Software Requirements Specification (SRS)

The SRS template is located in an **srs** subdirectory of each CSCI's document subsystem view. For example:

```
a_system\cscis\a_csci\a_csci.ss\view.wrk\srs
```

The SRS template is primarily used as a template in which to manually enter the document content.

Design Documents

For design, SoDA supports a project-wide IDD and CSCI-level SDDs.

Interface Design Document (IDD)

The IDD template is located in an **idd** subdirectory of the project-wide document subsystem view. For example:

```
a_system\a_system.ss\view.wrk\idd
```

The IDD template contains SoDA commands to automatically generate document sections for interfaces that are to be implemented as Ada program calls. Sections for interfaces implemented in other ways (such as pipes, sockets, and so on.) are entered directly into the document.

The information sources for the automatically generated document sections are the exported program units of the project's code-containing subsystems. If a file named **interface_diagram.*** is present in the IDD document directory it will be included in the document. This file can be in any graphic format imported by Word.

Software Design Document (SDD)

The SDD template is located in an **sdd** subdirectory of each CSCI's document subsystem view. For example:

```
a_system\cscis\a_csci\a_csci.ss\view.wrk\sdd
```

The SDD template contains SoDA commands to generate most of the document. The information sources for the SDD are the code-containing subsystems of the CSCI. In addition, if files named **system_architecture.***, **csci_architecture.***, and/or **states_and_modes.*** are found in the SDD document directory they will be included in the document.

SoDA Source Subsystems

The code-containing subsystems of the project are the information source for the automatically generated sections of the IDD and SDD. To generate either of these documents, you need to create specifications for all units that are to appear in the document. You do not need to do anything special to the Ada units for SoDA to generate these documents. However, if you follow the convention of attaching comments to declarations, those comments will appear in the documents. A comment is considered attached to a declaration if there are no blank spaces between the element and the comment. For example:

```
-- this comment is attached to A_Package
package A_Package is
-- this comment is attached to A_Package as well

    -- this comment is attached to A_procedure
    procedure A_Procedure;
    --
    -- this comment is attached to A_Procedure as well

    -- this comment is attached to A_Function
    function A_Function return Integer;
    procedure Another_Procedure (

        A_Parameter : Integer
        -- this comment is attached to A_Parameter but
        -- not Another_Procedure

    );
```

In addition to extracting information from subsystems and Ada units, SoDA looks for graphics in the following subdirectories of each source subsystem view:

- A CSC interface architecture diagram in SDD section 4.1.X from:
 <view>\sdd\preliminary\architecture.*
- Object scenario diagrams in SDD section 4.1.X from:
 <view>\sdd\preliminary\scenarios*
- A CSC implementation architecture diagram in SDD section 5.X from:
 <view>\sdd\detailed\architecture.*
- Object scenario diagrams in SDD section 5.1.X from:
 <view>\sdd\detailed\scenarios*
- Object instance diagrams in SDD section 5.1.X from:
 <view>\sdd\detailed\instance_diagrams*

Note: The subdirectory names for the location of graphics in the source subsystem views are defined in domain extensions. For more details, see the Domain Extensions topic in SoDA Help.